APPLIED REAL-TIME INEGRATED DISTRIBUTED CONTROL SYSTEMS: AN INDUSTRIAL OVERVIEW AND AN IMPLEMENTED LABORATORY CASE STUDY

Wael Khaled Zaitouni

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

August 2016

APPROVED:

Yan Wan, Major Professor
Xinrong Li, Committee Member
Shengli Fu, Committee Member and Chair of the Department of Electrical Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Victor Prybutok, Vice Provost of the Toulouse Graduate School

Zaitouni, Wael Khaled. *Applied Real-Time Integrated Distributed Control Systems: An Industrial Overview and an Implemented Laboratory Case Study*. Master of Science (Electrical Engineering), August 2016, 356 pp., 11 tables, 202 figures, 126 numbered references.

This thesis dissertation mainly compares and investigates laboratory study of different implementation methodologies of applied control systems and how they can be adopted in industrial, as well as commercial, automation and control applications. Specifically, the research paper aims to assess or evaluate eventual feedback control loops' performance and robustness over multiple conventional or state-of-the-art technologies in the field of applied industrial automation and instrumentation by implementing a laboratory case study setup: the ball on beam system. Hence, the paper tries to close the gap between industry and academia. The first chapter gives a historical study and background information of main evolutional and technological eras in the field of industrial process control automation and instrumentation. Then, some related basic theoretical as well as practical concepts are reviewed in Chapter 2 of the report before displaying the detailed design. Chapter 3 analyses the ball on beam control system problem as the case studied in the context of this research through reviewing previous literature, modeling and simulation. The following chapter details the proposed design and implementation of the ball on beam case study as if it is under the introduced distributed industrial automation architecture. Finally, Chapter 5 concludes this work by listing several points leaned, remarks, and observations, and stating possible development and the future vision of this research.

ACKNOWLEDGEMENTS

I begin with Allah's blessed name, praise him and glorify him as he ought to be praised and glorified. Then, I pray for peace and blessings for all of his noble messengers, in particular on the last of them all, the blessed prophet Mohammed, peace be upon him. After that, I would like to express my gratitude to my advisor, Dr. Yan Wan, for everything she has done for me during my graduate studies at UNT. Words cannot describe my gratefulness to Dr. Wan for all the lessons I have learnt working under her. Her doors were always open for students anytime. She inspired me, encouraged me and taught me how to conduct scientific research and to be a good researcher. Working under her supervision has been a great pleasure, and I am proud that I was a lab member of such an outstanding individual as Dr. Yan Wan.

Thank you to my family. My beloved mother has thought me the real value and the importance of education and seeking knowledge. Thank you to my father! Words cannot explain my appreciation for the continuous financial support you always provide. Thank you, my parents, for all of your prayers, wishing, and support. Special appreciation also goes to my brothers who have helped me through and before my health situation and diagnosis and who won't hesitate in assisting and supporting me. I would like also to particularly reward my wife for all of her cooperation and sacrifices throughout our professional and educational paths to earn success, especially since we decided to pursue our graduate studies in the U.S.

Finally, to all of my professors, friends, colleagues, and the nice staff of Electrical Engineering department at UNT, thank you for your trust and support, especially Dr. Shengli Fu, the department chair, and my committee member, Dr. Xinrog Li. I won't ever forget your support. Thank you all and God bless you. Special reward is also given to Mr. Bobby Grimes, laboratory manager/academic technician at Engineering Technology for the mechanical work.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

3-D	Three-dimensional
ABB	ASEA Brown Boveri
ACII	American Standard Code for Information Interchange
AD	Analog devices
Aff	Acceleration feed forward
AI	Analog input
AM	Amplitude modulation
AMS	Assets management solution
AO	Analog output
APC	Advanced process control
API	Application programming interface
ARM	Advanced RISC machines
AS-I	Actuator sensor interface
ASIC	Application-specific integrated circuit
AWG	American wire gage
AWGN	White Gaussian noise
BCD	Binary coded decimal
BD	Block diagram
BIBO	Bounded input bounded output
BJT	Bipolar junction transistors
BLDC	Brushless DC
BRAIN	Broadband radio access for IP-based networks
BRAIN	Broadcast rapid access intelligent network
C&E	Cause and effects
CAD	Computer-aided design
CAN	Controller area network
CAT	Category

CCR	Central control room
CCS	Centralized control system
CEO	Chief executive officer
CMOS	Complementary metal-oxide semiconductor
CMR	Common mode rejection
CN	Control network
CNC	Computerized numerical control
COM/DCOM	Distributed component object model
CPR	Counts per revolution
CPU	Central processor unit
cRIO	Compact reconfigurable input and output
CSMA	Carrier sense multiple access
CSMA/CA	Carrier sense multiple access with collision avoidance
CV	Control valve
DA	Differential amplifier
DAC	Digital to analog conversion
DAQ	Data acquisition
DC	Discrete current
DCS	Distributed control system
DD	Device description
DDC	Direct digital control
DI	Discrete input
DLL	Data link layer
DO	Discrete output
DP	Differential pressure
DSP	Digital signal processing
DSPs	Digital signal processors
DSSS	Direct sequence spread spectrum

DV	Delta-V
EIA	Electronic Industries Association
EMF	Electromotive force
ERP	Enterprise recourse planning
ESD	Emergency shut-down
F.F	Foundation fieldbus
FB	Function block
FBD	Function block diagram
FFT	Fast Fourier transform
FGS	Fire and fas system
FHSS	Frequency hopping spread spectrum
Fig	Figure
FIR	Finite impulse response
FLOPs	Floating-point operations per second
FPGA	Field programmable gate array
FSK	Frequency shift keying
GE	General Electric
GM	General Motors
GMACs	Giga multiply-accumulate operations per second
GND	Ground
H/W	Hardware
HART	Highway addressable remote transducer
Hex	Hexadecimal
HMI	Human machine interface
HSE	High speed ethernet
HTML	Hyper Text Markup Language
HTTL	Hyper Text Transfer Protocol
HVAC	Heating ventilation and air-conditioning

I/O	Inputs/outputs
I/P	Current to pressure convertor
IC	Integrated circuit
IEC	International Electro-Technical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMC	Internal model control
INA	Instrumentation amplifier
IP	Internet Protocol
IPv6	IP version 6
IR	Infrared
IRR	Infinite impulse response
ISA	International Society of Automation
ISO	International Organization for Standardization
IT	Information technology
KCL	Kirchhoff’s current law
KVL	Kirchhoff’s voltage law
LAN	Local area network
LAS	Link active scheduling
LASs	Link active schedulers
LCD	Liquid crystal display
LCTI	Linear continuous time invariant
LCTV	Linear continuous time variant
LDTI	Linear discrete time invariant
LDTV	Linear discrete time variant
LP	Linear programming
LPF	Low pass filter
LQR	Linear quadratic regulator
LSB	Least significant bit

LSE	Least square estimation
LTI	Linear time invariant
MAC	Media access control
MCP	Motor control panel
MCU	Micro-controller unit
MIMO	Multiple inputs multiple outputs
MLD	Manual loader
MOSFET	Metal-oxide-semiconductor field-effect transistors
MPC	Model predictive control
MPPT	Maximum power point tracking
MPU	Micro processor unit
MSB	Most significant bit
MSPS	Mega-samples-per-second
MV	Manipulated variable
NI	National Instruments
NL	Nonlinear
NoC	Network on chip
Op Amp	Operational amplifier
OPC	Object linking for process control
OPC A&E	OPC alarms and events
OPC DA	OPC data access
O-QPSK	Offset quadratic phase shift keying
OS	Operating system
OSI	Open systems interconnection
P&ID	Process and instrumentation diagram
P	Proportional
PAC	Process/programmable automation controller
PC	Personal computer

PCB	Printed circuit board
PCI	Peripheral component interconnect
PCN	Process control network
PCS	Process control system
PD	Proportional derivative
PDU	Protocol data unit
PHY	Physical layer
PI	Proportional integral
PIB	Process interface building
PIC	Programmable intelligent computer
PID	Proportional integral derivative
PID-PIV	PID with proportional velocity feedback
PID-PIV-VAff	PID-PIV with velocity feed-forward and acceleration feed-forward
PIN	Positive-intrinsic-negative
PIV	Proportional position loop Integral and proportional velocity loop
PLA	Programmable logic array
PLC	Programmable logic controller
PLDs	Programmable logic devices
PM	Phase modulation
PMDC	Permanent magnet DC
PN	Positive negative
POS	Product of sums
P-P	Peak to peak
PPR	Pulse per revolution
PSU	Power supply unit
PV	Process variable
PWM	Pulse width modulation

R&D	Research and development
RAM	Random access memory
RC	Resistors and capacitors
RF	Radio frequency
RFID	Radio-frequency identification
RIB	Remote interface building
RISC	Reduced instruction set computer (a type of a microprocessor)
RLC	Resistors, inductors, and capacitors
ROM	Read only memory
RS	Recommended standard
RT	Real time
RTD	Resistor temperature detector
RTL	Resistor-transistor logic
RTOS	Real time operating system
RTU	Remote terminal unit
RX	Remote transmitters
SAR	Successive approximation
SCADA	Supervisory control and data acquisition
SF	State feedback
SFC	Sequence function chart
S/H	Sample and hold
SIS	Safety instrumented system
SISO	Single input single output
SLC	Single loop controller
SNR	Signal to noise ratio
SOA	Service-oriented architecture
SoC	System on chip
SOP	Sum of products

SP	Set point
SPI	Serial peripheral interface
SS	State space
SSE	Steady state error
S/W	Software
TCP/IP	Transmission Control Protocol over Internet Protocol
TCP	Transmission Control Protocol
TDMA	Time division multiple access
TF	Transfer function
TSMP	Time Synchronized Mesh Protocol
TTL	Transistor–transistor logic
TX	Receiver
URAT	Universal asynchronous receiver/transmitter
UNT	University of North Texas
UV	Ultraviolet
UWB	Ultra-wide band
Vfb	Velocity feedback
Vff	Velocity feed-forward
VHDL	VHSIC Hardware Description Language
VHSIC	Very high speed integrated circuit
VI	Virtual instruments
WAN	Wide area network
WCN	Wireless control network
W-HART	Wireless highway addressable remote transducer
Wi-Fi	Wireless fidelity
WiMAX	Worldwide interoperability for microwave access
WLAN	Wireless local area network
WSAN	Wireless sensors actuators network

WSN	Wireless sensor network
ZN	Ziegler and Nichols

CHAPTER 1

INTRODUCTION

This research paper mainly compares and investigates laboratory study of different implementation methodologies of applied control systems and how they can be adopted in industrial, as well as commercial, automation and control applications. It is vital to introduce brief essential and basic information before getting into specific details of research directions and results. To better explain the motivation of this research, brief historical and background information about control and automation is given in the first section expressing main developments and eras in the area. This background is then followed by a detailed description of the latest advances and technologies available in the market of control and industrial automation.

Similar to any developing technology, a comparison study should be conducted in order for both designers and operators to know advantages and disadvantages of different configurations and implementations which may provide guidance in choosing suitable approaches for various applications. Similarly, in the field of control, this paper focuses on the main available technologies with an emphasis on the impact of communications and control mechatronics, embedded implementation, and digital architecture on eventual performance of control loops' dynamics response as parts of large distributed industrial systems. Therefore, in addition to the history and background of digital distributed control systems, this dissertation's introduction comprises of an adequate description of communication networks and protocols in industrial process control in general, wireless technology in control and WSN in specific, common and basic principles concepts and definitions used in the research, and finally a clear thesis statement represented by the motivation, purpose and scope.

1.1. Background and Historical Information

As any other engineering field, the science and technology of control systems has evolved through various periods of developments known as “history of control” [18]. The concept of applying control systems has been recognized since ancient times [18], also found in history in a number of civilizations before that, and then classically developed in old European times. However, the modern control theory has actually been scientifically and academically studied in the 19th century [16]. Since that date until today, many developments have taken place in both theoretical as well as implementation aspects [16], [17]. Control systems theory and applications are found today to be crucial to be studied, understood, and implemented in countless applications and industries such as transportation, automobile, biology, biomedical, aerospace, electronics, communication systems, and manufacturing process plants automation.

This research focuses mainly on one of the most important applications of modern control theory, industrial process control systems automation and instrumentation. In parallel with known improvements and developments of control systems as a theory and a field of engineering discipline, its industrial automation application went through several eras classified based upon implementation technology. But, before getting into brief discussion about those evolution periods, a brief discerption about automation, and process and manufacturing industrial plants should be presented.

1.1.1. Overview of Industrial Plants and the Need of Control and Automation in Industry

Any industrial plant facility around the world should have sets of goals and priorities. As per priority levels that vary from one organization to another, those objectives are productivity, safety, stability, reliability, quality, maintainability, redundancy, cost reduction, optimization, configurability, etc. Those requirements necessitate the need of automation in general, which is

derived as a term from the two words “AUTOMATIC OPERATION” and which offers a number of advantages:

- Increases plant and business productivity
- Requires less human intervention, and therefore, reduces the risk of lack of manpower.
- Can assure high levels of safety due to its automatic intelligence and configurability.
- Contributes to process stability and robustness; therefore, fewer equipment’s shutdowns which leads to higher productivity, reduced production, raw material, and maintenance costs, and as a result, increased reliability.
- will ultimately lead to improved product quality if properly engineered, maintained, and operated,

Automation can be technically defined as a control system that takes care of the various operations involved in a certain process, in an automated way with minimal human intervention. Consequently, a control system is practically a combination of various devices that are integrated as a structure used to sense, measure, indicate, and manipulate RT variables, which in turn controls processes to achieve the desired results.

Industrial plant sites can be generally classified into three main categories based on their varying environments: process industry, discrete and manufacturing industry, and mixed.

1.1.1.1. Continuous process industry

Process industry’s operation and progression tends to be continuous normally. Thus, it involves mostly regulatory (continuous) control. Examples of process industry are power, chemical, petrochemical and polymers, oil and gas and refinery, minerals and mining plants, etc. In most cases, process plants are comparatively slow in operation and execution where continuous RT variables such as level, flow rate, pressure, temperature, density, voltage, current, etc. are vital

to be measured and controlled for the plant to produce the aimed products [12]. However, process control does not mean regulatory control only at all; rather, it also includes some discrete devices and machinery such as ON/OFF valves, limit switches, fans, and motors, and mechanical equipment, e.g. compressors and turbines which, involve more Boolean logics, interlocks, and sequences. Fig 1.1 (a), (b) and (c) below show offshore and onshore petroleum plants.

Fig 1.1: (a) an offshore petroleum plant; (b) piping and connecting offshore with onshore; (c) installation of offshores plants.

Also see Fig1.2 (a) and (b) under that demonstrate one major example of oil and gas process industry, refinery. After that follows Fig 1.3 (a) and (b), which shows power plants.

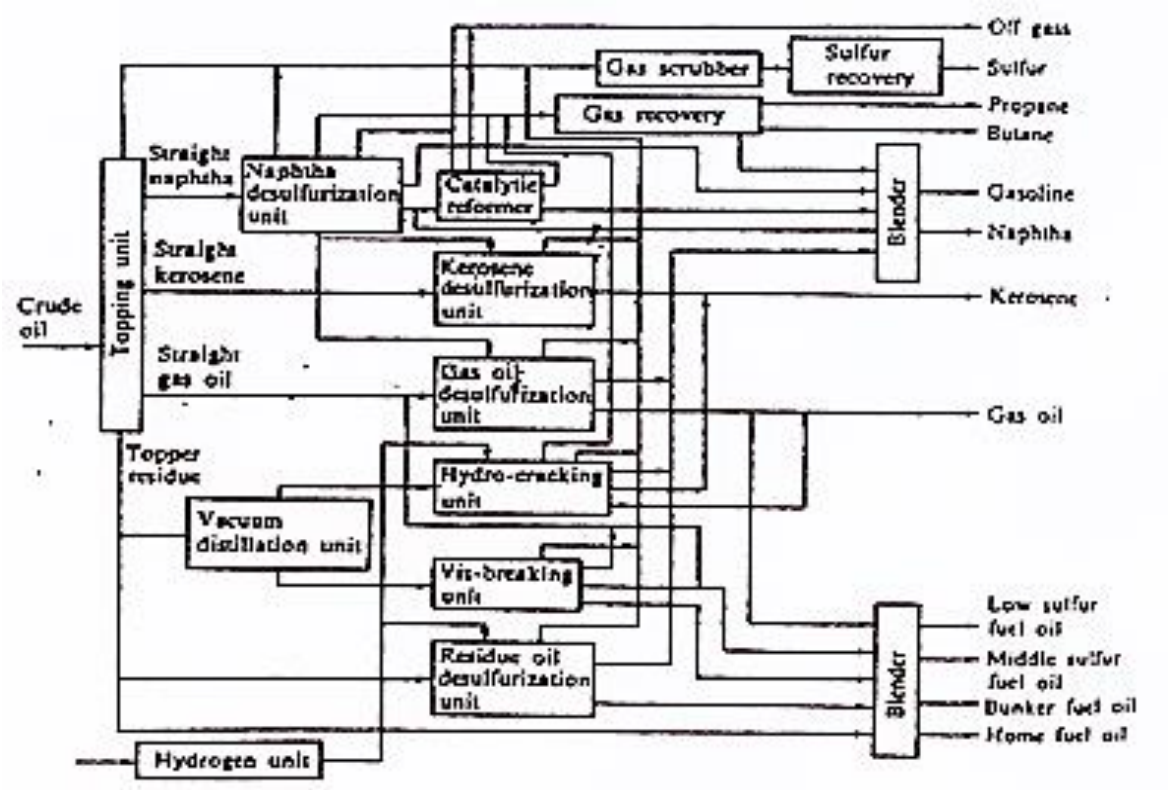


Fig 1.2 (a) From crude oil to refinery and production and (b) a common oil refining process flow

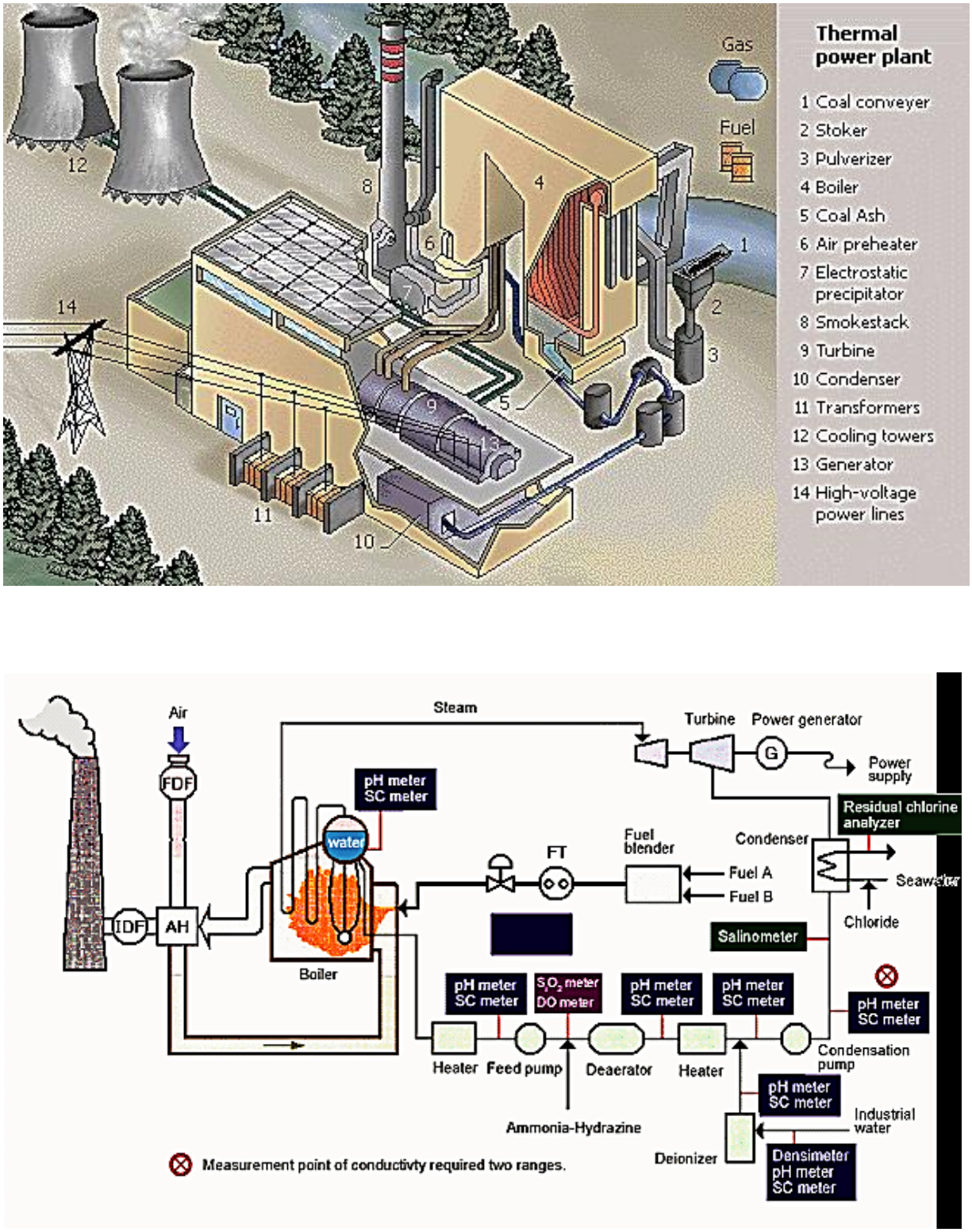


Fig 1.3 (a) a power generation plant's units and (b) main controls, P&IDs, and process flows of a typical thermal power plant.

1.1.1.2. Discrete and manufacturing industry

From its name discrete manufacturing requires more ON/OFF, machine, and sequential control, i.e. automatic packing, assembling, and CNCs, as in automobiles. This type of industry is usually faster than the process one; hence, it requires quick discrete responses and control actions that involve sequences and timing functions [12]. There, time, speed, weight, position, vibration, power, torque, etc. arise among the main variables to measure, monitor, and control. Likewise, other types of plants, being discrete in manufacturing, shall not imply ignoring other continuous variables and loops to be looked over secondarily. See Fig 1.4 (a) and (b).

Fig 1.4: Discrete manufacturing (a) packing and (b) assembling processes

1.1.1.3. Mixed (batch)

Mixed plants are a mixture of all continuous, discrete, and sequential, and which is mainly known as batch processes and industries, i.e. food manufacturing and pharmaceutical plants. This type of industry is more complex in terms of control and automation systems engineering since several final products can be manufactured from the same production line [13]. Hence, to control such processes requires sets of recipes sequences and permissive conditions besides normal regulatory and discrete instrumentation techniques. See Fig 1.5(a) and (b) and 1.6 (a) and (b).

Fig 1.5: (a) food manufacturing and (b) pharmaceutical are batch processes.

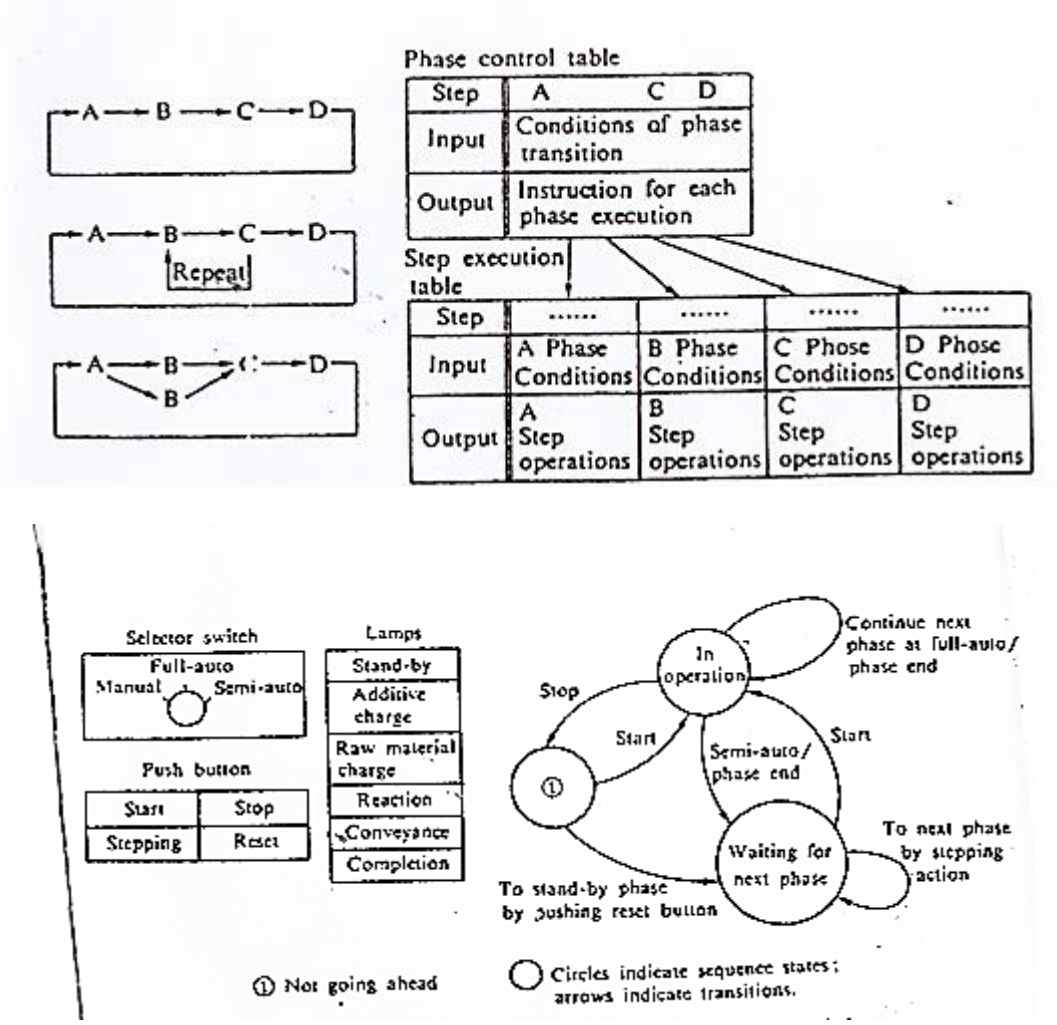


Fig 1.6: (a) sequences and execution steps and (b) states, phase, and modes in batch control.

1.1.2. Developing Eras of Industrial Process Control, Automation, and Instrumentation

The applied knowledge of measurement instrumentation has taken place in the industry back in the 18th century when the first pump pressure governors, voltmeters, ammeters, and thermometers were invented [17]. Since that time for almost a century, the term "Instrumentation" had been mainly meant by measurement and sensing instruments until 1920, when the idea of the use of automatic feedback control in industrial plants was first introduced as closing the loops for already installed process indicators [1], [17]. Since then, one of the most important practical implementations of modern control theory has been Industrial Automation and Process

Instrumentation, which has evolved rapidly since the last century. This evolution of process plants control instrumentation has gone through several modern eras: pneumatic systems, analogue controllers/systems, digital controllers, digital systems, centralized digital systems, distributed digital control systems, and finally the hybrid systems.

1.1.2.1. Pneumatic systems (1930s and later)

There are still retired or senior engineers who have seen and worked with pneumatic systems since back in the beginning and middle of the last century. At those moments of industrial automation and instrumentation history, every dynamic in process plants was implemented mechanically, including control systems. In other words, the way of building control systems for industrial facilities was purely using mechanical parts. Not only were controllers representing mechanical parts of process machines, but also the medium of communicating sensors and actuator signals was based on pressure manipulated compressed air, known as pneumatic sensing and actuating, and from that the 3 – 15 psi standard analog range is derived. [1], [17].

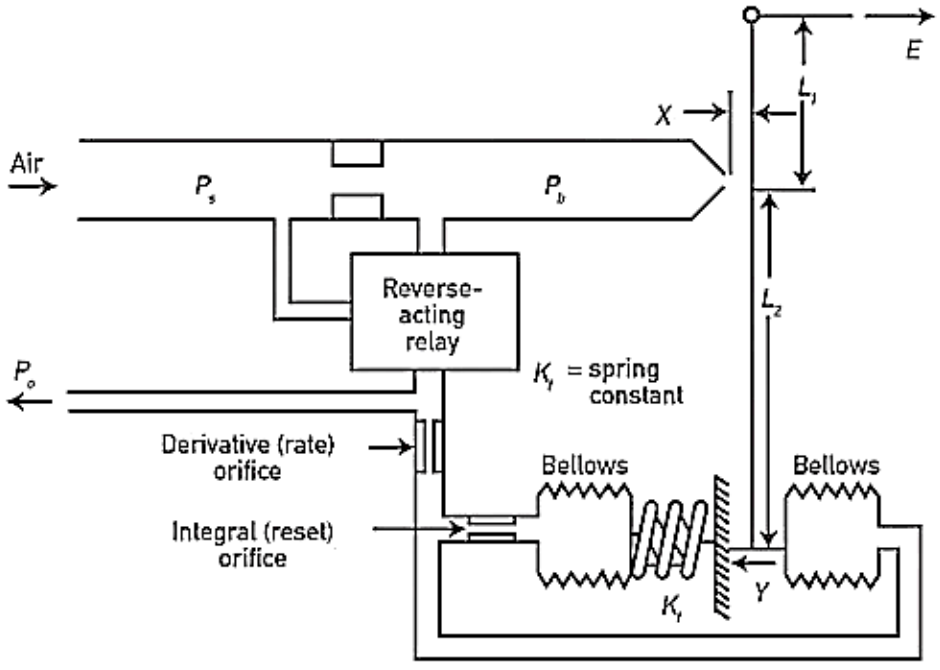


Fig 1.7: A mechanical PID controller (taken from [2])

Those types of automation systems do not exit any longer to control most of the industrial facilities today except for a very few. In contrast, they are still being implemented in some other applications as home automation and HVAC, although this field has gone through similar technological advancements [1]. Being mechanical in nature implies that those systems may perform very well in terms of control loop dynamic response, depending on the quality of the machine design, but they are huge in size and space, not flexible, not programmable, need regular periodic maintenance, require more frequent human intervention, not smart, or easily to be diagnosed, and so on.

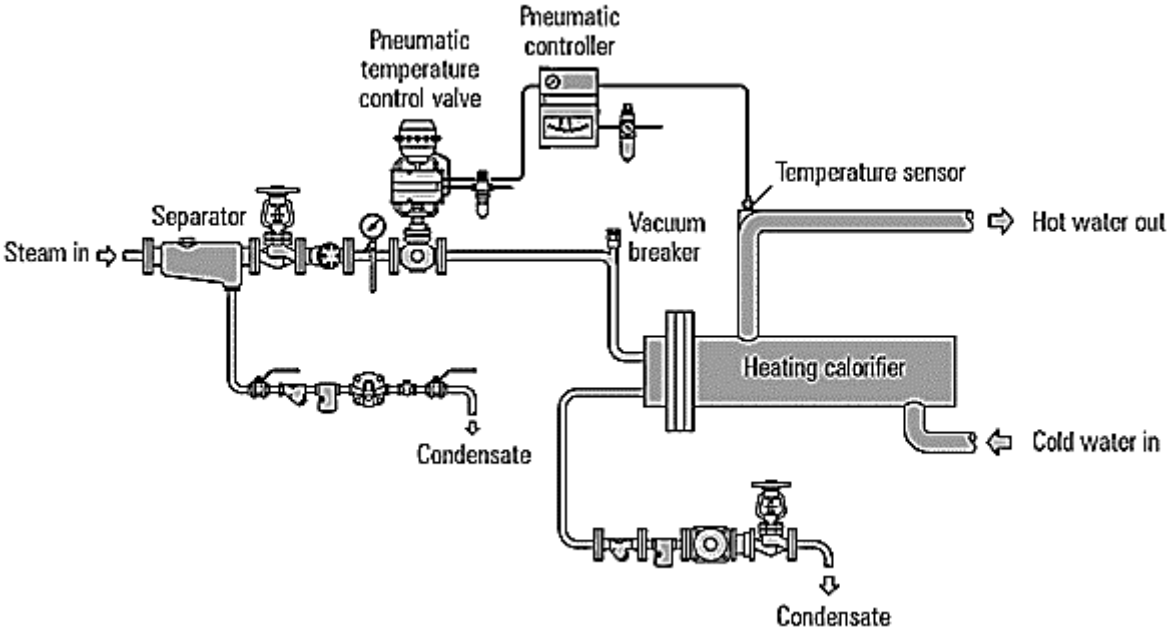


Fig 1.8: A mechanical pneumatic control loop and process flow diagram (taken from [3])

1.1.2.2. Electrical analog controllers (the Merged Era, 1940s)

The pneumatic period in general was accompanied with a revolution in the world of measurement instrumentation that followed significant inventions in the analog electronics, so later on, in the early; 40s, mechanical controllers were replaced by electronic ones. However, the plants' automation systems where still considered pneumatic in general because the way of the

field as well as supervisory monitoring communications and indications were still following the standard 3 -15 psi protocol, and so this era can be labeled as “the merged” period since both approaches were combined [1], [17]. Similarly, this technological period had its advantages and various disadvantages of the pneumatic devices, cables, and systems mentioned above. Besides, at those moments, electronics circuits and elements were relatively big in size. Several other drawbacks of analog systems in general will follow in the next subsection.

1.1.2.3. Analog systems (1950s)

Around 1950s, aligned with improvements in electronics theory and applications, not only had people started implementing controllers as analog circuits, but also the whole automation system was considered electronic. As for other dynamic electrical circuits, the simplest configuration would take the form of an RC circuit (more information is given in coming sections). The major difference between this era and the previous one is that the main manner of communicating field sensors and actuators to the control system was by placing conductor wires that carries electrical signals, voltage or current, rather than transmitting 3 – 15 psi pressure ranges. This method is known as “wired transmission”. Fig 1.9 (a) and (b) shows a control room/system of an analog industrial automation system. This IEC standardized, 3-15 psi, protocol was replaced by the famous 4 – 20mA in this period. If appropriately biased, designed and tuned, analog systems can always provide paramount performance due to their continuous time dynamics characteristics that matches RT process environments. Moreover, Analog systems/controllers are still preferable to be utilized until today in many applications; nevertheless, considering large scale industrial plants infers complying with the set of requirements and objectives stated earlier, and so analog controllers/systems in industrial automation could consequence in:

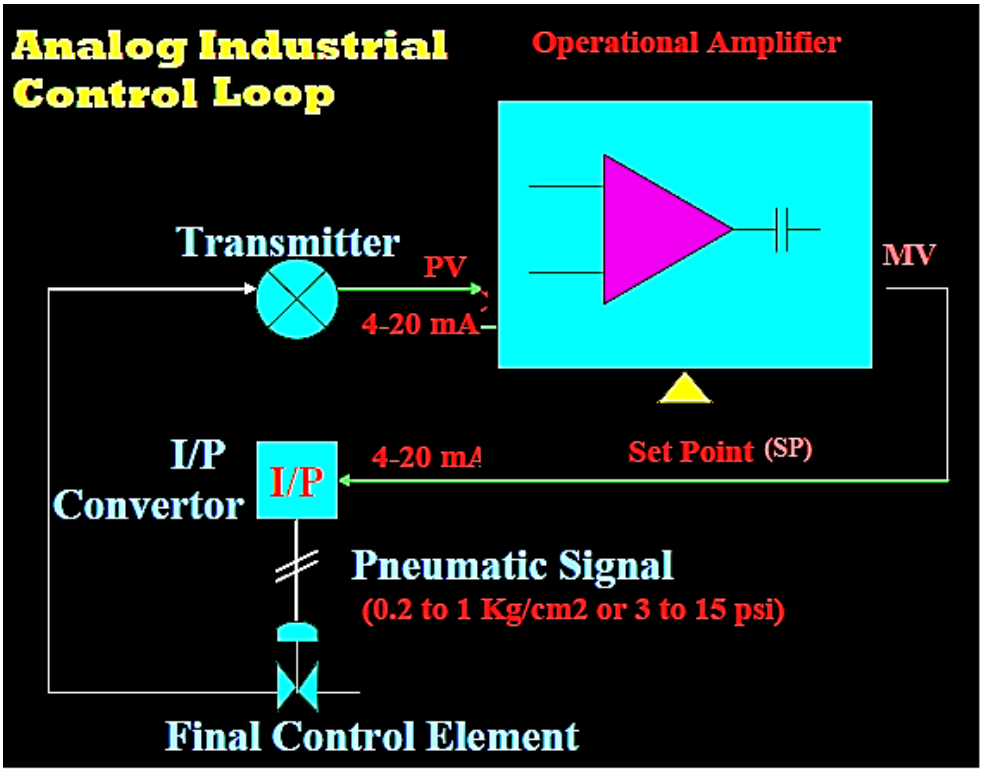


Fig 1.9 (a): A control room of a full electronic analog automation system where the indicators were pneumatic (taken from [1]) and (b) a diagram of a typical industrial analog control loop.

- Wiring problems: Increasing number of wires increases the risk of design and wiring errors, consequently higher engineering, maintenance time and cost.
- Circuitry space difficulties: The higher the number of plant's I/O, the higher the number of control loops; hence, extended capacity area of H/W circuitry becomes obligatory.
- Inflexibility and not being configurable (changes entails shutdown, rewiring, rebuilding the circuits, therefore more time and cost consumption)

1.1.2.4. The introduction of early digital computers in control and industrial automation

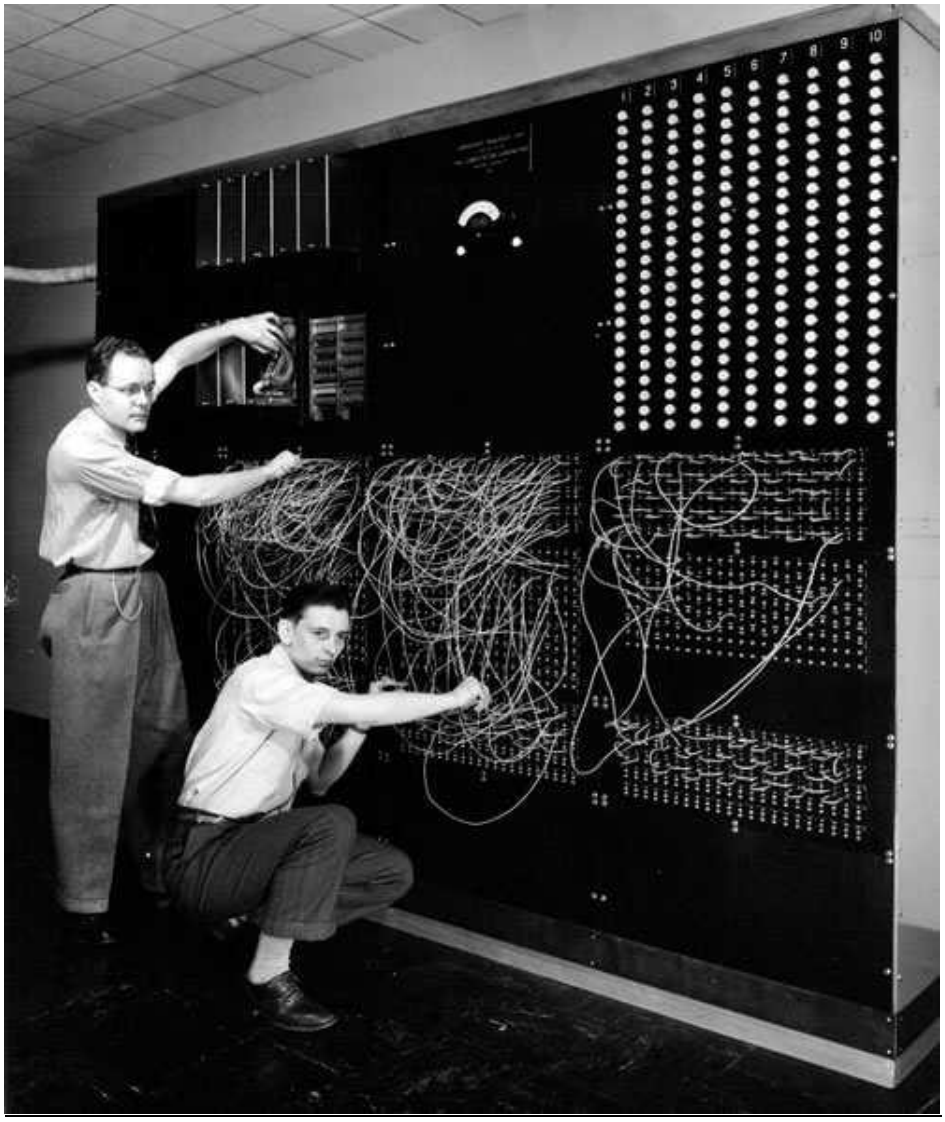


Fig 1.10: Early Digital Computers (taken from [1])

Later on, during late; 50s and the beginning of the; 60s, digital computers had begun to be widely used after being developed as operational (configurable) in the mid; 50s. This fact brings attention back to the history of digital computers themselves where they were less accurate, less efficient, and slower in speed and computational capabilities. As an example, computers took around 20ms to perform only one multiplication operation. They were much larger in space, compared to what people have today, and unintelligent such that more human intervention was required for operation and configuration [1], as appears in Fig 1.10 above. Briefly speaking, computers were still unable to perform RT control tasks; hence, during this period, they were used for other minor functionalities in industrial automation such as indication, displaying, recording, trending, printing and HMI SP change while the actual controllers were still analog [1]. Fig 1.11 displays an old plant's digital control room of an analog system.

Fig 1.11: The use of digital computers in control rooms as indications and operation while the actual control was analog circuitry (taken from [1])

Till this point, all that has been introduced represents primarily the evolution of process automation industry where plants are considered mainly continuous, which involves regulatory

control essentially. However, during those times and before, people used to implement discrete control using H/W circuitry for ON/OFF applications. This H/W circuitry was prepared by manual relays as seen in Fig 1.12 on the next page [1].

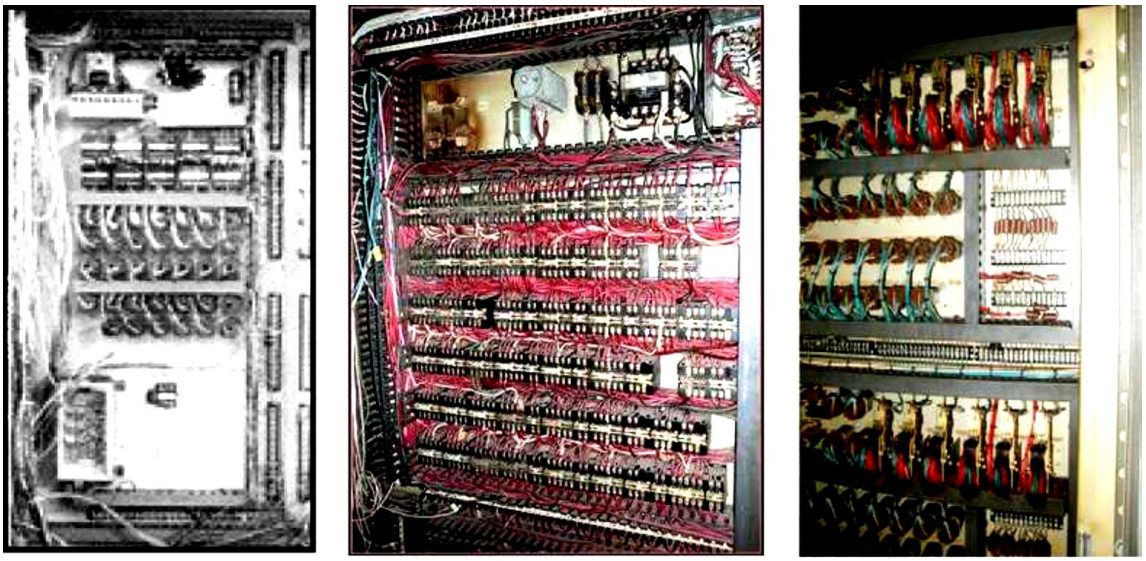


Fig 1.12 Manual relay panels from the early 1960s was the way of implementing digital ON/OFF discrete machinery control (taken from [1])

1.1.2.5. Direct digital control systems (centralized) and the birth of PLC

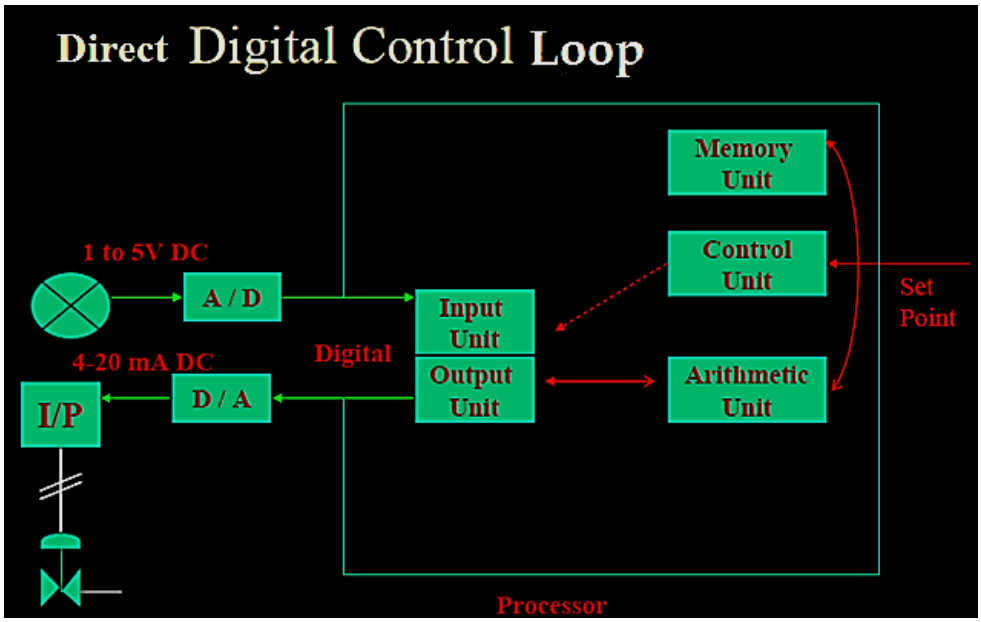


Fig 1.13: DDC and the use of ADC/DAC to interface the analog (field) world with computers

During the same decade, late; 60s and early; 70s, an imperative pace had taken place in the history of control. Digital control systems had emerged, but now instead of being deployed for minor tasks in control rooms and SCADA, computers were actually executing algorithms. In other words, digital control algorithms, implemented as computer programs directly, had replaced analog controllers; hence, the period was given the name DDC. Processors units' development was considered revolutionary in the whole world of engineering and technology that people have been living until the moment, including automation and control instrumentation. This invention of digital programmable and flexible computers had enabled efficiency, and fast computation and accuracy which significantly saved engineering, maintenance, as well as operation time, cost and effort if concerning thousands of loops and I/Os [1].

Fig 1.14: IBM minicomputer being used in controlling a plant, 1964 (taken from [1])

Though, at those historical moments, computers were still relatively large in volume and space; in addition, and most importantly, control was performed by one computer (CPU) only. This fact left computers loaded, consequently reducing both I/O capacity as well as efficiency. Not only were increased number of I/Os and programing logics loading CPUs, but also all other vital process control automation functionalities such as HMI, SCADA, trending and recoding all executed by only one CPU. For this reason, the name centralized control system arose as the apt label for this DDC method.

Leaders in the field of processing and computations had, later on in the; 70s, come with a higher capacity and developed centralized control computers that could handle industrial automation applications as the one that appears in Fig 1.14, IBM's minicomputers [1]. Yet, as per industrial control and instrumentation standards, the two core technical critics of CCSs rely on:

- If the CPU fails, the entire industrial plant gets affected and consequently shutdown. This fact results in CCSs not satisfying exactly a number of basic industrial requirements, i.e. productivity, reliability, and cost reduction.
- Redundancy concept was not available at that time. Redundancy in automation can be defined as having two parallel systems or controllers; one would be active and the other would be standby so that if one fails the other takes over.

These two main shortcomings and other mentioned above encouraged R&Ds departments in leading manufacturers and vendors in the field of industrial automation to seek enhanced and satisfactory solutions for non-stopping demands of process plants' end user business owners of more flexible digital systems that would insure maximum productivity, reliability, safety, and smartness. Fig 1.15 below shows a simple illustration of a centralized digital controller that were used before in industrial automation.

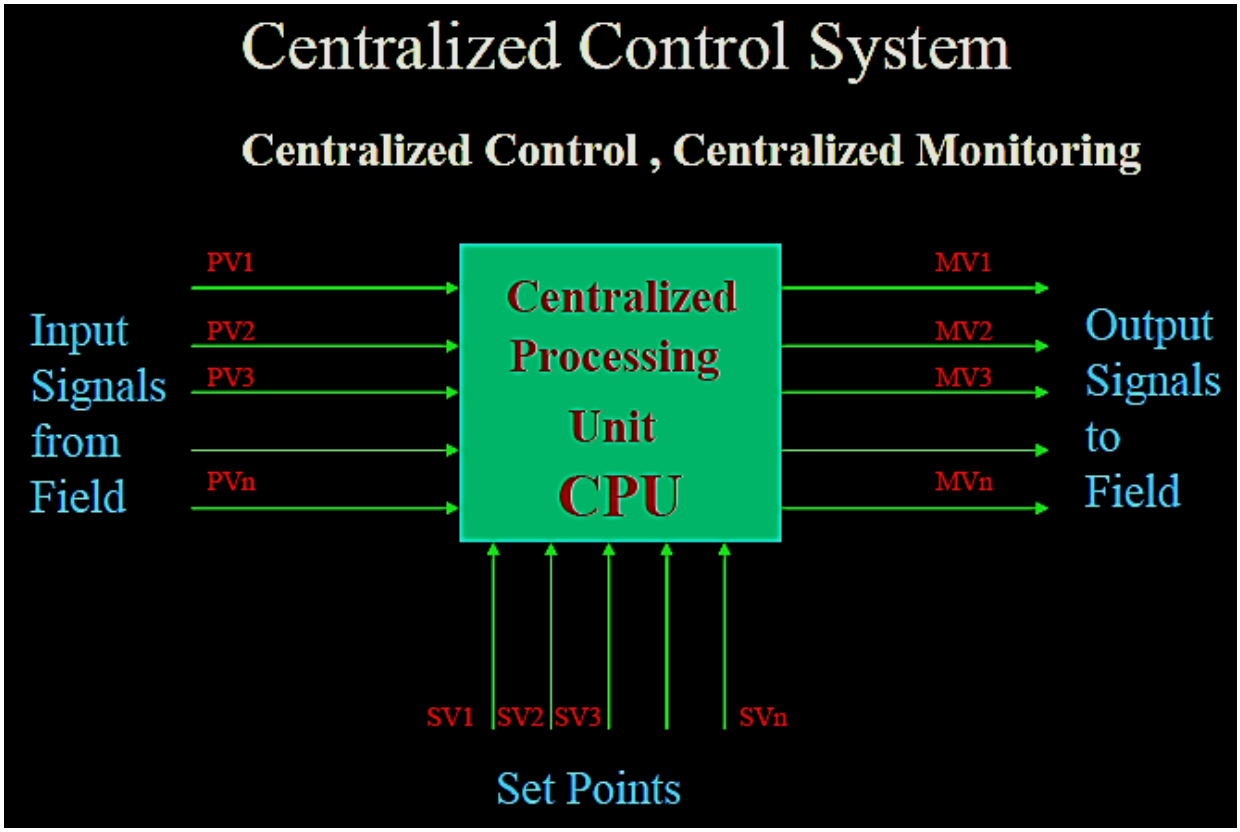


Fig 1.15: General centralized DDC architecture

Similarly, regarding discrete devices and machinery control, this period witnessed a stepping stone towards advancement in this area. The demand came down from one of the biggest end-users in the world, GM, to several machine controllers and motor builders such as Allen Bradley in 1968 to deliver a compact, programmable, and efficient solution to manual ON/OFF relay panels which were inflexible, large, and time and cost consuming. GM put general guidelines for a device that could satisfy the following:

- A standard machine controller that is compatible with globally accepted ladder logic diagram which emulates the relay panel arrangement.
- The device, or digital controller should be capable of performing dynamics logic besides static implemented by relay ones.

- A machine controller could reduce equipment's emergency shutdown and maintenance due time and increase the level of flexibility and reliability needed by accepting added I/O modules (modular) while managed by CPUs.
- It had to be more user friendly and rigid enough to adapt to a harsh industrial environment.

Fig 1.16 (a): The first PLC in history, Allen Bradley's the Bulletin 1774 and (b) the first Modicon PLC (taken from [1])

After almost a decade of trials, research, and prototyping of products such as the mini computers of the Digital Equipment, programmable matrix controller of Allen Bradley, and Modular Digital Controller of Modicon, the first PLC was born. So, the first device that was given this name, PLC, was an Allen Bradly's invention called the Bulletin 1774 PLC. It was created by two famous engineers: Odo Struger and Ernst Dummermuth. This PLC was a successful product that got accepted in industry, and it was a baseline for all other PLC products later on. Allen Bradley's competitor, Modicon, had then created its own PLC, The Modicon 184 [1]. Fig 1.16 (a) and (b) on the previous page shows snapshots of the first two PLCs: Allen Bradly's and Modicon's.

1.2.1.6. The emergence of DCS and development of PLC (late; 70s and; 80s)

As mentioned above the world of industrial process control and manufacturing were still demanding more universal solutions that could maximize plants' I/O capacity, profits, and safety by reducing computers sizes and interfacing with other forms of technological development such as S/W interface, communications and networks. Hence, the well-known Japanese leader in the field of instrumentation and measurement systems, Yokogawa, released CENTUM in 1975 as the world-first DCS [17]. This important stepping stone in the history of industrial automation saw the light after vital advancement in the domain of digital electronics, DSP and computer architecture where the famous microprocessor was invented besides the application of LAN network protocol IEEE 802.4, Token Bussing [1]. After that, in the early; 80s major DCS vendors released their similar product that quickly invaded the market. Those competitors are still known until today as the big DCS vendors and manufacturers, as for example Emerson (Fisher and Rosemont), Yokogawa, Honeywell, ABB, Invensys (Foxboro), Rockwell Automation (Allen Bradley), and Siemens, Omron, GE and others [1].

The main features, descriptions, and advantages of DCS over CCS stand:

- Control of large plants became distributed among multiple CPUs (controller units) so that if one controller fails it does not cause the whole plant to shut-down. Moreover, RT control and other functions got separated where the distributed controllers became embedded while other PCs on the same network are now used for other tasks such as SCADA, HMI, operation, graphics, etc.
- Every communicating device to the network such as field instruments with associated logics and interlocks became S/W connectable. Not only did network nodes and devices become software readable but also all other control and logic functions became standardized, user friendly, and easily programmable (with the advancement of higher level programing languages and FBD environment). As a result, flexibility that was missing in previous versions became possible with DCS, and so any change or modification requires simply reprograming then downloading.
- With features and technologies such as local LCDs, control room digital monitors and screens, charts and trending, graphical interface, and faceplates, continuous supervision and DAQ, information about the process is presented to operators in various formats in a centralized supervisory monitoring environment known as CCR. This way, operation, maintenance, diagnostics, and troubleshooting have become much easier.
- Field to system wiring had reduced significantly; also the new design had taken redundancy into consideration at different architectural levels which reduced costs and complexity.
- As a result of all the above, DCS offers more flexibility, configurability, supervisory monitoring availability, reliability, and redundancy; thus, more process stability, less shutdown, increased productivity, and of course cost effectiveness in the long run will be provided.

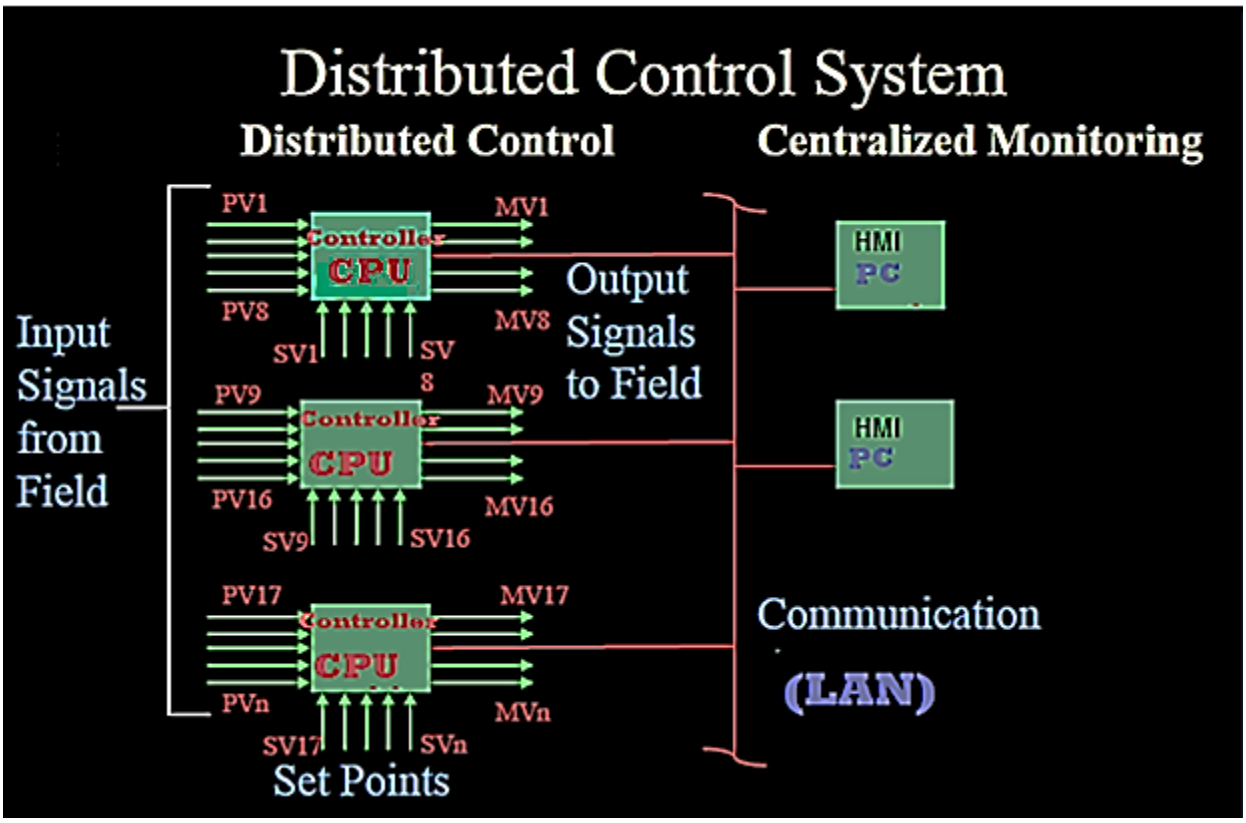


Fig 1.17: DCS implements a networked architecture of distributed embedded controllers and PCs

More detailed classifications on DCS's control units and communication aspects are given in the coming sections. DCSs today are found to be installed and functioning as "master" automation systems in thousands of plant sites around the globe covering various process, manufacturing, as well as services industries, for example, oil and gas (on shore and off shore), refining, petroleum and petro-chemical, mining, food manufacturing, pharmaceutical, power generation and distribution, and others. Fig 1.18 illustrates a typical modern digital DCS CCR today and how user-friendly plant operation, process control engineering, maintenance, and monitoring have become using computer graphics interface and higher level software and systems engineering tools. A typical DCS software displays a tree level configuration interface of the CN besides core control programing modules such as FBD, SFC, sequence tables, SIS logic solvers, C&E, Boolean logic gates, blocks and pages, ladder diagram, etc.

Fig 1.18: A typical DCS CCR and graphical interface operates plants easily (taken from [24])

Likewise, since that time, the PLC has become more standardized and widely implemented in not only machine and discrete device applications, but also in process control regulatory ones. This upgrades in the PLC usage and functionality were because of its evolution to have transistor logic, i.e., TTL I/Os (besides the mechanical relays ones), redundancy, operatory HMI, networking level expansion, and continuous control functions. PLC has become closer to DCS in features, yet each is still distinct and specialized in prime functioning areas. However, since day one, safety has been a real concern in such plants and factories, and as seen above that DCS is meant to chiefly improve quality and productivity after mainly looking to process control.

Since the PLC had developed to be more capable of providing similar communication, besides being fit for machine and sequential tasks, due to its architectural nature, fast logic execution rates, and so rapid I/O response, it has also been chosen to implement one of the plants' most important priories, i.e. SISs. One famous form of SIS is ESD systems which, from its name, is predestined to automatically takeover safety quick trips in case of emergency. Even though being PLC based and integrated with the master system, DCS, SIS/ESD has become a complete autonomous specialty today that has specified career path and certifications.

1.1.2.7. Digital communications (smart devices) era, late; 80s and early; 90s

Since the invention of micro digital computers, they were introduced to be used in control applications as seen before. However, although the emergence of digital control systems periods and the DCS technology that has provided more solutions, the mean of communicating the field process to the control system had been analogue by nature for a long time. Known as the conventional protocol, the 4 -20 mA analogue filed/system transmission had represented the industrial standard for decades until a series of research and developments had arisen in that matter, field/system communications, to add more smartness to this traditional way of measurement and control signals transmission.

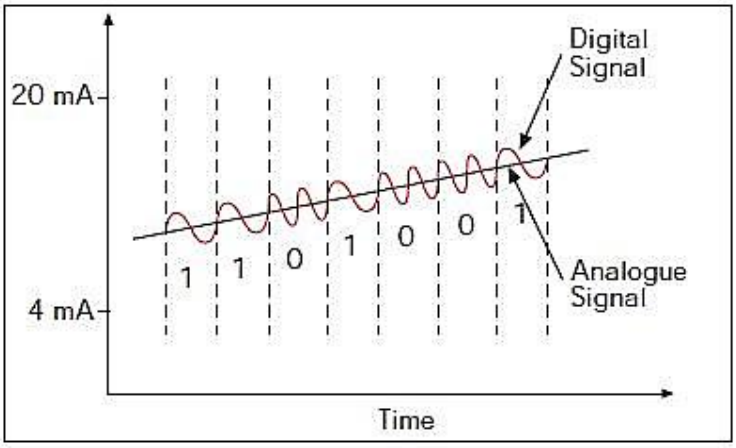


Fig 1.19: HART digital communication signal carried in the standard 4-20mA standard one.

For instance, starting from late; 80s, well-known standard protocols such as HART (the standard protocol of HART foundation), BRAIN (of Yokogawa Electric), and others have been introduced to process plants, in which transmitters and devices (field sensors and actuators) still send the 4 – 20mA analogue signals but accompanied with digital diagnostic signals as appears in Fig 1.19. More details of this type and classifications of field communication protocols and standards are given in Section 1.2 below, Communications networks and protocols. Those devices are called “smart” since they added more digital diagnostics features and transmission.

1.1.2.8. Full digital communications (fieldbus technology era) – late; 90s and; 00s

After the single point digital communications era, a revolution had occurred in that field, industrial automation communication protocols in which fieldbus technology had been introduced. Fieldbus technology made transmitters even smarter to send all signals, including process measurements' and controls' digitally. Not only is fieldbus fully digital, but also remotely configurable and, sometimes, smart enough to be independent of the control system. In other words, in fieldbus, basic control algorithms are built right inside field devices (sensors and actuators instruments), which allows more reliability in case of control system shutdown because the bus is powered up separately. Furthermore, in terms of wiring cost, fieldbus has been the most effective among all mentioned protocols because it took only one bus to connect a minimum average of eight devices (sensors and actuators) to the system, which is not the case for point-to-point wiring connection in 4 -20mA or even the wired HART [17]. Fig 1.20 underneath illustrates common DCS automation architecture showing several fieldbus protocols and the difference between them and the HART (digital point-to-point) filed/system communications.

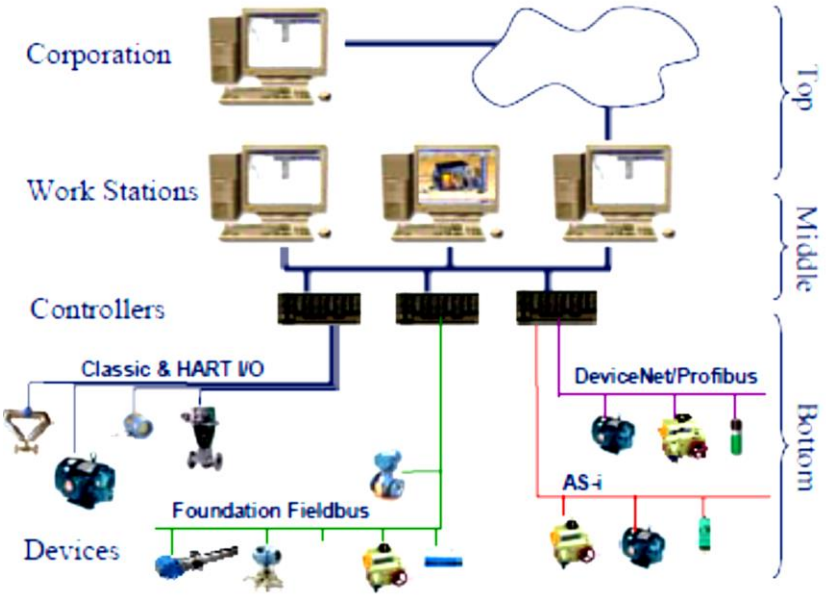


Fig 1.20: A typical DCS with various field communication protocols (taken from [10])

Recalling the basic idea, to control a large scale industrial plant with an automation system requires handling an enormous number of control loops by distributed (networked) controllers or systems. These systems, known as PCSs or Automation Systems, are attached to hundreds if not thousands of control loops which turn the problem into controllers' communicating field networks of sensors and actuators which ultimately compose, along with production management network and IT, a whole plant network as appears in Fig 1.21.

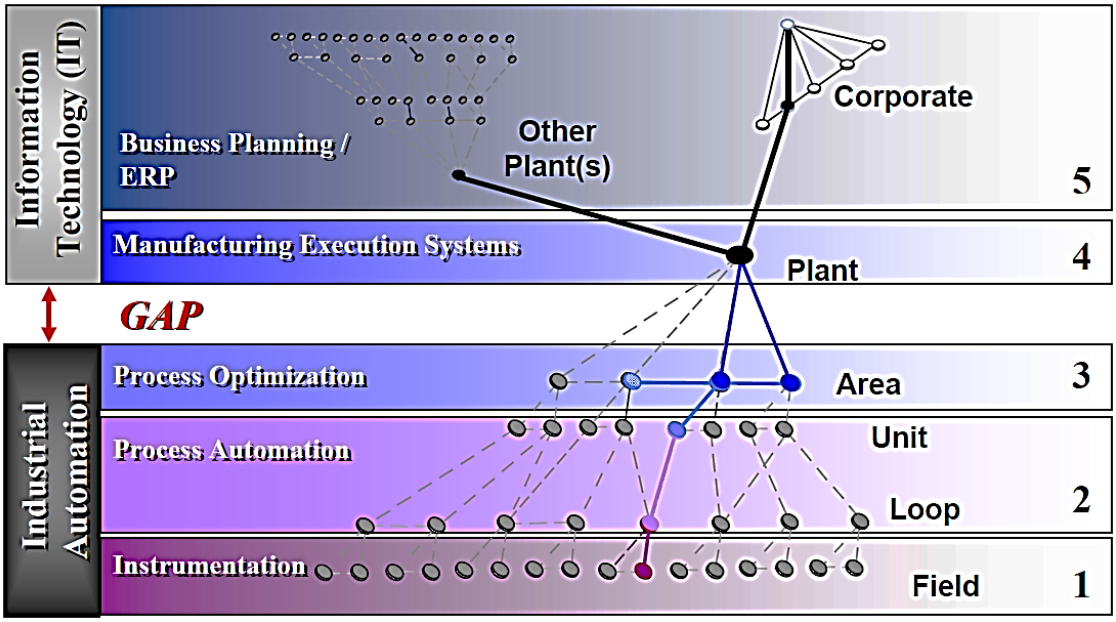


Fig 1.21: Indusial Communication Automation and IT networks architecture (taken from [21])

Controlling large plants started by making control distributed, had grown up to be different networks levels communicating with each other, and is still moving towards more integration until eventually comprising a complete integrated architecture as what appears in Fig 1.22. Two phases of integration exist in this era. First, integrating other third party subsystems such as PLCs, ESD, and other forms of SIS either to the CN directly or to be part of a higher level, PCN. The other form of integration is merging the plant intranet network under the IT scope with this PCN network. This era started to implement multiple hybrid fieldbus technologies in several levels but with a gap to higher levels.

1.1.2.9. Modern era and current R&D: hybrid systems (plant web networks levels architecture)

Industry is moving towards full integration as stated above, but as per very close personal experience with industrial automation engineering business field, the fieldbus technology has taken some time to be accepted eventually from process plants' end users. This means that industry is currently moving towards integration of multiple solutions in different network levels that form a whole "plant-wide networks architecture" of several hybrid standards and technologies. Yet, it indeed relies on the end user's philosophy, project budget, training, and even education concerning which technology and system architecture to adopt and deploy in which particular levels. With the advancement in communication and computer networks field protocols, CN, PCN, and plant IT networks integration has become possible such that a CEO of a certain production company can monitor, operate, and even tune and configure his plant remotely from overseas and abroad locations, for example, which can be understood from the figure below [5], [22], and [26].

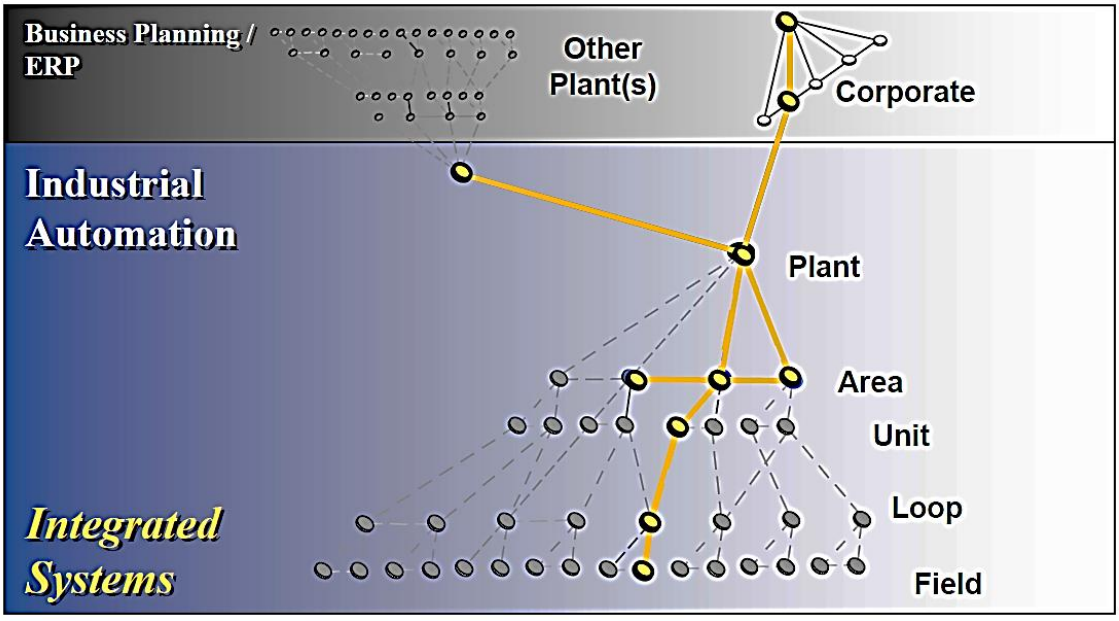


Fig 1.22: Industrial Automation and PCN is moving towards full integration with ERP and IT networks (taken from [21])

After the success of the fieldbus network era, which has been widely implemented in many plants worldwide, the development of wireless sensor networks, WSN, has taken researchers further in a different direction where there will not be even use of any sort of wires to control plants, especially on the field communication side. WSN in industrial process control sounds very promising; nevertheless, as any new technology under development, there has to arise a number of challenges which require more study and development to prove validity, especially concerning manufacturing facilities where safety, accuracy, reliability, and productivity matter significantly.

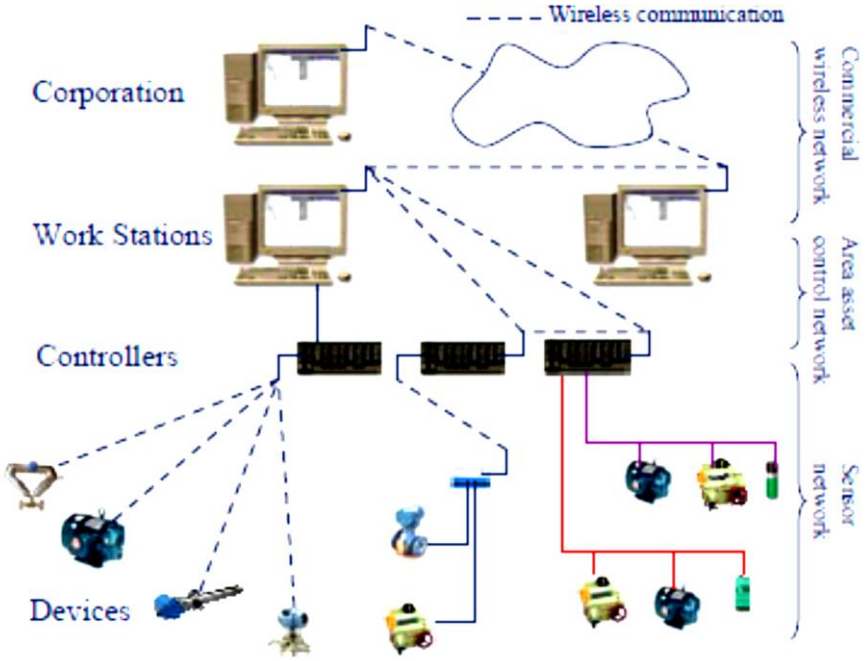


Fig 1.23: The R&D and latest proposed industrial hybrid architecture (taken from [10])

From that perspective, industrial specialist, as well as academic, groups have been working intensively on the use of WSN, WSAN, and WCN in industrial process control plants to come up with products and protocols that will persuade end-users to start measure and control their plants wirelessly. Aside from that, there appear a number of standards, technologies, and protocols considering replacing wireless networks in different architectural levels as appears in Fig 1.23. Section 1.3 illustrates the chief wireless standards and networks that might be utilized in industry.

1.2. Communication Protocols and Networks in Control and Automation

After the rough historical recall of main technological eras involved in applied process control systems, one could extract that industrial communications networks and protocols can be primarily classified into four main categories: filed communications and networks, control networks, higher level networks, and 3rd party subsystems communications. But before getting into detailed classifications and study of those different categories, [27] recalls the standard OSI model used to define most communication networks and protocols that appear in Fig 1.24 below.

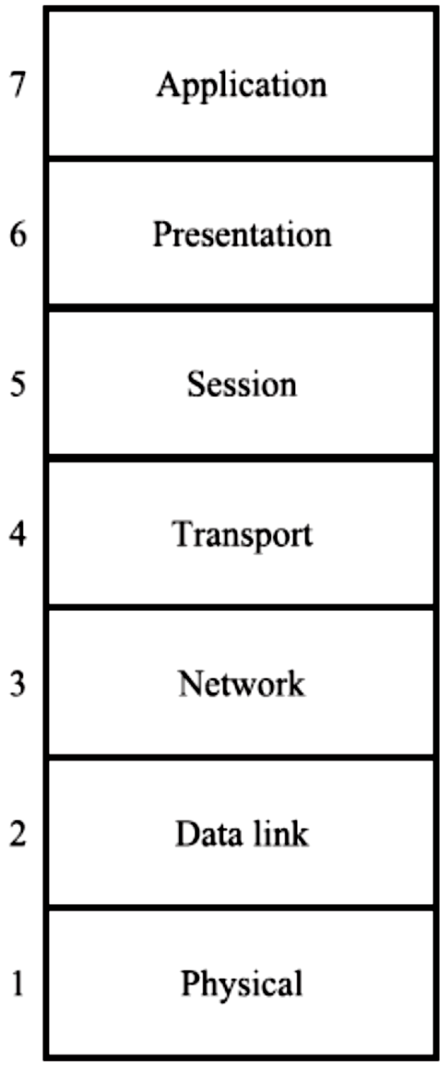


Fig 1.24: The ISO OSI model.

1.2.1. Field Communications and Networks

As seen earlier, automation systems technology has gone through historical evolution determined by measurement and control as well as communications implementation. As of today, communicating the lower levels of the modern automation architecture pyramid to the DCS is known as field communication. As per technical scientific transmission media and communication method, filed communication is further categorized into conventional point to point, digital (smart) single point, full digital (fieldbus), and WSN.

1.2.1.1. Conventional point to point

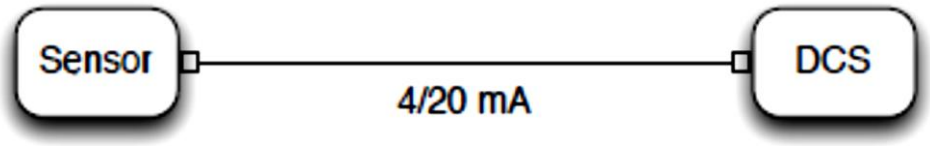


Fig 1.25 The conventional analog point-to-point protocol in plant automation (taken from [22])

The conventional field communication is so simple and abstract that it just requires wiring (cabling) to carry either voltage or current signals from/to field instruments (sensors and actuators) to/from instrumented control systems as DCS. As shown before and seen in Fig 1.25 above, the main standard in process control automation industry is the analog 4 – 20mA DC signals equivalent to continuous 0 – 100% PV or MV scales of ranges [23]. The main reason behind choosing 4mA to represent the zero value of PV, for example, relies on the ambition of differentiating between the minimum measured signals from the bad (no communication) ones. This type of field communication is distinguished as a one-way single variable where devices cannot be remotely configurable and deliver limited diagnostic information. However, this conventional protocol appears to be the most trusted technology with respect to control loop performance due to the advantage that an analog environment provides.

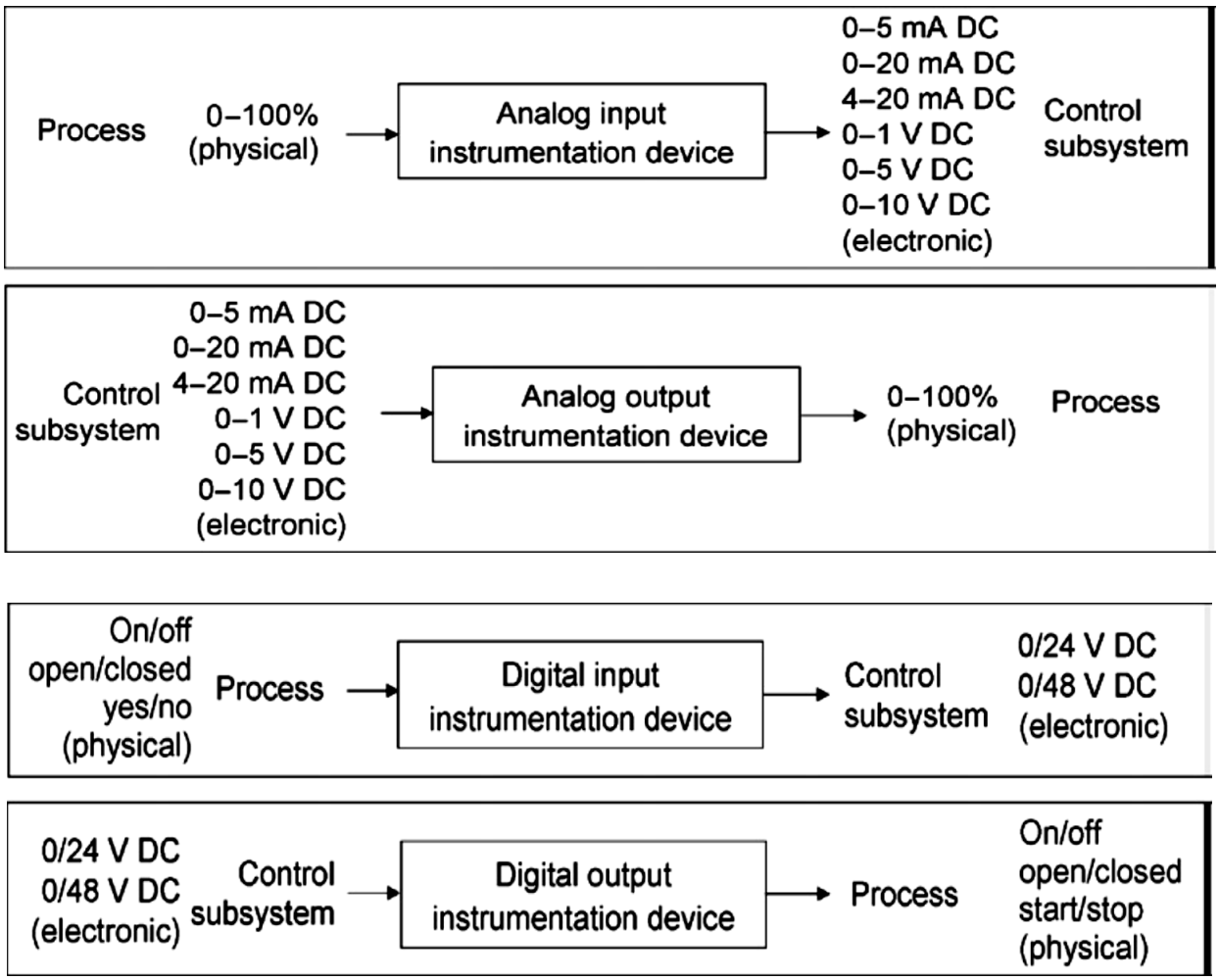


Fig 1.26: Various standards of the conventional field communication (taken from [23])

This type of field communication is a common method utilized since the vast majority of personnel as well as organizations possess the training and knowledge on how to deal with this standard protocol, which requires nothing but circuit connectivity analysis level, besides performance advantage of a very high transmission speed. Despite the advantage, this system involves a lot of wiring and cabling that may result in human error, therefore increased cost. Note that when referring to such conventional protocol, the 4 – 20mA is not the only one implemented despite being the most famous. There exist other automation standardizations, other than the industrial process control's, such as 0 – 5V, 1 – 5V, 0 – 10V, 0 – 2.5V, and others for both analog and discrete (ON/OFF) signals as appears in Fig 1.26 [23].

1.2.1.2. Digital (smart) point-to-point

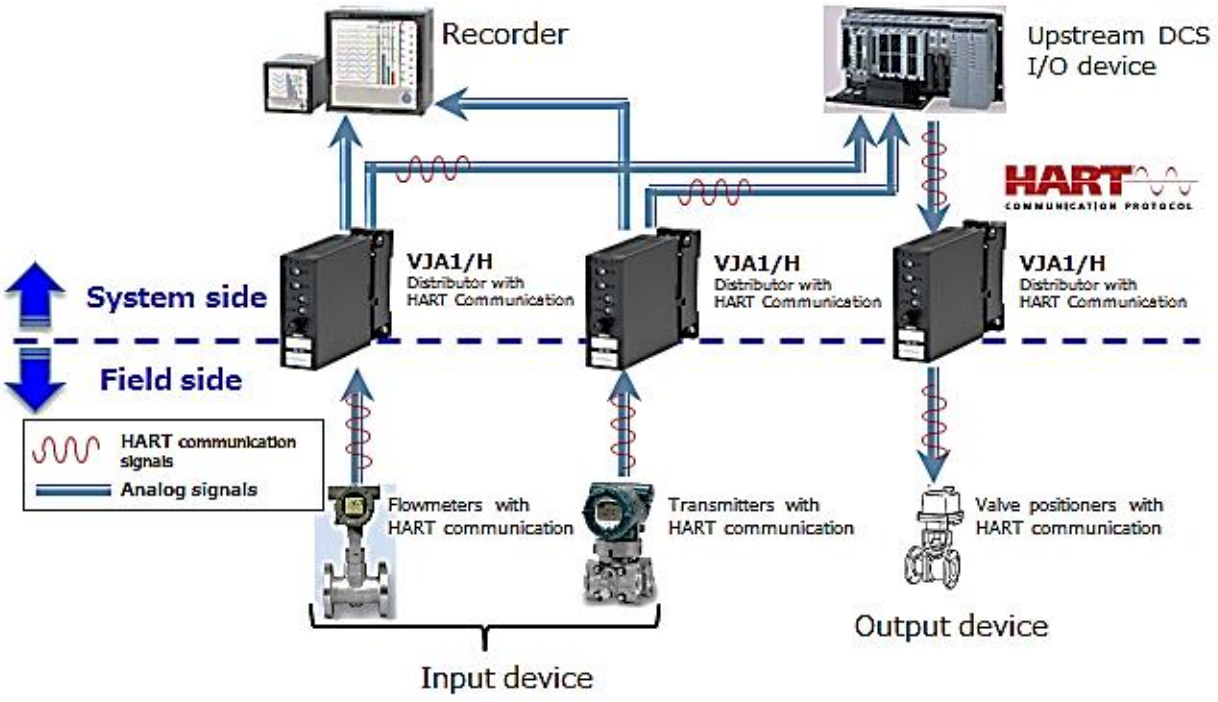


Fig 1.27: Field to DCS communication using HART protocol (taken from [25])

Wired Digital communications using known modulation schemes follow the second field communication category. In this type, measurement as well as diagnostics data are transmitted among the same bi-directional single point channel line to/from the system (see Fig 1.27). In other words, field communication is accomplished by digital modulation techniques which allow transmitting analog (measurement or actuation) signals carrying digital data. This implies that all field devices utilizing this kind of communication should include ADC/DAC and CPUs internally. The most popular protocol of this type is the HART standard that turned the conventional 4 – 20mA described earlier into FSK modulated signals where the higher frequency signal represents the digital data whereas the analog measured or manipulated one is the lower rate. This digital 0/1 data is represented by 2200 Hz and 1200 Hz, respectively, where the

amplitude of the digitally modulated signal is kept between -0.5 to 0.5 mA in the physical channel layer of the standard as appears in Fig 1.28 [27].

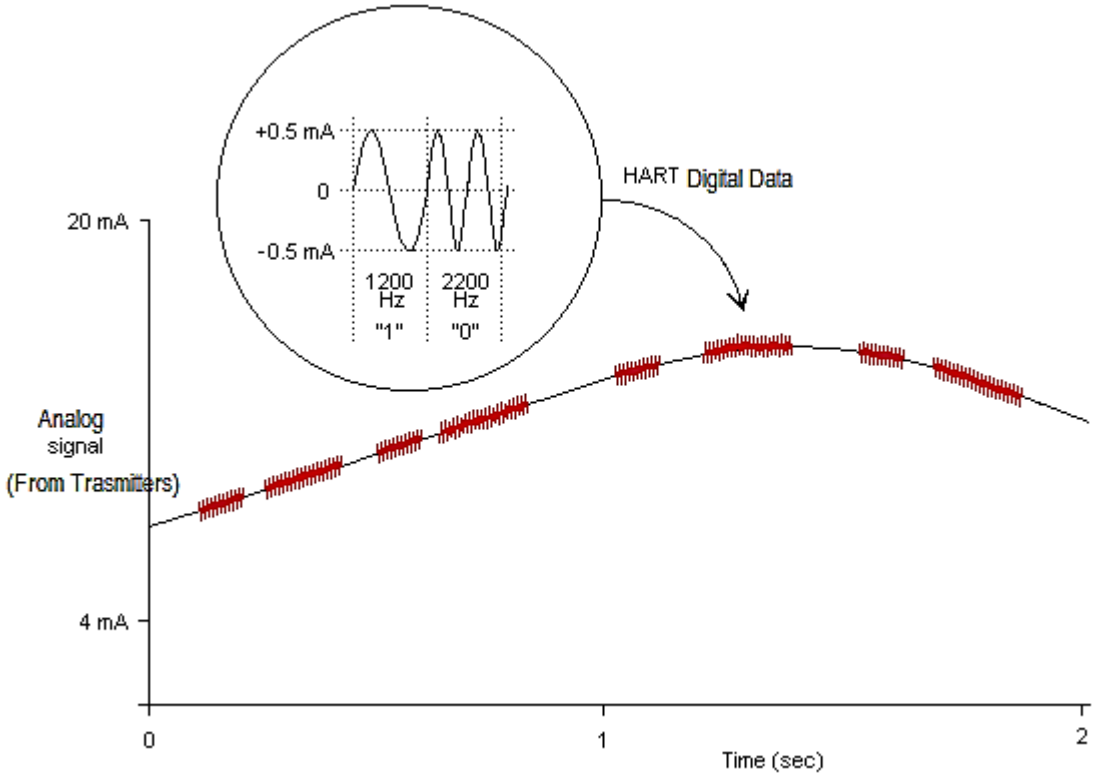


Fig 1.28: HART signal over physical (channel) layer

Describing the protocol itself, HART realizes data link, network, transport, and application layers on the top of the physical connection explained above. In total, the HART standard utilizes layer numbers 1, 2, 3, 4 and 7 from the standardized OSI model. Thus, besides HART's physical layer, the DDL follows master/slave communications where the system, frequently DCS, represents the first while the field devices do the latter. Similar to most protocols, network layer functions as a router, security manager, and end-to-end session operator, whereas the transport layer ensures the operation is healthy and is done successfully. Finally, the application layer defines the protocol's components itself such as types of commands, responses, data, and status. [27], [28] give more information about different command types and their usage defined by the

HART protocol. In general, a single HART digital data frame looks like the one that appears in Fig 1.29 where the message is divided the way: 5 and 20 bytes of hex FF preamble, a 1 byte start character, 1 or 5 byte source and destination address, 0 to 3 bytes called expansion, a 1 command byte, byte counting for status and indicating data of another 8 bits, a status response of 2 bytes from the slave, the sent/received data of 0 to 253 bytes, and finally a 1 byte checksum to detect errors [28]. This digital messaging HART protocol reaches a data bit rate of 1200 bits/s [27].

Fig 1.29: A HART message frame

Although this type of field communication is smart as seen above compared to the conventional analog, this approach is still considered half digital or in other literature is given the name "hybrid," which addresses several major limitations as:

- Offers limited communications compared to intelligent devices (full digital technology)
- Field devices are remotely configurable through the host application layer; yet, this feature is limited to vendor specific field instruments not universal to any device recognizing HART. Also, the data rates relatively tend slow in using this approach.
- The field remains depending on the system (controllers). In other words, if the system (DCS) goes down, the field will shut-down over this communication method although the devices (sensor transmitters/actuators) seem smart. This limitation is because of loop wiring and powering which remains conventional in this method. Besides, the limited digital access inhibits HART's, or other similar protocols', devices to execute control.
- This method still considered as point-to-point (single variable) wiring approach which sustains the extended cabling cost, time, and maintenance dilemma.

1.2.1.3. Full digital two-way multi-drop (fieldbus)

The most advanced wired field communication category is the fieldbus technology. The term Fieldbus in industrial automation environment can be technically defined as a full digital two-way multi-drop communication system for instruments and other plant automation equipment. Fieldbus is mainly applied though in the field communication level of the industrial automation/communications architectural structure. Looking at Fig 1.30 (a) and (b) below, one could notice the immense change that fieldbus has offered over the conventional methods since no more single-point connectivity is applied. Rather, several field devices can be connected to only one wiring bus, which saves cabling as well as engineering and maintenance cost [23].

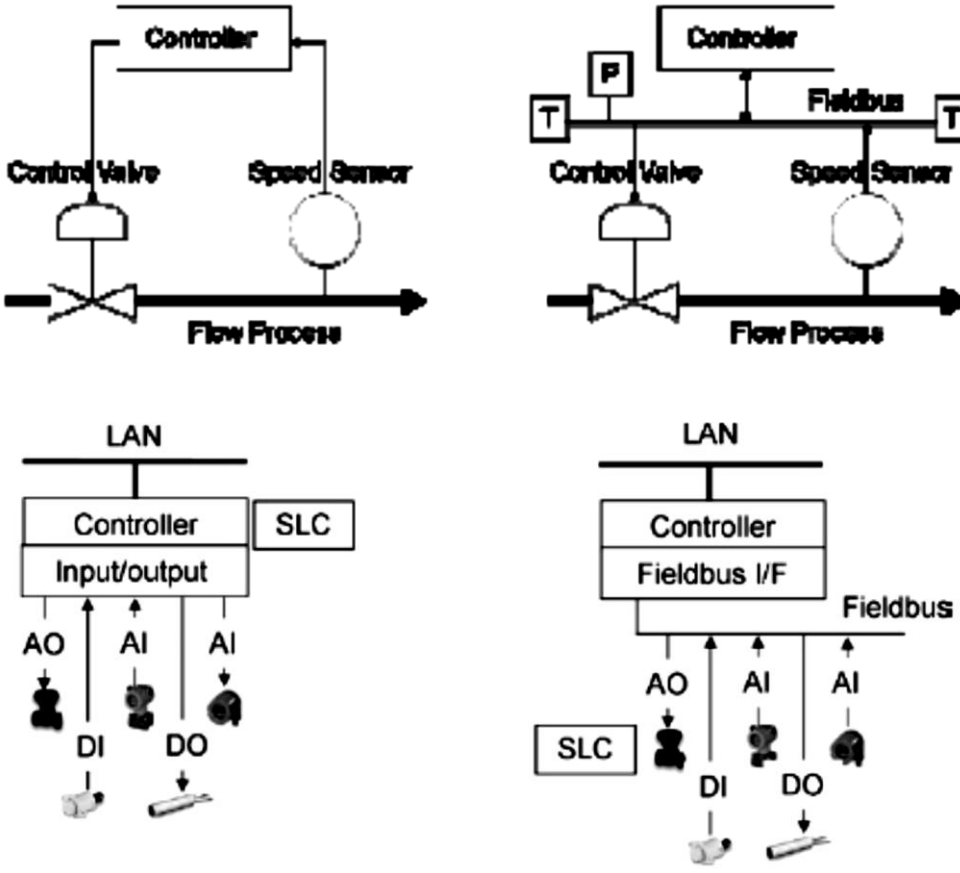


Fig 1.30 (a) (taken from [11]) and (b) (taken from [23]): The difference between point-to point and fieldbus wiring.

Not only does fieldbus require field instruments to be smart but also intelligent. In other words, fieldbus networks use digital transmission instead of digital modulation schemes utilized by smart protocols, i.e. HART. As a result, even the physical channel layer of the communication stack transmits pure zeroes and ones and not modulated data. This implies that full digital communication between the system and the process plant can be achieved; therefore, field devices are completely and remotely programmable, independent of the control system since digital control program data is downloadable into the instruments' CPUs, and can provide more diagnosis and preventive maintenance information [1], [27], [23], and [26]. There appear several protocols, trademarks, and brands that follow the fieldbus technology standards such as F.F, Control-Net, Device-Net, CAN, ASI, Profibus [1], and others as in Fig 1.31.

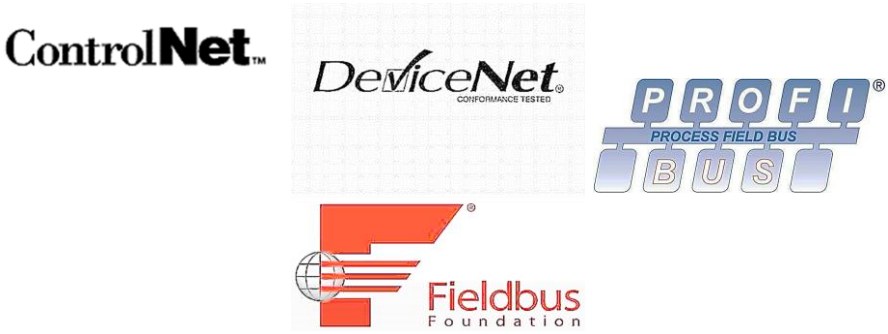


Fig 1.31: Several fieldbus protocols logos and brand names (taken from [1])

How data is being digitally transmitted over such a bus depends actually on the protocol applied. This paper considers F.F. as a model example for all other fieldbus protocols as it is the most famous and widely used worldwide whereas in some parts of the globe, i.e. Europe, Profibus perhaps appears more often. The F.F. physical layer consists of nothing but a two ends terminated twisted pair bus cable that contains maximum number of multi-drop field devices where digital data communications are established by the mean of "Manchester Coding" technique. The physical structure should also include what is called F.F. power supply [29].

In general, F.F protocols are further classified into two sorts:

- F.F-H1 protocol: this type realizes IEC 61158-2 in the physical layer [26], and it is implemented as the powered loop twisted pair copper wire cable revealed above. The standard F.F. is mainly used as field instrument bus while it can connect a minimum of 2 and maximum of 32 nodes (devices)/16 per segment. The bus length may reach up to 1900m, and the data rate is 31.25 kbps. Since the baud rate is not very fast, this type of F.F supports intrinsic safety functionalities only, in case of immediate response for emergencies. Also, as an advantage over HART, F.F-H1 is a universal open standard protocol so that it is vendor independent [26], [27].
- F.F-HSE protocol: is mainly established for connecting sub systems such as PLCs, analyzer shelters, other remote I/O systems as RTUs, and ESDs as a part of the field bus. It runs at 100 Mb/s or higher using fast Ethernet physical connection. It offers more safety features as it is redundant. It is also universal and vendor independent [26], [27], [45].

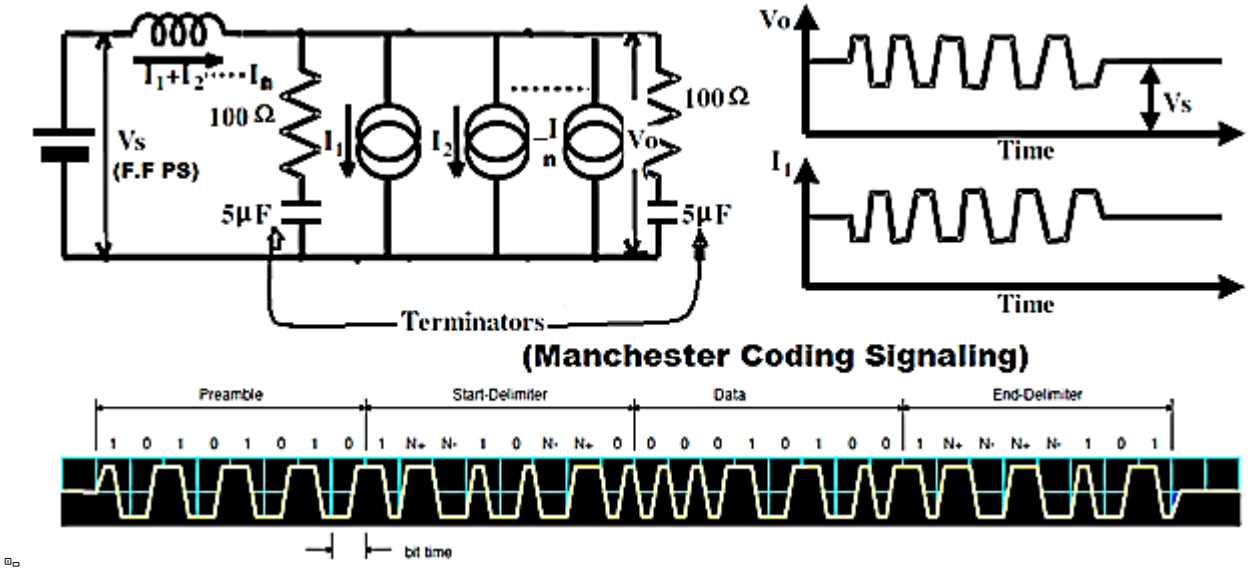


Fig 1.32: (a) F.F. PHY layer circuitry equivalence, (b) signaling method using Manchester coding, and (c) message frame in the DDL

Overall, Fig 1.32 describes the physical layer's circuit implementation in F.F. where the bus can connect n ≤ 32 number of nodes (field devices) and the signaling method is the bi-phase Manchester serial pulse transmission coding [29]. The field devices function as current sinks in the circuit, so, when they transmit data, they draw less current, 15 mA, representing the high signal or 1. Inversely, low signals (0, zero) are generated by drawing a higher current, 20 mA. The resulting output voltage range is then 0.75 V and 1 V P-P as a result of applying Ohm's Law:

$$V(\text{low}) = (15\text{mA})(50\ \text{Ohm}) = 0.75\ \text{V and } V(\text{high}) = (20\ \text{mA})(50\ \text{Ohm}) = 1\ \text{V}.$$

As a result, the voltage output signals are 0.75V/1V P-P. However, in Manchester coding, high peak does not necessarily mean binary Boolean equivalent 1 and vice versa. This method is known as non-return to zero, which suggests that high peak decodes logic 1 while low-edges denote 0s produces a drawback that the voltage on the line may remain constant if transmitting long sequences. Manchester coding, in contrast, overcomes this problem by decoding and transferring low edge to high as 0 and vice versa. This way a change in voltage per decoded bits can be assured [27]. The bus should have two terminators at two ends in order to work appropriately. In Fig 1.32 for example, the terminators are the equivalent of an impedance of 5uF capacitor and a 100-ohms resistor connected in series. Pushing the current flow in the bus arise as the main purpose of the terminators; they function also as protection against electrical reflections [29].

For best digital data rate performance, F.F standard suggests two rules. First, the number of devices on a fieldbus can vary between 2 and 32. Second, the cable as well as the spur length primarily depends on the number of devices connected to one segment. As a result, the optimal number of devices that can persist as acceptable communication speed as well as cost saving is 8 per segment. If exploring F.F as a whole communication stack, the protocol calls layers 1 (PHY), 2 (DLL), and 7 (application) from the standard OSI model. F.F, on the other hand, adds an

additional layer on the top called F.F. application. Other than the physical, F.F DLL consists of a subset of type 1 in IEC61158-3/4 in addition to ISA S50.02 part - 3/4. To summarize, in every protocol in general, DLL primarily transfers data from nodes to others. DDL functions also as a priority manger. Likewise, F. F's DLL has four main functions:

- MAC: it is also considered as a sub layer under the DLL. In F.F, MAC performs TDMA to insure only one device on the network is allowed to transmit signals at a time since multiple users share the same bus medium [27], [45].
- LAS: DLL under F.F acts as LAS to control the medium access. This step is accomplished by the concept of "token passing". The term "token" can be described as the right to send a data unit where the LAS owns this right and then passes it to another device to allow it to send messages. After that, the token then returns to the LAS for further MAC [27].
- Priority organization: F.F DLL defines and manages priorities [27]. Upon the case and the function, F.F protocol outlines three main priority levels: urgent, normal, and available.
- Node addressing: each device on the bus should have its own address between 0x10 and 0xff hex range. As per device type, whether a link master, default or temporary, the address range is assigned.

The application layer on F.F follows three models: connecting clients (host computers, DCS, users) to servers where the first request data and the other responds, publisher subscriber, and source link models. F.F software application stands on top of all the above where the programmer can locally or remotely interface with blocks inside sensors (devices). Inside each F.F device, exit three block types: a resource block which describes the characteristics of the device as DD files, a transduces block, which represents local connections with RT physical I/Os, and finally function blocks which are downloadable and provide the control of the device as AI, AO, PID, etc.

1.2.1.4. WSNs

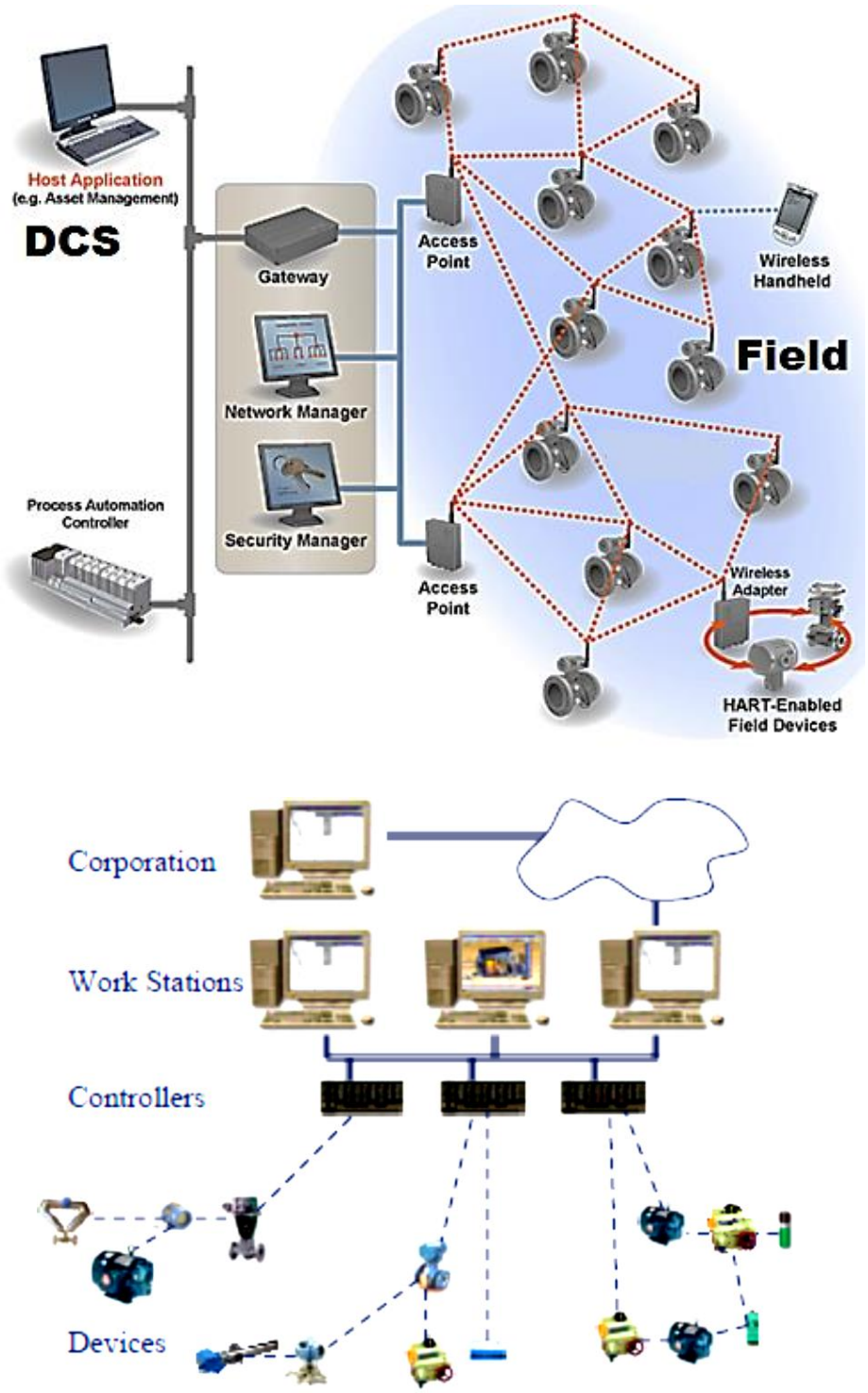


Fig 1.33 (a) Typical WSN components and architecture in industry and (b) WSN as level 1 while DCS represents level 2 in the modern industrial architecture (taken from [10])

WSN is emerging as the newest field communications technology in industrial automation networks. From its name, WSN replaces all the above, conventional, smart, and intelligent field communications with a wireless RF networking medium. In other words, sensors transmit measurement and manipulation signals data wirelessly to/from DCS or any other automation control system architecture. Fig 1.33 (a) and (b) illustrates a typical industrial WSN architecture and components and such technology as a part of DCS and higher plant's web levels. As seen in (a), typical WSN architecture consists of several components [7] and [35]:

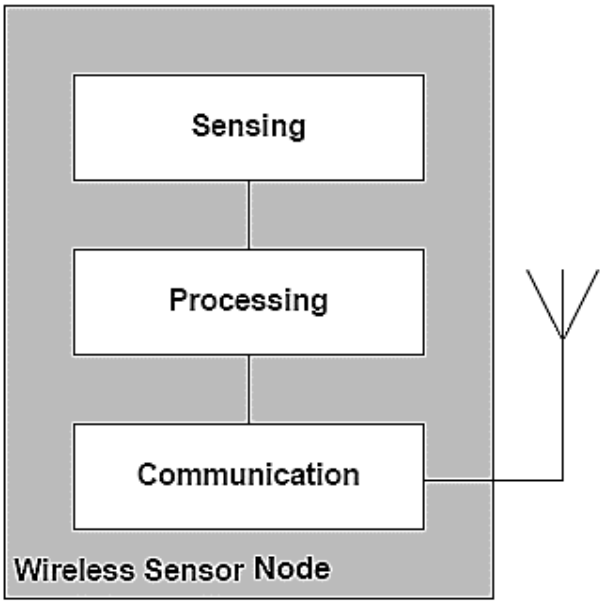


Fig 1.34: A typical WSN node [9]

- Wireless nodes [30]: WSN nodes are nothing but the field devices that provide RT measurement and other types of data to the system wirelessly. They are referred to as nodes because they compose the WSN PHY layer elements. A WSN node further consists of a number of main subcomponents [9], [32], and [33]. First, the actual sensor(s)/actuator(s) (WSAN). Second, in some cases where sensors are wire based, then wireless adaptors in between become required. After that emerges the PSU, mostly long life batteries. Then, a processor unit as a CPU or low power microcontroller as a part of

the field device contains part of the node. Also, the node should include AIs and ADC to interface with physical sensing elements. Finally, any WSN node should embrace communications and RF circuitries where the actual wireless physical connection of TX/RX besides modulation and antenna are all implemented. (See Fig 1.34).

- Gateway: the major functionality of the gateway on WSNs is to gate pass or join the field wireless network to the wired LAN CN, usually Ethernet based, as declared from the figure. As a result, the host application software such as AMS, which is usually part of the DCS can have access to the device and vice versa. Multiple gateways can be used in a WSN depending on the preconditions and configurations [30].
- WSN access points, routers and handhelds: WSN can be configured following various network topologies. For example, in the mesh topology, routers in the field are needed, while in the star topology the gateway acts as the only router in the network. Fig 1.35 below shows the major three to four WSN configuration topologies and their routing differences: star, mesh, and tree, besides peer-to-peer [14]. Each topology has its advantages and design tradeoffs as per power consumption, performance, latency, etc. [34], and it is up to the designer to select the best topology for his/her application. Access points are nothing but antennas located near to the system side interfacing the filed network with the gateway. In many scenarios, access points may represent part of the gateway physically. Moreover, from networking point of view routers can be realized as field devices (nodes); the only difference however is that the first do not interface with the process, the second do [31]. Routers work also as repeaters in such networks [7]. The handheld devices can provide local access to any WSN node field instruments manually, and they can be considered as part of the network components [35].

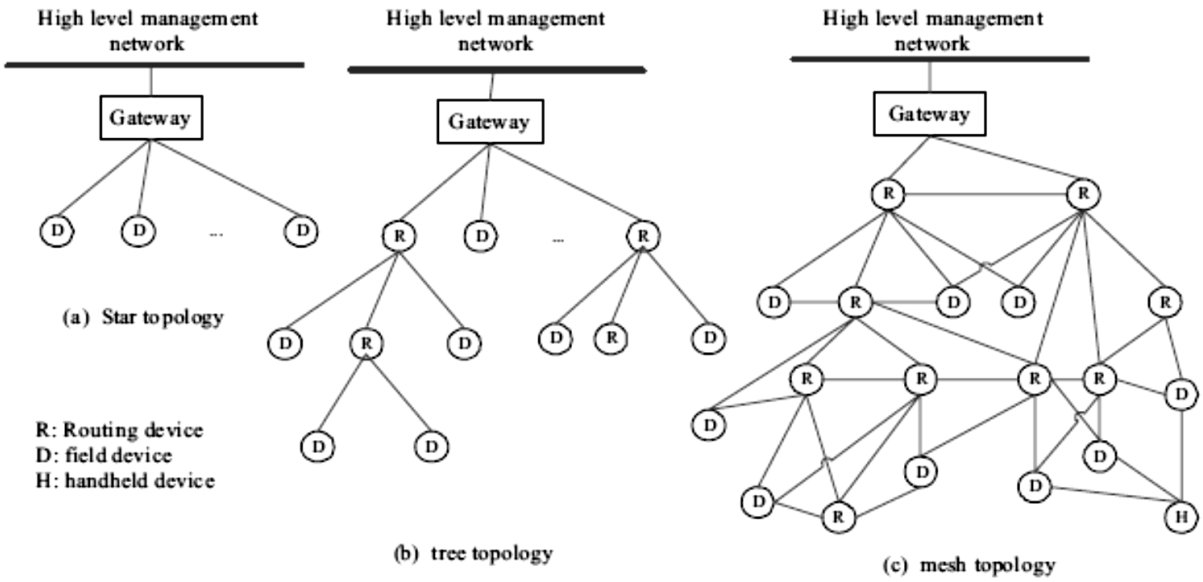


Fig 1.35: Different WSN topologies [14].

- Network manager: is responsible for coordinating the network such as scheduling, message verification, and buffering [30] and [35]. The network manager forms and manages the network topology such that the optimal routing and paths to end nodes are ensured. It also divides network assembled data into time slots and super-frames [20].
- Security manager: is repressible for the end to end communication and network security by allocating security keys, and it has the authority list of devices to join and link into the network [20] and [35].
- Host application: is the top level software component where engineers, operators and end users can have access to, remotely configure and analyze field wireless devices to collect data and read process measurements following vendor specific tools [30].

Usually, the network and security managers are all integrated along with the gateway as a whole physical unit. However, the above network architectural components may vary according to which protocol is applied in certain applications. The coming sections demonstrate more scientific and technical study of the industrial WSN standards and different emerging protocols.

Other than industrial field communications, WSN is found to be useful in several other applications such as building management systems and home automation, automobile, biomedical instrumentation and health monitoring, other service and commercial applications such as construction monitoring, and finally security and governmental defense field applications. As the focus of this paper is to examine and study industrial applications, WSN aids and is found applicable in various plant automation tasks such as:

- Remote site RT PVs measurement and monitoring [4] and [36].
- Analytical monitoring such as gas chromatograph, PH, air and water quality, etc. [36].
- AMS and condition based maintenance in parallel with RT monitoring [36].
- They can be used also to gather data about the overall plant production to higher level criteria for manufacturing planning, optimization, and ERP [36].
- Process control: not only does WSN provide field measurements, but also is capable of performing RT control [4]. However, although the advantage WSN may provide, due to the harsh industrial environment, therefore high requirements, this application is not known to be officially adopted by plant owners and end-users. Hence, there is still ongoing research regarding evaluating, proving, and studying the validity and capability of different WSN protocols to execute acceptable RT control performance, especially regulatory. Saying that suggests discussing WSAN which is referred to in this research.
- On the other hand, WSN is not suitable for some other process control automation applications such as safety interlocks and ESD. This is because of the immediate action response requirement of the ESD applications that does not tolerate delay time associated with wireless transmission [4].

[4], [36] and [37] list and sort possible advantages of replacing WSN as an alternative field communication. Common supposed benefits are:

- Satisfies the demand of numerous available remote field devices where wiring is difficult and costly.
- Offers a very high level of flexibility since no wiring is involved. Consequently, installation and relocation are easy.
- Minimizes automation project execution cost as a large amount of engineering time and work is usually devoted wiring side of instrumentation that requires cabling, connections, I/O, cabinets, and marching. In contrast, all WSN field devices may connect to one gateway node wired to the DCS.
- WSN technology may also help in advancing applications such as condition based monitoring maintenance, AMS, and interface with higher corporate levels, i.e. ERP and production planning.

However, there appear a number of unaddressed issues and challenges associated with WSN in industrial environment [4], [32], [33], and [37]. Those issues have kept specialized R&D departments, research, and academia busy attempting to answer open questions and find solutions and improvements to such problems summarized, but not limited to, as:

- Latency and delay: this challenge matters a lot when considering closed loop control.
- Safety and security: since its wireless, WSN stands subjected to external hacking dangers.
- Power: increasing network and nodes performance raises power consumption and vice versa. Thus, there are always design tradeoffs between activity, speed, and power saving.
- Noise, interference, and other wireless communication and RF issues: aside from being AWGN, a wireless channel is always subjected to fading and multipath effects.

1.2.2. Control network

CN simply forms the DCS nodes of controllers and workstations managed by one switch regularly. The controllers are usually located in special shelters nearby their process areas called PIBs or in other naming philosophy, RIBs. Each RIB typically carries the set of controllers needed to handle a single process area while mainly each controller is designated to control and take over I/Os of a whole unit, if possible. Figs 1.21 and 1.22 discussed in Section 1.2 show the difference between field, loop, units, area and plant. On the other hand, workstations are normal PCs that are located in the CCR where each one has different authority, security and access levels, and core functionalities. For example, one node may work as the master node on the network carrying the software database and logics and does functionalities as engineering and configurations; trending and alarms and events; and tuning and operation, besides graphics and display. Other nodes may have engineering access where a DCS engineer can configure control logics and download to controllers, whereas some others do not have that authority as they serve as operator workstations and so on.

From communication point of view [45], DCS CN is typically implemented as a redundant LAN network. Its switch is considered as layer 2 if accepting field networks, i.e. F.F, as layer 1. This LAN regularly takes the form of an Ethernet utilizing TCP/IP protocol on the top. In other words, CN is nothing but a LAN that follows IEEE 802.3 standard in the PHY and MAC layers where the physical connection is an Ethernet copper, or converted to fiber optic, cable. The MAC is implemented by CSMA/CD technique while the network layer follows IP routing and addressing at the switch level. These switches, usually dual redundant, use node (hosts) IP addresses to control the routing of the network by assigning data packets from sources to destinations. Thus, IP identifies host nodes on the LAN and then provide logical location service.

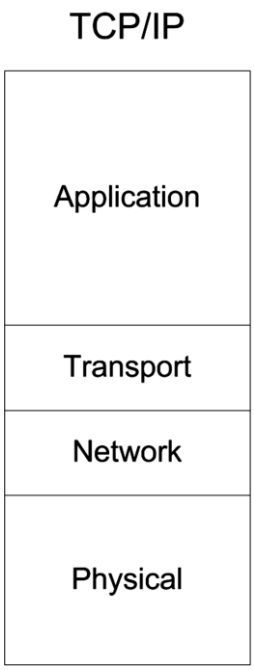


Fig 1.36: TCP/IP over LAN (taken from [38])

In general, the network layer provides packets routing and addressing. The transport layer on the other hand stands responsible of the end to end, node to node, or host to host communication. Within TCP/IP based LANs, TCP forms the transport layer. TCP then allows two controller nodes, for example, to create links, therefore interchange data streams. TCP ensures also the health status and reliability of data communication between host nodes (controllers and workstations). Fig 1.36 shows the layers realized by the TCP/IP LAN based CN [26]. Most modern DCS LANs utilize the above IEEE 802.3 and TCP/IP; however, depending on the vendor specifications sometimes, CNs may take a different version or protocol where controllers become separate from the Ethernet network of workstations. As a result, two sub network levels can be extracted from such separation: controllers' network (control level) and workstation Ethernet LAN (SCADA and HMI levels) as appears in Fig 1.37 below. Note that, although it has not been officially implemented yet, CN can be wireless as a WLAN as [38] and [5] suggest. More details about this point are given in the coming Section, Wireless Technology.

1.2.3. Higher-level networks hierarchy

Any communication level above layer 2 switch of CN can be considered as higher level. The architecture usually has a midway network level on the top of CN sometimes named PCN. It sometimes includes computer nodes that execute advanced process control and communications functionalities such as historians, data logging, and OPC server that connects 3rd party subsystems such PLCs, analyzers shelters, and SIS/ESD, in addition to some advanced applications software as AMS server and preventive maintenance. This level should form that intermediate interconnection that closes the gap between industrial automation networks and other production planning and IT ones, explained earlier, so functionalities as optimization and APC are done at this stage also [15]. Inter-zone communication that allows other expanded CNs to talk with each other, in case more than one DCS zone are located at one plant site, is also included in the PCN.

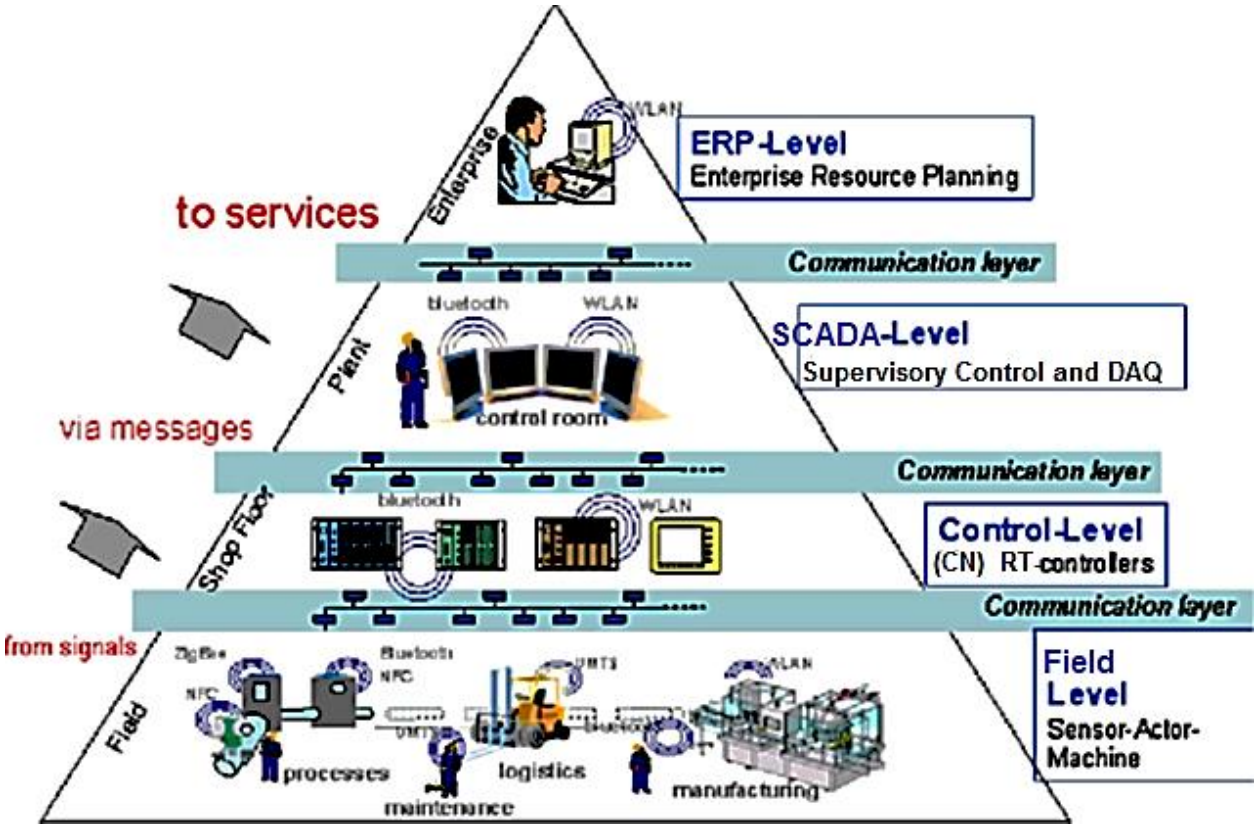


Fig 1.37: Industrial Automation and plant's networks levels architecture (taken from [5])

The higher level networks above the PCN may include production planning, ERP, and the corporate IT networks. They can be all integrated into the IT network where the main functionality, aside from being the company's intranet, is maintenance, production, scheduling, and human factor engineering planning and control. Fig 1.38 simplifies the overall image of different interconnecting industrial automation and plant networks from lowest to highest levels including layer switches.

1.2.4. Subsystems Communication

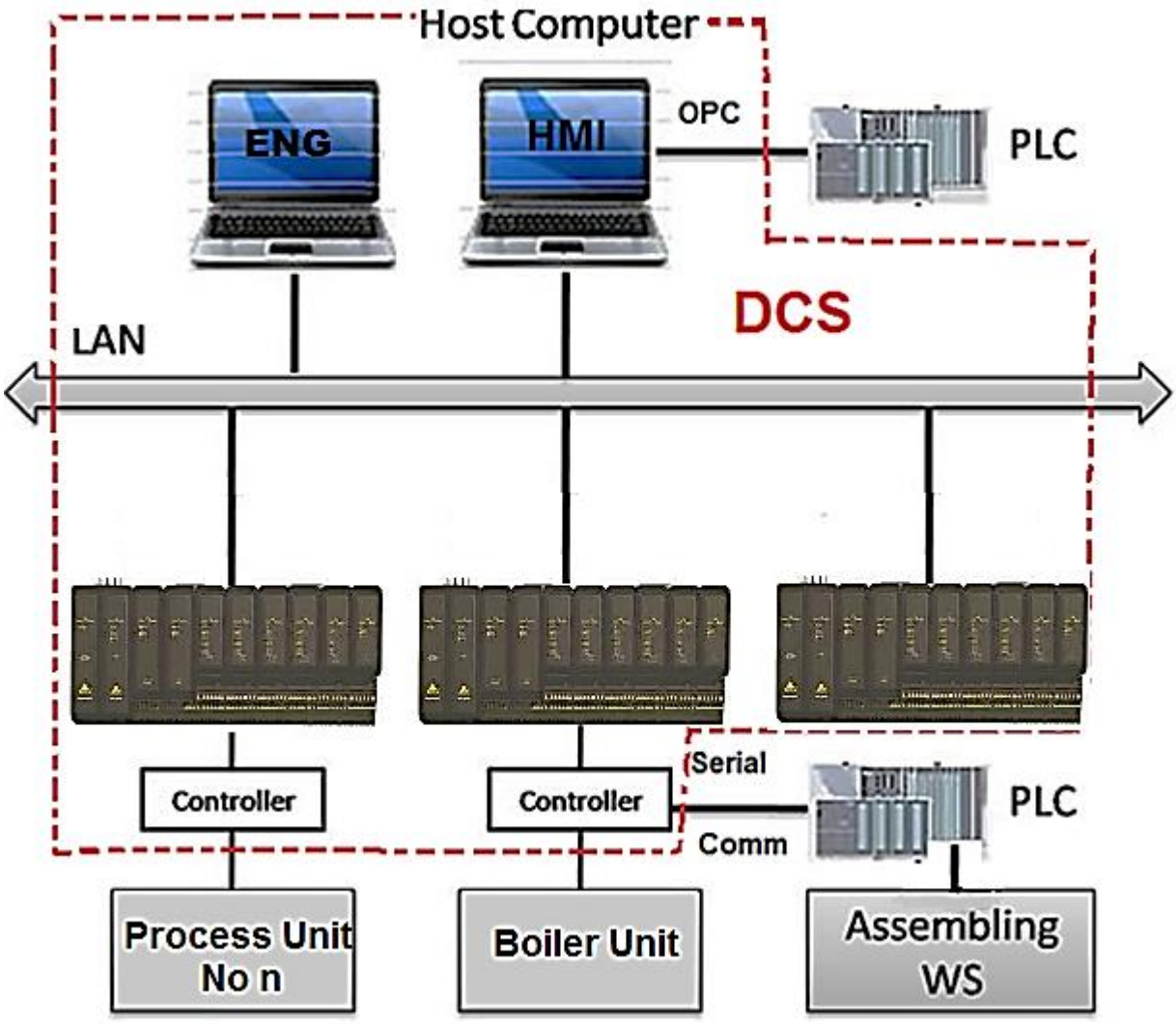


Fig 1.38: 2 ways of communicating master DCS CNs to 3rd party subsystems PLCs on the PCN.

Fig 1.38 above shows the difference between the master system, DCS, and other 3rd party subsystems communicating to it. As introduced before, there exist numerous machine controllers, discrete devices, and packages that are either accompanied with their own controls or require PLCs due to their sub process environment. In addition, SIS, which takes care of ESD mainly, is a parallel system to DCS, which does the process control tasks primarily, in most plants. Those subsystems as a result need to communicate with the master system, DCS, to form the integration. This type of process of integrating other vendor systems to the master one, DCS, is conservatively called subsystems communications. There exist three major possible ways to accomplish subsystems communications:

- A conventional single point H/W wiring: this method is the simplest as explained in the field communication before, but it requires a single wire per single variable or parameter communicated. So, the method is efficient yet still expensive and provides limited communication.
- Serial communication
- OPC

1.2.4.1. Serial communication (Modbus protocol)

Unlike parallel transmissions, the process of transmitting digital data serially, i.e. one bit at a time, on a sort of a physical cable (channel) or a computer bus is known as serial communication. Parallel communication, on the other hand, requires several channels sending data at the same time on shared or multiple physical links. There exist a number of PHY standards that can carry serial data such as Ethernet, SPI, RS232, RS 485, and so on. Serial communication may accept numerous higher layer protocols on the top, however; the most common utilized in industry is the

Modbus protocol. [39], [40] provide a thorough overview description of the protocol over serial line and TCP. That said, one could further divide Modbus into two major protocols:

- Modbus over serial line.
- Modbus over TCP/IP (Modbus TCP).

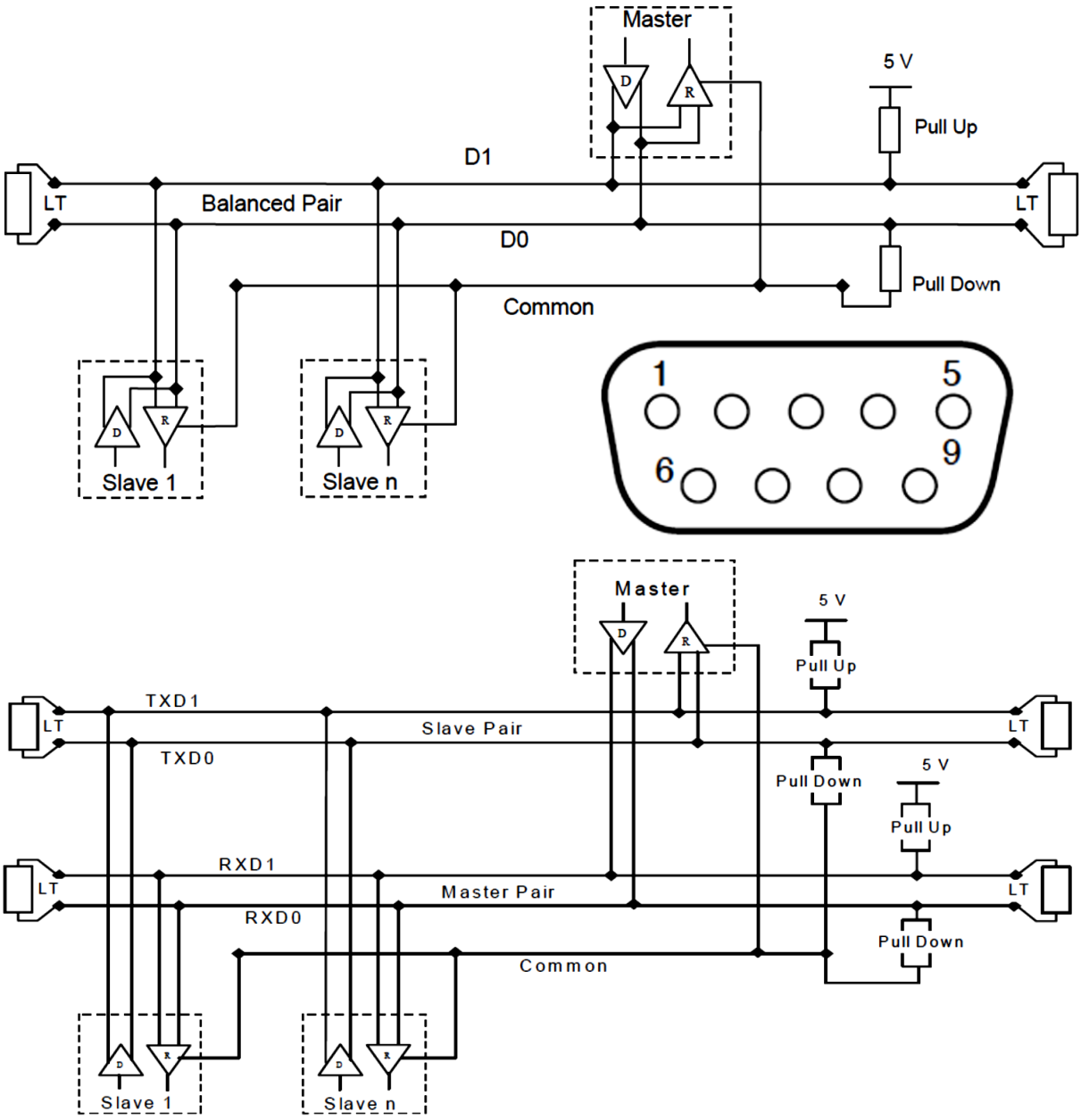


Fig 1.39: (a) The 2-wire (half duplex), (b) the serial bus (RS) connector, and (c) the 4-wire (full duplex) serial PHY layer specifications.

Modbus over serial line realizes layers 1 (PHY), 2 (DLL), and 7 (Application) from the standard OSI model. The PHY follows either EIA 232C or EIA 485 standards. More specifically, Modbus over serial line protocol can either have RS232 or RS485 physical connections between communicated controllers. RS 232 can be either configured as 3-wire connection (TX, RX, and GND) for two-way communication or as a 2 wire where one-way data transmission appears abundant. Similarly, RS 485 composes two connection configurations: 4-wire and 2-wire networking. The 4-wire mode allows full duplex communication where both ends can talk (both send and receive data) simultaneously by consuming two TX (inverting and non-inverting), two RX pins, besides the GND. On the other hand, the 2-wire mode is called half duplex since it allows one data direction at a time [39], [40].

Fig 1.39 illustrates from top to bottom: (a) the 2-wire (3 wires) connection, (b) the typical RS bus connector with 9 pins, and the 4-wire (5 wires) mode. The data signaling baud rate of the PHY layer is configurable at the application level; a typical recommended value may reach 9600 bits/second, and as any other communication link, there exist a maximum cable length depending upon the bit rate. For example, for the regular 9600 b/s rate, the maximum length can reach 1km [39]. Moreover, as it is for fieldbus, the serial bus should be terminated to eliminate the reflection that travelling waves may encounter when propagating in the transmission line as an effect of impedance discontinuity. The figure shows the use of pull up/down resistors that help in maintaining the right polarity of the signals [39].

Modbus over serial line DLL has two sub layers:

- First, Modbus master/slave protocol layer: at Modbus Serial Line DLL, communication is established though Master-Slaves protocol where only one master, at an instance, i.e., DCS, can be connected to the bus while it can have more than one slave (maximum of 247

hypothetically) [39]. Being the master, DCS leads communication by first sending requests to slaves, for example, PLCs. Then, slave nodes start sending data; they also do not have access to each other at all. Two different master/slave request modes occur: unicast and in broadcast modes. In the first the master node (DCS) identifies a specific slave address (from 1 to 247) then sends the request as a message. After that, the slave replies by another message. In the broadcast mode, the master system sends requests to all available slaves that they accept where no reply is required. This process is essential in establishing the communication commands, and the remaining address (0) is reserved for the initialization step [39].

- Second, Modbus serial line protocol's transmission mode: similarly, this sub-layer can have either one of two protocols or modes: Modbus RTU or Modbus ASCII. In the first, each 8 bits (1 byte) in the Modbus message consists of 2 bit hexadecimal characters while in the other, bytes have the form of ASCII codes, as can be seen in Fig 1.40 below.

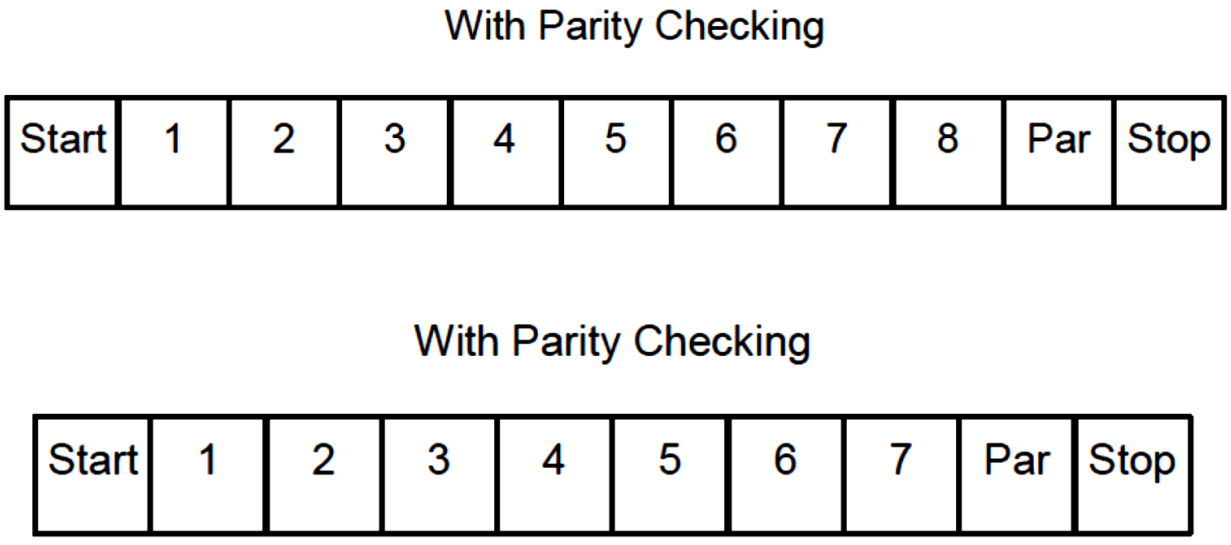


Fig 1.40: (a) Modbus RTU data byte (upper) vs (b) Modbus ASCII's formats (taken from [39])

General byte formatting takes a form of data bits preceded by a start bit and followed by party (odd or even) and stop ones as appear in the figure. Serial spreading from LSB to MSB sequential orders forms the transmission and identification of each byte. Overall, Modbus application protocol on the top of the DLL verifies data units such that they contain an address byte and function byte, in addition to data bytes, (minimum of 1 and maximum of N=252), described above. Fig 1.41 shows a Modbus RTU data packet frame. Besides the protocol data unit that includes the three fragments mentioned, a Modbus message frame (packet) comprises of 2 bytes (16 bits) error checking field as per 2 different methods: Cyclical Redundancy Checking (RTU) and Longitudinal Redundancy Checking in the ASCII mode. This is excluding the party checking step executed at the end of each 8 bits of data. Finally, each frame is separated by a minimum amount of 3.5 start and end characters to ensure no data communication overlapping occurs, 1 start and 2 end characters in the case of ASCII mode [36].

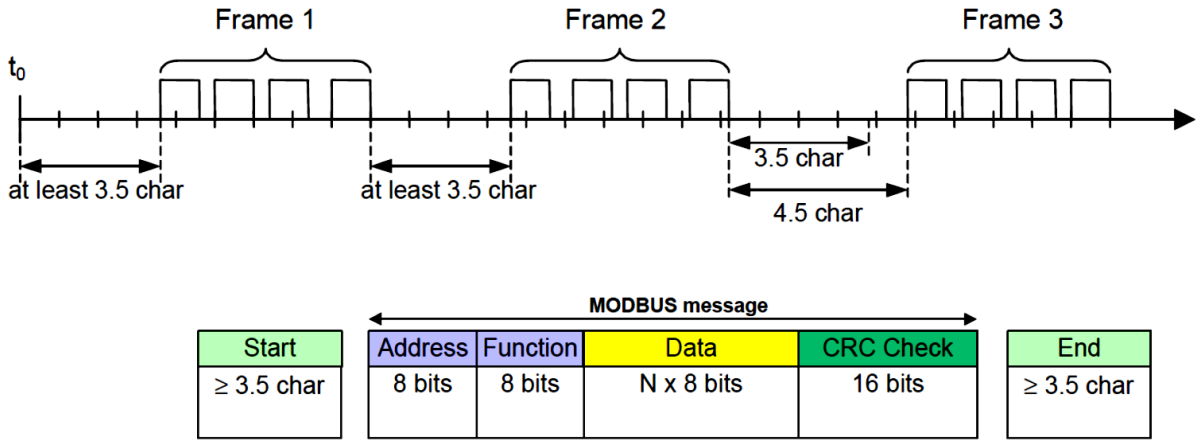


Fig 1.41: Modbus RTU packets (taken from [39])

On the other hand, Modbus TCP/IP forms a complete set of a different protocol. It utilizes exactly the same IEEE 802.3 PHY and data link, network of IP address, and finally TCP transport layers introduced in the TCP/IP protocol above. However, here adds Modbus application protocol header on top of the TCP/IP one. More information about this type is provided in [40].

1.2.4.2. OPC

OPC stands for OLE (a technology developed by Microsoft Corporation for any software computer machine working on windows operating system) for Process Control. OPC thereby is an industrial standard established in 1996 by a professional association (formed by a group of industrial automation and instrumentation besides some software and IT vendors), OPC foundation. The standard forms and preserves open protocol connectivity between different types of industrial automation systems and subsystems, such as DCS, PLCs, SIS/ESD, HMIs, analyzers, gateways etc. [41]. The main purpose behind such alliance that formed a standard group of a universal communication is to satisfy the query demand of communicating between systems, controllers, or subsystems without getting into traditional routine driver-based connectivity issues such as end-to-end communication, particular protocol compliance, I/O loading, different vendors' specifications, etc. [42]. Hence, the basic idea of OPC is moving communication connectivity between systems and subsystems to the software level provided that both ends realize windows OS at certain architectural uniform. As a result, the desired open and common connectivity between different types of systems and vendors can be achieved.

The OPC organizes unit grouping of data communicated in industrial process control environment, [OPC foundation], [41], [44] where:

- Process data: this type of data composes the core business of OPC DA group specifications.
- Historians: OPC historian functions to transport historical type of data.
- Alarms and events: OPC A&E applied for the purpose of transmitting of such data.

Conceptually speaking, unlike other forms, OPC differs from all other types of subsystems communications between various types of control units and systems by offering full exchange of data between sources and sinks without actually knowing about each other's ways of protocols

and data flowing. This can be done through forming a layer between sources and sinks called an "abstraction" which permits two-way communication so that both ends do not talk to each other directly. In other words, sources as well as sinks can communicate only with the intermediate, abstraction, layer which in return controls and allows data distribution to both sides.

This abstraction layer is implemented by inserting two important OPC system architecture S/W segments, which in total form the following basic four components [42], [44]:

- Data sources: are commonly the subsystem, i.e. a PLC. However, a data source is a general name of where the direction of communication initiates, since two-way communication is possible. Sources usually communicate to the abstraction layer through usual industrial communication protocols classified above such as Ethernet TCP/IP or Modbus. The side of the abstract layer that connects to sources is called OPC server described below. Commonly, OPC servers are physically located at the PCN level of the industrial automation communications architecture, so Ethernet TCP/IP based LAN is more likely to be implemented. Therefore, starting from data source and reaching OPC server, four standard ISO/OSI communication layers fulfill: PHY and DLL (Ethernet), Network (IP), and lastly Transport (TCP). However, unlike all other communication methods, the other commutated side does not need to know anything about how data sources connect to the abstraction level, which are the above four layers. This is established by the OPC server.
- OPC server: which is a S/W application, designated to OPC communication specifications. Other than known H/W computers, the term "server" refers to S/W program components presented in a physical network layer observing OPC. OPC server functions as data sources gateway to the abstraction layer. The OPC server can both read and write data to and from sources. Once entering intermediate world of OPC, the relation

between OPC servers and OPC clients takes the form of the stated master/slave relation.

- OPC clients: OPC clients are S/W applications designated to communicate with OPC servers by applying messaging defined by OPC specifications. Abstractly, OPC clients represent masters while OPC servers work as slaves. OPC clients take control over communications with OPC servers based on initiating requests. However, OPC clients need to take requests from the application above them then translate those commands into OPC environment through the concept OPC DA Microsoft's COM/DCOM, explained in [43], and vice versa. OPC Clients are usually installed in workstation applications such as HMIs and SCADA located in CCR that talk to OPC servers at PCN level.
- Data sinks: are mainly applications on the DCS CN network that have OPC clients' S/W installed on them. There exists a procedure of conversion that occurs at this level between OPC clients and those applications to translate between them and the abstraction layer of the OPC specification world. This step is accomplished by the concept of API commutation. Fig 1.42 below summarizes the OPC subsystems commutations.

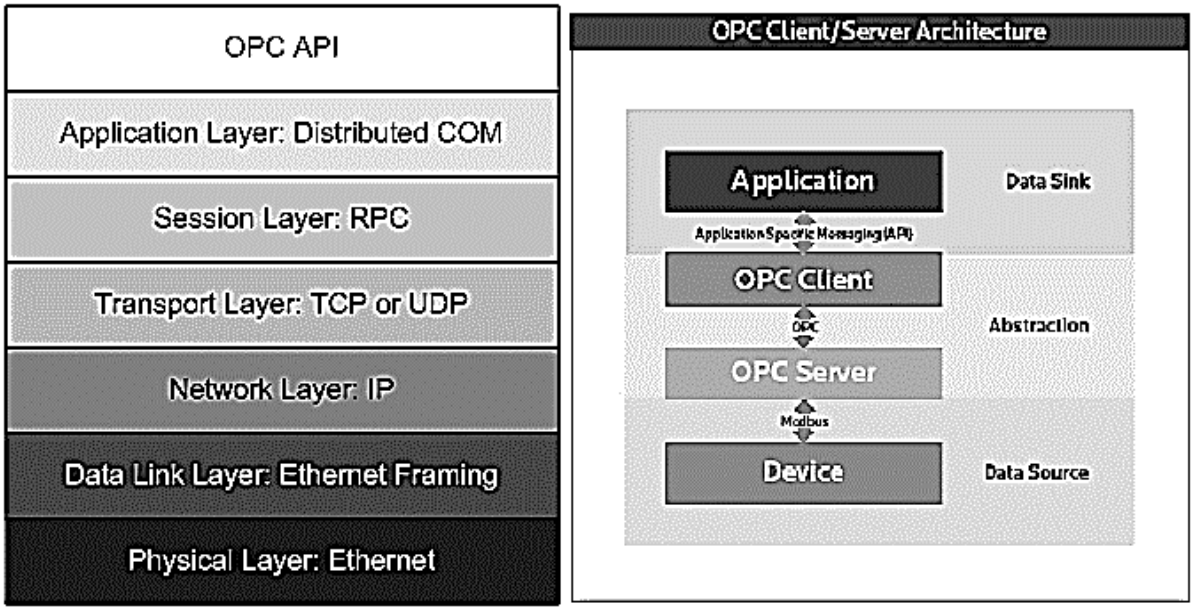


Fig 1.42: Layered OPC commutations (taken from [44] and [42])

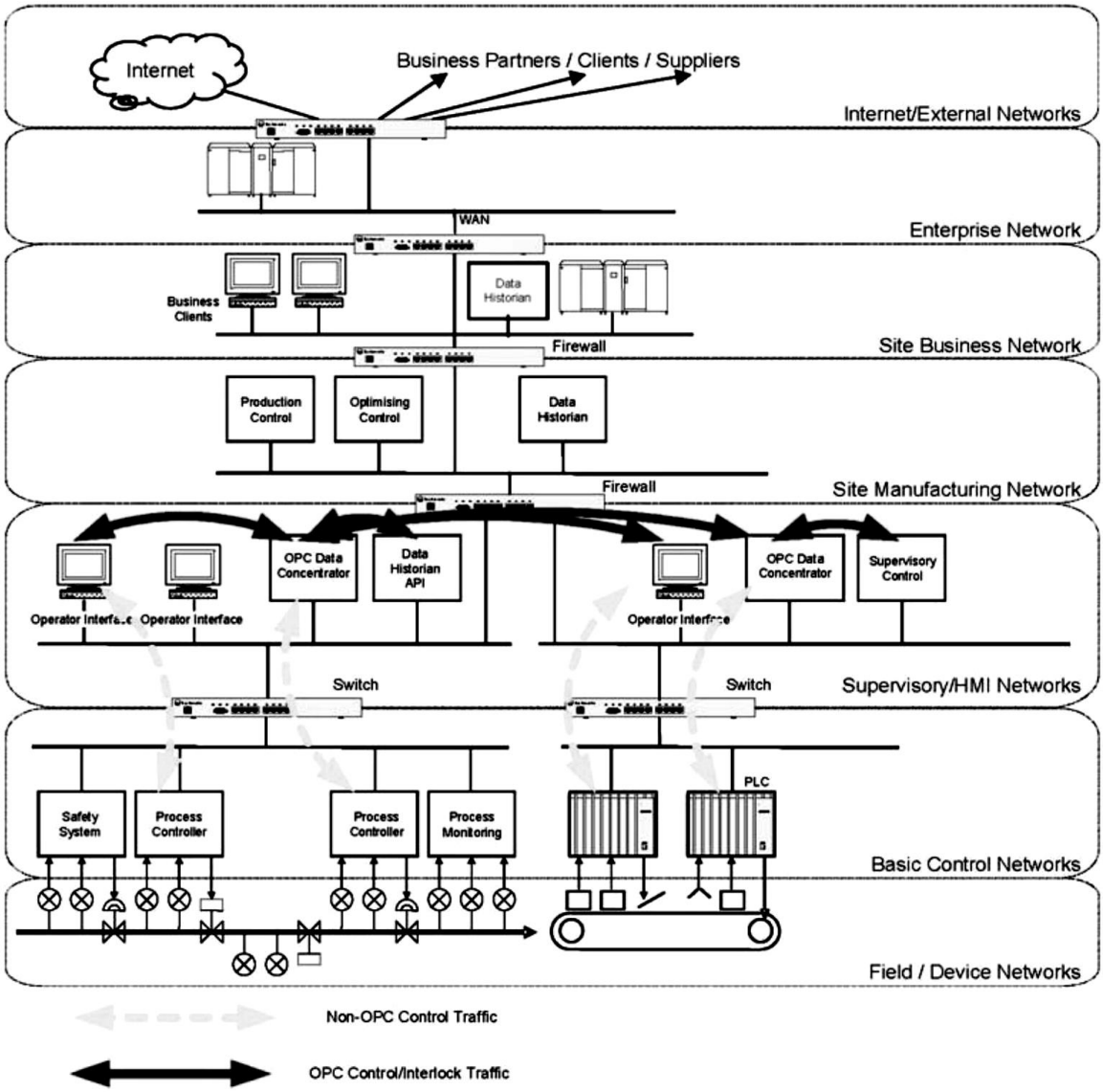


Fig 1.43: How OPC servers and clients fit in the DCS/subsystems CN/PCN architecture (taken from [44])

Fig 1.43 also illustrates how above discussed OPC communication fits into the plant-wide architecture of industrial automation. Moreover, OPC foundation has released developed specifications to the classical one above called OPC UA. It offers several advantages over the classical such as the use of Microsoft's emphasis of SOA instead of COM/DCOM, no more requirement of windows OS, and addition of firewalls security functions [44].

1.3. Wireless Technology in Control and Industrial Automation

Section 1.2.1.4 above briefly introduces WSN in industrial automation. However, when considering the use of wireless technology in industry in general, one could determine that it might go beyond WSN, as a form of field communication, surpassing that of other industrial automation network levels such as CN, PCN, and even higher level architecture besides other applications. [9], [32], [34], [45] provide thorough studies and classification of wireless technology, standards and protocols used and suitable for industrial control applications. This section provides a brief discussion of possible wireless technologies and standards utilized in process and manufacturing industry. In general, industrial wireless standards can be classified as per area of coverage and application into three main categories from wider to shorter: WiMAX, WLAN (Wi-Fi), WPAN, and others [45].

1.3.1. WiMAX, IEEE 802.16

WiMAX follows IEEE 802.16 standard in both PHY and MAC layers [45]. Since it covers long distances, up to 50 kilometers, with a relatively high throughput, of 75 Mbps, WiMAX is thought of primarily as a metropolitan access practice. Another key use of WiMAX is as an expansion to existing network access coverage, for example, remote areas. Regarding industrial plants, WiMAX may benefit in the following two main applications:

- First, it can be used in SCADA network based systems to connect the CCR to RTUs, located at remote site locations. More discussions about SCADA systems and RTUs is given in coming sections.
- Second, WiMAX could aid as a backbone for other RF wireless signals in plants, e.g. WSN, which arrange a hybrid architecture as a part of field automation communications.

As a result of such backbone structure, several advantages arise such as higher reliability, expanded area of coverage, and ease of network integration.

On the other hand, WiMAX does not appear as a common application in most plants out there for the harsh environment of watching such big network, the need of optimal placement of WiMAX towers in order to gain full advantage, security matters, lack of flexibity, and others. It is currently not widely applied in automation but may be interesting in the future [45].

1.3.2. WLAN (Wi-Fi), IEEE 802.11

Commercially known as Wi-Fi, WLAN follows the IEEE 802.11 standard in which it run at a frequency of 2.4 GHz (or 5 GHz) at a maximum data rate of 54 Mbps. As per IEEE 802.11 standard's specifications, a typical Wi-Fi access point varies between 45 m (indoors) and 90 m for outdoors. WLAN is usually configured as two different modes: ad-hoc and infrastructure. The first applies the peer-to-peer concept in which stations in which all controllers and workstations suddenly form an impulsive LAN wirelessly. In the infrastructure mode, the network follows the server/clients concept described before by acting as an access point [9].

The application of WLAN in industrial automation arise as the following:

- It may aid as a backbone for other wireless networks on the plant for sack of data concentration and avoidance of high lose probabilities. As an example, Wi-Fi is sometimes used in conjunction with short-range/low-power present networks on the plant such as 2.4 GHz WSNs [45], [34].
- Also, one major application of WLANs is they can act as CN or PCN, demonstrated before, replacing regular LAN. Suppose for example a WSN is implemented as field network, Wi-Fi can in return collect data, from the gateway, described earlier, and pass it to the CCR. Here, the gateway is considered as a node in the CN [45], [34].

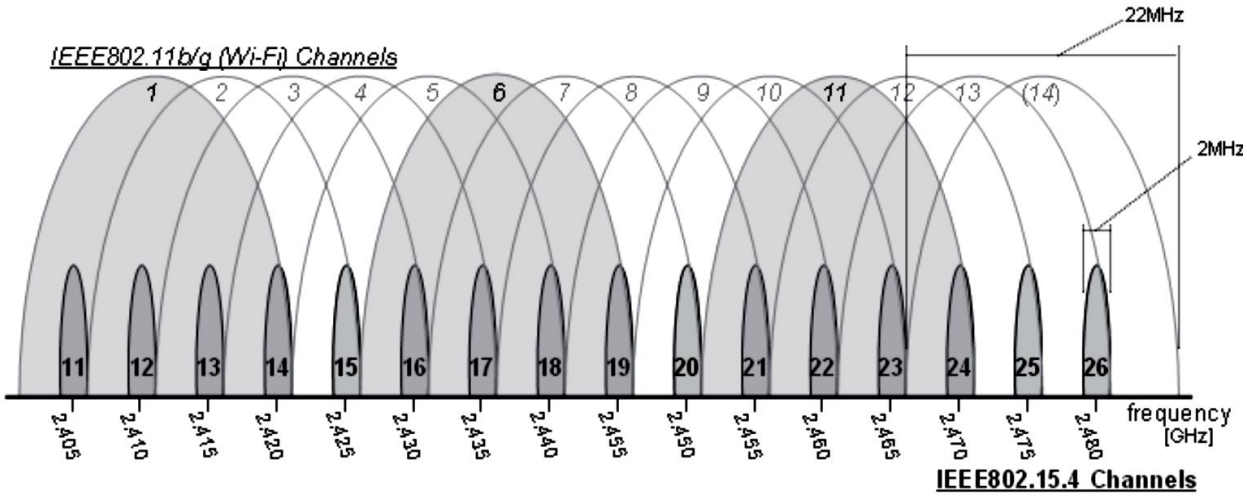


Fig 1.44: Wi-Fi as a backbone to 2.4GHz WSN channels (taken from [34])

WLAN offers number of advantages in process automation including the high covrerage range, its open and know standard, its abilty to add robustness to WSNs, its cost-effectiveness, and of course its high data rate. However, WLAN is not acceptable to be implemented as a WSN so that the need of device-to-device communication occurs. This is because of the comparatively high power consumption of Wi-Fi due to its fast throughput rate compared to other WSN protocols, i.e. ZigBee and W-HART.

1.3.3. WPAN

This subsection classifies the smaller range wireless networks known as WPAN standards that are applicable in industrial automation environment. They originate as: Bluetooth, UWB, IEEE 802.15.4, and others as this research deliberates [45].

1.3.3.1. IEEE 802.15.1 (Bluetooth)

The newest version of Bluetooth operates at a frequency band of 2.45 GHz with a data rate up to 1 Mbit/second and over a distance of 5 to 10 meters. Bluetooth employs (FHSS) in the PHY RF channel transmission. Because of its relatively low coverage range compared with other WPAN standards besides other issues such as high complexity and low power requirements,

process plants have decreased their interest in Bluetooth for WSN applications. Yet, Bluetooth is considered for high throughput applications for few numbers of nodes. The Bluetooth protocol's PHY and DLL completely follow IEEE 802.15.1 specifications [32].

1.3.3.2. IEEE 802.15.3 (UWB)

Ultra-wideband (UWB) is considered also as another short-range WPAN technology. UWB operates by transmission of nanoseconds duration impulses in periodic sequences forms. IEEE 802.15.3 standard outlines the PHY of UWB. Recently, it has gained some industrial attention whereas before it was mainly found in personal wireless networks and multimedia [45]. As an example, UWB can be used for non-process automation wireless communication tasks such as registering and locating equipment and devices on the plant. In addition, UWB might be temporarily utilized during site commissioning when wiring seems time consuming and difficult. Due to its unique operation characteristics, UWB benefits in a highly secure transmission of high throughput without sacrificing much of its power. However, the disadvantage of realizing limited number of nodes and short distances coverage range has decreased the acceptance of UWB in the harsh and unsafe process control environment [32].

1.3.3.3. IEEE 802.15.4 (WSN)

Unlike other WPAN, IEEE 802.15.4 is the standard that took over WSN tasks for its features that match process control requirements of extended number of nodes (sensors), low power, and reliability. Section 1.2.1.4 illustrated the common architecture and components of a typical WSN. This subsection tries to proceed a step down into the communication stack of WSN that principally adheres to IEEE 802.15.4 specifications in the PHY and MAC layers [34], [9]. However, there arise several WSN protocols that realize IEEE 802.15.4 standard, yet they differ in features, therefore usage, and specifications, especially in the layered structures. The next

section compares the most common industrial WSN protocols and technologies available for field instrumentation applications. In general, a typical WSN stack should have a PHY, MAC (a sub-layer of DLL), Network, and finally the application layers [36], [9] as appear in Fig 1.45 below where:

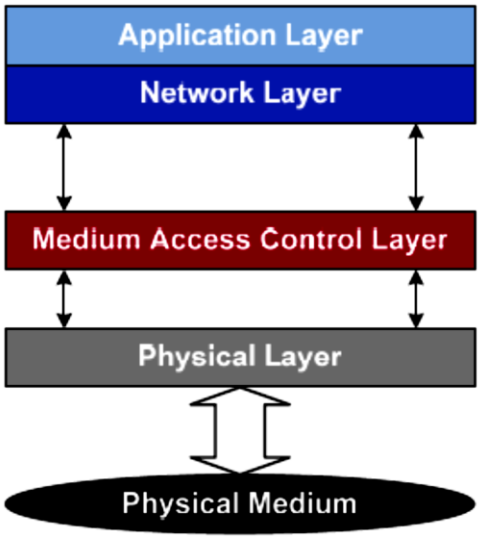


Fig 1.45: General WSN communication stack (taken from [9])

- The PHY layer's core task in WSNs is the RF channel, TX/RX, modulation of digital data, signals transmission, and spread spectrum. In other words, PHY in WSN provides the H/W and the way of transmitting the raw data (bits) over the actual RF channel line.
- Similar to most other examples, The MAC arranges the multiple users (devices or sensors) access to the RF link. It synchronizes the time between devices and the radio signals.
- Besides its network management and routing roles, the Network Layer in return delivers end-to-end packets of data from the sensors (sources) to the system (DCS).
- The Application Layer provides a translation in a S/W form to other parts of the system and the operator. The application layer shows the network tree and decides the function of each component and node on the network. The application layer also insures the health and reliability of data packets [9], [36].

1.3.3.4 Others wirless technologies

A new and different technology is the RFID. It can be used for example in inventory control and assembly line applications where the tagging and tracking appears important [5], [9], [34]. RFID uses electromagnetic fields to transfer data wirelessly, for the purposes of automatically identifying and tracking tags attached to objects. Therefore, RFID uses an electronically programmed tag. The tag is called RF tag and is read from a distance. The RFID runs on varieties of frequencies from several hundreds of kHz to GHz covering a range of 20 to 100 meters, depending on the type of the tag. Typically, the RFID reader transmits the required power and data to the tag. RFID information is considered to be static, that it must be programmed into the tag for each device in the network. Hence, RFID is not the best choice for WSN applications that require regular measurements or diagnostics and flexibility in configuration and reconfiguration. However, as seen in [5], the RFID technology is still under R&D process.

Other existing forms of wireless technologies are 6loWPAN, Satellite communication, and Cellular Networks. 6loWPAN utilizes the same low-power wireless 802.15.4 networks on the PHY, but the standard (IP) communication stands on the top. The upper layer technology utilizes IPv6. Thus, 6loWPAN offers an advantage of communicating directly with other devices via IP networks [32]. Satellite communication can aid in remote analytical monitoring of resources. However, satellite communication may accompany time delay disadvantages, wireless transmission issues, and acquire cost. Furthermore, one critical disadvantage of satellite communication in industrial automation is the lack of existing standard protocols. Cellular networks might be used to collect measurement data from remote locations in process automation. This can be thought of in SCADA systems applications. Some current R&D propose GPRS to improve the reliability and the usefulness of cellular networks in industrial applications [36].

1.4. Industrial WSN Protocols and Standards

As the pervious section introduces the wireless technology in industrial automation, this part of the research lists the most common emerging WSN technologies in industry and it roughly compares them. Until today, the most well-known three candidate WSN protocols are: ZigBee, W-HART, and ISA.

1.4.1. ZigBee

Commercially known as Xbee transceivers, Zigbee appears to be the most well-known or wide-spread WSN standard not only in industry but also in academia, building automation, aerial, and other commercial applications. IEEE 802.15.4 standard expresses the PHY and DLL of ZigBee that is 2.4GHz RF frequency and a bandwidth of 2MHz (250 kbps), DSSS, O-QPSK digital data modulation technique, and CSMA/CA in the MAC layer [33], [32], [35], [34], [7]. ZigBee targets for cost sensitive, very low-power, and low-speed wireless networking [32], [37], [7]. Using CSMA/CA, ZigBee can operate in a beaconed mode where some sort of time-synchronizations between nodes apply in a shape of super-frames divided into 16 slots. Default ZigBee, however, supports non-beaconed mode. On the top of the mentioned PHY and DLL layers, ZigBee protocol adds field device coordination and network topologies by providing mesh, or other forms of routing, scheduling, and sleep mode, which is the ability of a sensor node (field device) to conserve power and maintain reliable communication [33].

Usual communication channel of a typical WSN, including ZigBee, is formed by consecutive super-frames. In IEEE 802.15.4 ZigBee a single super-frame unit includes beacons, active portion, and inactive portion of data, in addition to reserved slots to some predefined periods of applications which guarantee both communication cycles and previously defined functions taken into account, as emerges in Figs 1.46 and 1.47.

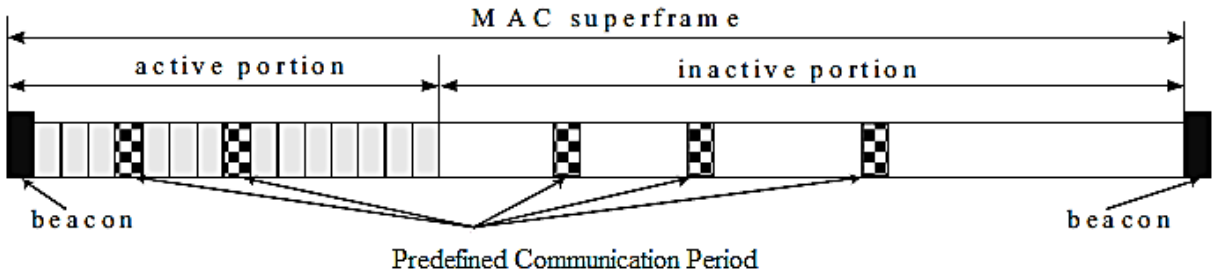


Fig 1.46: ZigBee MAC super-frame using CSMA/CA (taken from [46])

As stated, the multiplexing technique is the CSMA/CA where each channel is divided into slots with the same amount of time. However, for RT automatic applications, ZigBee MAC layer assigns priorities on the top of the CSMA/CA process. In other words, messages with high priorities have shorter periods of time while low ones are allotted longer intervals. Thus, when the channel is busy, only high priority messages can access the channel [46].

Due to a few reasons such as the inability to serve the high number of nodes within a specified cycle time and lack of FSHH, the ZigBee protocol has been considered not robust enough for harsh RF process control environments by some end-users. However, since 2007, the ZigBee Alliance has released an upgraded version of the protocol called ZigBee PRO in which the above two issues are resolved by adding the "frequency agility" concept outlined in [37], [33].

In summary, ZigBee offers excellent low power consumption and characteristics. Besides, the protocol provides some flexibility by supporting different topologies. In general, ZigBee/ZigBeePRO is a good candidate for several WSN applications; however, the technology suite applications such that a little data traffic and not too many devices are required.

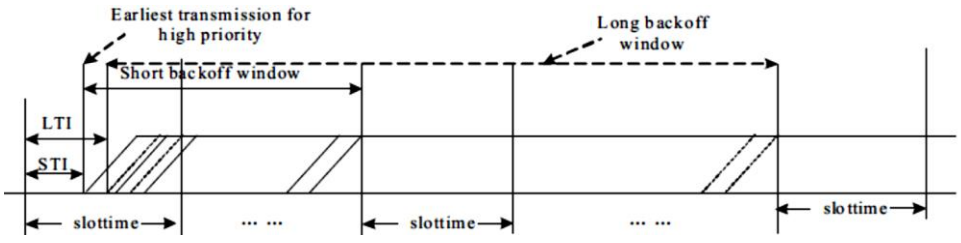


Fig 1.47: Prioritized CSMA/CA to increase RT transmission reliability (taken from [46])

1.4.2. W-HART

The W-HART specification was released in 2007 as wireless extension of the wired version introduced earlier. The technology also employs the IEEE 802.15.4, 2.4GHz-base band radio. However, W-HART differs from ZigBee by adding some slight modification to both the PHY and DLL of the standard. W-HART uses both FHSS and DDDS. In addition, the MAC layer is designed to accept co-existence of other networks, such as the backbone Wi-Fi example, and realizes TDMA technique of 10 ms time slot for each field device [9], [32], [33], [35], [37].

In the network layer, W-HART applies power and path optimized mesh routing and provides channel blacklisting, redundant routs, and retry mechanisms. In the upper layer, the W-HART specification protocol stands on top where users have S/W environment access and configuration interface to the network tree including gateways and field instruments. The W-HART application layer defines commands and responses, burst modes, and security by setting joint keys, network ID, and end-to-end reality [9], [7], [32], [33], [35], [37].

The advantage of HART over ZigBee relies on adding FHSS, which is more noise and interference immune, and therefore, more secure. Besides, the time synchronization mechanism that applies before the TDMA at the MAC layer marks W-HART more robust and reliable to accept more number of nodes required in industrial plants. Moreover, in parallel with the FHSS, W-HART performs a mechanism called channel black listing shown in Fig 1.48 below.

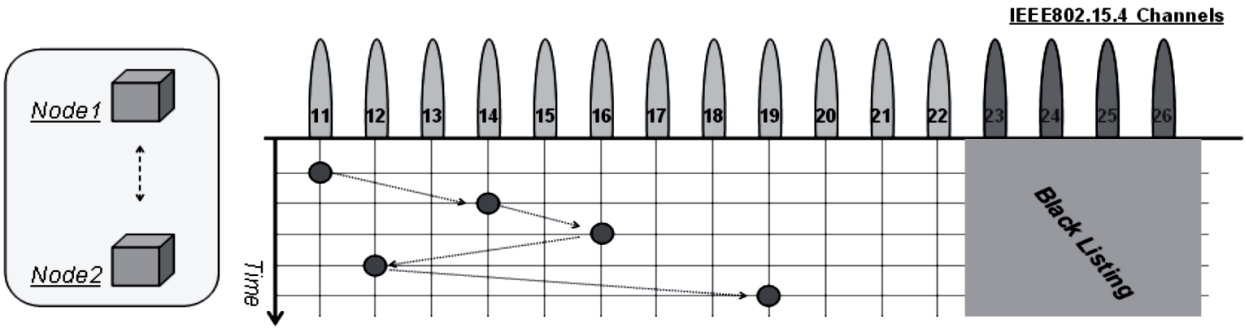


Fig 1.48: FHSS technique RF channel transmission with black listing (taken from [34])

The concept of channel blacklisting, together with FHSS, can be summarized as the W-HART PHY layer transmits modulated RF signals between nodes by dividing 16 channels that are continuously assigning alternating different frequencies each at a time around the broadband, 2.4GHz. Among the 16 channels, there might be several that are sensed as poor quality ones. If two nodes cannot communicate on those poor-quality channels in a given slot, they will be given another chance to communicate. Otherwise, those bad quality channels are blacklisted, consequently skipped from the FHSS slotted sequencing of channels [33].

Another modification to the IEEE 802.15.4, W-HART executes a fully time synchronized TDMA technique at the MAC level. This first involves time synchronization among devices by inserting defined time offset duration information included in the ACK packets replied by the receiver node as a confirmation of successful reception of data sent. This way, TSMP avoids the beaconing approach realized in the ZigBee, which consumes more power and is less cost effective. If in case a receiving node did not send back this ACK package to the sender, MAC TSMP automatically switches to a neighboring node to assign a specified time slot [7], [9]. Please notice Fig 1.49 below.

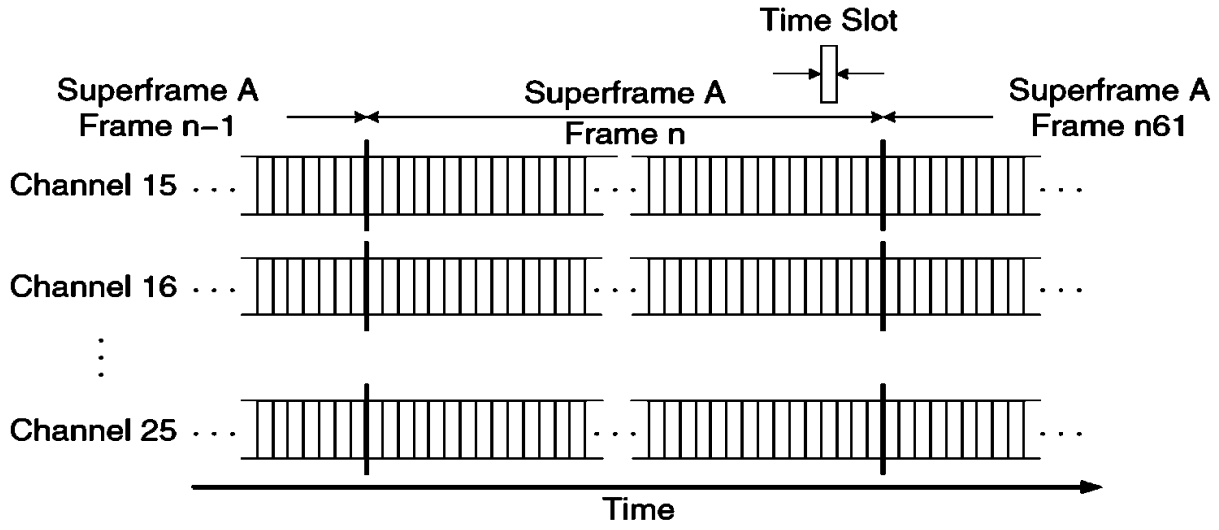


Fig 1.49 [47]: TDMA, time slots and W-HART's super-frames (taken from [48])

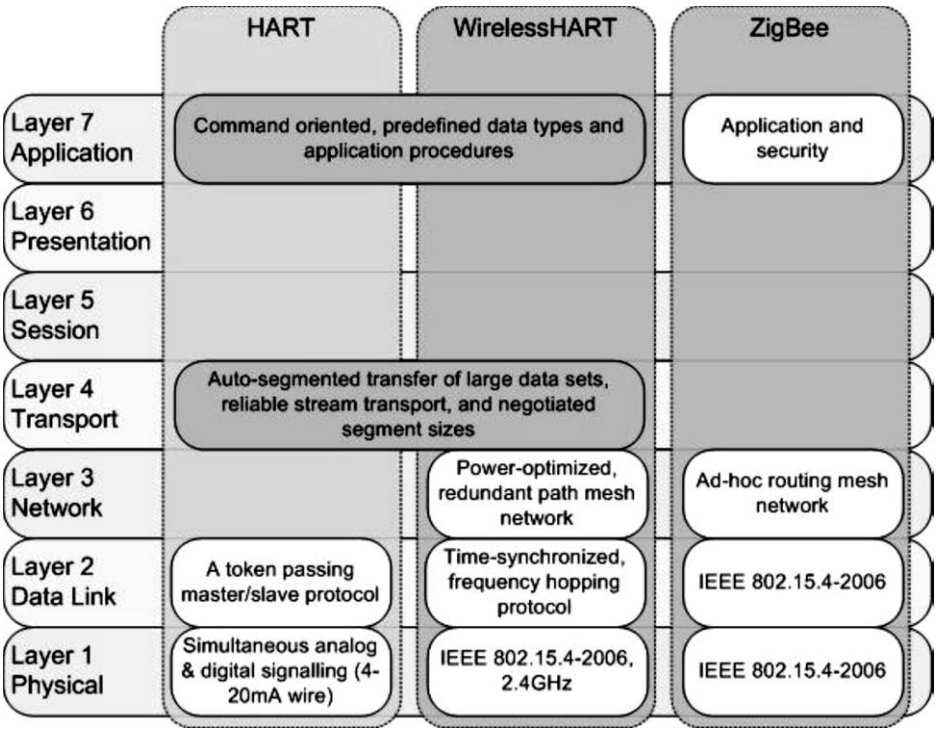


Fig 1.50: The main differences between HART, W-HART, and ZigBee protocols (taken from [35])

W-HART applies two main approaches of network routings as per the practice:

- Graph routing: which is the default and preferred method of routing in the W-HART protocol. Graph routing pre-scans a defined route for a particular message to be sent from a source to a destination. Therefore, path redundancy is attained by pre defining several graph routes for one data message [35].
- Source routing on the other hand, which is allotted mainly for network diagnostics rather than transmitting RT PV messages, applies the ad-hoc concept to create routes for communication with no redundancy applied [35].

Fig 1.50 above illustrates general protocol differences between ZigBee, W-HART, and HART. As a noticed drawback on W-HART, it appears that the protocol supports only wireless HART devices; however, for the wired smart HART and conventional ones (4 -20mA), wireless adaptors are needed. Other than that, the protocol does not support other field devices.

1.4.3. ISA 100

So far, ZigBee as well as W-HART have built a proven foundation as well-known trademarks technology standards in the area of industrial WSN applications. ISA-100.11 protocol, however, is the newest among the most common three industrial WSNs. ISA SP100 committee has released the first version of the protocol, ISA-100.11a, since 2009 to be a dedicated WSN standard protocol for industrial automation applications particularly. The features of the standard put into consideration the industrial wireless specifications such as factory automation environment, backbone networks, RFID applications, power sources, and other process control applications. [32], [9], [37].

The PHY Layer of ISA 100.11a relies on the same modified W-HART IEEE 802.15.4, 2.4GHz-band one of DSSS, FHSS. Yet, at the DLL layer ISA 100.11a offers both MAC modes CSMA/CA and TSMP TDMA (to allow more throughput when preferred) as a hybrid protocol with the same channel blacklisting technique explained before [9]. In the network layer, ISA 100.11a adopts the 6loWPANs IPv6 mesh technology grasped earlier whereas upper layers employ the ISA100.11a native protocol, which is publishing/subscribe client /server based, and it can be summarized as applying object mapping, tunneling, bulk and alert event notification functions, and others [34]. These features allow ISA100.11a to be a more universal standard that supports other available common protocols and vendors found in industrial plants such as HART, F.F, Profibus, Modbus, etc. [34].

If considering security, similar to W-HART, ISA100.11a applies symmetric key, join key, network ID, and end-to-end security [34]. Moreover, ISA offers an option of a public key cryptosystem [9], [32]. Aside from technical communication specifications, as a quick

comparison to W-HART as in features, one could find out several tradeoffs, analogies, and dissimilarities. They can be summarized as, but are not limited to:

- First, both W-HART and ISA100.11a apply the FHSS technique, which facilitates RF transmission to increase security and robustness; however, the second gives an option of switching TSMP TDMA, which ensures reliability, to CSMA/CA, which might provide higher bandwidth in more friendly environment cases (less number of nodes and lower noise and interference).
- Second, while all W-HART nodes have the feature of acting as routers, which increases the routing redundancy level, ISA100.11a supports a simpler type of non-routing device. This gives the user the trade-off option to decrease the cost of an alternative path while consuming less power or vice versa.
- Both standards co-exist very well with other wireless bands, i.e. IEEE 802.11a/b. Nevertheless, one very remarkable advantage of ISA100.11a over W-HART is that the first supports not only wireless but also wired networks in higher layers. This characteristic marks ISA100.11a open friendly with other fieldbus devices and protocols and CNs. [33].

In summary, ISA100.11a is a very promising technology in regard to industrial field communications using WSN approach. The ISA 100 group aims for a uniform open WSN standard, and the committee has come up with a first version of a protocol that tries to gather the best of both previously recognized ones: ZigBee and W-HART. However, regardless of ZigBee, which might not rise to the high industrial automation standards, in some applications, which protocol offers the best features among ISA100.11a and ZigBee might be debatable as it really depends on the purpose, practice, and the user.

1.5. Summary and Motivation

This section states the thesis motivation, scope, and the main idea behind such introduction. The section first summarizes all the historical background and technologies described in this chapter. Then the purpose and scope of the paper is specified.

1.5.1. Summary

This long introductory chapter can be summarized in the following points:

- Industrial automation and control was first implemented as analog systems; then the digital computers took over this field as seen in Fig 1.51 (a) and (b) below.

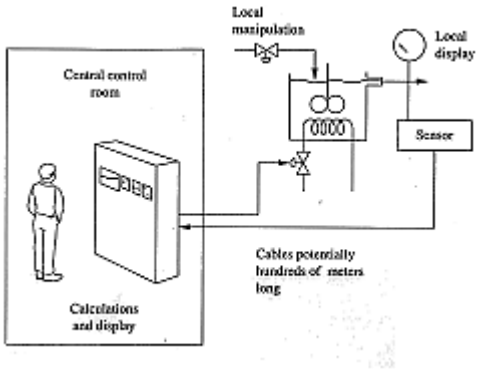


Fig 1.51 (a) Industrial control with analog calculations (taken from [119])

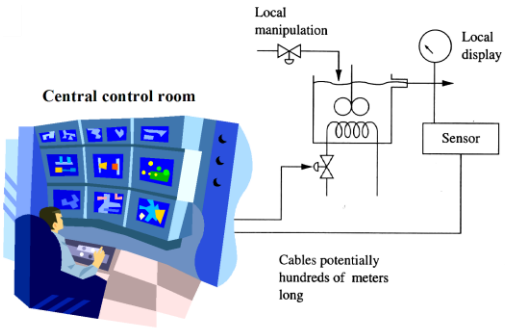


Fig 1.51 (b) Industrial control with digital taken from indications [119])

- Digital control systems were introduced later on where computers started implementing the control directly, instead of being used for SCADA purposes only. This digital control has undergone an evolution process (Fig 1.52 (a)) of being centralized first, then distributed, and now industry is moving towards the integrated systems where PCSs now include DCS and other subsystems such as PLCs on the same networked PCN architecture.

- This technological evolution process has not only concerned control processing units and digital architectures, but also a major topic in industrial automation, field communications, where systems are categorized as pneumatic, analog, smart digital, and fieldbus, as appears in Fig 1.52 (b). Also, WSN has been introduced in this context.

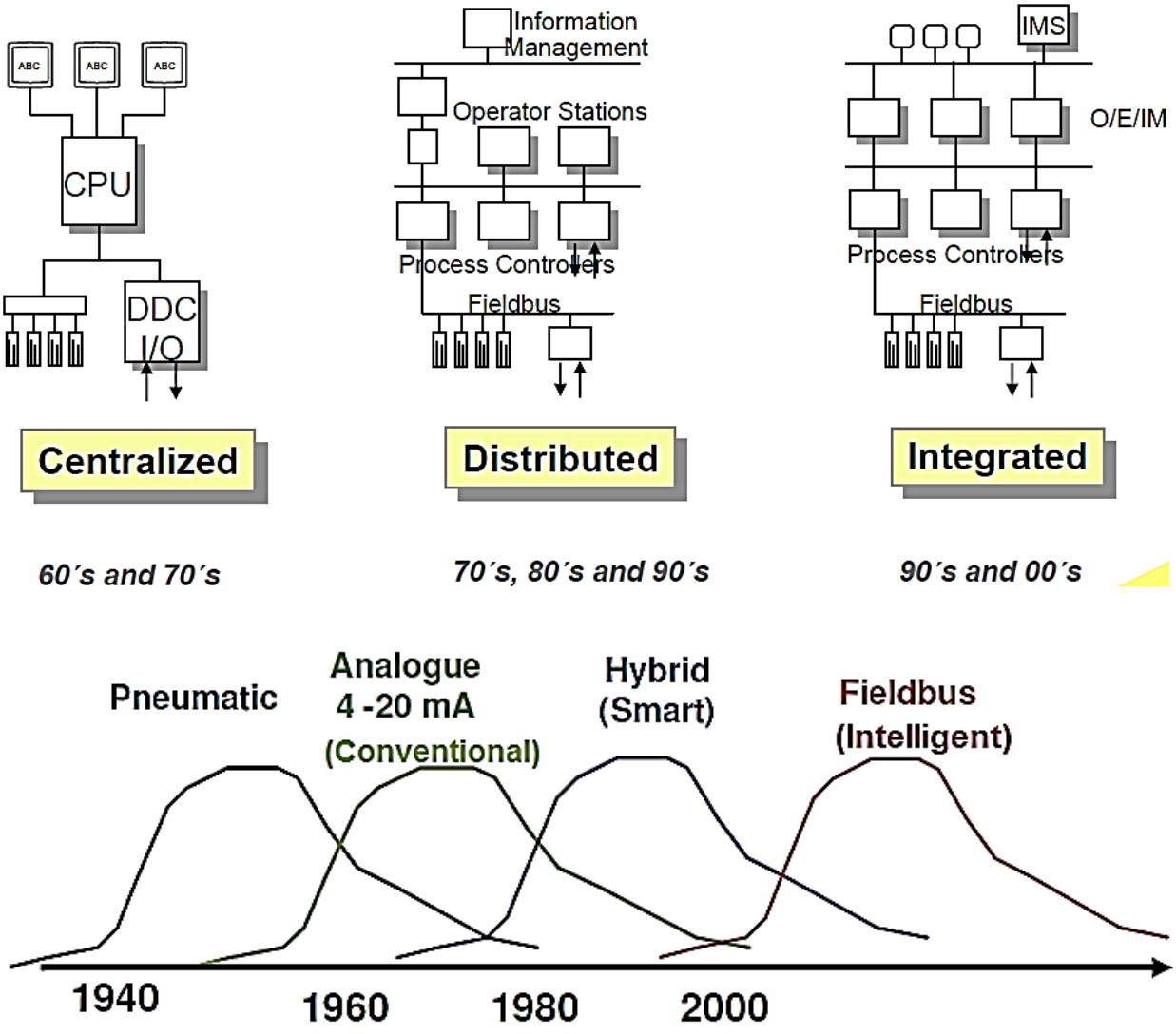


Fig 1.52 (a) Evolutional periods of industrial digital control systems (taken from [120]) and (b) field communications protocols

- Points one and two above suggest industrial automation and control systems as layered architectural networked levels as shown in Fig 1.53 below.

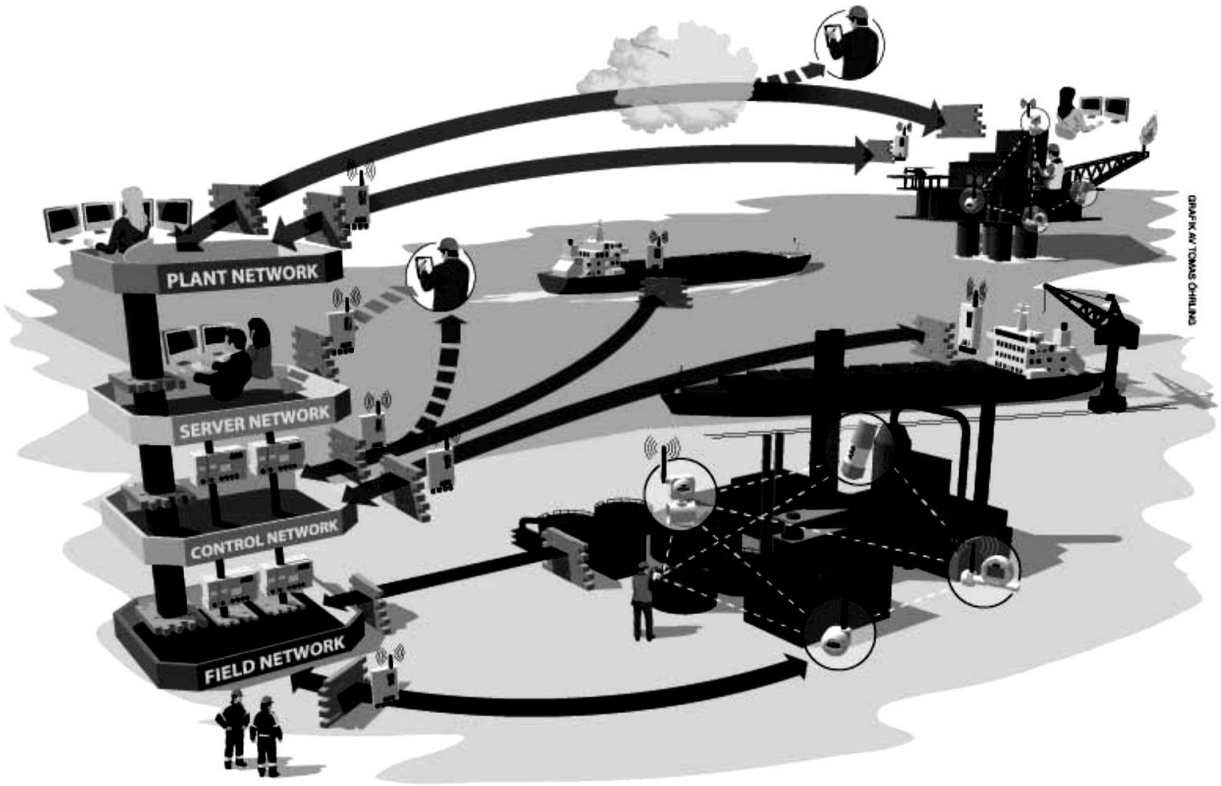


Fig 1.53: Modern industrial automation layered networked architecture (taken from [4])

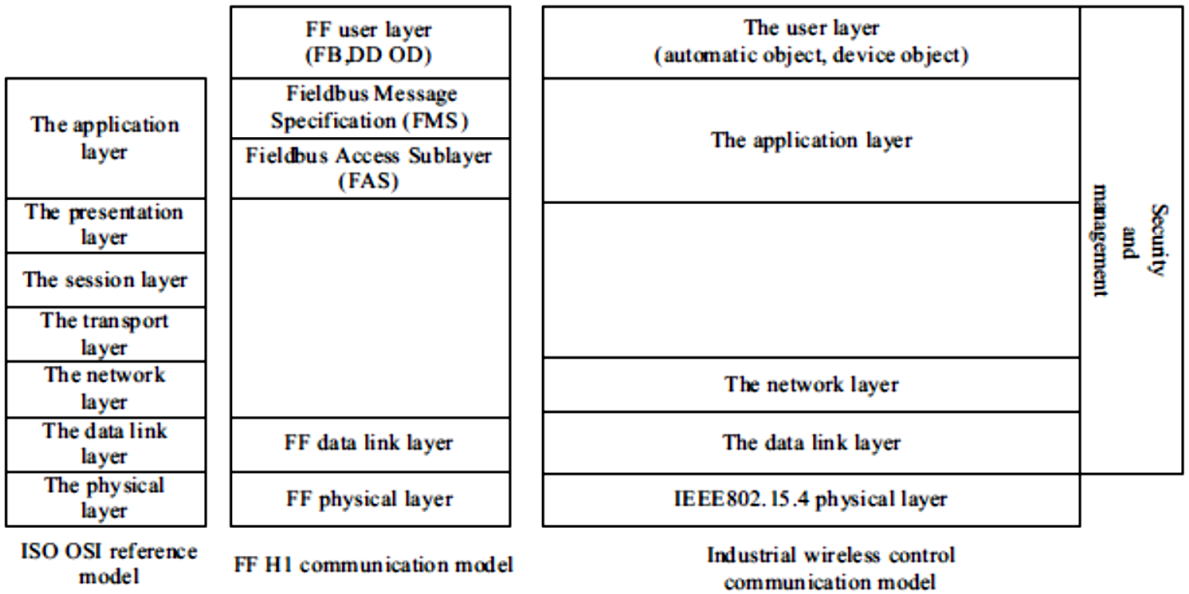


Fig 1.54: The latest two field communication technologies, Fieldbus and WSN (taken from [46])

- Different communications protocols in industrial architectures propose studying major features and limitations of each technology. A good example would be Fig 1.54 above.

- An important form and side of network integration are the subsystem communications where the two common approaches used are: serial transmission and OPC. The latter is more of S/W linking done on the network level whereas first is performed in the CN layer by the mean of physical connectivity. The OPC specifications require the two communicated ends, source and sink, master/slave, or in a common practice DCS and PLC to realize a form of Windows OS on certain levels. On the other hand, serial communication should define a specific bit of transmission protocol of communication between the two controllers. The most common of all appears the Modbus protocol of its two types: over serial line or TCP/IP. Fig 1.55 (a) and (b) underneath demonstrates the major deviations between both.

Layer	ISO/OSI Function	Modbus Function
7	Application	Modbus Application Protocol
2	Data Link	Modbus Serial Line Protocol
1	Physical	EIA 485, EIA 232C

Modbus Serial

Layer	ISO/OSI Function	Modbus Function
5,6,7	Application	Modbus Application Protocol
4	Transport	**TCP**
3	Network	**IP**
2	Data Link	IEEE 802.3
1	Physical	IEEE 802.3

Modbus TCP

Fig 1.55: Modbus protocol (a) over serial line and (b) over TCP/IP commutations stakes.

- Discussing different communications standards in automation leads to search in the newest technology in this matter, Wireless, with an emphasis on WSN as field communication.

- WSN has been suggested as an alternative for the wired versions and technologies of the field communications. However, due to several unaddressed challenges such as latency, reliability, and noise issues as well as security, until today, end-users have been installing wireless devices instead of the regular ones only for monitoring and indication purposes, (Fig 1.56). Therefore, a lot of recent research has evolved in the area where studying the capability of standard WSN to perform RT process control tasks efficiently, as if WSAN nodes are installed seen below in Fig 1.57. Three WSN standards emerge as accepted candidates so far in industry as field protocols or part of hybrid site networks: ZigBee-PRO, W-HART, and ISA 100.11a.

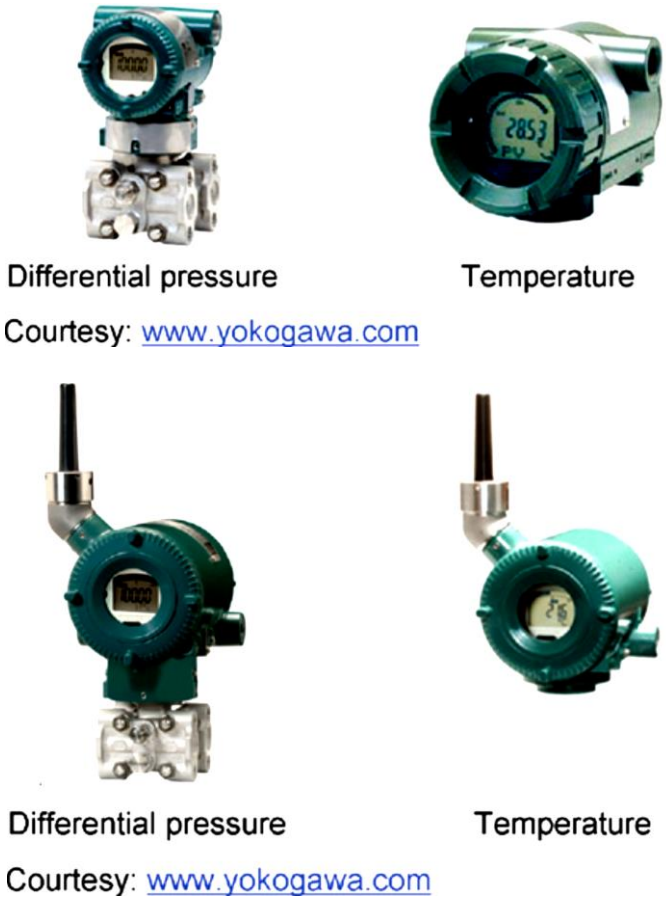


Fig 1.56: (a) A sample of industrial field instruments (transmitters) and (b) wireless nodes (taken from [23])

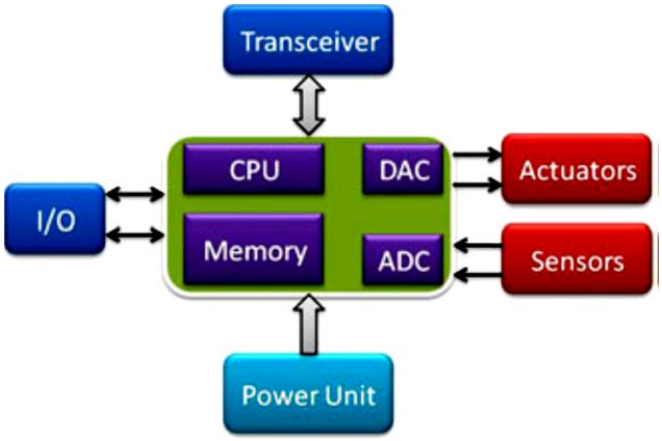


Fig 1.57: WSAN node block architectural diagram [6].

- The three WSN protocols are all IEEE 802.15.4 based. However, Table 1.1 as follows summarizes the comparison between them that makes each a unique standard [4], [7]:

	ZigBee	Wireless HART	ISA 100.11a
PHY Layer	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4
Radio Channel(s)	DSSS	DSSS/FHSS	DSSS/FHSS
MAC	CSMA-CA	TDMA	TDMA/CSMA
Channel Blacklisting	No	Yes	Yes
Routing & topology	Ad-Hoc mesh	Redundant TSMP	power optimize mesh
Power consumption	Low	Low	Low
Reliability	Low	High	High
Robustness	Low	High	High
Bandwidth	Higher for few nodes	High	High
Co-existence	Low	High	Higher
Latency determinism	Low	High	High
Suitable Application	Home automation	Industrial automation	Industrial automation

Table 1.1: Comparison between the most common WSN technologies.

Also Fig 1.58 simplifies the layered communication stakes of the three WSN standards:

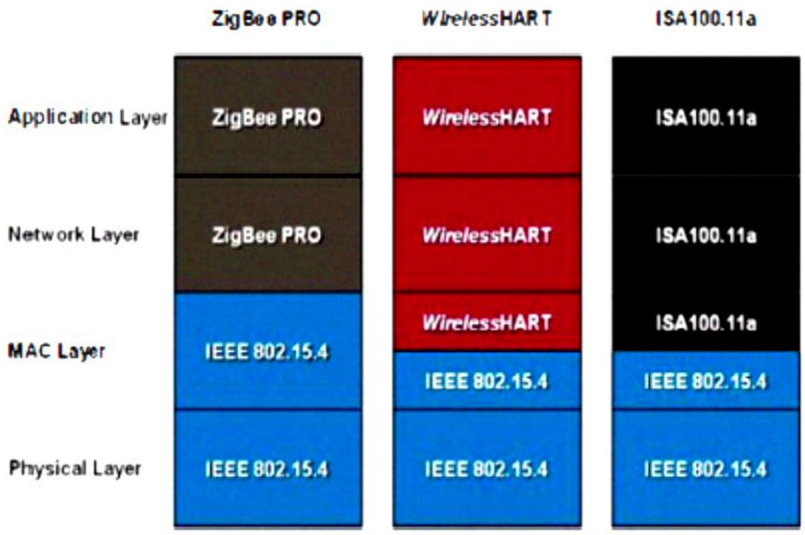


Fig 1.58: ZigBee, W-HART, and ISA100.11a communications stacks [7].

- Not only the possibility of implementing WSN in industrial automation has been studied, but also other wireless technologies are suggested in various architectural level to replace discussed wired standards. As an example, [5] of Fig 1.59 below proposes several wireless technologies at certain layers: WSN for field sensing, Wi-Fi WLAN as a backbone and CN instead of, or beside, the Ethernet, RFID for device tracking and inventory control and/or decentralized control, and also other cellular networks for higher level interface or emergency alarms such as GPRS.

1.5.2. Purpose and Scope (Motivation)

After summarizing the background information and current technological R&D in this field, industrial automation, instrumentation, and process control, this subsection states the focus of this research and justifies the motivation behind chosen topics. Also the scope of this paper of which the philosophy, methodology, and outlined context of the next chapters and their relation with the introduced topics will be mentioned as well.

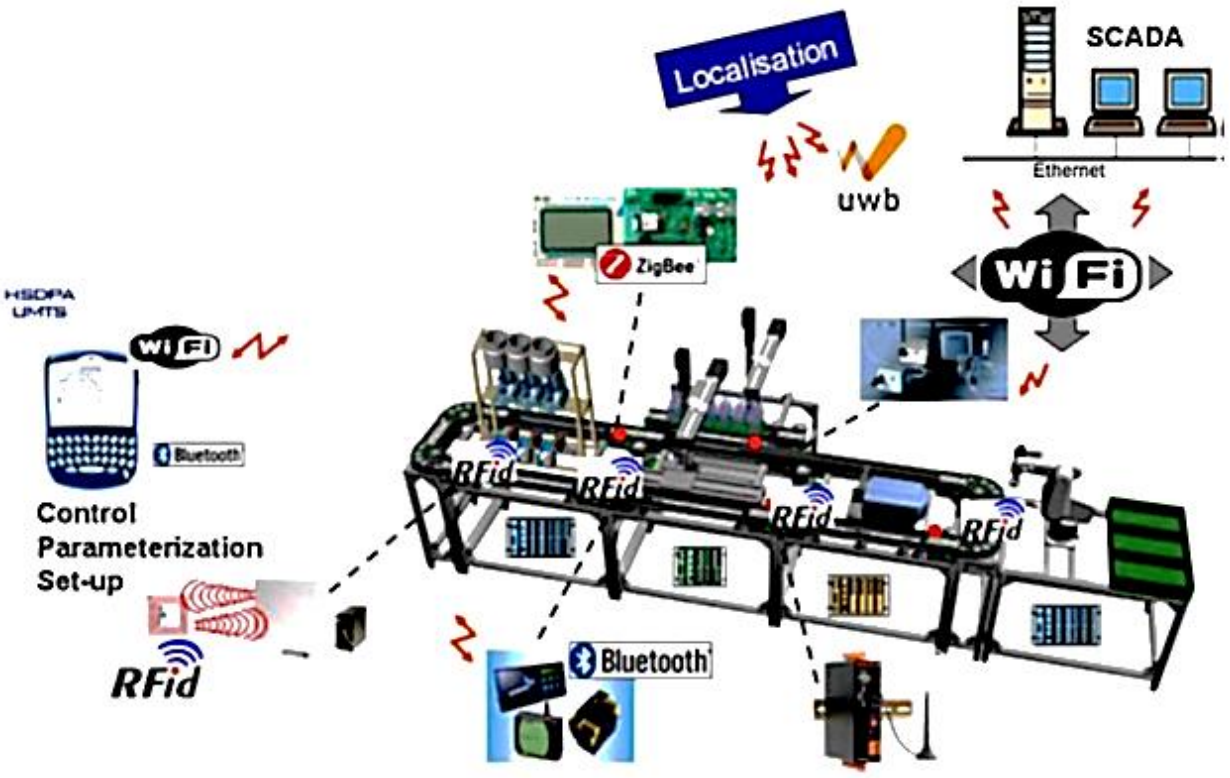


Fig 1.59: An example of proposing several wireless technologies in automation (taken from [5])

1.5.2.1. Purpose of the research

The purpose of this work lies in a couple of points as closing the gap between industry and academia, implementing useful applied control setup as per industrial standards to be used for educational, training, and laboratory research purposes, and answering a few open questions regarding assessing various instrumentation technologies, at different automation levels, and evaluating control loops quality and performance through experimental studies utilizing those.

1.5.2.2. Scope

The focus of the report is to experimentally study the effect of certain instrumentation technologies as sensing, signal conditioning, digital processing and computer architecture units, actuation, and finally field communication on evaluating RT control quality and performance by implementing a case study, the ball on beam system, after reviewing basic concepts and theories.

CHAPTER 2

BASIC THEORETICAL AND PRACTICAL CONCEPTS AND DEFINTIONS

This chapter reviews some basic concepts and definitions used in the research. Those are linear systems and feedback control theory, measurement systems and sensors, signal conditioning and processing in measurement control instrumentation, different types of control units and systems, control algorithms, actuators and final control elements, and finally motor driving fundamentals.

2.1. Dynamic Systems and Feedback Control Between Theory and Practice

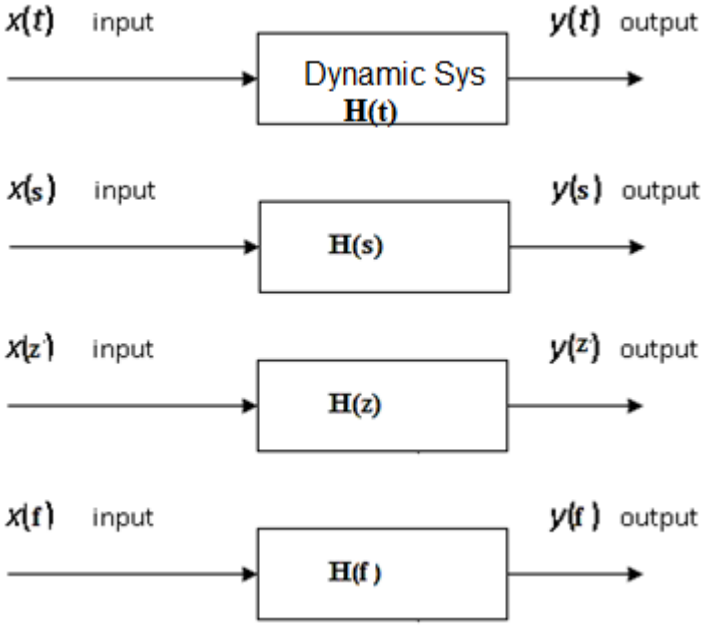


Fig 2.1: General and basic structures of dynamic systems and their various analysis tools

There are different kinds of systems. They could be classified as deterministic or stochastic, static or dynamic, linear and NL, continuous time vs discrete time, or discrete event systems. As this paper exposes the use of control theory in applications as industrial automation, embedded systems and mechatronics, most systems studied appear dynamic and have the form of

deterministic LTI with its two sub types: LCTI and LDTI. A dynamic system can be defined as any set of relationships where the output(s) depends on the current as well as pervious values of input(s) with respect to time change. Thus dynamic systems can be expressed mathematically as differential equations where the general form can be written as:

$$\frac{dy^{(n)}(t)}{dt^{n}} + a_1 \frac{dy^{(n-1)}(t)}{dt^{n-1}} + \ldots + a_n y(t) = b_0 \frac{dx^{(m)}(t)}{dt^{m}} + \ldots + b_m x(t) \qquad (1)$$

Where *y* represents the output, *x* is the input, and *n* and *m* denote the rate of changes orders of outputs and inputs respectively.

The above dynamic system is a SISO class as it has one input, *x,* and a single output. However, if there is more than one input or output, the system is referred to as MIMO. Also, a dynamic system is said to be linear if it satisfies the superposition and homogeneity principles [18]. Most RT dynamic systems are NL in nature as well as the ones studied here in this report; however, linearization approximation is applied to adapt to standard theoretical linear analysis such as Laplace, Z, and Furious transforms domains (Fig 2.1). Furthermore, a dynamic system is called time invariant if a delay time is exposed to its input(s) that causes the same amount delay in the output. Otherwise, the system is called time variant. Even when modeled as NL, all dynamic systems present in this study are considered LTI when analyzed.

A more collective representation of LTI systems is the SS with its following general form:

$$\begin{aligned} \dot{X} &= Ax + Bu \\ Y &= Cx + Du \end{aligned} \qquad (2)$$

Where $\dot{X}$, Y, *x, and u* are all functions of time and represent the system *n* state(s), q output(s), variables, and *p* input(s) respectively, and they can be scalars or vectors depending on the order and the number of I/Os. Similarly, *A, B, C,* and *D* matrices of $n \times n$, $n \times p$, $q \times n$, and finally $q \times p$ or feedthrough (zero) dimensions respectively.

Most RT dynamic systems are of the above LCTI; however, some take the form of a LDTI, as revealed in the stepper motor example described later. Discrete time systems can be represented by difference equations which are approximately equivalent to differential equations in continuous RT systems. Also, as far as digital control systems are involved, it appears more relevant to discretize modeled LCTI systems, and convert them into LDTI ones. This can be achieved by several finite series mathematical approximation methods. One known method is the Taylor series expansion where *k* is an integer sampling iteration number and *I*, the identity matrix:

$$\dot{X} \approx \frac{X[k+1]-X[k]}{\Delta T} \leftrightarrow X[k+1] = (A+I)X[k] + Bu \tag{3}$$

As previously mentioned, the SS in (2) provides detailed information about the internal dynamics of LCTI systems whereas the TF approach describes the relation between inputs and outputs in a SISO analysis approach. Though, there might be times when analyzing SS models LCTI systems through an equivalent SISO TF approach seems more helpful in differentiating between several internal blocks of the overall structure and understanding the I/O relations between them. TFs of SISO LCTI systems can be extracted by taking the Laplace transform:

$$L[x(t)] = X(s) = \int_0^\infty x(t)e^{-st}\,dt \qquad (s = \sigma + j\omega) \tag{4}$$

Where S is a complex term in the Laplace domain's plan of real magnitude and an imaginary part. The Laplace transform helps represent SISO differential equations of *n* derivative order into a simplified algebraic relation which simplifies solving (analyzing the output response) of such systems. Similar comment applies for Z transform in the case of LDTI systems. Those transforms along with Fourier aim to transfer systems from time to frequency domains as in Fig 2.1.

$$X(z) = \sum_{n=0}^{\infty} Xn\,Z^{-n} = \sum_{n=0}^{\infty} X[n]\,Z^{-n} \tag{5}$$

The above equation represents the definition of the Z transform of a discrete time signal $X(n)$. However, if the Laplace and Z transforms describe LCTI and LDTI systems in frequency domains respectively, then it seems logical to relate them directly without getting back to differential/difference equations. There exist many ways of converting a continuous Laplace TF to a discrete Z one, [56], such that the basic relation between the two domains attains:

$$Z = e^{ST} \tag{6}$$

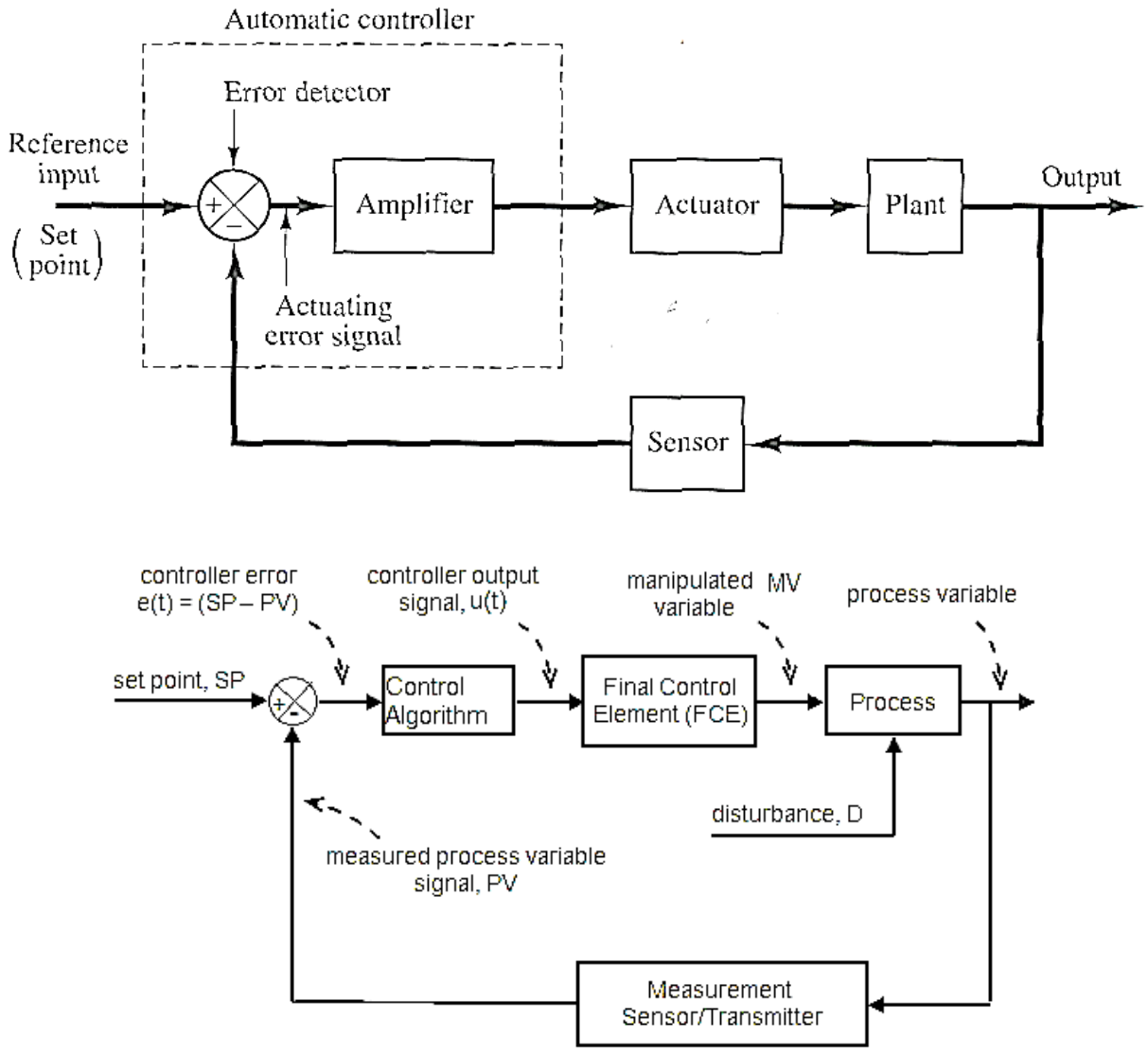


Fig 2.2: (a) The general simplest form of a closed feedback control loop and (b) its corresponding industrial methodology and naming philosophy adopted in this paper.

The above briefly describes how dynamic systems are mathematically represented to be theoretically analyzed. Conversely, since this research is all about RT control systems, the most basic theory of all is the feedback control loop as appears in Fig 2.2 (a)/(b). A feedback control system is a closed loop type [18] where the output variable, PV, desired to be controlled is measured, by a form of sensor, and then feedback to a control unit. The controller block in turn has three main function steps:

- First, it compares this measured PV with desired reference value, SP. The difference between PV and SP is called the error signal such that Error = SP – PV.
- Second, the controller passes this error signal to a decision internal block, more accurately referred to as control algorithm. This control algorithm sub-block, in its simplest form, functions as an amplifier where its output remains proportional to the error amplitude amount. Note that much of the literature denotes the control algorithm block as "the controller," while this paper's naming philosophy considers the whole control unit block as a controller, which better fits the RT implementation settings. More discussions on control algorithm is coming later.
- Third, an important functionality step of the controller is the control action. This term means the actuation signals going, out of the controller, to the actuator or the whole plant process dynamic. For example, in the SS model representation of plant processes, its input, *u(t),* represents nothing but the output actuation signal of the controller. *u(t)* can take a form of relay pulse (step) function, a ramp, or even a more complex shape; it is also known as "the control law" in many other conventions. Regardless of the naming methodology, in practice, there exist two main types of control actions: forward and reverse [47]. To better explain the two actions, consider the level control system example in Fig 2.3.

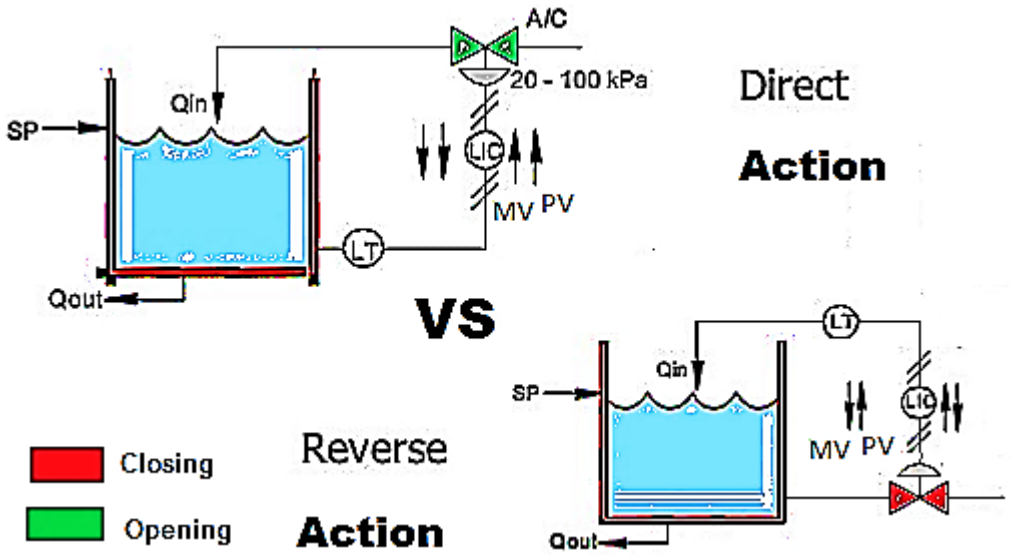


Fig 2.3: An example of a direct, (a) verses, (b) revers control actions

In the figure part (a) above, if the level (PV) is less than desired SP, the valve will proportionally open. Hence, an increase in opening the actuator position would lead to an increase in PV and vice versa from where the term direct action is driven. On the hand, in part (b), it is the other way around. An increased opening of the actuator would lead to a decrease of PV and vice versa. In other words, if the level is less than SP, for example, the valve will proportionally close to allow more flow inlet to fill up the tank so that PV increases more and vice versa. Thus, the action is called reverse action.

The actuator after that takes the controller output signal and acts accordingly as per the commanded control action. From its name, the actuator actuates the process in order to reach desired states. Much of the literature considers both the actuator and the process as a whole block, represented by the plant SS; however, this research separates the two for the sake of studying the effect of various actuation designs and technologies on the eventual performance of the control loop. The output of the actuator is given the industrial name MV since manipulating it changes the process output, which is the ultimate controlled variable. However, MV can be also considered as an output in some MIMO SS settings as may emerge later in the case studies.

Finally, the process block represents the actual dynamic system desired to be controlled. In numerous other conventions, this block is denoted by the name plant. To avoid confusion, this paper sticks with the practical naming philosophy, so if the term plant is already given to the large-scale factory processing unit, the actuator along with the process blocks are given the name process plant in this paper. Fig 2.4 shows a typical time response of a feedback control loop with respect to a desired SP, position in this example. To remind, the performance of a control loop is measured by several factors [18]:

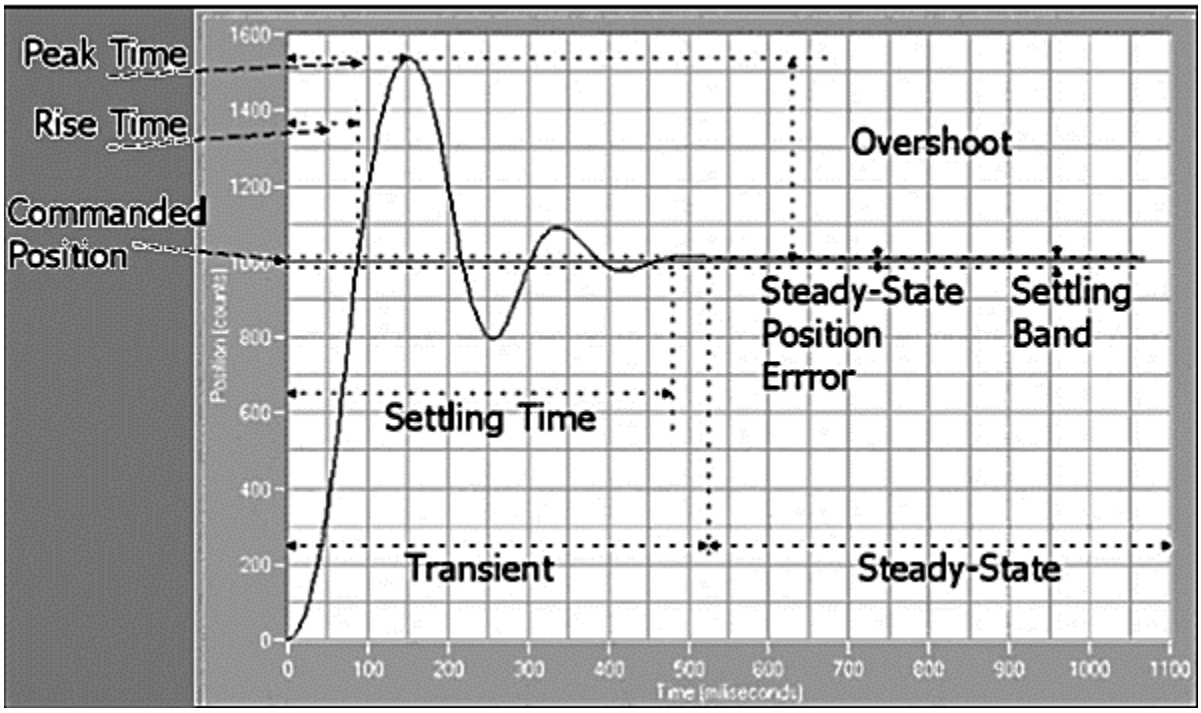


Fig 2.4: A typical closed loop feedback control system response (taken from [49])

- Settling Time: is the time required by the PV to approximately reach or settle near the SP after a sudden desired reference change, usually occurs within 2% to 5% of the final value.
- Rise Time: which is also known as the time constant, is the time required by the output PV response, of changing the input SP, to rise from 10% to 90% of its value, which is almost 63% the peak overshoot. Hence, the shorter the rise time, the faster the response.

- The Peak Time: is the time required for a PV response to reach the first overshoot peak due to sudden change in the SP.
- Overshoot is then the maximum peak value of the PV response right after setting the SP. A higher overshoot may result in a faster dynamic response. However, an increase in the overshoot may result in an unstable or less robust process [18], [49].
- The SSE is the minimum difference between the actual PV and the SP curves when control loop reaches its steady state that occurs when the system stabilizes as the output response goes to infinity.
- Another important definition when considering control loops performance is the dead time. It is the delay time period from the instance of changing the SP until the system actually responds. The time delay is critical in studying RT control performance characteristics such as in wireless measurement and control.

The above control loop performance measures indicate different kinds of response of any system. Hence, a system is said to be stable if it tries to track the SP yet does not oscillate after a given input. An unstable system on the contrary is the one that oscillates and never reaches the SP. In fact, an unstable system is also different than the oscillatory one. In the first, the oscillation increases with time while in the second the instability remains constant in magnitude [49]. There appear then three important stable control system responses:

- An over-damped system: has no overshoot as it goes smoothly and slowly to the SP. Being slow in response implies that such a type of systems produces a longer settling time. However, an over-damped tuning of a control loop ensures stability while sacrificing the dynamic response time performance.

- Unlike over-damped, under-damped systems outputs oscillation for a period of time but attains stability eventually. An under-damped system can be characterized by long settling time and higher overshoot that is not preferable in many applications.
- A critically damped system is more balanced. It provides mid-points between response time and smoothness. Therefore, such a type of response produces intermediate values of overshoot and settling time.

A ratio at which a system overshoots is called the damping constant, ζ. It is defined as the ratio of the actual amount of damping of the system over the critical one, $\zeta = 1$. The lower the damping ratio, the higher the overshoot and vice versa, and an unstable system has a damping ratio of $\zeta = 0$ [18], [49]. [18] provides detailed basic guideline equations in which a control system can be theoretically designed to satisfy predefined overshoot percentages and settling times.

Aside from designing control loops performance, considering MIMO higher order systems, as appears in countless industrial examples, suggests further analyzing plant processes models of such sorts before designing their controllers. This phase infers complexity levels and expected behavioral evaluation of the system intended to be controlled, and the step can be combined along with simulation. This brings up the concept of controllability and observability described in [18]. Determining controllability of a system indicates the ability of a designed controller to adjust dynamics and eventually be able to control the process whereas observability decides whether the interior state variables of that model can be measured.

Consider the general model of a SS form given in (2). A system is said to be controllable if and only if the states can be changed by altering the input, *u(t).* In other words, a system is indeed controllable at initial condition of states x_0 at time t_0 in the SS if there exists an input u(t) that can transfer all state x(t) from x_0 to the origin at time t_1. On the other hand, a system is said to be

observable if and only if the value of the initial state x_0 at t_0 can be observed from the output response *y(t)* through the interval $t_0 < t < t_f$. Thus, if this initial condition cannot be determined, then the system is unobservable.

Both controllability and observability can be analytically obtained after an initial deterministic model construction of a SS system. Controllability is determined by finding out the rank of the following matrix equals the order of the system, number of states, *n*, such that

$$n = Rank\begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{7}$$

If the above condition is satisfied, the system is controllable; otherwise, it is not. Similarly, to find out whether a system is observable requires ranking its observability matrix such that

$$n = Rank\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{8}$$

A step before the above would be systems modeling. In order to construct deterministic SS models following from (2), a set of physical properties and laws are essential in which each model is built upon. This brings an additional classification to dynamic systems based on their physical nature; they can be classified as either mechanical, electrical, or combined chemical, thermal, or fluid, etc. The following lines list the most basic physical laws and modeling rules used in this paper, namely mechanical, rotational, and electromechanical systems models since this research exposes case studies of such sorts.

Starting from mechanical systems that involve linear motion, four basic variables arise: linear position or displacement, *x*, velocity,$v = \frac{dx}{dt} = \dot{x}$, acceleration, $a = \frac{dv}{dt} = \frac{d^2x}{dt^2} = \ddot{x}$, and finally force, *F* in Newtons. When modeling mechanical systems, the most basic law to be

remembered is D'Alembert's [50], which is simply a restatement of Newton's Second Law of Motion. It states that all forces around a mechanical body of mass, *M*, should sum up to zero at equilibrium such that:

$$\sum_i Fi = 0 \tag{9}$$

Hence, there appear several mechanical systems elements forces laws utilized in this paper. First, the movement of a single mass; this law represents Newton's second directly.

$$F = \frac{dv}{dt}M = Ma = M\ddot{x} \tag{10}$$

Another linear force in such a dynamic environment is the friction effect. The force of friction is modeled as the following equation:

$$F = b\Delta v \tag{11}$$

Where *b* denotes the friction constant in Newton's seconds per meter units and Δv is the difference between velocities of two masses where friction occurs. If one of the two masses does not move, then the friction force can be expressed as:

$$b\frac{dx}{dt} = b\dot{x} \tag{12}$$

Similarly, considering rotational mechanical dynamics, an equivalent variable to force is torque, τ. Besides their weights, when dealing with rotating bodies requires considering mass moment of inertia, *J*. Therefore, applying Newton's Second Law in (9) considering rotational motion yields:

$$\tau = J\frac{d\omega}{dt} = J\alpha = J\dot{\omega} = J\ddot{\theta} \tag{13}$$

This is assuming a fixed moment of inertia where $\omega = \dot{\theta}$ is angular velocity in rad/s units, θ is angular position (rad), and α is the angular acceleration such that:

$$\alpha = \dot{\omega} = \ddot{\theta} \tag{14}$$

As a result, consider a rotational mechanical system, *H(t)*. To model the system involves reforming D'Alembert's law stated above to its angular form as the following:

$$\sum_i \tau_i = 0 \tag{15}$$

One more relationship that needs to be kept in mind when modeling rational systems that involves gearing is the gear ratio. It is defined as the ratio of the radius of an output to the radii of the input of a pair of gears. Alternatively, it also equals to the ratio of number of teeth of the same pair respectively, knowing that the spacing between contacting edges of both gears must be the same [50]. It is also given that the input and output angles when attaching two gears are not equal such that:

$$r_1\theta_1 = r_2\theta_2 \tag{16}$$

Hence, combining what we have yields the gear ratio, *N*, that is given by the equation:

$$N = \frac{r2}{r1} = \frac{n2}{n1} = \frac{\theta 1}{\theta 2} = \frac{\omega 1}{\omega 2} \tag{17}$$

Where r_1, r_2, n_1, and n_2 are the radii of the input gear, the radius of the output one, the number of teeth of the first and the second's, respectively.

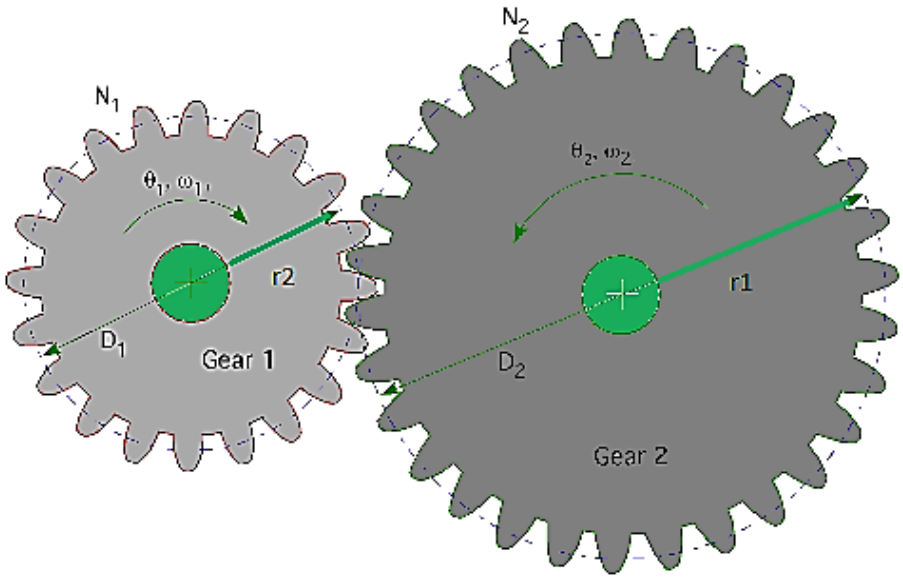


Fig 2.5: A pair of gears and the relationship between both of the radii, gear ratio and angles.

Likewise, electrical dynamic systems can be modeled using the well-known KVL and KCL [50] Keeping in mind, besides Ohm's law, the dynamic elements of capacitors as well as inductors such that:

$$i = C\frac{dV}{dt} \tag{18}$$

Where i is the current flowing into the circuit and across the capacitor, C, is the capacitor in Farads, and V is the voltage across the capacitor. Similarly, for voltage across an inductor:

$$V = L\frac{di}{dt} \tag{19}$$

Here, *L* is the inductor in Henries (H). Some other systems are electrometrical as develops later.

Once a dynamic system is modeled, it usually goes though further analysis to determine whether it is linear, controllable, observable, and stable. The first three have been covered; the following lines briefly describe stability of an LTI from a closer to theoretical perspective than the practical above. There exist a number of different types of stability analysis in literature, which may vary in their conceptual meaning such as BIBO, Marginally, Conditional, Uniform, or Asymptotical stability besides Instability. Generally, an LTI system is said to be BIBO stable if every bounded input to the system results in bounded output over an interval from t_0 to infinity [72]. In other words, an unstable system infinitely responds when a finite (bounded) input is applied to it. There are many ways of determining and analyzing LCTI or LDTI systems, stability conditions. One of the famous methods is the poles analysis by the means of characteristic equations taught in [18] and obtained from knowing the TF of the system model. However, for MIMO SS models described earlier a common method usually uses Eigenvalues. Considering a Laplace transformed form of the SS representation in (2), the following condition must be satisfied using the identity matrix and the Eigenvalues, where *D(s)* denotes the TF denominator:

$$D(s) = |(sI - A)| = 0 \tag{20}$$

2.2. Measurement Systems and Sensing Elements

As observed in the general closed loop control system BD, the part that provides measured feedback is one of the most important components to pay attention to, if not the most of all. Most theoretical control systems design problems assume either unity, correct, or accurate feedback measurements. Nevertheless, in RT practice, no matter how optimum the controller design is, simply if the sensor does not provide clean and accurate measurements due to noise, communications, and calibration issues, the control system will never behave as desired. From that perspective the importance of studying measurement systems arises. It is not just the sensor, as might be imagined, but a complete set of components forming an entire independent "measurement system".

Measurement systems instrumentation is found to be studied as a complete discipline and specialty itself. It involves the knowledge and techniques of physical properties and elements that are used to measure various RT variables, such as in [27], and how to relate those natural relations into useful signals in order to use them for further processing before transmitting them to control systems or HMI indications. Thus, any measurement system should consist of [27]:

- A sensing element: which is the actual physical sensor attached to the process
- Signal conditioning: which deals with converting sensor outputs into more conditioned, useful, and standard types of signals, i.e. voltage, current, or frequency, to be able to transmit them and use them in further processing. The application in this research and the significance of signal conditioning in control instrumentation is briefly described in a separate upcoming section. An example of signal conditioning elements would be voltage dividers, Weston bridge circuit [27], and analog filters and amplifiers.

- Signal processing: which takes the transmitted conditioned signal and output changes them into a domain where processing such as calculation and computations becomes easier. For example, ADC converts transmitted signals into digital ones in order for a microcontroller to calculate related values extracted from electrical ones so that the output turns out to be ready for display or indication. Fig 2.6 shows a BD of the four phases.
- Data presentation: which is the final stage of the measurement system. This part provides a user friendly reading for the observer, HMI. An example of data presentation would be LCDs, recorders, and computer monitors.

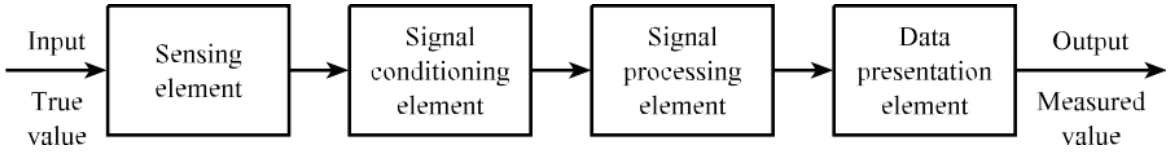


Fig 2.6: A measurement system consists of four main stages (taken from [27])

This research implements a number of measurement applications in the case studied, but before getting into classification of different types of sensing elements, some general basic definitions and concepts should be addressed first. They can be characterized as static and dynamic characteristics of measurement systems [27]. The first measures sensors' transducer performance and signal quality by defining several concepts as:

- Range: the input range of a measurement element is simply all possible values it can have between the max and the min. Thus, an input range is specified by the minimum and maximum values of inputs. For example, a temperature sensor/transducer can be specified to have a range from 0 to 100 °C while its output range is between 0 to 10V. Therefore, an output range is specified by its minimum to maximum set of values.
- Span**:** the span of a measuring instrument is different than the range in some sense. The span is the difference between maximum value and the minimum. Thus, the span indicates

the maximum variations in input or output. Considering the temperature sensor example above, the span of the input is 100 – 0 = 100°C. Similarly, the span of the output is 10V.

- Linearity and non-linearity**:** if the static relation between the input and the output of a certain instrument can be drawn as an "ideal straight line" [27], this element is considered linear. Contrarily, if a different curve is obtained other than a straight line as a result of plotting the output of a given sensor as a function of input, then this instrument is said to be NL. Thus, unless signal conditioned, most real life sensors have static nonlinearity I/O characteristics. However, many of them can be adjusted to be closer to linear. An instrument that is statically linear can be simply modeled as a straight line equation of a slope *m* as:

$$y(x,t) = mx(t) + a \tag{21}$$

  Where $y(t)$ represents the output, $x(t)$ is the input, and a is constant and can be determined by finding line intercept. Non-linearity on the other hand can be measured as a percentage with reference to the ideal linearity [27].

- Hysteresis: hysteresis is defined as the difference between a certain output of a given sensor when the input increases and decreases at a fixed measurement point. Not all instruments involve hysteresis; however, many do. Maximum Hysteresis of a given instrument can be also obtained as a percentage measurement of the output span [27].
- Sensitivity: it is the rate of change of a measuring device's output with respect to its input. Sensitivity is obtained by taking the ratio of the difference between two selected values of outputs over the subtraction of their corresponding inputs, $\Delta O/\Delta I$ [27]. Thus, for a linear sensor model given above, sensitivity is simply the slope of the line, *m*.
- Error band: defines the accuracy +/- error in the instrument output as a result of all effects.

- Resolution: this is used in the case of devices that involve discrete measurement or environment, i.e. gears, rotary encoders, some types of flowmeters that provide pulses frequency outputs, etc. Also in other parts of measurement systems or control signal processing where conversion to digital domain, i.e. ADC, occurs, the term resolution is used more frequently. It is then a measure of how close to continuity the discrete measurement is compared to the supposed continuous behavior [27]. The higher the resolution, the closer to continuity, and so the better the measurement device.

Most sensing elements and instruments have dynamic response and characteristics besides the mentioned basic static ones. In other words, countless sensors can be demonstrated as two series components: static gain and dynamic. The dynamic characteristics of sensors are usually modeled as a SISO TF of a continuous time (analog) Laplace transform. Most sensor dynamics responses follow either one of two models: first order or second TF [27]. The first order sensor TF is given by:

$$G(s) = \frac{O(s)}{I(s)} = \frac{\Delta O}{\Delta I}\ \frac{1}{1+\tau s} \tag{22}$$

Where τ here is the time constant of the first order element dynamic response and $\frac{\Delta O}{\Delta I}$ is the steady state static sensitivity. For an ideal (statically linear) element, $\frac{\Delta O}{\Delta I}$ represents the slope of the line. Therefore, a more general notation of (22) approaches as:

$$G(s) = \frac{O(s)}{I(s)} = K\ \frac{1}{1+\tau s} \tag{23}$$

Where K is known to be the static gain. Similarly, the second order sensor dynamic element is given by:

$$G(s) = \frac{O(s)}{I(s)} = \frac{1}{K}\ \frac{1}{\frac{1}{\omega n^2}s^2+\frac{2\zeta}{\omega n}s+1} \tag{24}$$

Where ζ is the damping constant introduced earlier and ω_n is known as the natural frequency of the measured signal, $\omega = \frac{1}{\tau} = 2\pi f$.

Here it is critical to mention that there are techniques in which some of NL non-ideal effects as well as interfering environmental effects on measurement instruments can be adjusted and improved, other than signal conditioning techniques coming later. Those error correction methods stand for example: isolation, using differential measurement, using opposed environmental inputs, and using high gain negative feedback [27].

In general applications, measurement systems' sensing elements can be classified based upon the nature of the physical and conceptual type of measurement as: resistive, capacitive, inductive, electromagnetic, thermoelectric, elastic, piezoelectric, piezo-resistive, electrochemical, and others [27]. Nevertheless, besides common measurement techniques, concepts, and sensing elements used in industry, this section emphasizes methods and sensors applied here in this research. They are displacement and distance sensors, rational position measurement, and other instrumentation.

2.2.1. Displacement and distance sensors

This research implements the practice of linear position or distance measurement. Position measurement appears to be essential in robotic and motion control types of application. Position sensors are also found in elements in the control of actuators of both types: linear and rotary. This subsection concentrates on the linear position measurement (distance). There exist several ways of position sensing in general. Four main types of position sensors can be categorized as:

- Resistive
- Optical
- Inductive

- Sound based (Ultrasound)

The simplest form and most common resistive type position sensors are potentiometers. They can be further divided into: rotary potentiometers and linear resistive wires as in Fig 2.7. Rotary position measurement is described in the next subsection. However, in the linear resistive wires, the change in position produces slight step variations in the output that follows the concept of linear variable resistance exemplified in [27], [51].

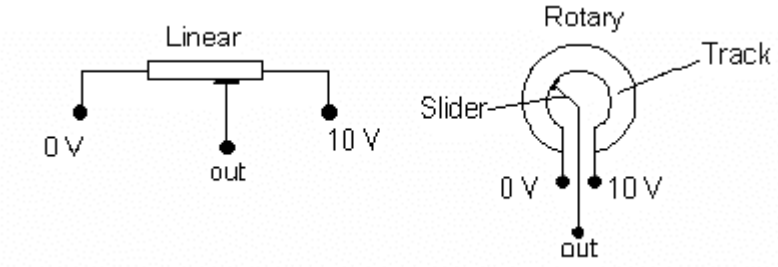


Fig 2.7: Linear and rotary resistive sensing elements (potentiometers)

Inductive position measurement uses the concept of inductor's reluctance and magnetic flux [27]. This method is known as "the wire wound type." How good the analog measurement is in such a method depends on how fine the wire is besides how closely it is coiled and attached to the track. Regarding the other two types optical and ultrasonic, both are known as depth gauges where they are not only used in distance sensing, but also level measurement, as in process tanks. The ultrasonic type is briefly covered within presenting level measurement in a coming subsection. However, this part focuses on the optical type of displacement, namely IR distance sensors, measurement as it is utilized in the implementation of the case studies of this research.

Optical type of measurement follows light radiation properties of covering several wave lengths. Thus, based on wavelength ranges, optical devices can be classified into [27]:

- Visible (0.4 μm to 0.7 μm).
- IR (0.7 to 100 μm).

- UV (0.01 μm to 0.4 μm).

Any optical type sensor should consist of at least three main components: a source, transmission medium, and detector [27]. The basic concept of optical measurement is not limited to distance sensing as the power of the transmitted signal, *S(t)*, from the source is a function of wavelength, λ. Thus, the amount of signal power, "proximity," at the detector, RX, end after reflection equals:

$$P(\lambda) = \int_0^{\infty} s(t,\lambda)\partial\lambda \tag{25}$$

Indeed, Fig 2.8 below shows if the transmission optical between the source (TX) and a reflecting body occurs in a solid linear (horizontal or vertical) fashion, the distance will be nothing but the amount of energy between the two ends such that:

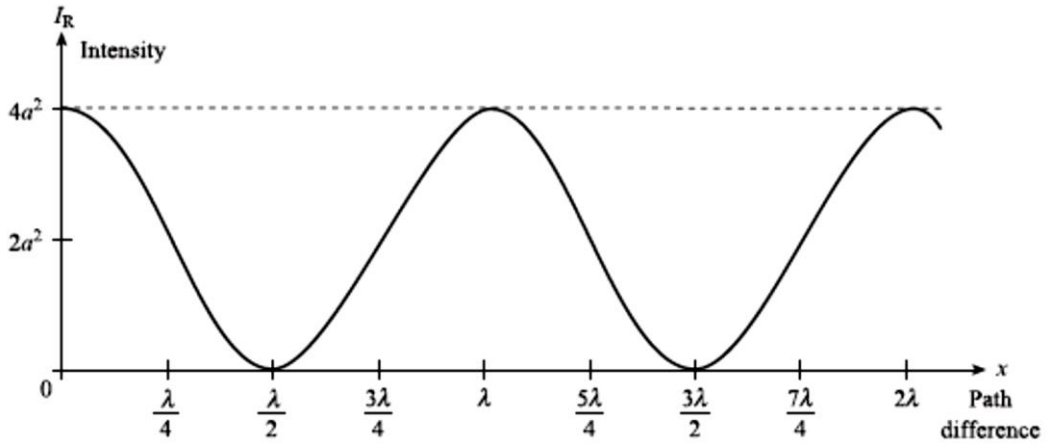


Fig 2.8: Linear distance between the source (emitter) and an object in IR sensing (taken from [26])

$$P(\lambda) = R(\lambda)Cos\theta = R(\lambda) = x \tag{26}$$

For $\theta \approx 0$, where x here denotes the distance and R represents the radius from which the source transmits until a surface area as general notation used in case of other than linear displacement is aimed as in Fig 2.8. A more general form involves some geometry in the coupling between sources and detectors. Fig 2.9 underneath shows one common emission form, using LEDs sources: Lambertian emitter [27].

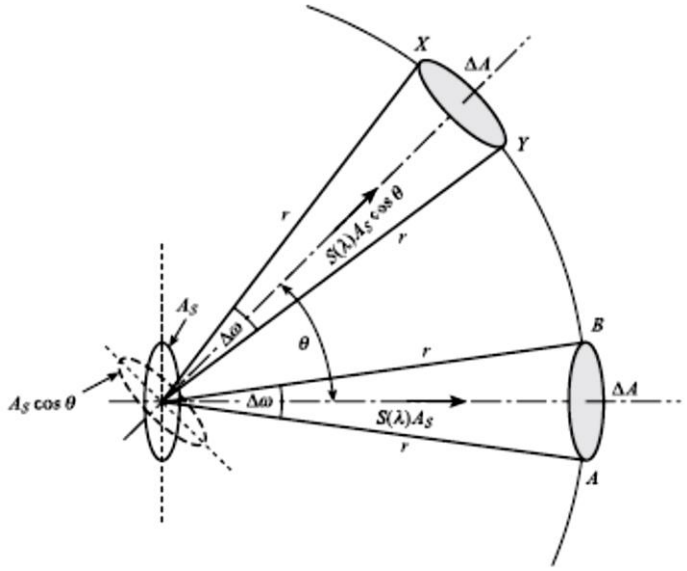


Fig 2.9: The linear displacement and one common form of emission, Lambertian emitters (taken from [27])

Thus, there appear several different types of sources and detectors such as:

- Hot body sources: these kinds of sources use the fact that any object body at above 0 K in temperature emits radiation. Hence, black bodies are considered the ideal emitters [27].
- LEDs: these are commonly implemented sources in commercial distance sensors. LEDs consist of a PN junction semiconductor element (diode), and they follow the Lambertian form of emission shown above. Some LEDs transmit IR waves while some others emit visible light. Both are found in distance sensors found on the market.
- Laser type of sources: Laser sources are further classified into several types based on their lasing medium: gas, liquid, solid crystal, or semiconductors [27].

Also there exist two kinds of detectors, RXs:

- Thermal: this type uses the fact that the power emitted from hot sources is also a function of temperature. Here, the detector heats up when it receives the signal power. This heating temperature is usually above the surrounding one. The detector then acts as a resistive or

thermoelectric temperature sensor with either resistance or voltage out depending on temperature difference [27].

- Photon detectors: this is the type used in this implementation. There are two classes of this type: and Photovoltaic detectors. In the first, a PIN junction, excited electrons at the RX increase in electrical conductivity and a decrease in electrical resistance upon increasing in transmission power (proximity) [27]. On the other hand, Photovoltaic detectors are photo PN junction semiconductor diodes, known as photodiodes. Their output power simply increases with increasing the current flowing through them. Hence, the reverse proportion actually happens here; with increasing the power signal detected, the sensor's output current increases.

Regarding the transmission medium, it can be gas, liquid, or solid [27]. In most frequent applications air, water, or fiber optics represent those mediums. The general concept involved with transmission medium is that each media has its own absorption as well as transmission features. They are analyzed by absorption and transmission spectrums. Absorption spectrums are defined as the power absorbed by over the power of the signal entering the medium, both of which are functions of λ. Oppositely, transmission spectrum represents the ratio of the power leaving to the one entering the medium [27].

Coupling sources to detectors encompasses different geometry rules and methods. Three methods of coupling are focusing system, optical fiber, or combination of both [27]. This paper exposes the use of IR distance measurement through implementing the case study. Fig 2.10 below shows a common IR sensor/transducer that uses LED type of sources and photovoltaic detectors. The output of the sensor transducer, after the signal conditioning amplifier, is a voltage standard signal, commonly 0- 2.5V, 0 -5V, or 0 -10V.

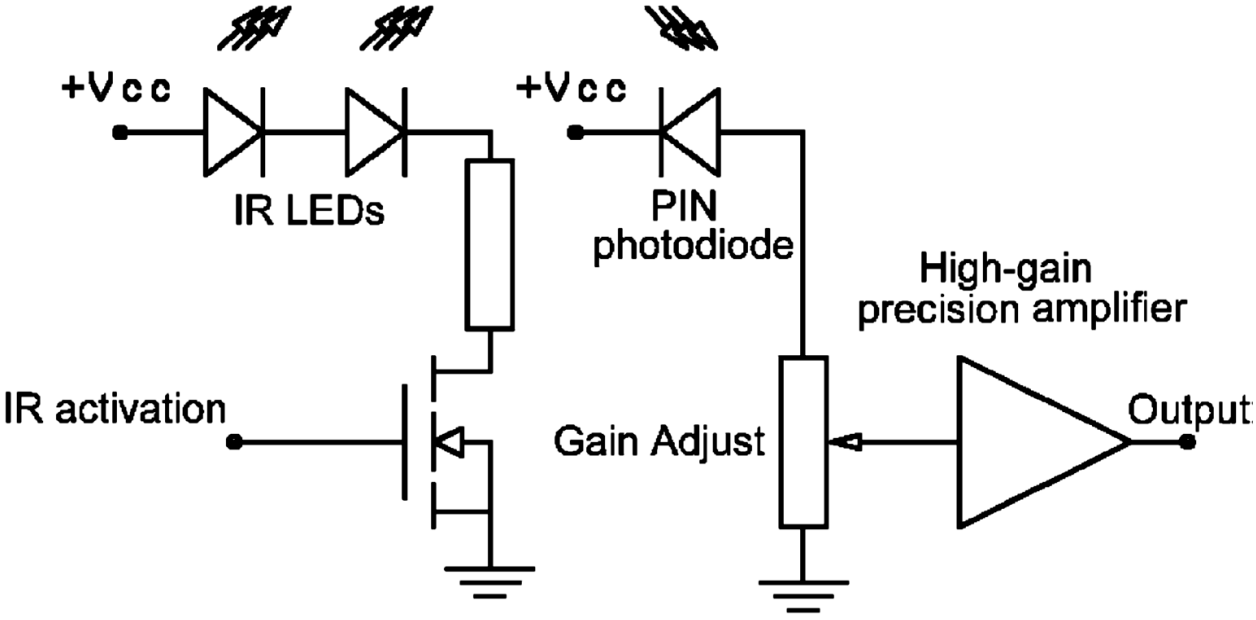


Fig 2.10: The IR proximity sensor used in the implementation (taken from [52])

Using principles illustrated above, a more uniform IR distance sensor/traducer model as a function of distance, and the phase angle of the reflecting surface is given by [52] and can be rewritten as:

$$V(x,\theta) = \frac{(As)(K)(G)(\mathcal{R})}{x^2}\cos\theta + \beta \tag{27}$$

Where x represents the distance measured and As, K, G and R are all constants denoting the intensity amplitude of the transmitted signal, the static sensitivity $K=\Delta O/\Delta I$ of the detector, the amplifier gain, and the reflection coefficient of the surface of the measured object respectively. β here signifies both the amplifier's offset and ambient light effect. By applying some error correction and signal conditioning methods, the effect of β can be minimized in a cleaner environment. Also the set of the four constant coefficients can be combined into one term called C_1, where $\mathcal{R}$ varies between 0.1 (black body) and 1 (white surface). Therefore, assuming measuring the distance of a white surface, in addition if linear displacement is obtained then (27) turns into:

$$V(x,0) = \frac{C1}{x^2} \tag{28}$$

Solving for x as a function of the transducer output signal gives

$$x = \sqrt{\frac{C1}{V}} \tag{29}$$

2.2.2. Rational position measurement and encoders

As seen before in Fig 2.7, a potentiometer is a variable resistance. Depending on the size and the length of the resistance material, the voltage applied over its ends may produce an output, using voltage dividers for example. A slider change in a rotary movement along this resistor makes it variable. The track (resistance body) may be made from carbon, resistance wire, or piezo-resistive materials.

Besides the resistive potentiometers, rotary motion in general including angular position, speed, and acceleration are commonly measured by either one of the following principles [27] that most of them produce electrical outputs such as:

- Optical measurement type: Similar to optical linear displacement measurement described above, this type uses light beams and light sensitive cells. The light beam can be either reflected or interrupted as pulses train production per revolutions. Encoding these pulses requires counting over time intervals so that angular position, speed, and even acceleration can be obtained. These counted signals may pass through conditioning and processing units in which time pulses are converted into useful standard analogue or digital outputs.
- Magnetic Pickups: Also similar to the type of distance measurement using induction presented above, this type uses inductive coils located close to the rotating body. By mounting a small magnet on the rotating body, pulses can be generated. Each time the magnet passes the coil, a pulse is generated. Some rotating bodies' material is ferrous, then magnet becomes required this case, where a tiny notch, hole, or any shape of

discontinuity in the surface causes change in the magnetic field, therefore generating pulses. Then pulses might be processed to produce standard analog or digital output.

- The third common type arise Tachometers: they tend to be unlike the previous as they produce analog outputs directly. Two classes of tachometers exist, A.C. and D.C. The first generates voltage directly proportional to the speed while the second generates sinusoidal outputs. In the AC type, the speed of rotation represents nothing but the frequency of the voltage signal. This frequency is then counted and processed. Both types attach the to back of rotating bodies. Hence, Tachometers are most commonly found built-in along with electric motors for the sake of measuring their speed or position.

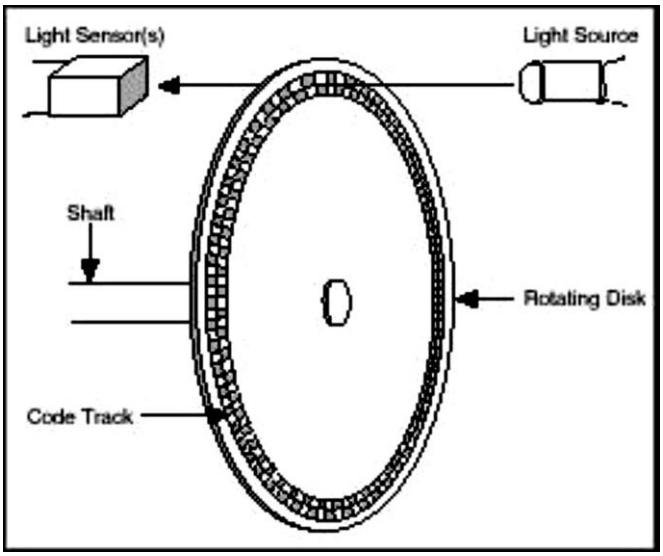


Fig 2.11: Rotary encoders implement the concepts of optical measurement and hall-side wall effect (taken from [23]) to generate pulses counted to interrupt position or speed

This paper exposes the application of rotary encoders to measure angular position in the ball on beam case study presented later. Encoders are electromechanical devices that measure position in motion systems [53]. They can be mechanical, optical, magnetic or capacitive as per measurement implementation technology. However, most common types of encoders apply the optical technology, then magnetic. Optical encoders use the first rotation measurement type

shown in Fig 2.11 above, which is the same concept of LED sourcing, light detectors, and a disk attached to the rotating shaft explained previously besides the Hall Effect, [27]. The square wave pulse sequence outputs are interrupted as angular position, θ, as per the following [53]:

$$\theta = \frac{Counts}{xPPR}.360° \tag{30}$$

Where *counts* represent the counter value as a result of pulse output interruption. This number is usually given as a device specification in the datasheets as CPR. There exist varieties of ranges of encoder CPR resolutions. They vary from 100 to 6000 CPR [53]. The higher the CPR number, the higher the resolution and, of course, the more precise the measurement. CPR and PPR could be related as:

$$PPR = (CPR)(x) \tag{31}$$

X here, as well as in the equation above, denotes the encoding type.

There are three encoding types: x_1 (single), x_2 (double), and x_4 (quadrature) illustrated in Fig 2.12 (a), (b), and, (c), respectively [53]. Regardless of the measurement technology implementing them, encoders can be also categorized as either absolute or incremental. Absolute encoders have multiple predefined set codes of various binary increments providing data words representing absolute angular positions in a single revolution. Absolute encoders provide position information immediately even when power turns off. Since the relationships between input (position) and outputs (coded words) are predefined during assembly, this type of encoders does not require periodic recalibration.

The other type of encoders is incremental encoders, which is the type realized by the equations above as well as the figure above. It is the also utilized in the implementation of the case study, the ball on beam control system coming after in Chapter 3. Incremental encoders operate by providing two pulses train outputs *A* and *B*. Unlike absolute ones, pulse trains do not

provide counting information independently on their side the in incremental encoder. Rather, an external up/down counter is usually implemented whether as H/W electrics IC or S/W code. Hence, in order to provide useful position information, the encoder positions should be referenced to the rotating device to which it is mounted. This usually happens by using index pulse, *Z*, besides the two square wave trains, *A* and *B*. Incremental encoders in summary can be distinguished by recording incremental changes in position of the rotating device to counting information [53].

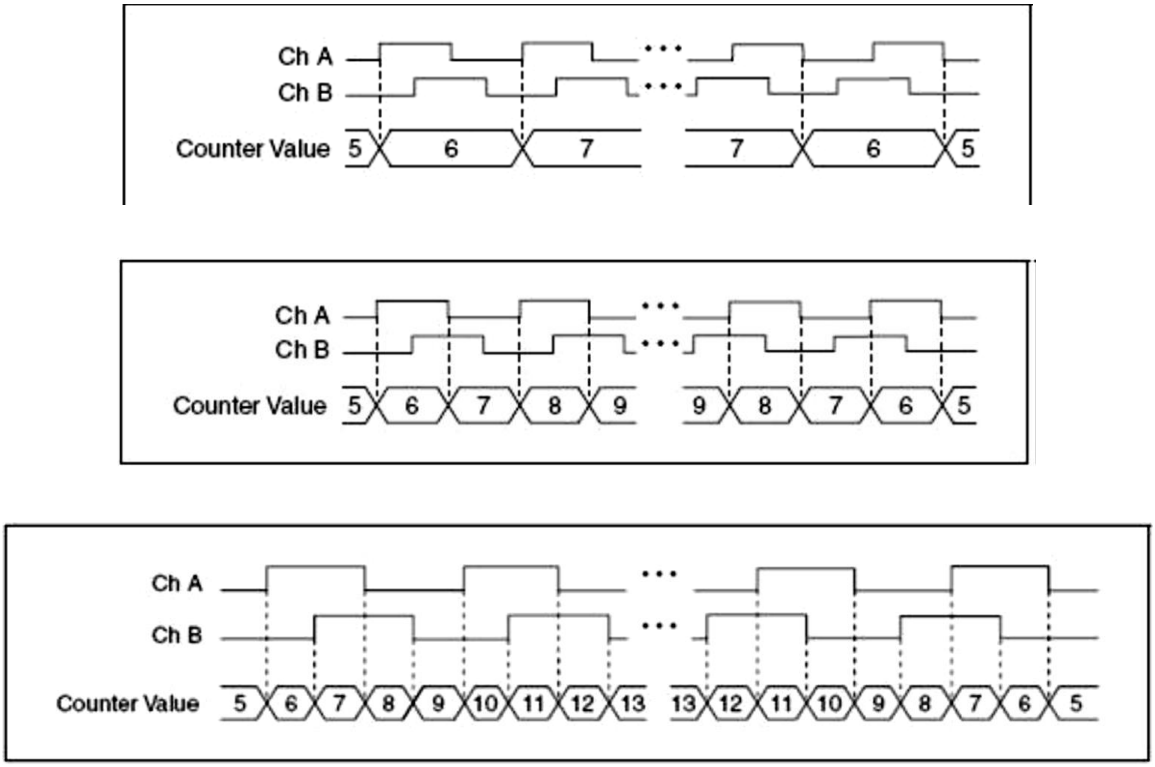


Fig 2.12 (a) Incremental x_1, (b), x_2, and (c) x_4 encoding and counting types (taken from [53])

Hence, this incremental counting information could be used as not only position readings, but also velocity and acceleration [53]. There exist two common ways of encoding (counting) encoder signals: binary and gray coding [53]. Encoders tend to be noisy; thus, differential measurement could be used along with incremental encoders to provide more noise and interference immunity [53]. There, a set of six signals is obtained, which, in addition to power and GND, add up to eight: A^+, A^-, B^+, B^-, Z^+, and Z^-.

Furthermore, not only do encoders provide rotary angular position measurement but they can be used also in linear motion applications. This can be determined by the way pulses are interrupted when encoders are attached to devices that move in linear motion, wheels for example. Linear encoding follows the same principle of the rotary ones; however, the mechanical construction part varies from the first in addition to the relation of counts to the linear distance and the tilt angle of the wheel. More information about linear encoding is given in [53].

2.2.3. Other Common Industrial Instrumentation

Four to five most common RT variables measured in process industry are temperature, pressure, flow, level, and force. Sensing elements and measurement methods of such variables appear standard in industry, so they should be briefly described even though most of them have not been implemented in this research.

2.2.3.1. Temperature

Starting from temperature measurement, it follows basic physics thermodynamic principles asserting that a change in temperature can cause deviations in some physical and molecular properties of some materials. For instance, temperature may either cause changes in electrical resistance or metal expansion. Thus, two main temperature sensing types are resistive and thermoelectric (thermocouple) [27]. There exist two main resistive temperature sensors: RTDs and thermistors [27]. RTDs follow the basic law relating resistance as function of temperature, *T*, in a polynomial form,

$$R(T) = R0(1 + \alpha T + \beta T2 + \gamma T3 + \dots) \quad (32)$$

Where α, β, and γ, constant temperature coefficients, and *Ro* is the resistance at 0°C. Another type of resistive temperature sensors are thermistors. They are made of small semiconductor pieces and follow an exponentially proportional relation (in Kelvins):

$$R(T) = ke^{\frac{\beta}{T}} \tag{33}$$

Here k and β are specific thermistor constants.

Thermocouples, on the other hand, operate as per the basic principle that two wires with dissimilar electrical properties are joined at both ends; thus, when one junction is made hot and the other cold, a small electric current proportional to the difference in the temperature is produced [27]. This temperature difference ($T_1 - T_2$) relation is expressed as voltage output EMF function linearly approximated as:

$$V_{emf} = \alpha\,(T_1 - T_2) + \beta\,(T_1^2 - T_2^2) + \ldots\ldots \tag{34}$$

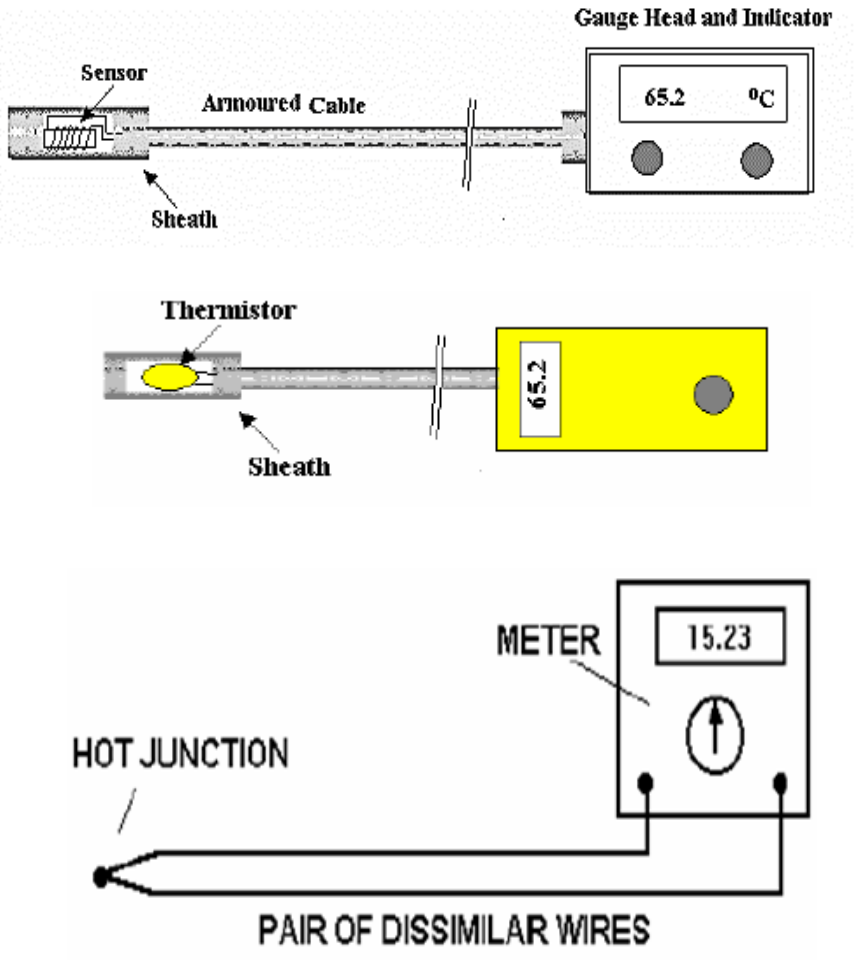


Fig 2.13: From top to bottom, (a) RTD (b) Thermistor and (c) Thermocouple based temperature measurement systems

Fig 2.13 above shows the three common temperature sensing system types. Aside from temperature sensor types mentioned above that are compatible with electrical/electronic environment of systems, there exist numerous other kinds that provide mechanical outputs. Those types of sensors were used in the pneumatic control systems introduced. However, most of them are still realized in the process industry as local field indicators besides the HVAC applications. For example, such systems include liquid expansion and heat pressure thermometers. These types of sensors are filled with either liquid, i.e. mercury or evaporating fluid where changes in temperature cause expansion or evaporation in those indicating materials. Other forms are bimetallic types of sensors and glass thermometers.

2.2.3.2. Pressure

Another essential industrial RT PV is pressure. Pressure transducers could be classified according to sensor type of output as either convert the pressure into mechanical movement or electrical signals. Four known examples of the mechanical pressure transducer are: bourdon tubes, spring and piston method, bellows and capsules, and diaphragms as appear in Fig 2.14.

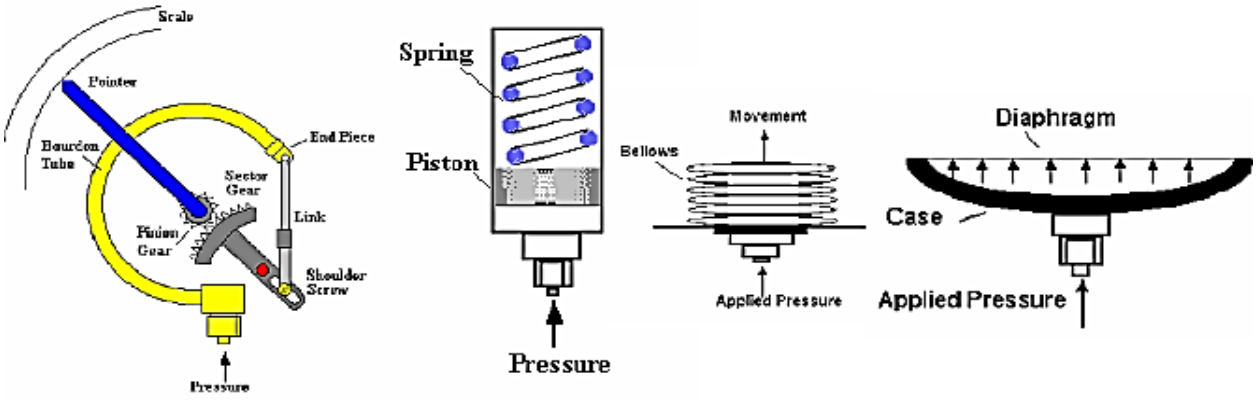


Fig 2.14: From left to right, (a) bourdon tubes, (b) spring and piston, (c) bellows and capsules, and (d) diaphragm pressure transducers

More importantly, there appear various ways of converting the mechanical movement of the preceding types into an electric signal such as the following:

- Strain Gauges [27]: they are small elements attached to surfaces being strained. Some typical pressure transducers associated with strain gauges would be diaphragms that bend under applied pressure. When the length of the element changes, as a result of applied pressure, the outcomes change in electrical resistance this could easily be signal conditioned and further processed into voltage afterwards. Hence, the mechanical strain of a surface is defined as

$$Gm = \Delta L/L \tag{35}$$

Similarly, the electrical strain is

$$Ge = \Delta R/R \tag{36}$$

L and *R* represent the length and the resistance of the gauge respectively. Then a constant relating both directly proportional to each other is called the gauge factor such that

$$Gauge\ Factor = \frac{\Delta R/R}{\Delta L/L} = L\ dR/R\ dL \tag{37}$$

- Piezo-electric [26] types: These elements use crystalline material pieces that produce electrical signals (charges) when mechanically stressed. Those electric charges can be easily interrupted as voltage or current. This principle is used in microphones for example.
- Others: other effects are mostly electrical, namely capacitive and inductive types [27].

2.2.3.3. Flow

One very important PV found in most kinds of process plants is flow. Whether gases, liquids, or both, volume as well as mass flow rates appear very essential information in order to monitor and control chemical, fluid, and thermal processes. Flow measurement in general relies on several basic physical principles as the relationship between fluid flow and pressure, temperature, viscosity, density and specific gravity. A very basic equation [27] relating flow rate, *Q*, to fluid's velocity, *v*, in a pipeline of a cross sectional area, *A*, is:

$$Q = Av \tag{38}$$

Another important basic principle is Bernoulli's theorem and Reynolds number [27].

There are a countless number of flow measurement types. They vary depending upon their principle of operation, make, and application. One rough classification [27] could follow:

- Positive displacement (flow quantity)
- Inferential types of measurement.
- DP and variable area,
- Mass flowmeters

Positive displacement flowmeters should have mechanical elements that make shafts rotational speed and number of revolutions directly proportional to the flow of fluid. Those speed and revolutions can be then measured using mechanical or electronic devices from which signals after that are processed further to display flow readings. Examples of such measurement type are rotary piston, vane, lobe, or meshing rotors (Fig 2.15) reciprocating piston, and fluted spiral gear flowmeters.

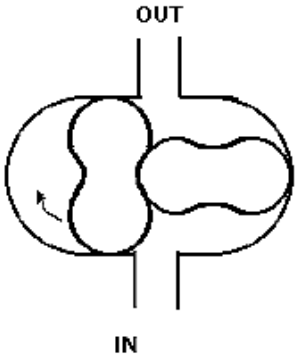


Fig 2.15: Meshing rotors flowmeters type consist of two rotors with lobes

Unlike positive displacement method that outputs speed values directly proportional, when flow of the fluid becomes contingent or indirect from some other natural effects produced, this type of measurement is called “inferred” [27]. Usually, a spinning rotor’s speed is measured either

mechanically or electronically then processed. Main examples of such flowmeters are: turbine rotor, rotary shunt, rotating vane, helical turbine, and many other types. The following lines describe some common types of flowmeters that group under the inferential category of measurement.

- Turbine flowmeters (Fig 2.16): which have axial rotors that spin by applying fluids while their speed represents the flow rate. An electrical rotational speed sensor (tachometer described before) is usually coupled to the shaft of the rotor to sense the speed where a magnetic slug is fixed that generates pulse outputs.

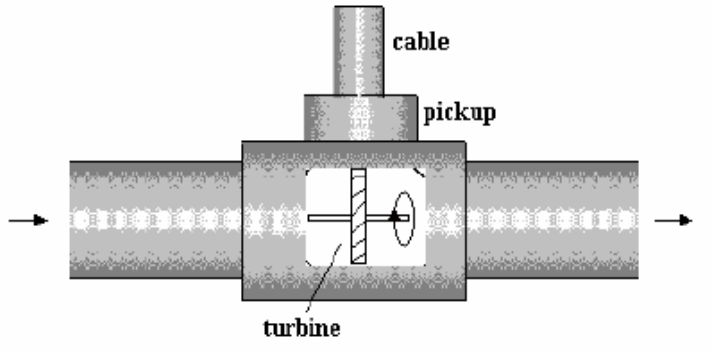


Fig 2.16: A turbine type flowmeter.

- Rotating vanes (Fig 2.17): this is similar to turbine flowmeters in some sense. The fluid flowing spins around the rotating vane where the speed of the rotor is measured mechanically or electronically [27].

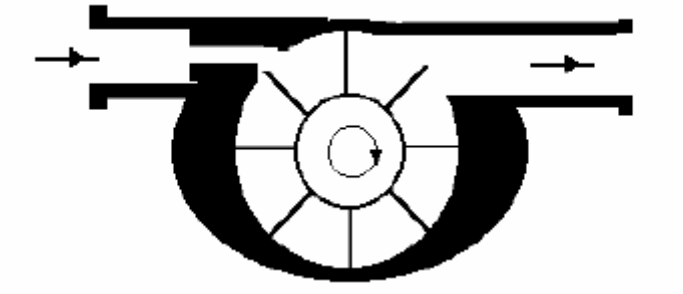


Fig 2.17: A rotating vane.

- Magnetic flowmeter: magnetic flowmeters are based on the principle of Faraday's Law of Induction; it states that "the voltage induced across any conductor as it moves at right angles through a magnetic field is proportional to the velocity of that conductor" [27] such as the following:

$$E = \frac{1}{c} BDv \tag{39}$$

Where *E* is the induced electrode voltage, *B* denotes the magnetic field strength (flux), *D* is the diameter of the pipe, *v* is the flowing velocity, and *c* is a constant. Then the volume flow, Q, can be obtained, after some derivation, from:

$$Q = \frac{\pi cD}{4} \cdot \frac{E}{B} \tag{40}$$

- Vortex Flowmeters: vortex flowmeters come as three different types commercially available: swirl-meter, vortex shedding, and fluidic meters. The swirl-meter for example operates on the principle of vortex precession [27]. Vortex swirl-meters are digital volumetric devices that have no moving parts. Pulses form frequency outputs proportional to fluid flow rate are and produced as outputs [27]. Vortex shedding on the other hand appears slightly different. The basic general principle they operate on is the Karman Vortex Shedding Street Principle [27] used to measure the volume flow rates of gases and vapors. The vortex flow meter is based on measuring the velocity average of the flow in the pipeline. Everything works according to the Karman Vortex Shedding Street Principle where frequency, *f*, of the vortex shedding is proportional to the flow velocity, *v*, as:

$$f = \frac{Sv}{w} \tag{41}$$

Where the non-dimensional Strouhal number, *S*, describes the relationship between vortex shedding frequency, *f*, the width of the body, *w*, and the average flow velocity, *v*, output.

- Ultrasonic Flowmeters: the measurement of flow rate can be determined by variation in parameters of ultrasonic oscillations in ultrasonic flow measurement. The basic principle depends upon the well-known speed of sound wave propagation, set distance (pipe diameter for example), and time instances that provide information about frequencies that are proportional to the flowing velocity, therefore flow rates. There exist two main usable types of ultrasonic flowmeters: Transit time and Doppler flowmeters [27]. The first type uses two ultrasonic sensors (2 TXs and 2 RXs) that are clamped at two opposite sides of the pipe with an angle, θ,with respect to the pipe axis. The flow is measured by estimating the average time taken for the two transmitted/received ultrasonic waves to travel across the pipeline section, both with and against the flow of liquid within the pipe. The difference in the two time intervals is proportional to the flow rate measured.

  The second realizes Doppler's principle [27]. Unlike the transit time type, it uses one ultrasonic transceiver only. The TX transmits continuous signals into the fluid. As a result, particles included in the flow such as gas or air bubbles besides disturbances usually reflect part of this sent signal. The remaining part of the signal reaches the edge of the pipe, then reflects. This reflected signal is then sensed by the TX transducer and the difference between the transmitted and received frequency is calculated. Here frequency is directly proportional to the liquid's velocity.

One very broad category of flow measurement in general is DP (variable head) based besides variable area methods of measurements. Both could be considered inferential, since flow is not measured directly but inferred from other indirect relations. Here it is either DP or variable area that is primarily aimed from which flow rate can be obtained or sensed accordingly. However, since both are widely used, especially DP (variable head), in industrial process plants

as common methods to measure flow rates inside pipelines, it is preferred to count them as a separate category.

DP flow measurement follows the principle that when the flowing fluids in the pipeline are subjected to a restriction or obstruction, they produce DP that is proportional to flow rate across this restriction or obstruction element [27]. Thus, the very basic relation is

$$Q \propto \sqrt{h}$$

This relation shows that the flow rate is proportional to the square root of the head, *h,* or DP. However, the details of the upper relation in which to algebraically relate the head and other associated parameters to the output flow may vary according to the method in which the DP is created inside the pipe beside the way of measuring the pressure. Thus, any DP flowmeter should consist of two parts: a primary and a secondary.

The primary elements are actually the parts that are used to restrict fluids' flow in pipelines in order to produce DP whereas the segments that sense this pressure difference using any of the measurement types introduced before are called secondary. To remind, the secondary element could be a manometer, a bellow, or a diaphragm transmitted mechanically, electrically, electronically or even pneumatically. Regarding the primary element, the most well-known and common method implemented is the Orifice Plate [27]. It is a thin, circular metal plate with a hole in it, and it is held in the pipeline between two flanges called orifice flanges. This method might be the simplest amongst other pipeline restrictions used in the DP method of flow measurement.

Segmental Orifice, Concentric, and Eccentric orifice plates compose three different types of this part as seen in Fig 2.18. In general, the fundamental equation of the DP, ΔP, flow, Q, measurement using orifice plates as primary elements develop the following where:

$$Q=\varepsilon A_0\sqrt{2g\Delta P} \tag{42}$$

Where ε is the efficiency, A_0, the area of orifice in square feet, and g is the acceleration gravity.

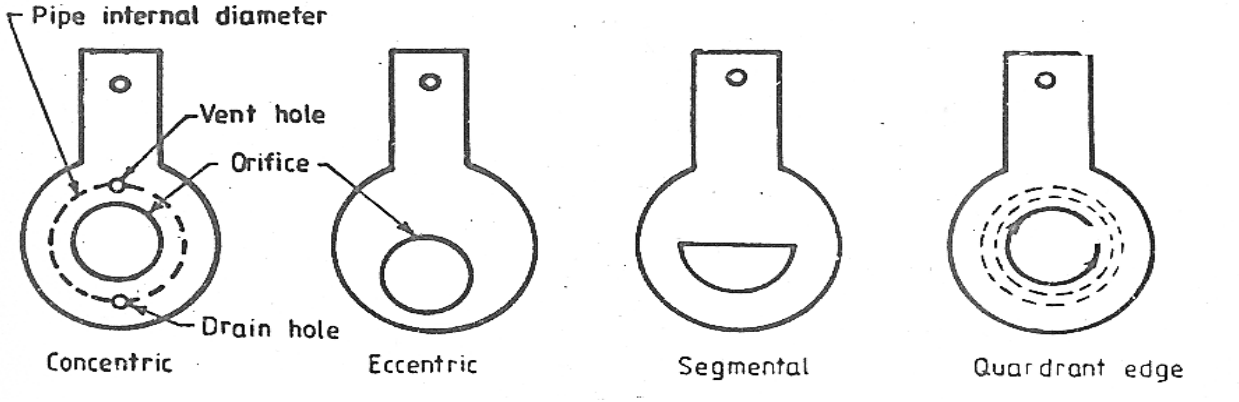


Fig 2.18: Orifice Plates

Other forms of primary elements of DP transmitters are venire tubes, flow nozzles, pitot tubes, annubars, elbow taps, weirs, flumes, open nozzles, and many others [27].

In addition to the variable head, DP, method briefly described, there are variable area flowmeters. As a distinction between both, in the DP method, the flow restriction is of fixed size while the pressure difference across varies. However, in the Variable Area flowmeters, it is the other way around; the differential pressure is made constant and the size of the restriction varies. There are two main types of variable area meters: Float (Rotameter) and Tapered plug.

Furthermore, there exist types of flow-meters that are mass flow based. They directly measure the mass flow rate of a flowing substance and not volume one where:

$$Q_M = Q_V \times \rho \tag{43}$$

Here Q_M is the mass flow rate, Q_V represents the volume flow rate, and ρ is the fluid density. One very famous mass flow meter in industry is the Coriolis meter. It works as per principles of Newton's second law of motion and the Coriolis Effect [27] where sensors contain vibrating tubes driven by electromagnetic coils all enclosed in a housing. The mass flow rate is proportional to the tube's frequency of twisting measured by electromagnetic velocity sensors.

2.2.3.4. Level

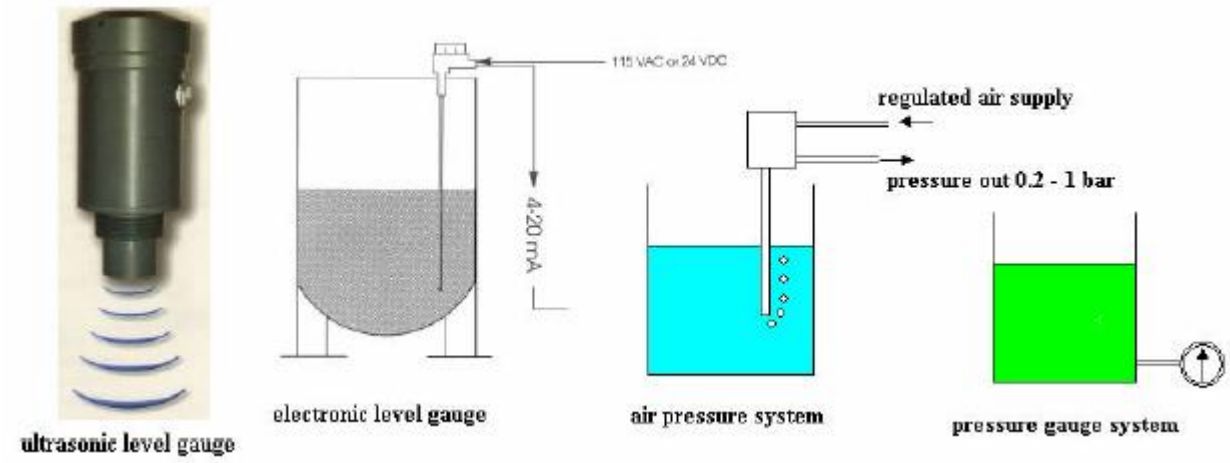


Fig 2.19: Several methods of level measurement.

Fig 2.19 above shows some methods of level measurement which represent one of the most essential and common processes to control in industrial as well as other applications. Those gauges measure the depth of liquids and powder in tanks. They use various principles and produce outputs in either electrical or pneumatic forms. The type to use depends on the substance in the tank. Level measurement in general could be classified into two main categories which themselves are further divided into types:

- Direct method: float type and visual level sensors are both considered classified under the direct way of level measurement.
- Indirect methods: this could be divided into three main types:
    - Level measurement based on displacement (distance position) level sensors
    - Head pressure based level measurement: examples of this sort are gauge and DP measurement.
    - Electrical types: recognized examples of electrical level measurement are the capacitive, inductive, ultrasonic, and radar gauge level measurement types.

One very common level measurement example in industry practices is DP. To determine the liquid level in a tank, the transmitter should be either connected to the high pressure side or the low on of the container. Depending on the take's configuration, the low sides may vent to the atmosphere sometimes. The DP represents the height of the level in the tank multiplied by the specific gravity of the liquid, and so, the output is proportional to the liquid level.

2.2.4. Calibration of measuring instruments

An experiment in which the static characteristics of a particular measuring instrument are obtained is called calibration. This can be done by plotting the output of the device as a function of input after measuring corresponding values keeping in mind environmental inputs and other effects. In industry, this experiment is usually executed at specialized labs as the results must be accurate with respect with some standards and techniques. However, here in this research manual educational laboratory calibration experiments are performed, and results are compared with the manufacturers' sensors datasheets and other previous literature. The results may not be so precise though, but the calibration experiment step is important in order to know the static relation of sensors inputs and outputs used, to reflect this in the processing programing S/W, and to determine whether further signal conditioning is needed.

2.3. Signal Conditioning and Processing in Measurement and Control Instrumentation

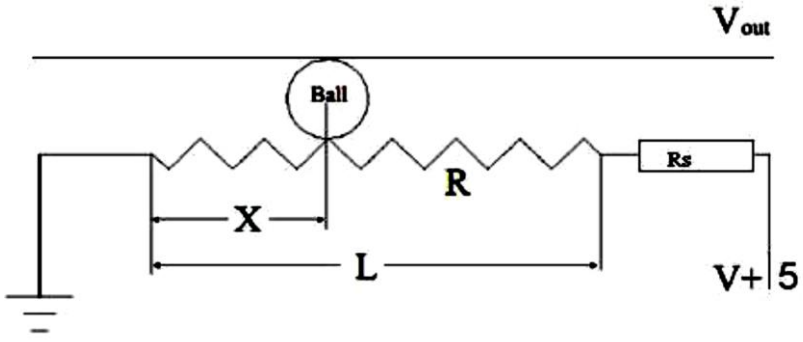


Fig 2.20: An example of signal conditioning to a linear resistive type position sensor (taken from [51])

Fig 2.20 above shows one of the simplest and abstract examples of signal conditioning methods. In the example, a ball displacement position, *X*, is sensed by a resistive wire sensor. Supplying the two ends of the wire power terminals and applying Ohms law yields varying outputs of resistance. However, adding this variable resistor to a voltage divider circuit of a fixed resistor, *Rs*, conditions the original sensed signal and converts it into a more usable form that would simplify dealing with it in further processing as voltage outputs such that:

$$V_{out} = \frac{R}{Rs+R} V_{supply} \tag{44}$$

Considering the sensor variable resistor *R* as an input and $Vout$ is the output.

Besides signal conversion and adaptation, there appear motives why signal conditioning in analog domain is crucial in measurement and control instrumentation as [27]:

- RT signals are usually exposed to noise and interference, consequently effects as distortion, harmonics, fading, and multipath due to many root causes as the harsh environment for example; therefore, steps like filtering and isolation are essential.
- Most original sensors outputs are small and not compatible with the standard ranges of neither the communication transmission, i.e. 4 - 20mA or 0 – 10V, nor AIs/ADCs DAQ cards requirements and resolution. Thus, amplification to those signals is compulsory.
- As seen in the non-ideal characteristics of sensing elements and transducers, many of them are associated with such effects. Therefore, correction techniques, such as linearization, I/Os impedance matching, and others, become beneficial and preferable.

2.3.1. Analog Signal Conditioning

In general, there are various purposes and applications of signal conditioning in control instrumentation. They could be mainly categorized as per their application in this report as filtration, small signal amplifications, and others.

2.3.1.1. Filtrating (smoothing)

Most if not all RT sensors signals accompany noise. There are two main sources of noise: external and internal [27]. Examples of external sources of noise would be AC power circuits and lines nearby, electromagnetic fields, wireless channels and interference, and rotating machines, i.e. motors, turbines, and generators. Internal sources are commonly random temperature that encompass thermal noise [27]. Noise is generally modeled as a statically distributed system, usually WAGN. There are solutions to minimize noise effects. The first always thought of and the most common of all is filtering. This phase prevents or cuts out unwanted frequency components of the signal according to predefined designed of points called filters parameters. Filters help in overcoming the effect of external noise sources besides the aliasing phenomenon [54], [56]. Also, there are different modes of filters that vary according to the application and purpose of filtering: LPF, HPF, BPF, and notch filters [55], [56].

Usually LPFs are applied right after sensing in measurement and control instrumentation and DAQ. Analog LPF passes defined continuous set of frequency components of signals lower than a chosen or designed limit. LPF could be designed as a 1st, 2nd, or even higher order of dynamic systems, and they are implemented as analog circuitry. The simplest form would take an RC (1st order) or an RCL (2nd order) circuit shape. This designed limit after which higher components are attenuated or filtered out is called cutoff or corner frequency, f_c, where in its simplest form:

$$f_c = \frac{1}{2\pi RC} \tag{45}$$

R and C are designed resistors and capacitors where $RC = \tau$, the time constant of the first order TF given in (23). Op Amps can also implement LPF and other forms of filters listed above as active given in [54]**.** Filters that are implemented by Op Amps are called active while RC or RCL

ones are considered passive [54]. More details about the applications of Op Amps in instrumentation and control are given in the next section. In control system applications, filtering out noisy signals benefits in smoothing the overall closed loop performance. Thus, when the measurement feedback signals tend to be noisy, the control oscillates and might reach instability regions, which affect the performance, and so the process of sensors signal, PV, filtering is also called “smoothing”.

2.3.1.2. Small signal amplification

After filtering rough PV sensor signals, amplification usually occurs. This is because most sensor outputs described above are usually small in amplitude, for example in mV, even after converting rough variables into voltage or current signals. Amplifiers are implemented to excite this task. Electronic amplifiers are made of transistors, regularly CMOS or BJT. However, a special and useful kind of amplifiers, that applies feedback internally, in various applications such as analog electronics, signal processing, RF, and instrumentation is the Op Amps. Op Amps are ICs that are built of transistors, mentioned above, resistors, diodes, and capacitors [55]. As other electronic devices, Op Amps have characteristics and specifications that tend to be non-ideal in practical applications. However, the performance of an Op Amp is always referred to and compared with the ideal conditions. Thus, an ideal Op Amp should have [27], [55]:

- An infinite amplification gain
- Zero common mode gain
- Zero offset voltage
- Zero bias current
- An infinite input impedance
- Zero output impedance

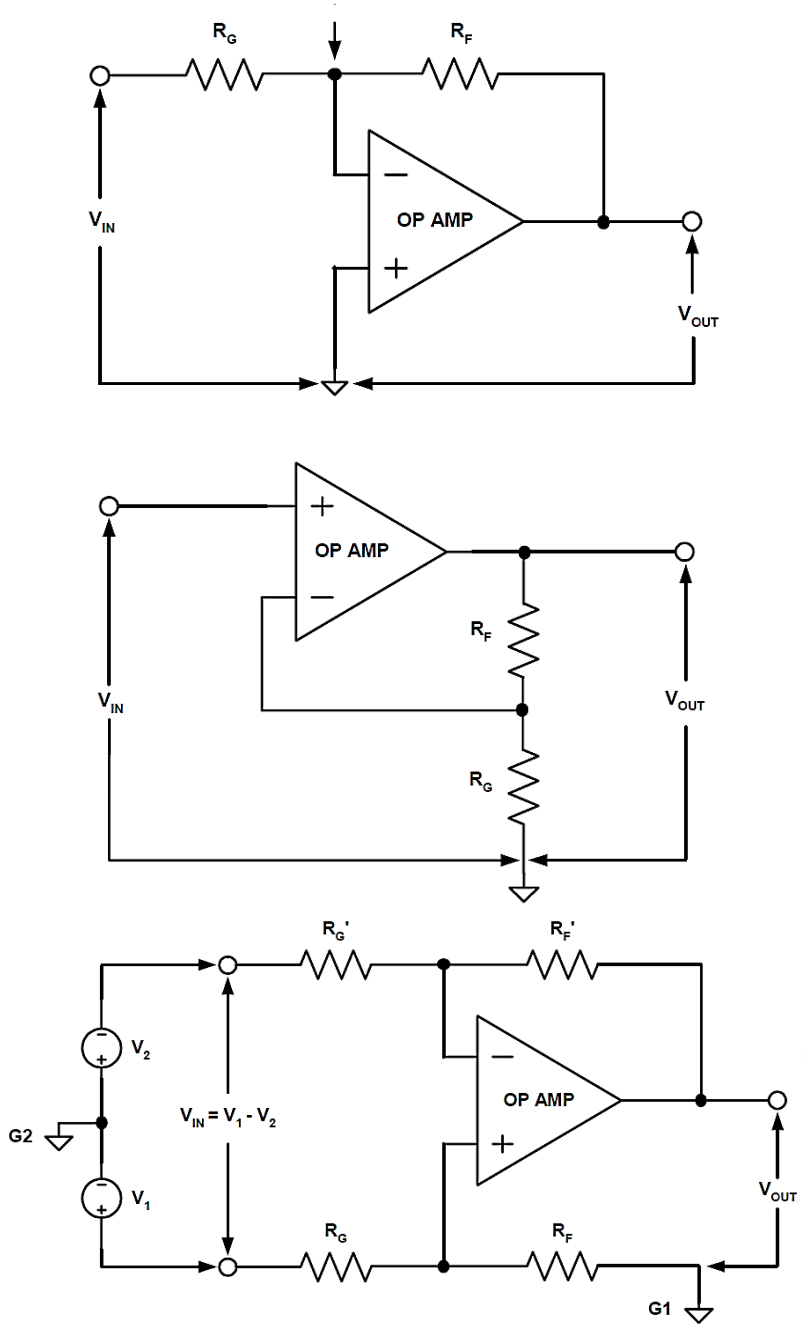


Fig 2.21: From top to bottom (a) an inverting, (b) non-inverting, and (c) DA Op Amp modes.

Op Amps are implemented as three main amplification modes: inverting, non inverting, and diffrential amplifiers [55]. The non-iverting amplifier is shown in Fig 2.21 (a). The magnifcation gain, *G*, in such a configuration is given by:

$$G = \frac{Vout}{Vin} = -\frac{Rf}{RG} \tag{46}$$

However, for the non-inverting Op Amp circuit in (b) the gain can be drawn as:

$$G = \frac{Vout}{Vin} = 1 + \frac{Rf}{RG} \tag{47}$$

Finally, for the differential mode appears in (c), the output voltage can be derived as:

$$V_{out} = \frac{Rf}{RG}(V2 - V1) \tag{48}$$

Therefore, the differential gain is

$$G = \frac{Rf}{RG} \tag{49}$$

2.3.1.3. Other forms of signal conditioning

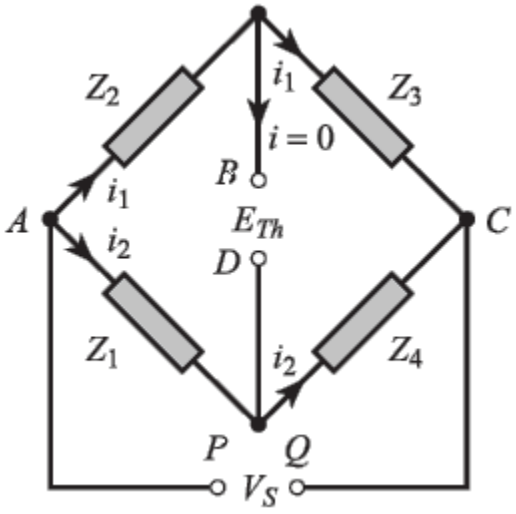


Fig 2.22: The Wheatstone detection bridge has several applications in signal conditioning

Fig 2.22 above shows a typical bridge detection circuit which has a countless number of applications in analog signal conditioning instrumentation such as converting signals of resistive, capacitive as well as inductive sensors into voltage outputs. Similarly, Fig 2.23 below illustrates the additional important Op Amp implementations used in analog signal conditioning and control systems.

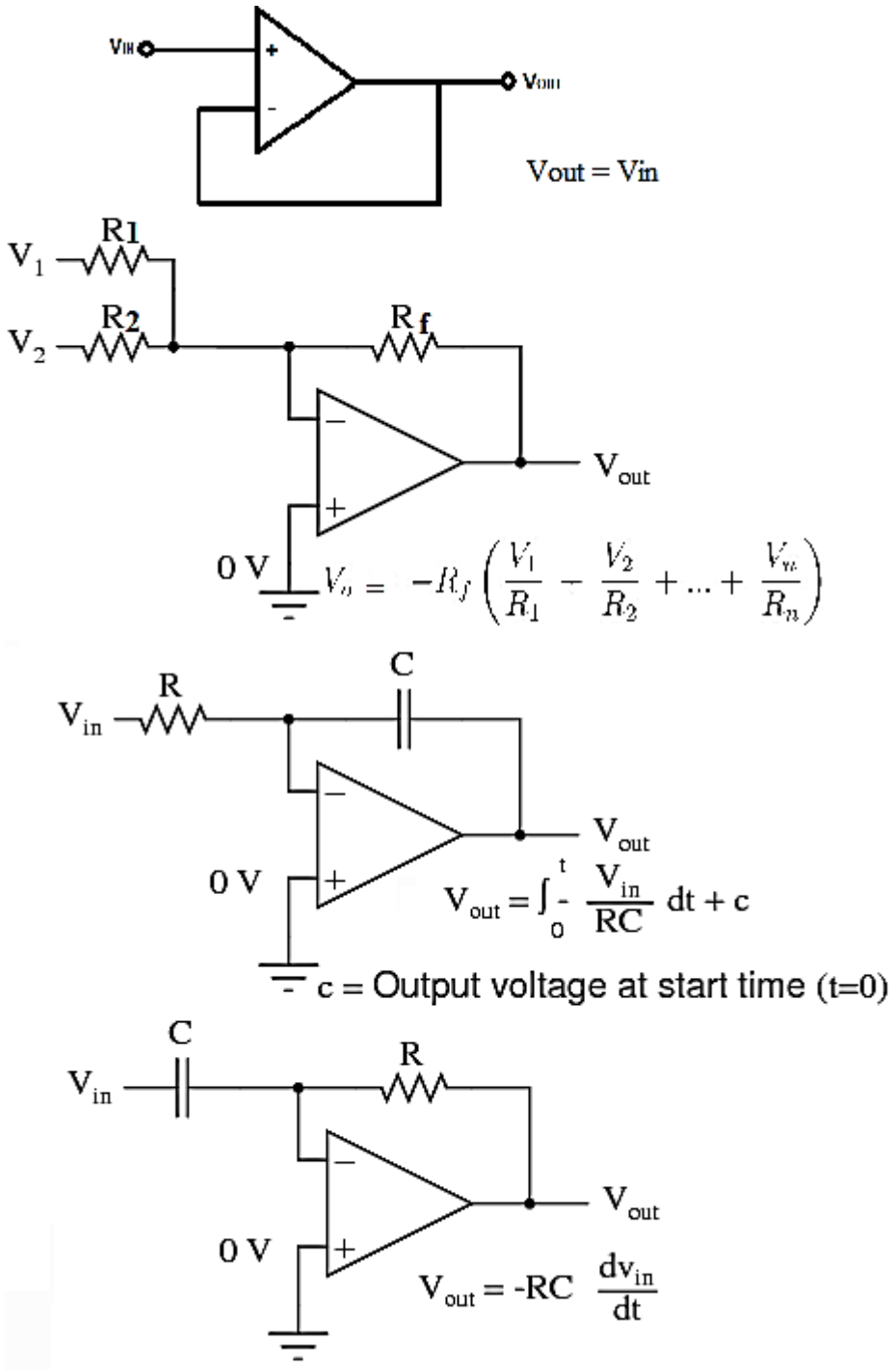


Fig 2.23: Four common important Op Amp realizations in analog control and instrumentation applications, from top to bottom, (a) a high input impedance buffer, (b) a summing amplifier, (c) an integrator, and (d) differentiator circuits.

Other signal conditioning examples would be Wheatstone bridges, bias change, linearization techniques, impedance matching, phase matching, voltage to current convertors, current to voltage convertors, AC carrier systems, oscillators and resonators, and many others [27], [55]. Except for a few such as the Wheatstone bridge, which implements resistors or impedances, and probably impedance matching, which uses passive circuits, most other signal conditioning techniques use Op Amps as primary elements in their implementation. Not only does Op Amp appear essential in most analog signal conditioning, but also in other electronic circuitry such as continuous H/W controllers. Some additional common Op Amp implemented modes are voltage followers, integrators, differentiators, and summing amplifiers (Fig 2.23).

2.3.2. Signal processing and DAQ

The signal processing stage begins with converting analog domain signals into digital, ADC. More often, this might happen at two different stages, depending on the control and communication technology implemented. For example, most measuring field instruments have their own processing and display units, so they include ADCs. If the field communication protocol is the conventional analog one, i.e., 4-20mA, then to feedback the transmitters signal to the DCS requires ADC too. In some exceptional cases, however, as the fieldbus, analog data are converted at once at the transmitter level; then digital signals are propagated as introduced before. Nonetheless, signal processing is essential in today's digital control and instrumentation applications. Whether it's the field instruments, control units, HMIs and SCADA, or even higher level application servers, most digital systems that involve CPUs or other forms of digital computing architecture should include three to four main phases: ADC, processing, DAC, besides the display unit. However, in practical industrial control systems more universal terms are used instead of the ones above; they are AIs cards or modules, controllers or CPUs, AOs, and HMI.

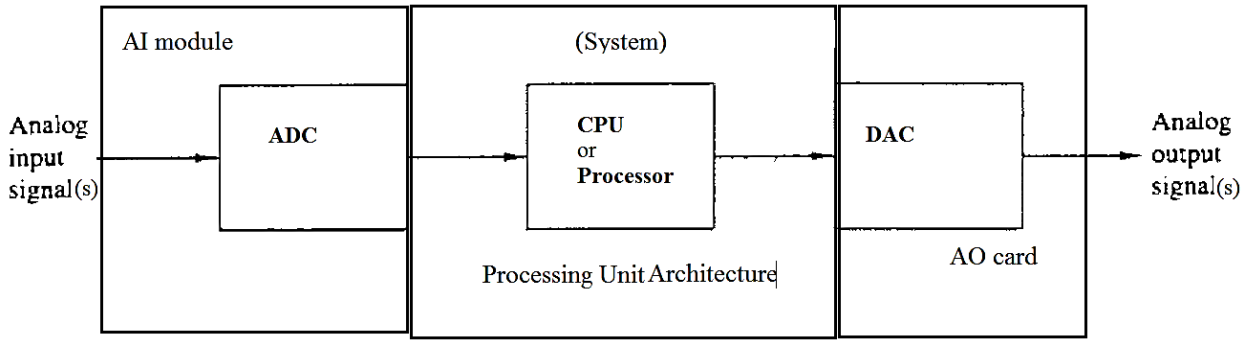


Fig 2.24: Three main Signal Processing stages.

Fig 2.24 above shows the main three stages of signal processing and DAQ in addition to displaying HMI. Industrial AI cards do not deal with one analog channel only. Instead they apply multiplexing to acquire multiple field transmitters' signals. The number of pins, acquired as AI channels transmitted from signal conditioning, varies from four up to 32, or maybe higher sometimes, according to the type of card and usage. Fig 2.25 shows a typical AI module that consists of *n* channel pins, an overvoltage protection circuit, multiplexer, buffer (for impedance and loading matters), and finally isolated ADC block. It is known that the ADC is a process of three to four stages: S/H, quantization, then finally coding, in which digital zeros and ones are interpreted from the quantized signals [56] (Fig 2.26). Note that serial to parallel conversion and vice versa should be present in such H/W in order to distinguish multiple sensors.

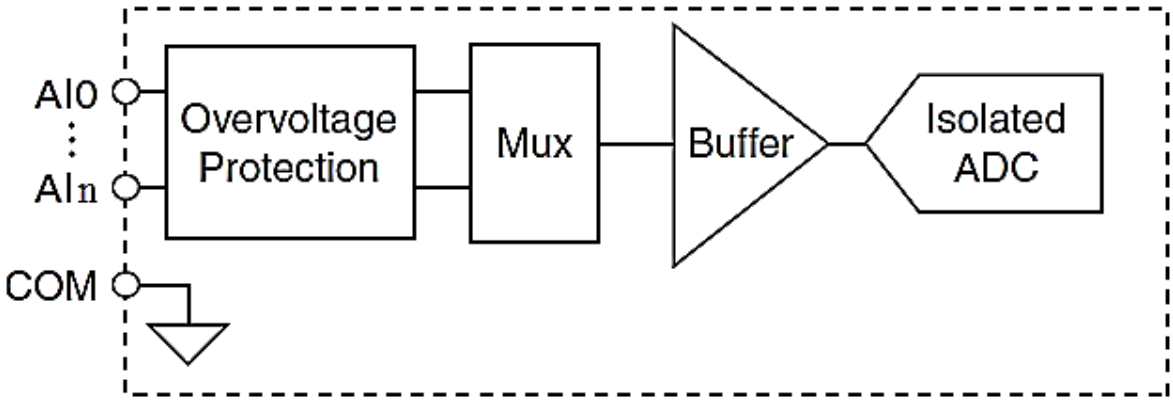


Fig 2.25: A BD of a typical AI card with *n* = 4, 8, 16, or 32 channel pins.

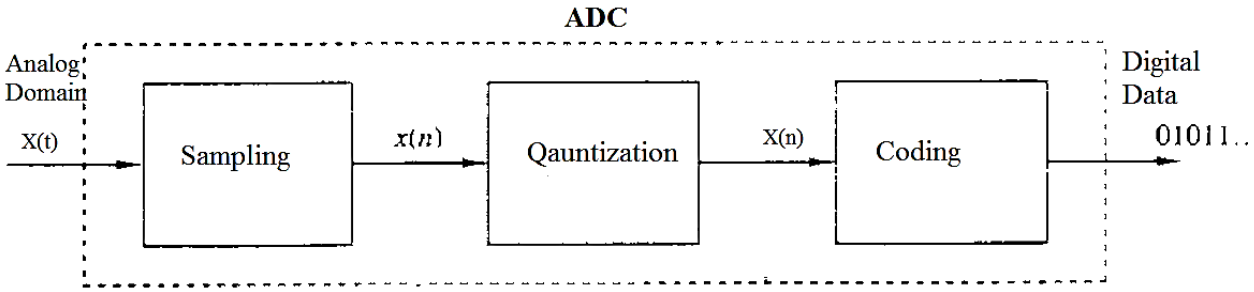


Fig 2.26: ADC is a process of S/H, quantization, and then binary coding.

The performance of a given AI can be characterized by the speed of conversion and the resolution. Both are combinations of several factors: sampling rate, ADC clock rate, and others [55], [56]. Most industrial AI modules have fixed sampling times per channel, hence conversion durations. As a system engineer or a designer, it remains critical to select the proper AI card that suits a specific application. More specifically taking the topic of bandwidth into consideration that the sampling rate condition should at least satisfy the Nyquist limit [56]:

$$fs \geq 2B \tag{50}$$

Both terms in Hz, f_s is the sampling frequency of an AI and B is the acquired signal bandwidth. Most AI cards are manufactured to provide adequate sampling rates to accept most process signals as they appear naturally slow, having low bandwidth, such as temperature, pressure, level, etc. However, there exist other types of signals that may involve higher bandwidths, i.e. optical position, flow, and encoders. Also, sometimes special types of AIs should be selected for very fast types of signals as sound acquisition and vibration.

$\Delta T = 1/f_s$ stands for the sampling period which is used in expressing discrete time systems; however, conversion time emerges as another term when considering I/Os propagation delay. Also, as seen, an AI is more than just ADC. The conversion time is an additional delay to sampling period. Hence, the conversion time is the period that is required to accomplish a single word conversion of M bits [55]. This is not including the set TDMA acquisition time either.

For example, an AI that realizes the SAR method [55], [56], and acquires more than one channel simultaneously has a propagation delay equals to sampling period ΔT plus conversion time (that varies according to the number of channel used simultaneously) in addition to TDMA time.

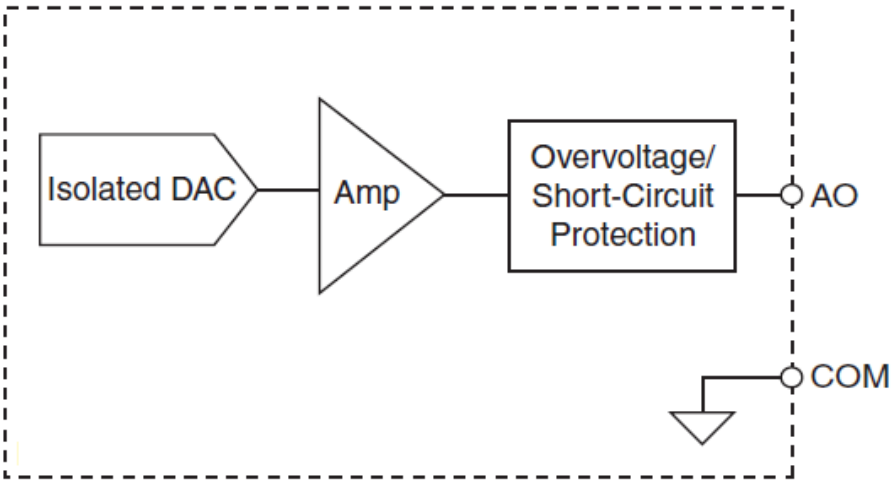
Another factor to consider in AIs is the resolution. The voltage resolution of an ADC is a percentage that equals to its overall measurement range (span) over the number of discrete values:

$$\%Resolution = \frac{Span}{2^M} \times 100\% \tag{51}$$

Where 2^M is the number of voltage intervals or step size. The resolution infers the magnitude of the quantization error. To represent or code the quantized signal into digital data, there exist several standards, [27], [56] as the radix of 10, binary numbering system, 1's and 2's complements, sign magnitude, BCD, octal coding, hexadecimal, floating point, fixed point representations, etc. The ADC circuitry could be implemented using several possible ways [27], [55] such as successive approximation, parallel (flash) ADC, ramp and compare, and the dual-slop methods. The overall digital binary output of *n* digits of the ADC should approximate to:

$$b_1 2^{-1} + b_2 2^{-2} + \cdots + b_n 2^{-n} \leq \frac{V_{in}}{V_{REF}} \tag{52}$$

Where V_{REF}, is a reference voltage and b_1, b_2, …b_n are the binary digits.



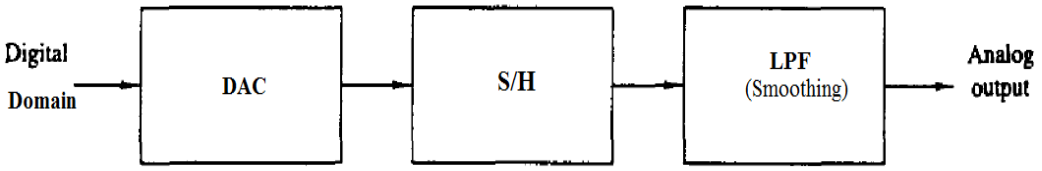


Fig 2.27: (a) A typical AO module's BD and (b) DAC sub-blocks

Similarly, as appear in Fig 2.27 (a)/(b), AO modules contain the DACs, gain amplifiers (for different range conversions and scaling), and other circuitry as protection. The DAC process consists of blocks too. First, the DAC block in which *n* bits parallel digital values are summed up by means of a summing Op Amp introduced above or the use of an alternative, ladder networks [27]. As a result, the output of this summing amplifier takes an analog form. Then, this analog output is passed to a S/H block to avoid overlapping consecutive digital words' conversion. The rectangular sharp edges of the output of S/H is after that smoothed by a LPF [56], [27]. The operation of the DAC can be mathematically considered as the opposite of the ADC's such that:

$$V_{out} = V_{REF}[b_1 2^{-1} + b_2 2^{-2} + \cdots + b_n 2^{-n}] \tag{53}$$

Besides I/Os, all digital processing architectures should include a clock signal source in addition to memories such as RAM, ROM for CPU based systems. The clock rate is usually determined by the H/W platform or the processor in which a particular AI acquires signals to. Some processors are considered fast because they have clock rates in GHz, others are intermediate, in MHz, and many have less clocking capabilities. However, the performance of a given computing H/W cannot be determined by the clock frequency only. Rather other factors such as the organization of the architecture, its realization, H/W capacity, computing precession system, i.e. integer, fixed point, floating point, etc., and others such as internal communications, all play major roles in assessing processing units' performance. If considering the computational time as the only measure of a processing unit's or an overall digital system performance, then other terms defined in [57]:

- Overall execution (elapsed) time: also known as response time, it is the total time delay a given H/W architecture spends to perform a whole task including I/O propagation, CPU and memories access, internal communications, RTOS management, and others. This measure matters significantly in assessing RT digital control loops performance.
- CPU time delay, which is the period taken for the processor to execute a task. This is including system time (RTOS, memory accesses, CPU execution, etc.) but excluding I/Os and other delays. If ignoring system delays, then the CPU execution time, which is the time it takes to execute a program, hence one soft control loop algorithm, is given by [57]:

$$\textit{Execution Time (seconds)} = \textit{clock cycles per program/clock rate (cycles/seconds)} \quad (54)$$

- System throughput: which is the measure of how many tasks a particular processing unit can perform in parallel (all at once at a given time) [57].

Most CPU based architecture include Address bus, Data bus, and Control bus; thus, there exist internal serial and parallel signaling and communications between H/W blocks [27]. This implies digital communications measures as the baud (bit) rate, R in bits/s, where:

$$R = nf_S \quad (55)$$

Recall, n is the number of bits and fs is the sampling rate. Therefore, for multiplexed signal of m channels,

$$R = mnf_S \quad (56)$$

Any signal processing and DAQ phase of a control or a measurement system should include HMI for user interface in which two main tasks are performed:

- Programming, configuration, and operation: which differs according to the tool used, situation, and stage. Conventional microprocessors programing use known machine, assembly, and low or high level languages where signal processing calculations, and other

tasks such as data presentation and operation, are programmed and downloaded to CPUs. However, more sophisticated and higher level S/W tools have taken the lead in industrial automation environments in which windowed applications suitable for computer monitors appear; for example, workstations in DCS CCR SCADA present underlying programs into FB and graphical interface. To emulate industrial environments, this work employs the NI's VI S/W programming tool program H/W processing platforms.

- Display: as introduced earlier, analog systems used to have pneumatic indicators whereas in the modern industrial automation architecture there exist two places of digitally displaying measured and controlled RT data; first, field instruments' PV local LCDs, and second, the supervisory level graphical display located at the CCR. However, there exist other purposes of data presentation other than indication or monitoring; they are recorders or historians where historical records of measured data are printed or logged in S/W servers [27]. The implementation of this research uses PC screens and VI's front panel of NI LabVIEW S/W tool as a mean of a graphical process interface and display. Fig 2.28 below shows an example of a signal processing and DAQ.

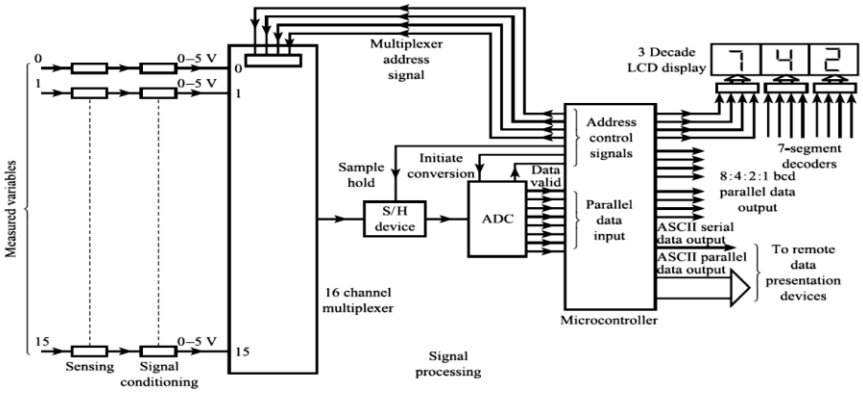


Fig 2.28: An example of signal processing and DAQ (taken from [27])

Other modes of signal processing and DAQ include DI, DO, frequency based convertors, communication interface cards such serial, fieldbus, and Ethernet, and others [27], [55].

2.4. Control Units and Systems

As the introduction gives brief background description about control instrumentation evolution ages and different technologies adopted since the first application of automation in industrial plants, this section classifies possible realization of control units and systems in particular, some of which are especially developed for industrial automation purposes and applications. A very broad classification divides control units and systems in general into two main categories: analog, and digital controllers and systems.

2.4.1. Analog controllers and systems

Analog controllers that were used before the emergence of the digital period have been briefly introduced earlier. Despite all the digital and communications advances, analog controllers are still found to be produced and rarely used in few industrial plants currently. Analog controllers can be realized as “single loop controllers.” Since they provide very high dynamic response performance, some process manufacturing end-users prefer to have some of the SLCs installed to control critical loops requiring the highest response possible to be maintained.

Courtesy: www.yokogawa.com

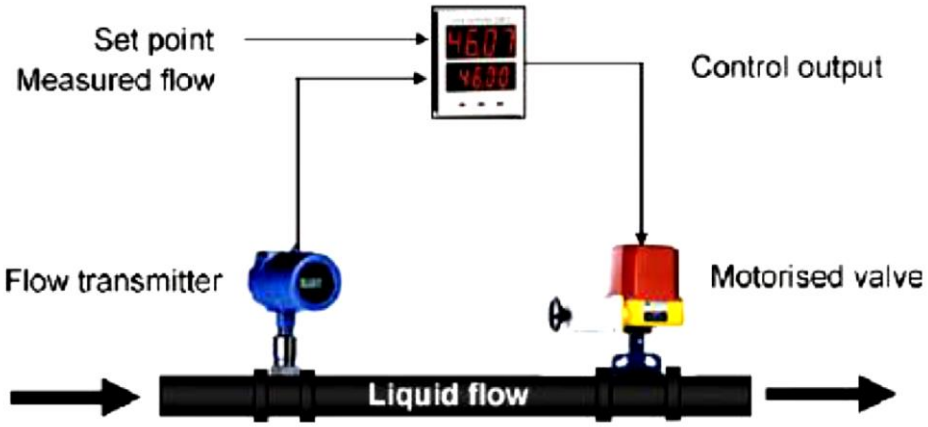


Fig 2.29: (a) A front look of a typical SLC and (b) single industrial analog control loop (taken from [23])

Fig 2.29 (a) above shows a SLC of one of the leaders in instrumentation and control system product vendors, Yokogawa, and how the loop is simply wired in (b). Though, the figure shows that the display and the local HMI are in a digital environment (LCD and buttons); this is for configuration as well as operation, i.e. SP and tuning. This feature does not deny the fact that the control is entirely implemented as an analog electronic circuitry [23] using RC, RCL, or Op Amp circuits introduced above. For example, a PID control algorithm remains standard in industry. It is a summation of a proportional amplification of an error signal, an integrator, and a differentiator where gains (variable resistances) are tunable as in Fig 2.31.

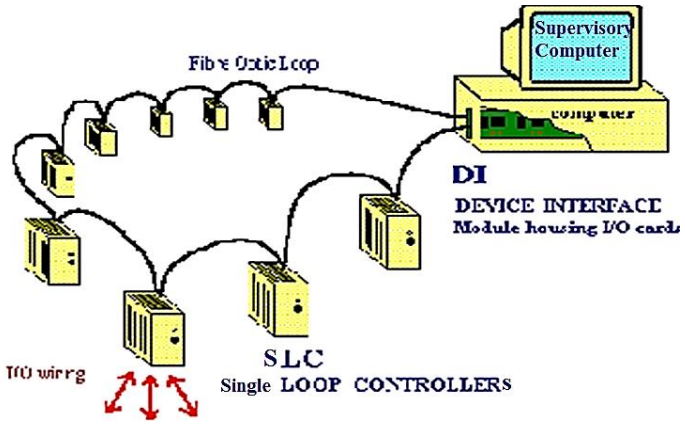


Fig 2.30: Controlling multi-loops plants with SLCs

In order to answer the question how these kinds of controllers were used to control large-scale industrial plants back in the analog age, the answer derives as it is hypothetically possible; however, the larger the plant, the higher the number of I/Os, the bigger the amount and distances of wiring required, the less the flexibility, the higher the cost, therefore the lower the feasibility. Fig 2.30 displays a possible way of controlling a process plant using SLCs.

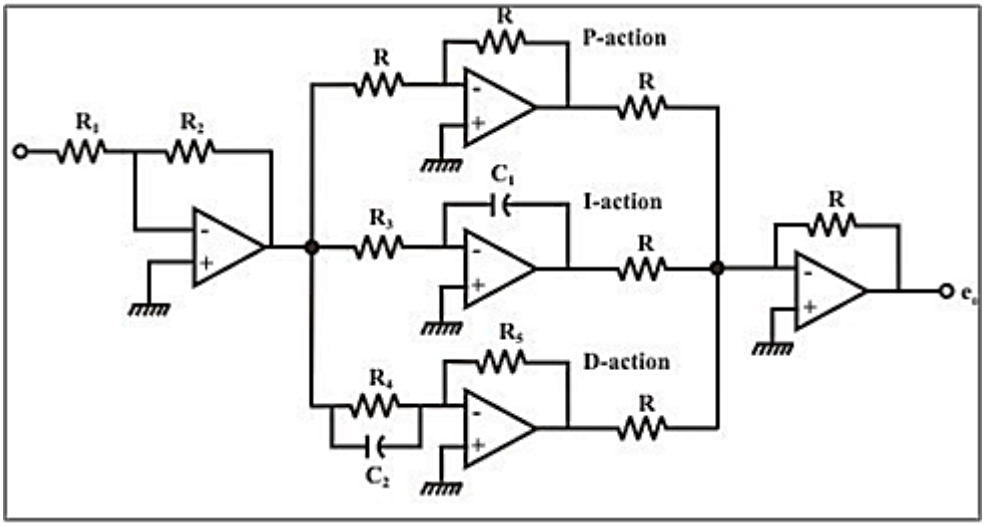


Fig 2.31: An analog PID controller realizing Op Amps (taken from [55])

2.4.2. Basic and Industrial Digital Systems and Controllers

Digital control could be implemented using several approaches and technologies. One could divide those ways into three main classes: fully H/W digital circuits, processor based programmable systems, and PLDs. However, since this research focuses on industrial applications the above mentioned three main classes, a more detailed classification would emerge in which also some of the terms used for such types of systems in industry are defined. They are ASICs, PCs, MCUs and MPUs, DSPs, FPGAs, PLCs, PAC, RT controllers, DCS, SIS or ESDs, RTUs, and lastly SCADA systems.

2.4.2.1. Digital ASIC

Digital circuits could be implemented as H/W gates only using CMOS technology for example, and so control units. This kind of circuits provides very high performance being H/W in nature, and they can be found in a countless number of applications. However, digital ASIC ICs are not programmable or reprogrammable, which does not allow the minimum flexibility required in industrial automation environment governed by regular changes and reconfiguration in the processes or throughout systems engineering project execution phases [58], [59]. Therefore, ASICs appear to not be suitable to implement distributed control; they can be placed at other locations though, where no need for programing and interface emerge such as in some parts of the field instruments electronics and discrete devices drivers. Most of the SLCs described above are now ASICs where they provide very accurate digital control results instead of speed [23].

2.4.2.2. PC (desktop)

Computer controlled systems are initially imagined as desktop based where the PC's CPU performs control functions directly. This is implemented by interfacing RT fields' I/Os by an external H/W tool, DAQ. There are types of DAQ interface boards; some are commercial and educational while many others fit industrial applications where chasses platform blog slots of different I/O modules as shown in Fig 2.32. The scope of those H/W interface is DAQ whereas the core duty of the PC is to perform the actual control and HMI [23]. This type of digital computer controlled system is also known as supervisory control where there appear no embedded or independent H/W control units of the supervisory (host) one. In other words, both control units and the supervisory HMI functionalities are integrated and not separate. Therefore, if the computer goes down, for some or any possible reason such as power or communication, the control in the field will be lost, consequently ultimately the plant or the process will trip or shut-down. This scenario is not preferable as per industrial standards and specifications.

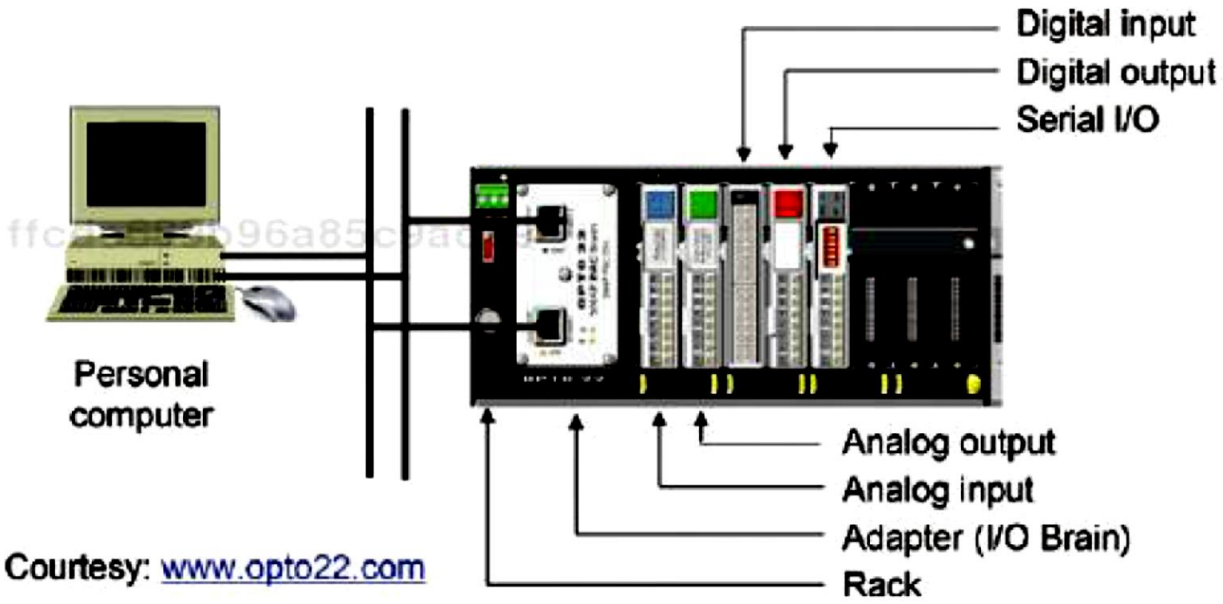


Fig 2.32: PC (desktop) based controlled system (taken from [23])

2.4.2.3. Embedded processors (MCU and MPU)

The desktop control unit introduced above is processor based, however, as concluded, not embedded. Embedded processor systems are known as MCUs and MPUs. Both are processor based, contain CPU, and are similar in the overall functionality and purpose, yet the two terms involve distinguished variances. Their descriptions are:

- MPU is a more general expression that includes both and can be used refer to other applications as computer systems and DSP. Microprocessors were first developed in 1971 as a substitute for early centralized digital computers mentioned in the introduction, which were called CPUs. Hence, a MPU is a complete embedded architecture set that includes CPUs that are entirely combined in the same IC ship [60], [61]. Therefore, the architecture of MPUs should include [60]:
    - A CPU: which is considered as the “brain” of the system that executes programs in a sequential logic manner, one after another, by reading first from memory

locations on the board, performing the calculations and required functionalities secondly, then passes, exchanges, and writes data back to peripherals such as I/Os, RAMs, and ROMs.

- A memory unit: which saves data and instructions and exchanges them when needed.
- A clock circuit: that generates the digital square wave frequency cycles. Any dynamic logic circuit requires a clock counting signal to operate besides power supply. The topic of the relationship between a digital architecture clock rate and performance has been discussed. Also, the clock cycle of given architecture plays a major role in power consumption.
- Peripherals: that are mainly I/Os, timers, UARTs, ADC/DAC and other communication interface modules. Peripherals provide the interface with external domains of the MPU as the RT analog world or other digital blocks.
- The bus systems: which are transmission medium of all the internal as well as external communications from and to the MPU. Three types of bus used in such architectures have been discussed before.

- MCU: MCUs are MPUs designated specifically for embedded systems applications. Thus, an MCU appears smaller in scale and is limited by the application, performance and the number of I/Os it can accept. However, a more technical difference that distinguishes MCUs from MPUs in general is the architecture organization itself. MCUs contain the CPU and other blocks defined above all integrated into a silicon chip while MPUs do not as they involve buses to connect to other architecture components. Fig 2.33 below demonstrates the key architectural difference between (a) MPUs and (b) MCU.

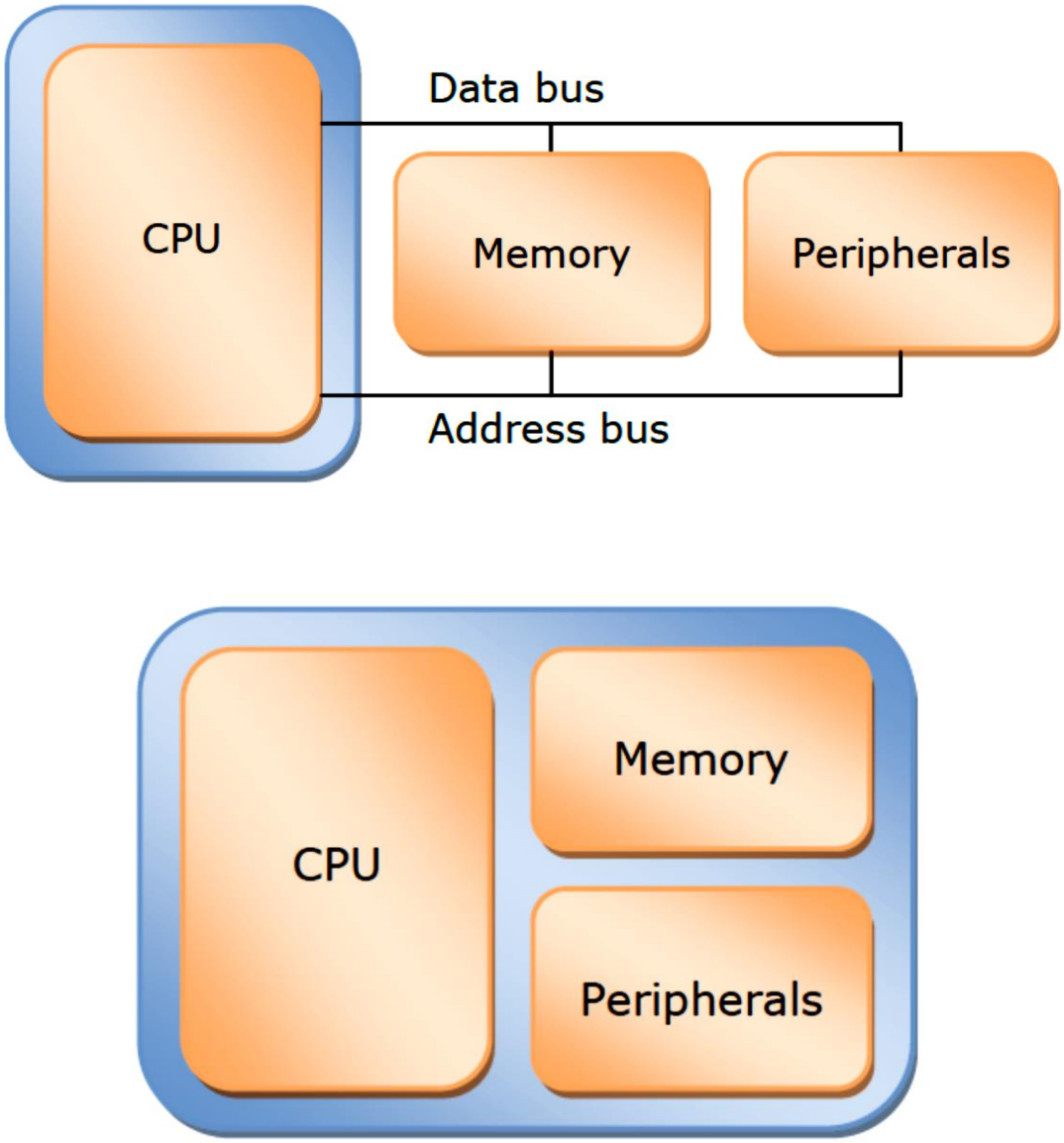


Fig 2.33: (a) MPU (top) and (b) MCU (bottom) (taken from [61])

With regards to industrial automation applications, MCUs cannot be realized as control units as they fit small prototype tenders such as educational laboratory, toys, or other machines or devices. MCUs might suit processing units, other than controllers in the DCS architecture, such as sensor nodes. On the other hand, MPUs are found to be more adopted in industrial automation. They compose a part of larger control units' architecture as industrial controllers. Since MPUs provide more area and performance, their cost is logically higher than MCUs [60].

2.4.2.4. DSPs

DSPs are processor based systems (MPU) but specified for DSP applications. Thus, DSPs' architectures are designed and manufactured to have very high computing capabilities. DSPs handle fast computation, so they fit several numbers of applications that involve extended signal processing and analysis as telecommunication systems electronics, medical and image processing, audio and video processing applications, FFT analyzers, computer boards, printers, copy machines, etc. [60]. DSPs can be also realized as control units when implementing control loops other than industrial automation environment such as air-born flights and navigation controls [60], antenna tracking, DC-DC convertors, and power electronics applications. DSPs, programing and implementation is classified into two prime realizations of algorithms: FIR and IIR filters. Fig 2.34 below indicates an illustration of an FIR filter realization of a DSP algorithm which in turn, as per the DSP ship used, may follow a fixed point or a floating point system. The architecture of DSPs is derived from general purpose MPUs as the first are considered to be special case of the latter [62].

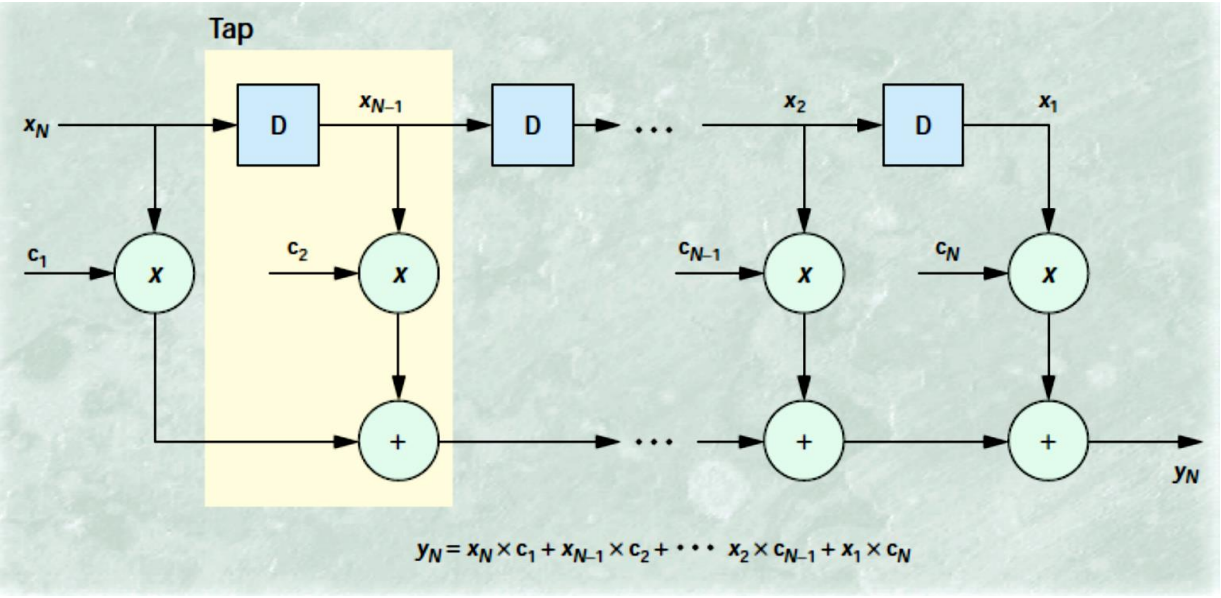


Fig 2.34: DSP algorithms is ultimately implemented as FIR or IRR filters (taken from [62])

In industrial automation, DSPs are getting more accepted in applications that require fast computations, especially in discreet manufacturing and machine controlling plants such as MCPs, industrial networking, cameras and machine vision, safety, etc. Though, the process control industry's controllers are known to have general purpose microprocessors due to the nature of the process, flexibility, reliability, and determinism required in such environments. Some MPU manufacturers have begun moving towards hybrid architectures where DSPs extend microprocessors [62]. The reference [62] concludes several differences and tradeoffs between general purpose MPUs and DSPs regarding performance measures, applications, power consumption, and cost. A few remarks could be noted here:

- In general DSPs may offer higher computational speed performance due to their sophisticated architecture and the programming approach while general purpose MPUs' tend to be more standard, easier, and more flexible for users and implementation.
- Most DSPs operate on fixed point numbering precision while MPUs may have more accuracy since they realize the floating point system.
- Since DSPs work on fixed point, they appear less expensive and faster than MPUs in general. However, this statement depends on other factors, e.g. the clock rate and others.
- Originally, both architectures were considered to be sequential, that their CPUs execute logic serially. Some advanced MCUs, however, could perform up to four instructions per cycle. Also newly modified DSPs can now divide instructions of 256 bits into eight 23 bit segments, which adds more parallelism.
- MPUs require high level programing language to be programmed; nevertheless, optimized performance DSPs are usually programmed in lower environments to control internal instruction sets of the CPU.

2.4.2.5. FPGA

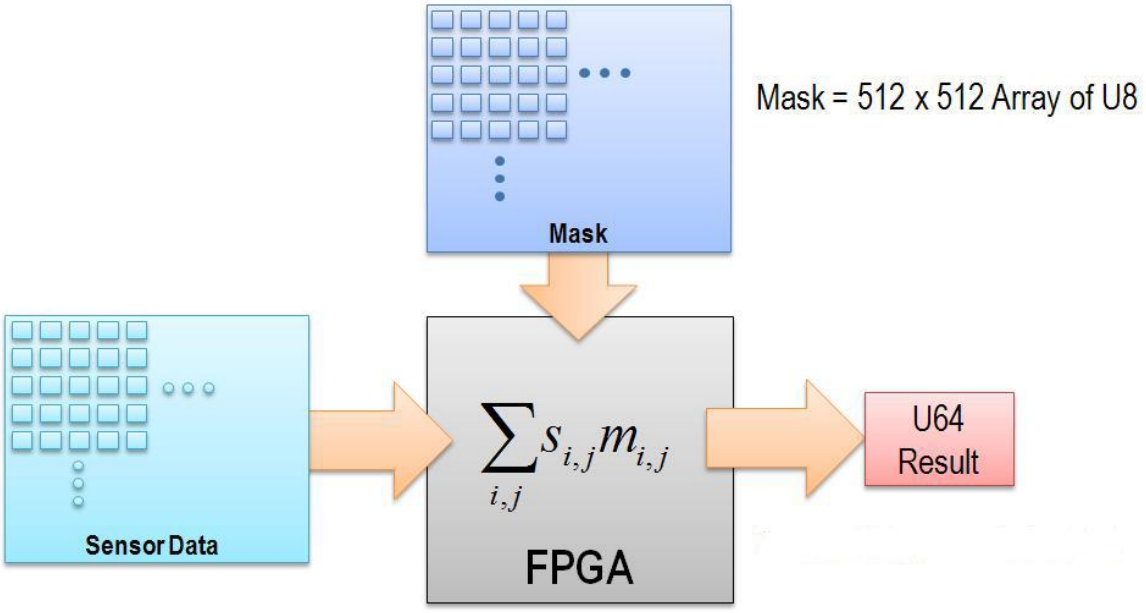


Fig 2.35: FPGA logic gate mapping realization as SOP digital functions (taken from [63])

FPGA is a new developing technology that has emerged in the world of computing architecture since 1985 by Xilinx. FPGAs are part of a bigger family, PLDs. They are programmable silicon chips that were developed to reach middle solutions between ASICs and processor based systems. More specifically, FPGAs are programmable digital H/W units [58]. They are similar to PLAs [64] where digital logic is implemented on the gate level, but with an advantage that FPGAs and other PLDs are reprogrammable unalike ASICs [58]. FPGA implementation relies on the basic concept that any digital logic could be ultimately realized as either SOP or POS [65]. Fig 2.35 illustrates the idea that FPGAs mask I/Os with logic gates by means of interconnects as SOPs functions. This way, any realization of DSP of digital control algorithm and logics would be possible as long as the maximum (finite) number of interconnects, gates, and memory elements are not exceeded. Low level logic blocks in FPGAs consist of flip-flops (basic for sequential and dynamic functions) and lookup tables (to map predefined functions and truth tables [65]) [58].

Regarding control unit implementation, FPGA is an interesting technology to study its effect on loop performance. FPGAs have been studied for this purpose in much of the literature and control applications as in communication electronic controls, and power electronics, and other non-industrial process control environments. FPGA may provide more parallelism, reliability, and maintainability as computing H/W itself, which may enhance control dynamic response [58]; however, to be comprehensively accepted and adopted in industry as control units in certain automation stages requires satisfying several other factor stated in the introduction earlier other than processing advantages. FPGAs are used to be programmed using low level machine languages as VHDL [58]; nonetheless, developers have started developing higher level tools to interface to the FPGA world, for example NI's cRIO H/W platform through LabVIEW VI's graphical interface. This report tries to implement an FPGA control design using this approach in the case study and compares the technology and results based on loops' response and performance with conventional industrially adopted advanced processor mode systems. More detailed discussion about the potential of FPGA in control systems applications will follow later in multiple occasions.

2.4.2.6. PLCs

PLC's evolution and significance in the history of industrial automation are introduced in Chapter 1. The term PLC is very common in process control industry as well as other engineering mediums. Yet, the question that arises when discussing this topic along with different industrial control processor based units and system, is what makes it a PLC? To answer this question from a technical point of view necessitates discerning between different architectures designated for industrial control purposes. PLCs are processor (CPU) based architecture that are customized to perform rapid control logic execution [61].

PLCs differ from other microprocessor based systems in two main internal operational and architectural organization features in order to minimize control loops execution time as much as possible. First, PLCs execute the efficient input scans and round-robin control [61] to the programing functions. Secondly, as appears in Fig 2.36 below, PLCs' internal H/W architecture allow faster I/O execution by integration and attaching them directly to the processing unit. Similar to any processor system, PLCs employ small RTOS on the top of the processing unit to manage the whole process and ensure reliability of data [61].

Fig 2.36: A simplified BD of PLCs architectural I/O advantage (taken from [61])

In addition to the fast execution feature that let them be specialized in large areas of industrial automation world such as discrete manufacturing, sequential, machine, and motion control and SIS/, PLCs have rigid frames and structures that suit harsh plants' process environment. Several other industrial system standards, some of which are discussed later, are considered to be PLC based, so common deductions could be drawn here. PLCs were originally developed to replace relay ON/OFF panels as introduced in Chapter 1. Since that time, PLC based systems have become more sophisticated and advanced so that they could cover networking, industrial communications, FBDs programming, HMI interface, etc. Hence, process engineers and end-users have recognized the best applications of PLCs compared to other systems, and so different process natures are assigned to distributed (hybrid) types of integrated systems, i.e. DCS as master major PCS and subsystems PLCs for machine packages, ESD, FGS, and some utility motors and pumps control. Fig 2.37 shows a famous general purpose PLC brand, Siemens S7.

Fig 2.37: Siemens S7 general purpose PLC (taken from [23])

2.4.2.7. PAC

As mentioned on several occasions earlier, the PLC functionalities have evolved rapidly so that they can be compared with sophisticated DCS features. Therefore, advanced PLCs could perform not only discrete manufacturing, ON/OFF, and timing motion functions rapidly and efficiently with expanded modules, I/Os, HMI and more user, fieldbus, network, and other vendor systems friendly, but also regulatory process control. Thus, PAC are PLC based systems that possess those features. Since the term PLC has been given to the conventional general purpose product that uses ladder logic for programming, similar to the one that appears in Fig 2.37 above, PAC has the same architecture but is customized for process automation purposes. Thus, they offer all the advantages mentioned above in addition to the environment of their programming and configuration that adds FBD and other standard regulatory blocks such as the PID. Fig 2.38 shows a typical PAC system that supports Ethernet LAN and other types of fieldbus protocols [23].

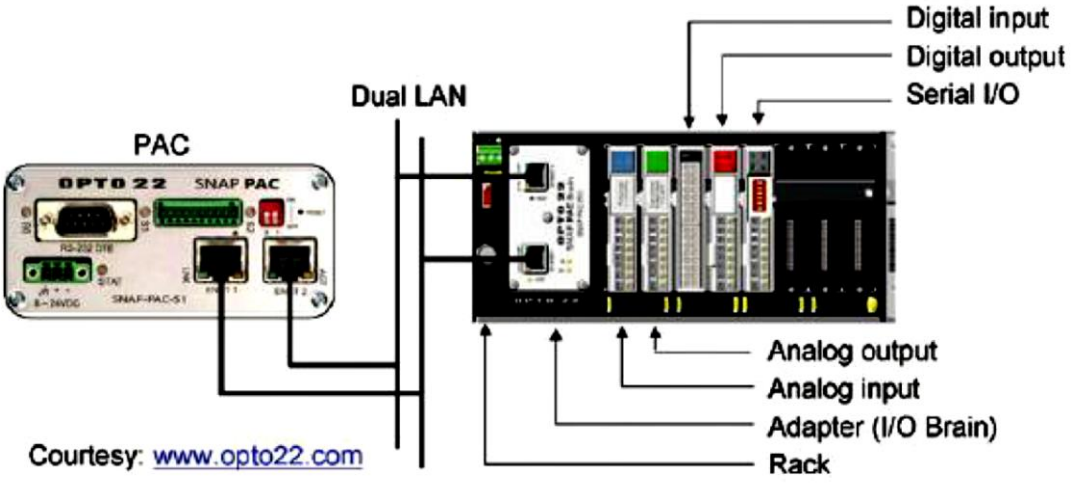


Fig 2.38: PAC is a special PLC that supports networking, fieldbus, and some DCS abilities (taken from [23])

2.4.2.8. RT-controllers

This paper labels regular industrial controllers as RT controllers. They compose basic units in the industrial automation architecture described already. Thus, RT controllers are MPU based systems that satisfy process control industry standards called earlier as reliability, configurability, flexibility, redundancy, scalability, embeddedness, and protocols compliance. Moreover, RT controllers' functionalities are considered to be general purpose with an emphasis on process control objectives. In other words, unlike PLCs or PACs, RT controllers provide relatively slower response and execution time but more accurate and smoother control results which fit general continuous processes, i.e. temperature, pressure, level, flow, force, etc. However, RT controllers support interrupts and interlock functionalities also, which implies that they can handle discrete devices and sequential control processes too. Hence, one crucial difference distinguishing PLC/PAC from RT controllers is that the latter is time based whereas the first is event triggered based. This fact led experts to select the best practices associated with the proper system [23]. The term RT has derived from RTOS, which runs on those systems on scan time [98].

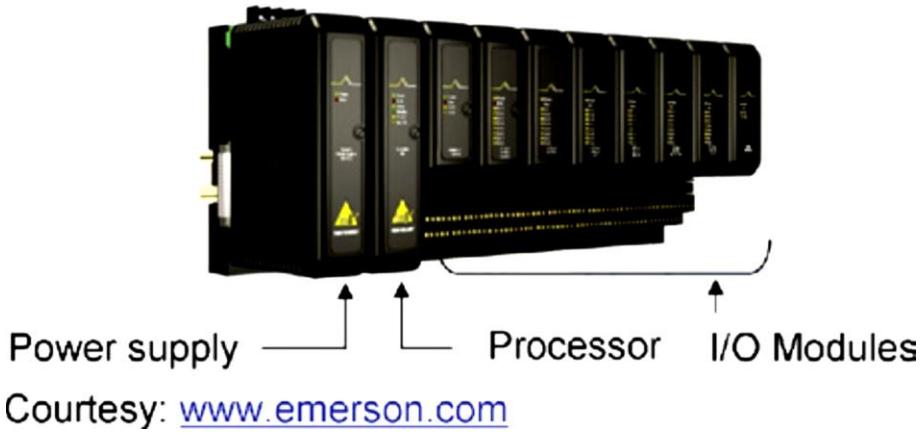


Fig 2.39: Emerson's DV RT controller forms a single DCS node

Fig 2.39 illustrates a typical industrial RT controller composing a DCS node, more specifically Emerson Process Management's DV system. What enables RT controller basic units of the PCS/DCS automation architecture beside functionalities that adhere to industrial specifications is their extended H/W environment and distributed I/O capacity. It is not only the controller rack or chasses and its I/O modules that compose the unit. It is a whole cabinet what makes a field control station that consists of the controller rack in addition to their extended multi I/O chasses. Thus, one RT controllers feature extracted here is their bulky memory and I/O capacity that could handle large scale industrial processes. For example, one controller unit can accept *n* number of extended racks while theoretically being capable of acquiring *m* amount of I/O cards and pins [23]. However, in practice, to avoid CPU loading problems which may affect its performance, reliability, and determinism, usual industrial standards limit the maximum I/O assignment for each controller to 50%. But, how each cabinet is configured and the I/O racks and modules are organized depends on the engineer and the design specifications of each project itself. A single controller carrier is a PCB that the MPU plus memories (where LEDs indicate diagnostics, power ON/OFF, communications, etc.) PSU, and included I/O modules are plugged onto. A single extension rack in turn contains, besides the I/O cards, a bus interface and a repeater module [23].

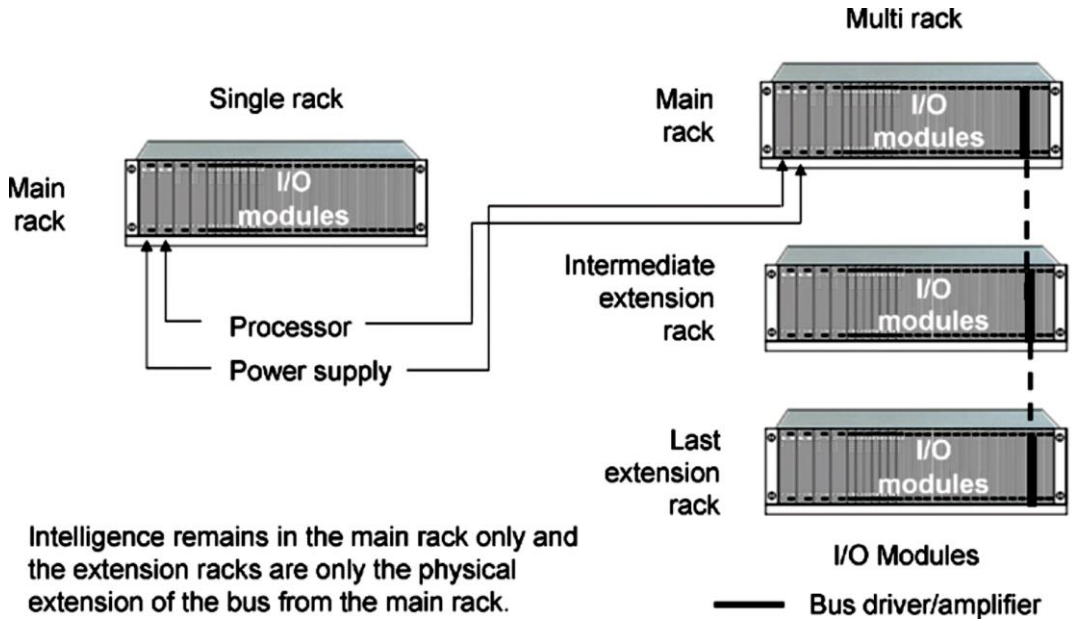


Fig 2.40: Industrial RT controllers I/O expansion racks form control stations (cabinets) (taken from [23])

Fig 2.40 shows a simple structure of a controller cabinet comprising of the processor rack that contain the CPU, PSU, I/O modules, the extension chasses including bus interface and amplifiers cards and communications busses connect them all together. Thus, there exist two types of busses for this purpose of internal control cabinet communications: serial interface and CAT [23]. Note, all follow plug and play setting on the carrier boards, which increase the configurability and flexibility levels required. Also, there are other electrical wiring and circuitry usually enclosed in each cabinet such as marshaling I/Os that directly interface with the field, protection relays, temperature, light, humidity sensors, and power failure switches [23]. Furthermore, regular industrial automation settings necessitate redundancy. Thus, besides all the features mentioned above, typical industrial RT controller cabinet, field control station, should apply dual redundancy concept in its architecture. For instance, each controller cabinet actually contains two processors (CPUs), DC PSUs, and I/O carrier so that if one (primary) fails, the other takes over (secondary).

2.4.2.9. DCS

If the RT controller composes a single unit in the architecture, then a clearer view of the DCS now emerges that it is technically the distributed networked control through those nodes. DCS is not only the LAN of the controllers and workstations, but also the complete redundant set of H/W, S/W, network, indication, communications, database, security, and functions components primarily designated for process control applications [23]. The DCS embraces two levels in the industrial automation pyramid, CN and SCADA. The PCN on top integrates DCS with other subsystems, PLCs and others. Recall the industrial plants hierarchy divides processes down into I/Os (sensors and/or actuators). Then loops may contain two I/Os or at least one (for open loops, i.e. indications or MLD). After that, many loops compose a process unit. Several process units then form an area. The overall process plant comprises of a number of areas and so on. Starting from the DCS nodes level, each RT controller regularly controls a single process unit. As a result, one area is usually handed by one PIB or RIB whereas overall CN/DCS partially takes over the whole plant. Fig 2.41 (a) below shows a typical DCS system architecture and how each controller takes over a process unit. Similarly, (b) illustrates three to four different automation levels managed by the DCS:

- Level-0, the plant: which is the large scale complex process of mechanical, chemical, and electrical equipment.
- Level-1, the field: which are the field instruments (transmitters and actuators) attached to process lines and machines.
- Level-2, the CN: which is the LAN of DCS RT-controller cabinets located at the RIBs/PIBs. If the process plant consists of more than one area, then additional RIBs geographically allocated close to their associated process will be present.

- Level-4, the SCADA: which is the CCR located at different (safe) sites on the industrial facility's campus. Sometimes the workstations are nodes on the same CN while in other configurations, they form a different Ethernet network.

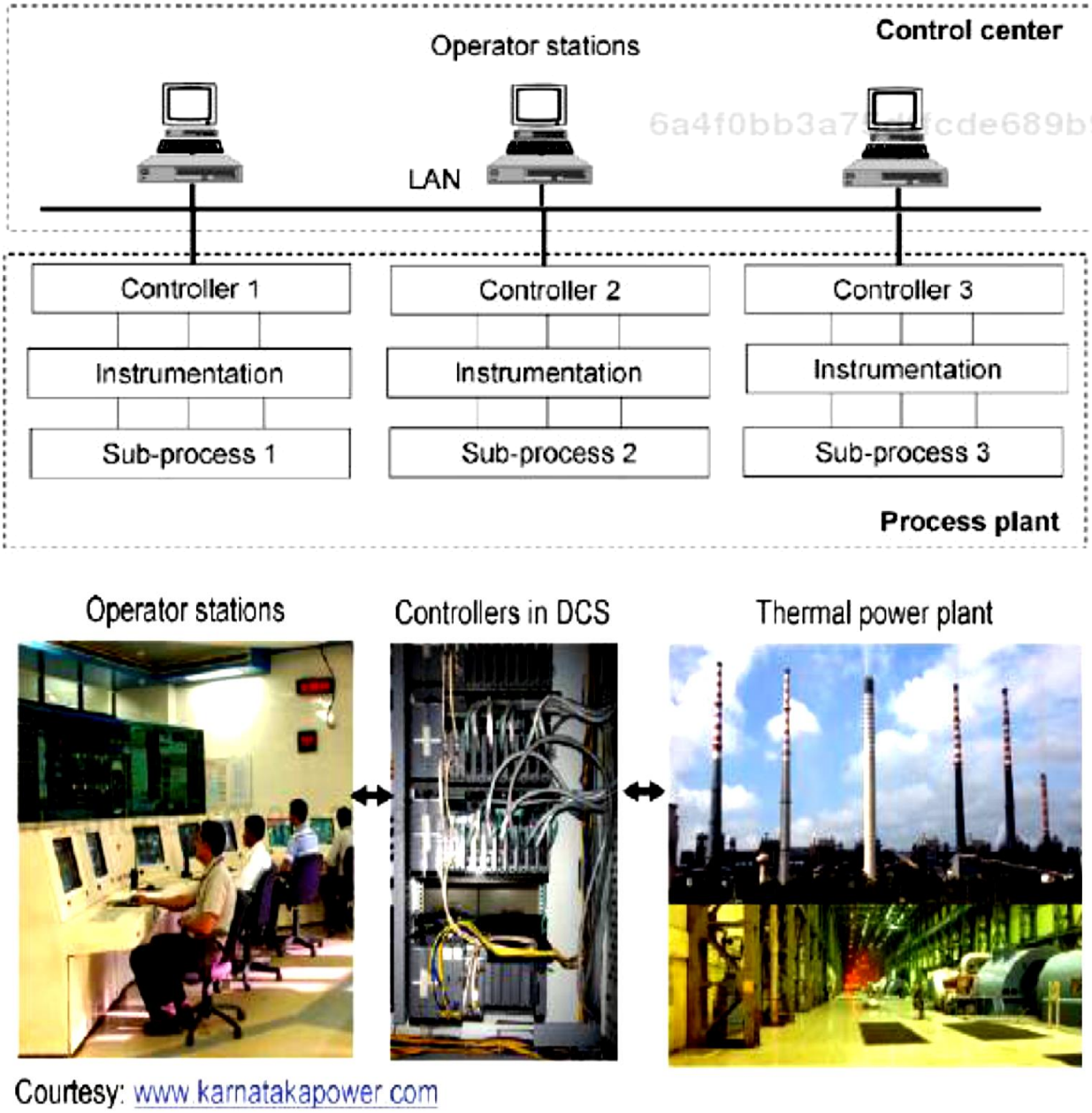


Fig 2.41: (a) DCS forms a LAN of RT controllers, (b) located at the RIBs/PIBs, and operators and engineering workstations, (b) found in the CCR (taken from [23])

2.4.2.10. SIS/ESD

SISs are PLC based systems customized for safety purposes such as ESD, FGS, and others. The term ESD is more frequently used in industrial mediums; however, SIS is the universal naming of such systems. This is another application or deployment of the features of the PLC since fast execution rates are obligatory in SIS. Recall that safety should remain the first priority in each industrial facility. Therefore, based on designed C&Es scenarios that depend on several factors such as the nature of the process, preset engineering conditions, and other expected hazards, SIS logics should guarantee immediate shut-down actions if either one of the causes is true [66], [70]. However, SIS/ESD appear disparate to general purpose PLC or PAC in the sense that the first do not support fieldbus communications at all. The conventional H/W analog or discrete field communication is the only standard that is allowed to transmit/receive measurement and actuation signals to and from SISs as in [68] project example.

This indicates that industrial standards recognize the effect of communication technology on control performance, and so the most trusted and efficient field connection method to serve that purpose appears to be the conventional point-to-point one introduced earlier. Furthermore, if DCSs should provide dual-redundancy, then some SIS may offer triple. In other words, SISs are H/W engineered and organized into cabinets and RIBs similar to DCSs, but triple redundancy tends to place three transmitters/actuators at the same single PV/MV in the field, three CPUs, PSUs, and I/O carriers per one functionality at the system cabinet level. Some other SISs are dual redundant though; however, they should follow certain industrial standards constituted by leading groups and organizations in the process automation field, ISA and IEC. The standards have systematically established a certain standardized assessment when evaluating SIS/ESDs. This standard is called SIL. It divides the probability of the SIS failure to prevent emergency hazard into four levels, SIL-1 to 4. SIL-4 the closest to perfection where most SISs reach SIL-3 [67].

2.4.2.11. RTUs

RTUs are majorly PLC based control units. From their designation, RTUs are placed at remote locations to monitor and control devices. The application for RTUs is not continuous process control. Rather they are designed to actuate pumps and discrete devices and monitor remote pipelines and equipment that are not located at the same facility site as grid stations, crude oil lines, and others. Fig 2.42 below illustrates an example of an RTU that is located at a remote electrical grid substation while the supervisory monitoring and operation of such distant site is located at a mother company's facility, SCADA.

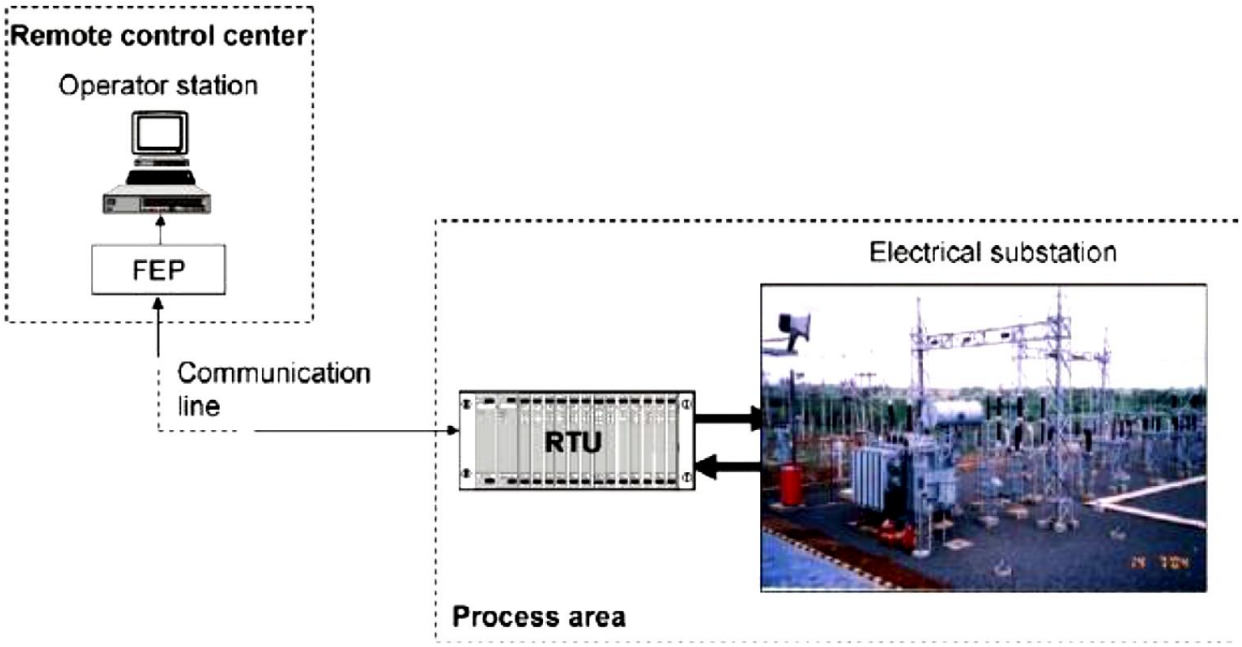


Fig 2.42: An RTU used for monitoring and supervisory control a remote site

2.4.2.12. SCADA systems

If an RTU forms a single unit, then SCADA systems are the whole networked based system. Similar to DCS, a network of RT-controllers and computer workstations form the CN, RTU here compose the SCADA system along with supervisory workstations located at a far-away site. Here, two confusions may emerge: First, the term SCADA has been used to describe one level in the

industrial automation PCS architecture, which DCS derives as a part of it, various times. Hoverer, here the name is slightly different, SCADA system, which means that the whole architecture is supervisory monitoring and control based, not for PCS purposes.

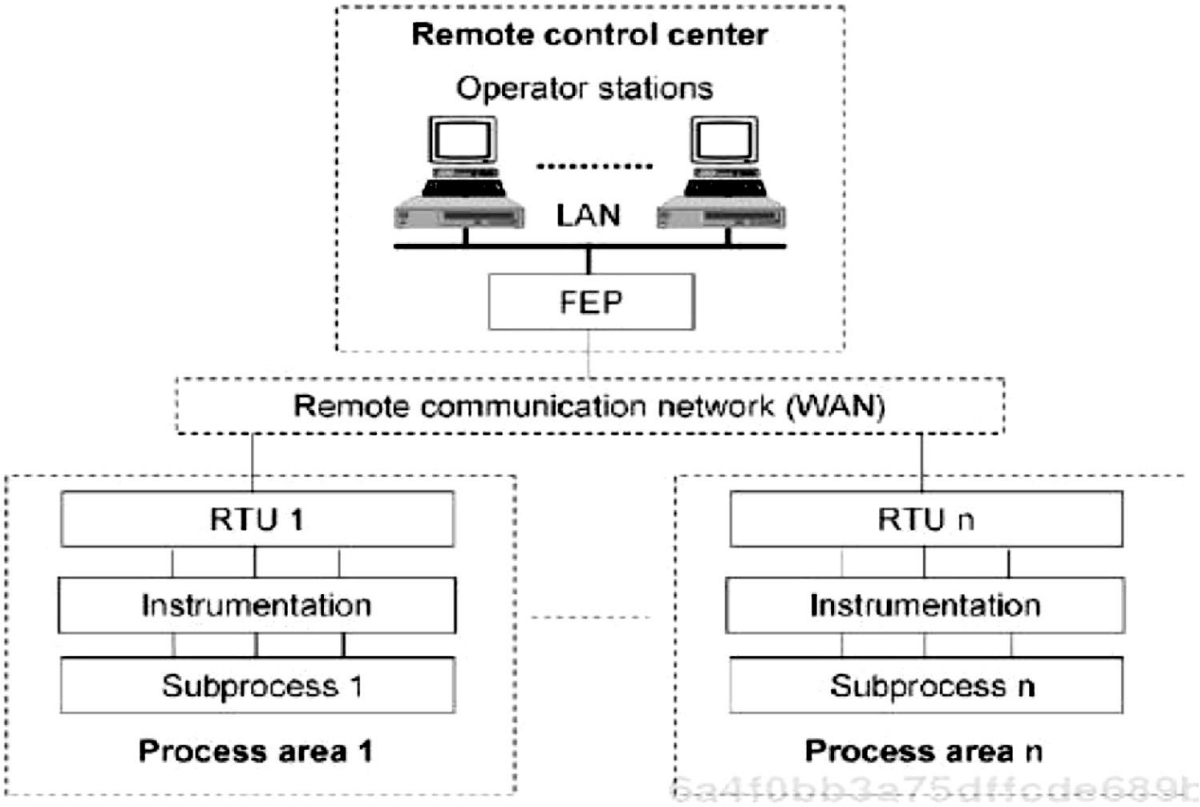


Fig 2.43: A SCADA system connects a single LAN CCR of HMI workstations to a number of remote sites through implementing WAN (taken from [23])

The second query that may arise regarding SCADA systems is what would distinguish them from DCSs if both are technically networked control units and HMI workstations. The answer to this enquiry from a more technical perspective can be summarized in the following points:

- First, although both are made by networking, SCADA systems are called networked based in industry while DCS are not. This is because of the fact that even though it is distributed, a single DCS system should be located geographically at the same process plant site while a SCADA system may communicate to several remote sites located at different locations.

- Second, as Fig 2.43 demonstrates since they connect remote sites to control rooms that require longer distance ranges, SCADA systems are WAN based while as presented DCS are LAN ones.
- Third, as stated above SCADA/RTU systems are PLC based whereas DCS are regular RT-controller processor units.
- Lastly, given that SCADA systems are mainly PLC based, it is settled that the latter are event driven while the DCS is time determined [23].

2.5. Control Algorithms

Control algorithm represents the calculation laws or regulator designs that are executed in the controller block described earlier. It is the part that has been thoroughly studied in the literature. It is still the most important step in control loop implementation as well. If assuming ideal instrumentation implementation, i.e. field communication, processing unit, system architecture, and mechanical setup, control algorithm is the key phase that determines stability, performance, and robustness of the closed loop. Control algorithms vary from the simplest to more complex. The simplest control algorithm of all is the ON/OFF. Alongside ON/OFF, this research classifies regulatory, non-regulatory, and mixed algorithms in general into four groups: PID controller, cascade and feed-forward, advanced algorithms, and sequential control. Those concepts are the foundation of further designs presented later. Here recall that the basic first step in feedback controller function before passing to the control algorithm block is comparison of PV and SP, such that:

$$e(t) = \mathrm{SP} - \mathrm{PV}$$

That produces the error function, *e(t)* as a result of applying negative feedback (subtraction).

2.5.1. ON/OFF control

The simplest and the cheapest of all is the ON/OFF control. It works as per the simple criteria that

$$u(t) = \begin{cases} u_{max}, & if \text{ PV} < \text{SP}, e(t) > 0 \\ u_{min}, & if \text{ PV} > \text{SP}, e(t) < 0 \end{cases} \quad (57)$$

Given that *u(t)* is the controller output. This will lead to tolerance, δ, error, or offset which may cause overshoot or oscillatory response of PV as well each time the controller acts:

$$u(t) = \begin{cases} u_{max}, & if\ e > \delta \\ u_{min}, & if\ e < -\delta \end{cases} \quad (58)$$

Thus, the algorithm appears simple to be designed and implemented. It is also called a two position or "bang-bang" controller as the control signal is either 0% or 100%. It does not involve any regulatory functions as the controller block can be modeled as a discrete event system. The controlled process should not necessarily be a discrete event system though, but it can be a continuous time as well. ON/OFF control is ideal for discrete processes and plants as already presented. However, for continuous time processes, ON/OFF associates shortcomings as the practical implementation rehearse dead-band tolerance that the transmission between the two states incorporates delay time. As a result, the system never reaches the steady state, and hence control at SP is not achievable. ON/OFF control may derive useful for large sluggish systems, particularly those incorporating electric heaters where the typical sinusoidal cycling periodic time is large. Less periodic, in contrast, results in excessive cycling, and hence instability [55], [70].

2.5.2. Regulatory Control

Regulatory control is classified under PID, cascade, feedforward, and advanced algorithms.

2.5.2.1. The classical PID controller

The most common and applied regulatory industrial control algorithm remains the PID controller. It is observed in industry as a single block with three, or more, design or tuning

parameters. Thus, the PID might take various configurations, P, PI, PD, or PID. The PID modifies the ON/OFF control disadvantage mentioned above by replacing its action to proportional gain.

In other words, the control signal or output, *u(t)*, gain is directly proportional to the magnitude of error. The P mode appears basic in all regulatory actions as it rejects the discontinuity of the ON/OFF. If the parameter is properly selected, the P mode has a feature of speeding up the dynamic response and not changing the order of the closed loop control system. Thus, the P control does not introduce overshoot. However, a sustained control offset or SSE characterizes one disadvantage of the P mode only. The plain P controller performs sensitive to noise also. To resolve the problem of restoring the offset or SSE as a result of the P control requires adding an integral action, I [18], [70].

The parameter, I, (industrially given the term "reset") mode helps upgrade the dynamic order of the loop by adding more overshoot to the response, therefore eliminating or reducing the SSE. By adding the I gain and function to the P, PV approaches SP rapidly. In other words, how fast the PV can reach the SP is determined by the tuned I parameter. However, the I mode increases the risk of the overshoot; therefore, this introduces oscillation in the response and consequently may reach instability. This issue of an excessive overshoot and oscillation associated with the I action can be suppressed by adding a D function to the resulting control law. Thus, the D mode compensates the control action by anticipating deviations in the controlled variable and acts before the full change occurs. The D parameter then enhances the dynamic response but does not change the order of the loop. The D mode is also called the rate action in industry. Summing it all up produces the following time domain PID controller [18], [55], [70]:

$$u(t) = K_P e(t) + K_I \int e(t)\mathrm{d}t + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t). \tag{59}$$

Taking the Laplace transform yields the known PID TF in S domain where $C(s) = \frac{u(t)}{e(t)}$

$$C(s) = \left(K_P + K_I \frac{1}{s} + K_D s\right). \tag{60}$$

The PID controller TF can also be expressed in z domain given in digital control system [69] as:

$$C(z) = K_P + K_I \frac{(z+1)}{(z-1)} \frac{\Delta T}{2} + K_D \frac{(z-1)}{z\Delta T} \tag{61}$$

The PID controller is a summation of the three basic regulatory modes (Fig 2.44 (a) and (b)).

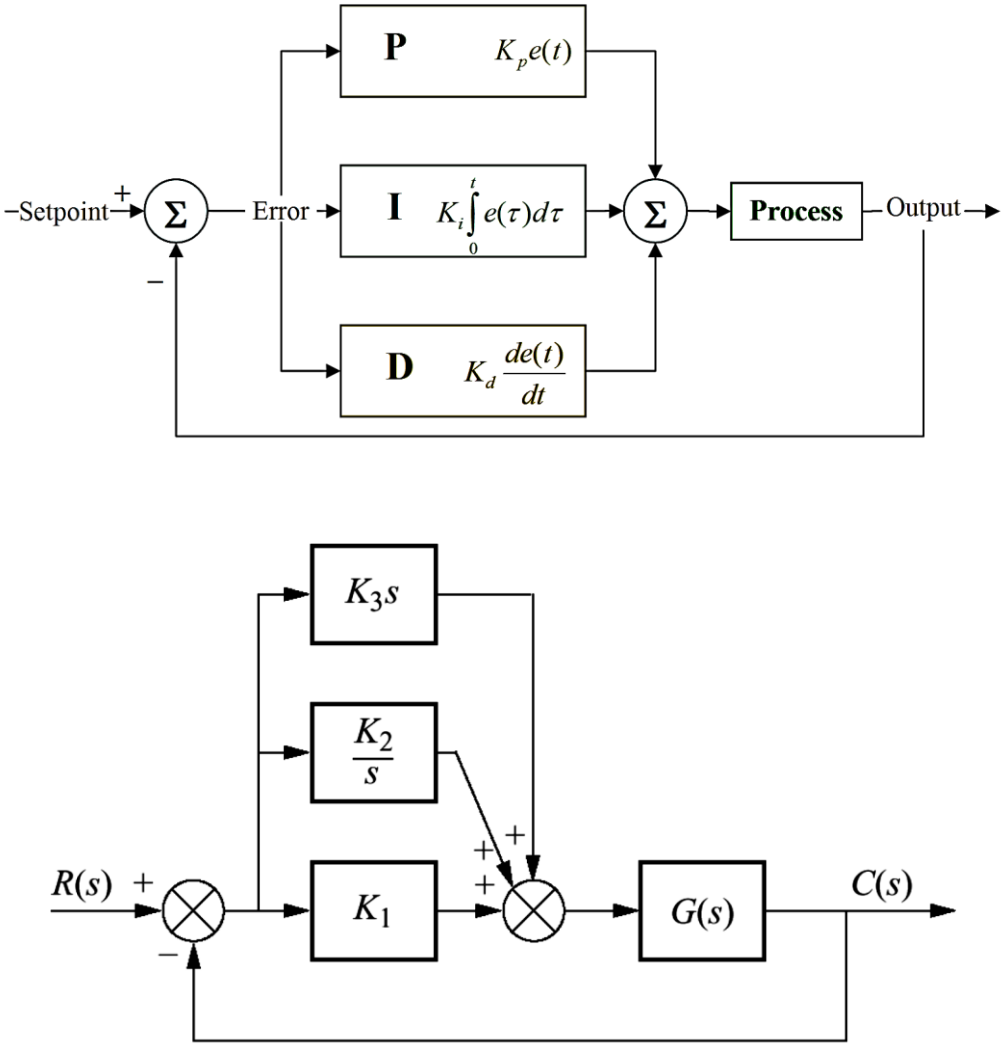


Fig 2.44: The PID controller BD in the feedback (a) (upper) time and (b) Laplace domains

(taken from [51])

2.5.2.2. Cascade and feed-forward control

The basic regulatory feedback closed loop described so far could be enhanced by more advanced control techniques such as the cascade and feed-forward. Starting with cascade control, it is also a common, widely implemented, and useful technique that adds more robustness and compensation as well as overall quality to the loop dynamic response and performance. Cascade control is defined as *"*a control system in which a secondary (slave) control loop is set up to control a variable that is a major source of load disturbance for another primary (master) control loop" [70]. The controller of the primary loop determines the SP of the summing controller in the secondary loop" [71]. The general BD structure of a system realizing cascade control appears in Fig 2.45, where the process is divided into a primary variable and a secondary. Two feedback control loops are cascaded here, an outer and an inner for the primary and secondary PVs respectively, provided that there exists a direct relation between both RT variables [70].

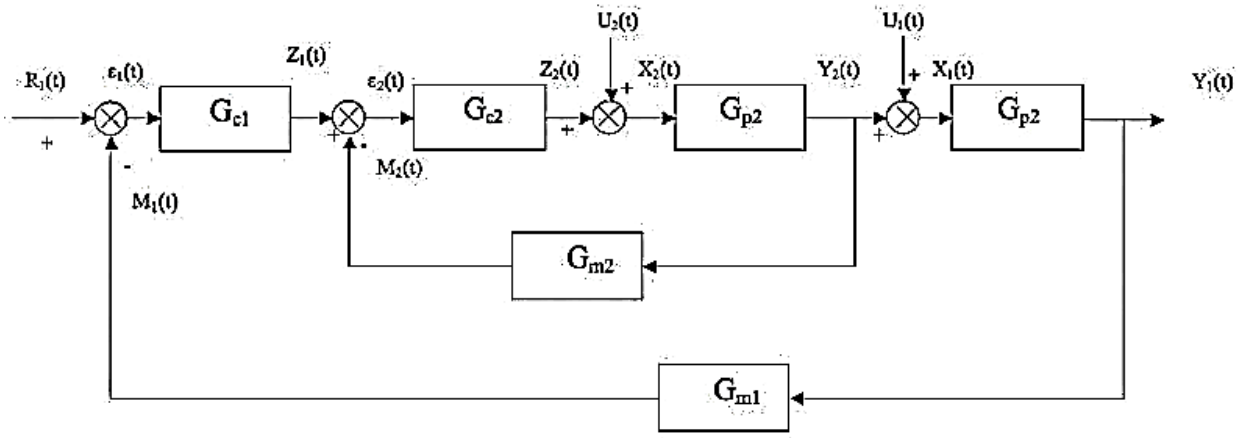


Fig 2.45: The general cascade control structure BD (taken from [70] and [72])

In the right circumstances, i.e. proper tuning, bandwidths, scaling, and relationship, the cascade strategy can significantly improve the overall performance of the control system of the ultimate controlled variable. So the question that might arise here is what technical advantage does the inner loop provide? It actually reduces phase lag of inner process setting limits produced

from the master controller. In addition, inserting an inner loop reduces the effect of the dead time introduced to the overall response as an effect of the secondary PV related to the main one. Thus, the cascade control technique works for compensating disturbances of the inner loop before they affect the outer one, therefore, preventing non-linearity, phase lag, and uncertainty of the secondary PV to reach or to affect the primary [70].

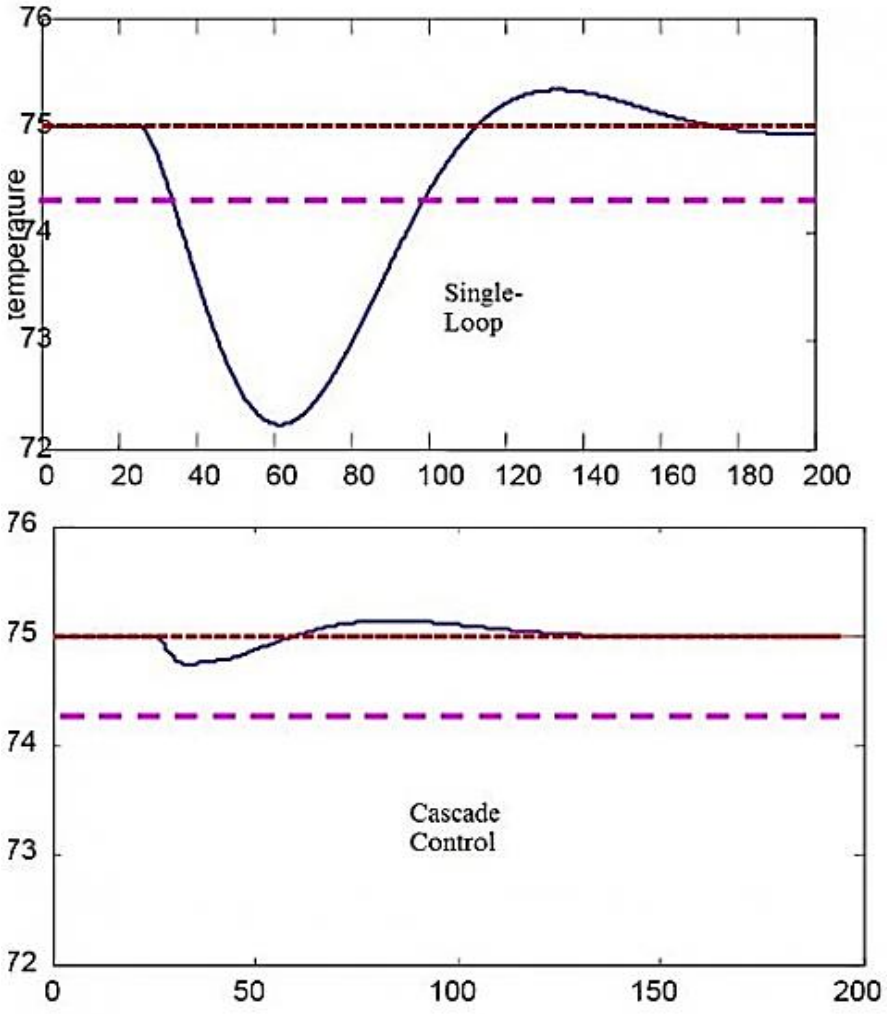


Fig 2.46: The advantage of cascade strategy on control performance (taken from [72])

Fig 2.46 demonstrates a proven experimental example of the positive effect of applying cascade control strategy. The figure clearly compares the difference between a single loop's (feedback) and the cascade control's performance presented in [72]. The advantage is noticeable.

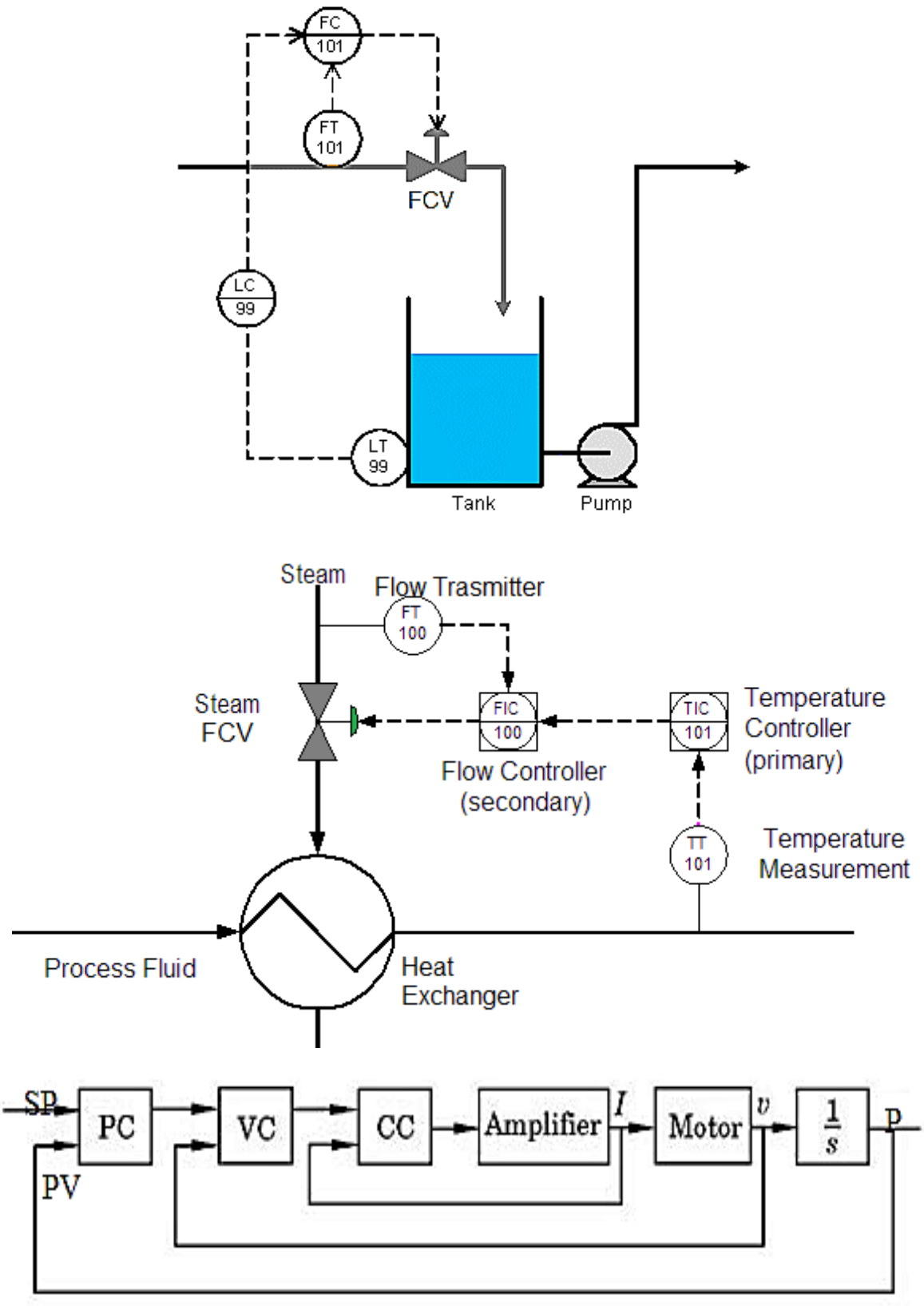


Fig 2.47: Three examples of cascade control applications from top to bottom, (a), (b), and (c).

Similarly, Fig 2.47 above shows three different examples of realizing cascade control. In (a), controlling the level in the tank is directly related to manipulating the inlet flow rate. Thus, the primary variable in this example is the level while the secondary (inner) is flow. This cascade strategy is implemented by taking the output of the level controller and passing it as a SP to input to the flow loop where the actuator, CV, acts accordingly. The higher the flow, the higher the level and vice versa. (b) demonstrates a similar example, however, incorporating different RT variables: temperature as primary and flow (secondary). Here, increasing the steam flowing to the heat exchanger raises the temperature of the fluid or gas in the pipeline.

Cascade control tends to be useful in other control applications other than the process industry as in the third example, (c). Three staged cascaded loops are found applicable in servo motor applications. In this motion control example, since the induction current is operational to torque, which is related to the velocity in some setup, the speed of the motor is measured and fed-back, which poses an inner loop to the master (position). It is known that the rotational speed is the nothing but the rate of change of the angular position; therefore, integrating the first produces the latter. Thus, identifying the relation between interconnecting variables is the first step in this strategy. There are other essential parts of such design as loops rates, scaling, and conversion.

However, there appear several cons of applying cascade control. First, it increases the complexity level of the dynamic system. Second, cascade control requires more field instruments (sensors) which increase the cost of the implementation. Lastly, to tune and design a cascade loop is a more difficult task [72]. For instance, if the parameters of the inner and outer controllers were not selected properly, the two loops will fight each other and, hence, the expected performance will degrade. Thus, an engineer designing cascade loops should pay attention to several factors. There are other common strategies as ratio, split rang, and override control [70].

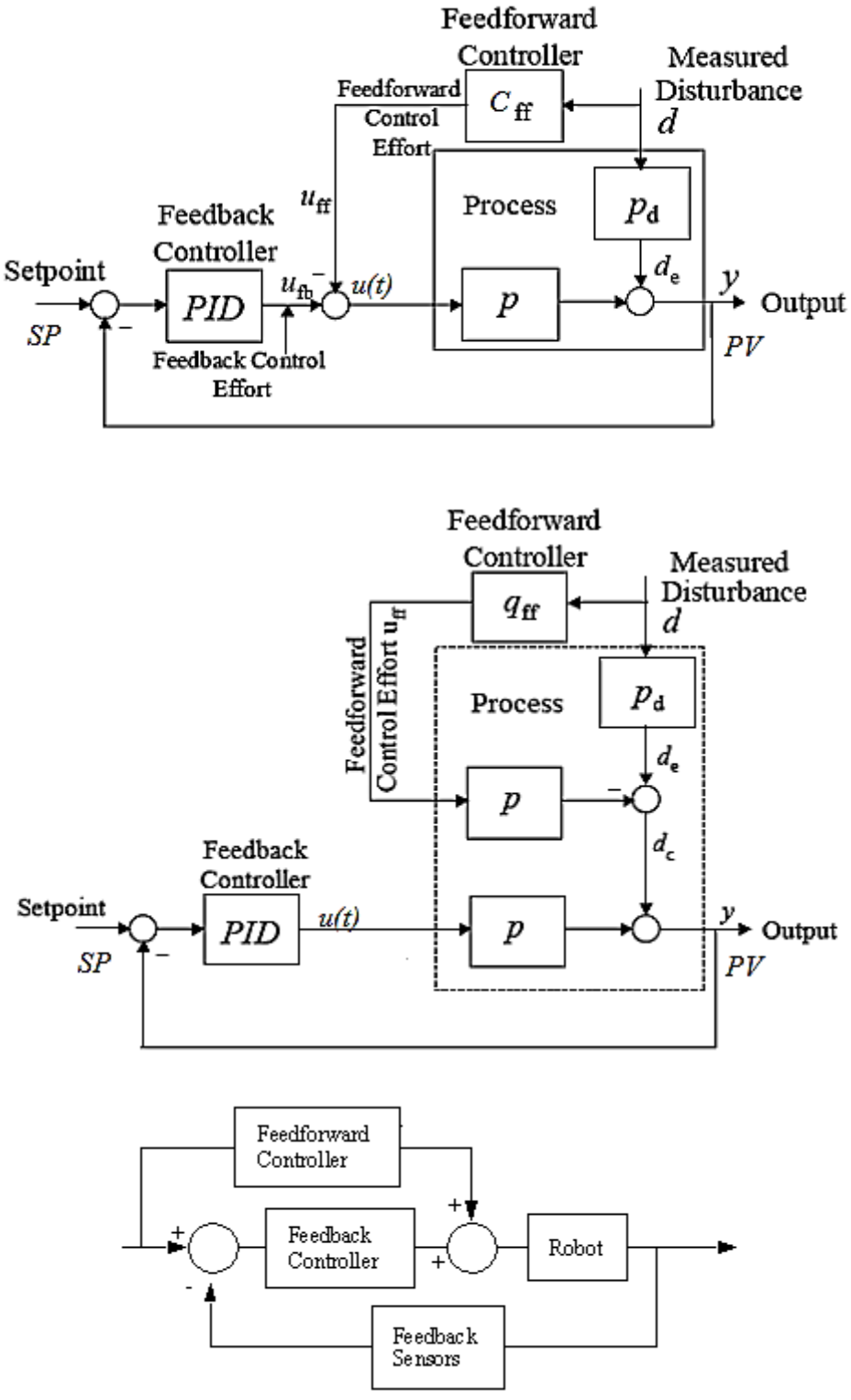


Fig 2.48: (a) BD methodology of disturbance measuring and feeding it forward to reduce its effect, (b) an equivalent diagram, and (c) an example of feed-forward control in robotics

Another technique that is usually applied in parallel with feedback, and could be individually, is the feed-forward control. Feed-forward control contributes to the overall control performance by concentrating on the disturbance. By measuring the disturbance before it affects the process output and placing a controller that its output is subtracted from the controller's, the feed-forward emerges different than the feedback that holds and measures the output then feeds it back [70]. Combining both, feedback and feed-forward would produce desired results in most cases. Fig 2.48 illustrates (a) the process of feed-forward control in disturbance reduction, (b) an equivalent method, and (c) a working example in robotics where the load change of position commands, SP, is estimated and fed-forward to the controller output to enhance the response.

2.5.2.3. Advanced algorithms

The parameters of the basic and modified regulatory control algorithms described above could be designed using several conventional, scientific, and academic methods studied in [18], i.e. stability analysis, root locus poles and zeros plans, frequency response, etc. However, due to lots of altering conditions of the process plants as initial designs, startup and commissioning, shutdowns, and online disturbance variations, the practical industrial circumstances call for more flexibility, less time consuming designs (especially online), and more advanced algorithms. Thus, control theory researchers have come up with many advanced algorithms concerning approaching the best designs for particular applications.

Many theories have emerged: fuzzy logic, robust control, adaptive control, optimal control, NL control, IMC, MPC, and others. Furthermore, to study applicable examples, analyzing and comparing results realizing those advanced methods seems motivating since they aid in RT control loop performance, hence advancement in the overall quality of automated plants. The vast majority of literature studying applied control focuses on this direction, finding out efficient

algorithms that enhance control performance. However, this paper considers other RT non-ideal implementation facts besides applying advanced algorithms to improve the overall control system problem. It is also vital to mention that most if not all stated algorithms above realize the basic PID either fully or partially as foundation while executing the advanced designs on top.

All control design methods and algorithms can be classified as either model-based or non-model-based. The non-model based does not require constructing plants' models. Rather such a method deals with online tuning of regularity controllers' parameters based on the RT response. Common examples of the non-model based controller design would be the PID tuning methods based on trial and error, open loop tuning, ZN, Cohen-Coon, and Tyreus-Luyben methods. Some of those are described in detail in [70], [72]. A set of tuning methods mentioned here follows:

- Open loop tuning: the open loop tuning method is simple. It suggests turning the loop to the manual mode (disconnecting the automatic feedback action and triggering the controller as a MLD) during the tuning process. After that, an online input step response, as the one that appears in Fig 2.49 below, should be monitored where information as the maximum slope, $y_2 - y_1/u_2 - u_1$, dead time, τ_d, and time constant, τ, are estimated. Then the design parameter should follow the ones that appear in Table 2.1.

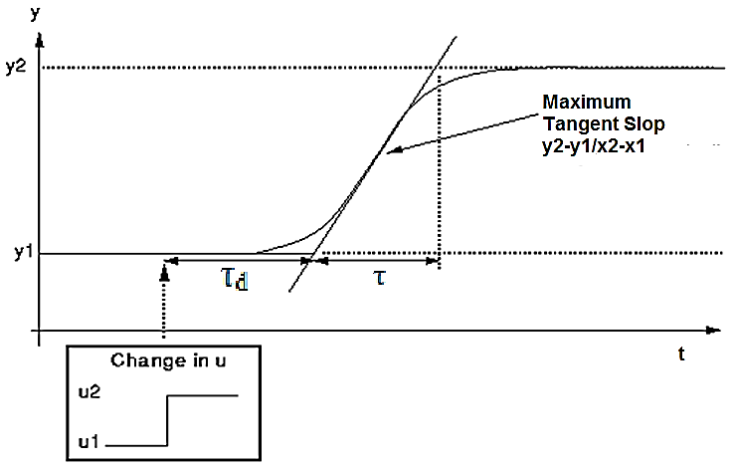


Fig 2.49: An open loop response to a step change in SP or plant's input controller output, Δu.

	K_c	T_i	T_d
P	K_0		
PI	$0.9K_0$	$3.3\tau_{dead}$	
PID	$1.2K_0$	$2\tau_{dead}$	$0.5\tau_{dead}$

Table 2.1: The open loop tuning method's parameters (taken from [72])

Where $K_0 = \tau\Delta u/\tau_d\Delta y$. The advantages of the open loop tuning method rely on the fact that it is quick and simple to be implemented, considered to be robust, and does not cause a lot process loss or damage while tuning. On the other hand, the method might not be so accurate sometimes when estimating values online, especially in case of fast processes [70], [72].

- Closed loop ZN: instead of turning the loop to manual mode for online tuning, the ZN method uses the feedback setup directly in order to determine the control parameters. The method suggests increasing the P gain slowly until a similar closed loop response, of Fig 2.50 below, reaches an oscillatory period of the repeated same amount of time, P_u.

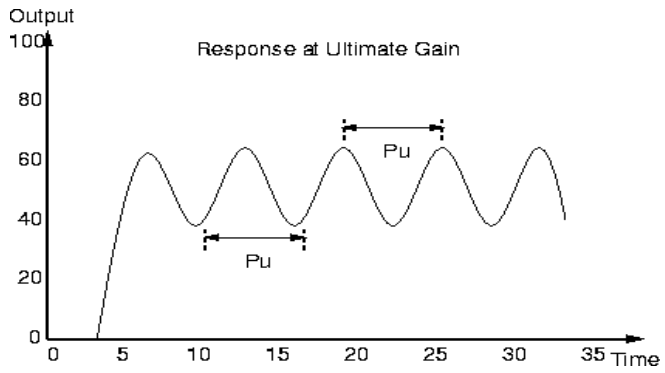


Fig 2.50: The closed loop (ZN) tuning method's constant oscillation gain determination

The tuning parameters then follow the resulting expressions in Table 2.2.

	K_c	T_I	T_D
P	$K_u/2$		
PI	$K_u/2.2$	$P_u/1.2$	
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

Table 2.2: Closed loop ZN tuning method's parameters (taken from [72])

- Cohen-Coon Method: the ZN tuning method presented above has proven its validity and effectiveness in countless applications. However, the ZN involves some disadvantages also as it is the time consuming and may drive the system unstable if tolerating constant oscillatory responses while tuning. The ZN does not take into account the time delay in some process associated with long dead time because systems respond slowly and inaccurately under such conditions using this method [72]. Therefore, the Cohen-Coon considers the time delay factor, but the method is deliberated an offline tuning that it requires reaching the steady state in order to launch the procedure. The following parameters in Table 2.3 are obtained from a step response of the system:

	Kc	Ti	Td
P	(P/NL)*(1+(R/3))		
PI	(P/NL)*(0.9+(R/12))	L*(30+3R) / (9+20R)	
PID	(P/NL)*(1.33+(R/4))	L*(30+3R) / (9+20R)	4L / (11+2R)

Table 2.3: The Cohen-Coon tuning parameters (taken from [72])

Though, this method assumes a first order closed loop response to work correctly, which does not reflect the practical facts that a lot of systems out there are more complex and of higher orders. This drawback limits the capability of Cohen-Coon method in addition to the necessity of attaining the steady state [72].

The model based designs though necessitate modeling the process as a dynamic system first then determines the controller parameters based on theoretical or simulation analysis of the whole dynamics. If considering advanced control laws only, this research may refer to a number of algorithms and concepts in several locations whether in simulation on implementation. The description of such advanced methods follows:

- Robust control: robust control is a general principle that deals with designing high gain control parameters that target to prevent the controlled system from uncertainty and sensitivity due to disturbances or unmeasured changes in the process. Thus, methods that deal with reducing the disturbance and aim for robust designs such as the feed-forward or cascading can be counted under this advanced approach. Simpler examples of robust control are high PID gains, adding a SP filter to cancel zeros in overall transfer function, the robust purposed IMC, and others. However, robust control theory is studied as a systematic method in control materials and texts [18].
- Adaptive control: this technique frequently adjusts the control designed parameter with sudden changes or uncertainty in the process. There are several classifications and types that derive under adaptive. Some methods rely on the concept of system identification and model estimation based on RT empirical data to retune the control parameter while others depend on a referenced defined model. Some other adaptive techniques do not refer to modeling, but they apply auto-tuning control using the methods described above [70]. Adaptive control methods also practice the basic feedback and feed-forward in calculating their adaptive parameters. Examples of adaptive control applications would be auto-tuning PID based on gain scheduling [51], ZN, IMC, and others.
- Optimal control: unlike all other methods, the optimal strategy considers the control effort's cost besides optimizing controllers' parameters. Optimal control deliberates the multivariable or the MIMO model of complex systems and then uses mathematical optimization methods to design the control parameters as per kwon set constrains. The most common example of optimal control is the LQR algorithm which aims to minimize a term known as quadratic continuous-time cost function,

$$J = \tfrac{1}{2}\int_0^{\infty} [\mathbf{x}^{\mathrm{T}}(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^{\mathrm{T}}(t)\mathbf{R}\mathbf{u}(t)]\ \mathrm{d}\,t \tag{62}$$

Given the constrains

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$

Which is nothing but the general system SS model given in (2) with an initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. This is done by designing a SF controller of gains matrix K,

$$\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t) \tag{63}$$

Such that an H matrix is obtained from,

$$\dot{\mathrm{X}} = \mathrm{A}\mathrm{x}(t) - \mathrm{B}\mathrm{K}\mathrm{x}(t) = \mathrm{H}\mathrm{x}(t) \tag{64}$$

Then the problem turns to minimizing

$$J = \int_0^{\infty} \mathrm{X}^T \mathrm{X} dt = \mathrm{X}^T(0)\mathrm{P}\mathrm{X}(0) \tag{65}$$

Where P is a matrix that stratifies:

$$\mathrm{H}^T\mathrm{P} + \mathrm{P}\mathrm{H} = -\mathrm{I} \tag{66}$$

Recall *I* is the identity matrix. This is called a general SF optimal controller [18]. However, accounting cost or energy of control effort besides the performance requires solving (62) where R, the scalar weighting factor, should be selected $R > 0$ and minimized such that:

$$\mathrm{K} = \mathrm{R}^{-1}\mathrm{B}^T\mathrm{P} \tag{67}$$

Where Q is an $n \times n$ matrix, and it determines the following using the eigenvalues, λ:

$$\mathrm{Q} = \mathrm{I} + \lambda\mathrm{K}^T\mathrm{K} \tag{68}$$

P is obtained from below, and a such method is known as the infinite horizon LQR [18]:

$$\mathrm{A}^T\mathrm{P} + \mathrm{P}\mathrm{A} - \mathrm{P}\mathrm{B}R^{-1}\mathrm{B}^T\mathrm{P} + \mathrm{Q} = 0 \tag{69}$$

Also [69] studies optimal control algorithms as the LQR in discrete time domains.

- IMC: IMC was first developed as a robust control algorithm because it keeps in mind the robustness of the system and rejecting disbranches and uncertainty in the process. As a result, IMC uses similar structures to the cascade and feed-forward in its strategy. It can also be considered though as an online tuning method since it depends on empirical data from the RT loop response. To compare the performance curves of the IMC with other online tuning methods such as the ZN and others, one could infer that the first does not cause an overshoot or oscillation in the closed loop response while considering dead time delay constraints [72]. This paper classifies the IMC as a distinct advanced control algorithm. It can be viewed as both model and non-model based design methods. The IMC algorithm is non-model based in the sense that no theoretical model construction is required prior to designing the controller that relies on RT trends. The IMC is also model based since estimating the dynamic of the system appears essential in such a method. The principle of the IMC is based on the statement, "control can be achieved only if the control system encapsulates, either implicitly or explicitly, some representation of the process to be controlled" [73]. Fig 2.51 shows the general structure of the IMC algorithm containing a plant and its estimation with a delay time model, e^{-Ls} besides the controller, $C_{IMC.}$

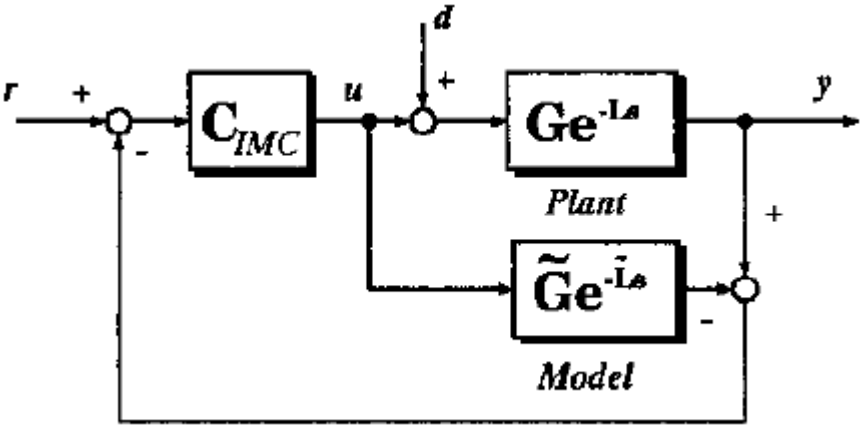


Fig 2.51: General structure of the IMC (taken from [74])

Looking at the BD above, the algorithm estimates the plant model, $\tilde{G}_p(s)$ that associates delay time from RT data and then plugs it in the algorithm shown in Fig 2.51. Thus, it is clearly noticed that the perfect control can be theoretically achieved if the estimated model exactly represents the real plant, i.e.$\tilde{G}_p(s) = G_P(s)$, and the inverse of that, i.e. $\tilde{G}_p(s)^{-1} = G_C(s)$, [70]. However, in RT practice no perfect model estimation is possible, so a LPF TF of an *n* order, $G_{LPF}(s)$, appears always attached with the controller block to increase robustness by attenuating model mismatch which in total results in the internal model controller, $G_{IMC}(s) = G_{LPF}(s)G_C(s)$. Hence, the output response of such an algorithm is

$$Y(s) = \frac{G_{IMC}(s)G_P(s)R(s)+\left[1-G_{IMC}(s)\tilde{G}_p(s)\right]d(s)}{1+\left[G_P(s)-\tilde{G}_p(s)\right]G_{IMC}(s)} \tag{70}$$

Where $R(s) = SP$ and $d(s)$ is the disturbance. Hence, with time delay:

$$Y = G_P(s)G_{IMC}(s)e^{-Ls}R(s)\,[1 - G_P(s)\,G_{IMC}(s)]G_P(s)e^{-Ls}d(s) \tag{71}$$

Now assuming perfect control, $\tilde{G}_p(s) = G_P(s)$ and $G_C(s) = \tilde{G}_p(s)^{-1}$, and subtitling $G_{IMC}(s) = G_{LPF}(s)G_C(s)$, results:

$$y_{IMC} = G_{LPF}(s)e^{-Ls}R(s) + \tilde{G}_p(s)e^{-Ls}[1 - G_{LPF}(s)e^{-Ls}]d(s) \tag{72}$$

From this result, assuming no delay time, $e^{-0s} = 1$, it is clear that the IMC suggests that the perfect tracing,$Y(s) = R(s) \leftrightarrow$ PV = SP, can be achieved when $G_{LPF}(s) \equiv 1$ [74]. The IMC can be implemented on top of a regular PID controller [70], [73] (Fig 2.52). The internal model controller, $G_{IMC,}$ can be written in terms of loop regulator, $G_C(s)$, and the estimated plant model as $\tilde{G}_p(s)$ as (71) below, from which the IMC-PID follows [73].

$$G_{IMC}(s) = \frac{G_C(s)}{1+ G_C(s)\tilde{G}_p(s)} \tag{73}$$

$$G_{PID}(s) = \frac{G_{IMC}(s)}{1+G_{IMC}(s)\tilde{G}_p(s)} \tag{74}$$

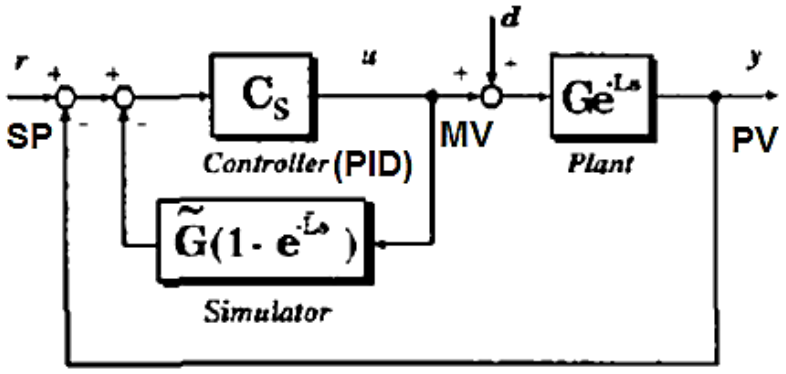


Fig 2.52: the IMC on top of PID

- MPC: MPC is different than the advanced techniques described above by taking into consideration besides current time's and past's parameters and SPs, future ones. Therefore, MPC depends on empirical or constructed models of the plant to anticipate future behavior of the process and act accordingly. Although the MPC is a general principle, it is primarily applied for multivariable problems in industry where complex loops arise and relations between SPs and control parameters of multiple sensors and actuators depend on further calculations and the history, nature, and future of the process. Thus, MPC involves three main steps. First, estimating an acceptable plant model learned from the behavior of the RT system. Second, predicting future PVs from past and current MIMO relations. Third, optimization of the control parameters, for example PID, based on the identified model. The number of predictions is called "the prediction horizon." Moreover, not only does MPC care for control laws but also optimal SP calculations for the controlled process. This step is done through optimization methods based on various objectives and sets of assumptions as maximizing the operation profit, minimizing reference value deviation, or maximizing the rate of production [70]. If the above process is implemented automatically, then the MPC can be considered as an adaptive tuning method also.

2.5.3. Sequential control

As mentioned earlier, some processes require sequential type of control to be automated. Sequential control appears essential in those types of processes and plants where timing, batches, phases, conditions, and preconditions are the bases that derive the automation system. However, sequences also exist in the continuous types of process plants as an automation functionality above the multivariable control, i.e. MPC. This is true in many petrochemical, oil, refinery, and power plants where a fully automated operation of a whole continuous process unit is possible after satisfying some permissive conditions. Then the sequence may include phases or steps, holds states, internal conditions, and termination [70].

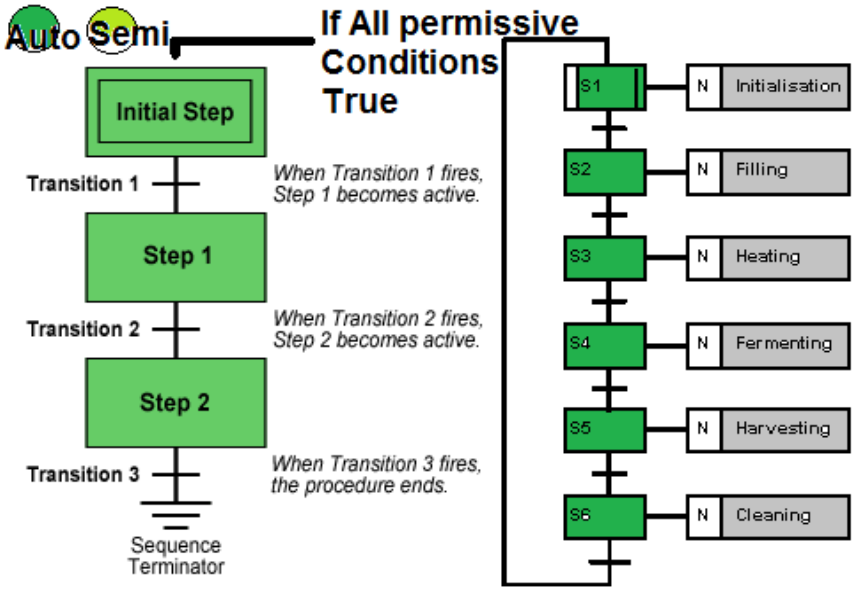


Fig 2.53: (a) The IEC SFC model (left) and (b) a process example (right)

Programing sequences in industrial automation follows the IEC-61131-3 standard that defines a special FB for the purpose, SFC. It suggests dividing the sequential control of batch/semi-batch processes into steps and transitions [70]. Upon stratifying initial conditions, the sequence starts. Then each step executes if the transition conditions are true and no holds emerge as in Fig 2.53.

2.6. Final Control Elements (Actuators)

A very important part of the control loop is the actuator. It is the part where it translates all the control laws and signals into RT physical actions on the process. Thus, understanding the physics, dynamics, and mechatronics of the final control element appears vital in designing and implementing a healthy feedback control loop. In other words, ignoring sizing and selecting a proper actuator for the application would result in unwanted experimental results when running the loop in RT even though the best and optimum control algorithm is designed. Since this paper implements laboratory cased suites of topics discussed, the main actuators used are defined and introduced here. There are various kinds of final control elements in instrumentation for process control. Some are continuously proportional to the control signals such as CVs, motors, analog lights, fans, etc. while others follow switching ON/OFF bases, i.e. discrete devices as pumps, heating elements, open/close three states valves, and solenoids. This paper divides them all into two categories: motor, and other instruments, since the first is further discussed and implemented in the case studies and the latter appear basic in industry.

2.6.1. Motors

Source [75] gives basic information about different types of electrical motors used as actuators in RT control applications and their theory of operation. As introduced earlier, dynamics systems can be classified according to their physical nature. Motors are electromechanical devices that in general take electrical current as an input, work as per electromagnetic and induction principles, and then produce torque or mechanical rotation as outputs. Thus, motors have electrical as well as mechanical parts. The basic physical law that forms the underlying foundation of motion physics of all motors is the Newton's law that is described in equation (10),

$$F = ma$$

Where F denotes force, m is mass of an object, and finally, a, the acceleration in which the object is traveling. However, the above equation matches linear motion only from where it has an equivalence with respect to rotational motion, and that is (13) also introduced earlier. To recall,

$$T = I\dot{\omega}$$

Where T here is the torque and $I = J$ is the mass moment of inertia.

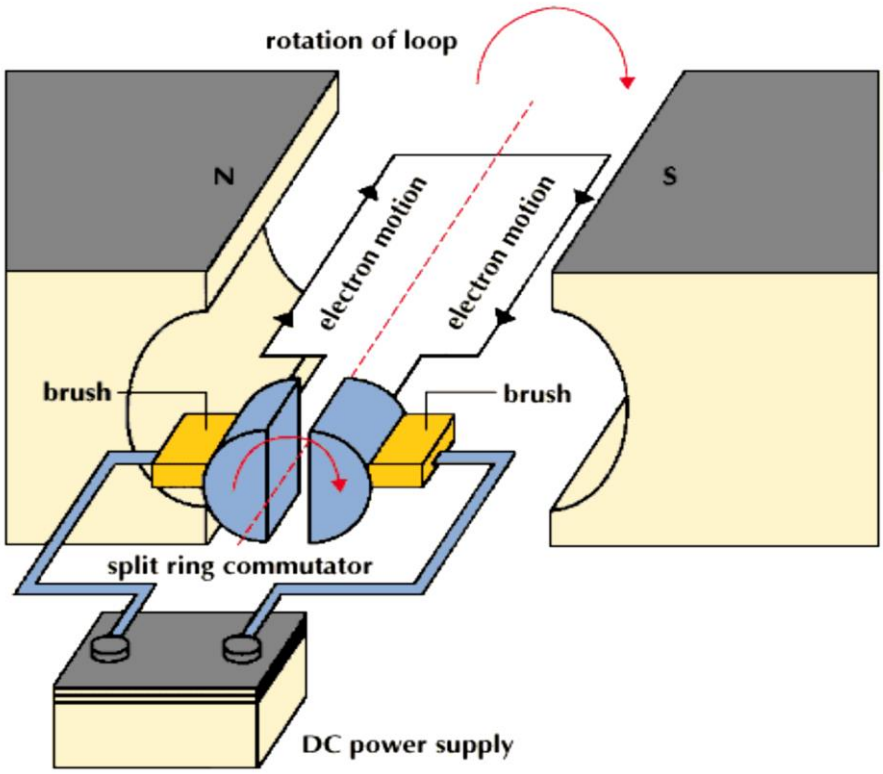
There are types of motors. They can be basically classified based on their power sources, internal current, and operation, such as AC, DC, and special types of motors. AC motors are further classified under Induction and Synchronous motors. Both AC and DC motors are widely applied in industry control, but this paper focuses on the second as it is utilized in the implementation. There are also special types of motors. One very famous and commonly used in motion control applications is the stepper motor.

2.6.1.1. DC motor

DC motors are further classified into two main types: brushed and BLDC:

- Brushed DC motors: due to its low cost, availability, and basic operation, and ease of driving, the brushed DC motor appears to be the classical and commonly used type of motor in control systems applications. Any brushed DC motor consists of four components: a rotor, stator, commutator, and brushes. The part that rotates as a result of exciting its magnetic poles by energizing the windings (coils) surrounding it is the rotor. It is usually made of one or more windings of *n* turns. The two stator ends produce constant magnetic field, generated by either permanent magnets or electromagnetic windings, surrounding the rotor. The rotor moves in a way that its magnetic poles attract the opposite ones of the stator. The brushes slide on different segments of the commutator, a copper cover, which is attached to the two axes of the rotor such that a dynamic magnetic field is generated when applying voltage input.

Fig 2.54 shows (a) a diagram of the DC brushed motor and (b) an inside cross section view.



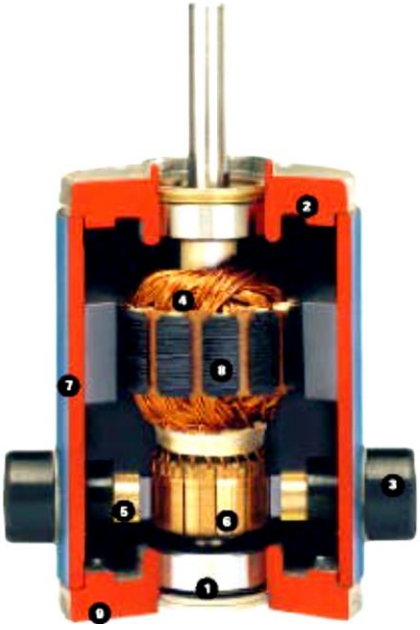


Fig 2.54: (a) The DC brushed motor components and (b) a cross sectional view of a PMDC

(taken from [75])

Looking at Fig 2.55 below and as stated several times, DC motors are electromechanical type of a dynamic system, so they can be modeled using both KVL as well as D'Alberts law introduced earlier. Hence, the following two state variable equations can be obtained [18], [50]:

$$J\ddot{\theta} + b\dot{\theta} = Ki \tag{75}$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \tag{76}$$

Recalling that $\dot{\theta} = \omega$ and $\dot{\omega} = \ddot{\theta}$ and b is the viscous damping constant of the motor, under the assumption that the torque constant equals the EMF's of the motor, $K_\tau = K_e = K$, and given that for DC motors, torque is proportional to the amount of current flowing into the motor [18],

$$T = K_t i \tag{77}$$

Also the speed is directly related to EMF voltage such that:

$$e = K_e\dot{\theta} \tag{78}$$

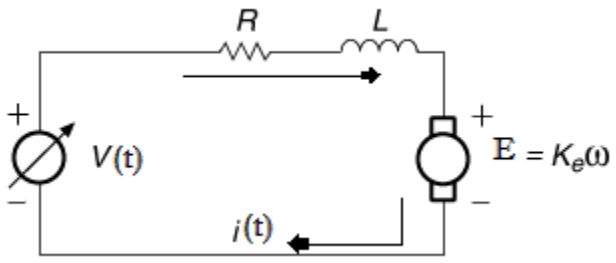


Fig 2.55: A DC motor model comprises of an electric and a mechanical part

The DC motor model can be viewed as a SISO LCTI system where the control voltage signal $V(t)$ input and the rational speed of the rotor, ω, that represents the output are related. Then applying the Laplace transform to (75) and (76) and then solving for $\frac{\omega(t)}{V(t)}$, yields the TF,

$$\frac{\omega(s)}{V(s)} = \frac{K}{JLs^2+(bL+JR)s+(bR+K_e{}^2)} \tag{79}$$

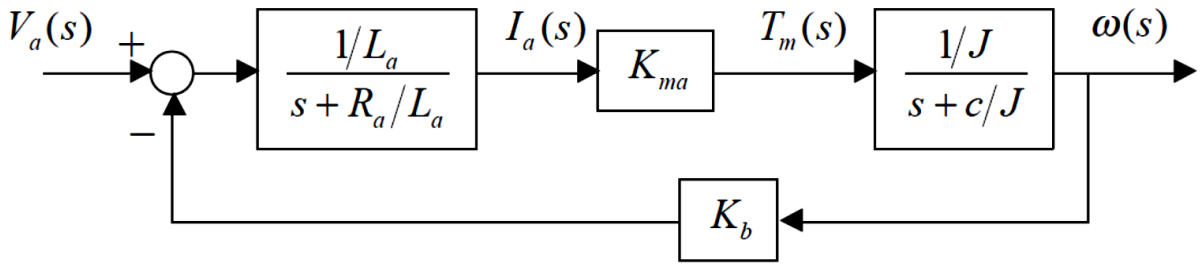


Fig 2.56: A DC motor can be viewed as a closed loop system where the output speed is proportional to input supplied (control) voltage (taken from [18])

More specifically, further analyzing the system model of the DC motor (Fig 2.65) one could notice the TF between $V(t)$ and $\omega(t)$ includes internal I/O relations. First, the driving voltage input produces a first order RL relation in Laplace domain, $\frac{I(s)}{V(s)}$. Then, the current output is proportional to torque by a factor of *K* as seen above. This torque is related to the output rotational speed of the motor by another first order TF relation, $\frac{\omega(s)}{T(s)}$. After that, the speed is proportional to the voltage by a factor of K_e by the production of back EMF according to the principle of electromagnetic induction such that,

$$I = \frac{V - EMF_{back}}{R} \tag{80}$$

So that the following relation is true. Fig 2.58 shows the torque, speed, current, and power curves.

$$EMF_{back} \propto \omega \xrightarrow{yields} \uparrow V \uparrow \omega$$

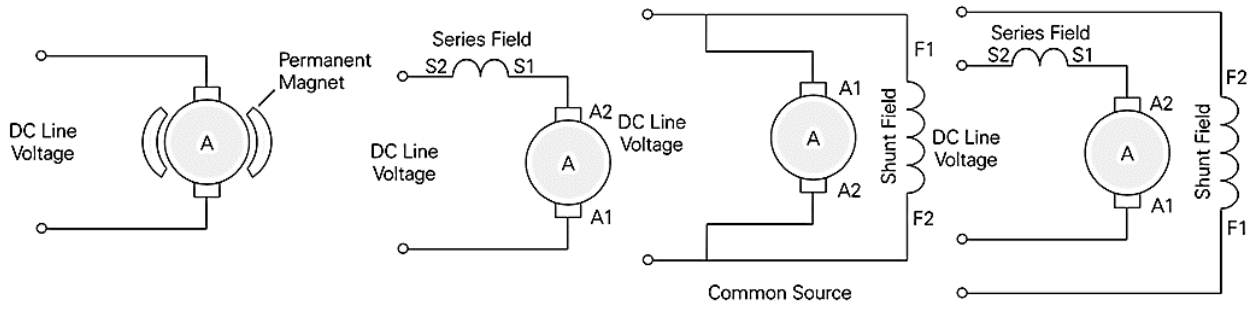


Fig 2.57: Four different mechanical configuration types of the brushed DC motor

With regard to the way of generating magnetic field, four types of DC motors arise: PMDC, shunt-wound, series-wound, and compound-wound brushed DC motor, which is a combination of both the shunt and series wound. Fig 2.57 shows the difference in configuration between the four types of DC brushed motors [75]. Every type has its feature, advantages, and limitations [55].

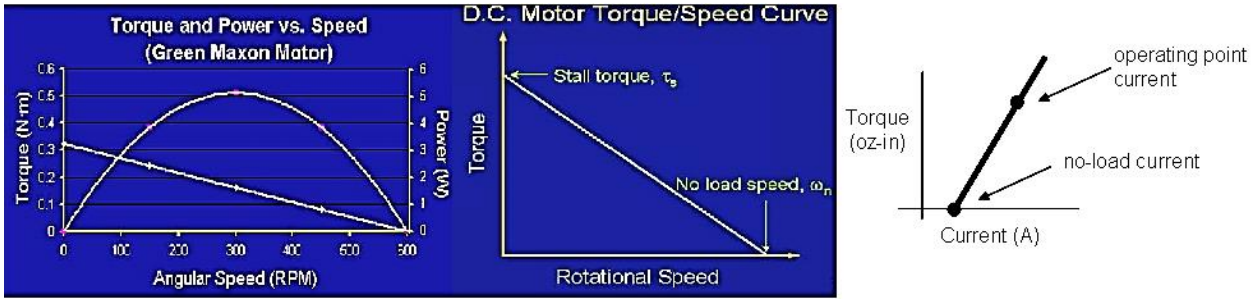


Fig 2.58: Performance curves of a typical DC motor

A special and common configuration of the DC brushed motor is the servo application where the position feedback is measured externally to the control. This is required in positioning applications because as can be seen, the output of DC motors is originally speed, so in order to be at position an internal feedback is required, which is known as “servoing.” Incidentally, that being said, some robotics types of actuators are called “servo motors,” which are nothing but internally and mechanically servoed DC motors used for small motion applications. However, for larger industrial loads, AC, DC or BLDC motors are usually installed and configured as servo.

Another type of DC motors is the BLDC. It differs from the brushed on the internal mechanical construction besides the commentator method (brushless). In other words, the mechanical commentator used in switching current directions and control its flow through the DC brushed motor is placed with electronic ones here. This gives BLDC more controllability and reliability in a countless number of applications, especially servo ones, where feedback to the control system should be present. Thus, most servo applications currently have moved in favor of BLDC [75]. However, those advantages predetermine higher cost than the brushed DC.

1.5.6.2. Stepper motor

A special and common type of actuator used in control applications in general and positioning, motion, and robotics in specific is the stepper motor. Similar to all motors, a stepper requires electrical current as input and produces rotation. However, the key difference here is that the stepper motor requires pulse inputs in order to be driven and produce discrete movement (steps), unlike the DC motor for example, which works in a continuous fashion. This fact gives stepper motor more accuracy and controllability, but less continuity. Yet, with the advancement in technology as the micro-stepping drivers, it is possible to have stepper motors today that perform close to continuous as possible with high range of resolution [75].

There exist three main available types of stepper motors:

- Permanent-magnet: this kind is like the PMDC as rotation is created by the forces between the permanent magnet and an electromagnet created by electrical current.
- Variable-reluctance**:** the variable-reluctance produces rotation with electromagnetic forces and does not include a permanent-magnet.
- Hybrid stepper motors: this kind includes both configuration modes.

As appear in Fig 2.59 below, a stepper motor is made of two phases or more, creating the discrete steps movements. The internal mechanical construction of a stepper motor consists of a rotor, stator, and coil windings. The rotor is a central shaft that spins during use while the stator is a surrounding stationary part. The characteristics of these components and how they are arranged determines whether the stepper motor is a PM, VR, or hybrid as mentioned above. Fig 2.59 shows (a) a schematic of a two phase stepper motor and (b) its electric equivalent model [76]. The basic idea of operation is simple. As the windings are activated, as a result of a pulse sequence, the stator will be magnetized and form electromagnetic poles that cause repulsion to the rotor.

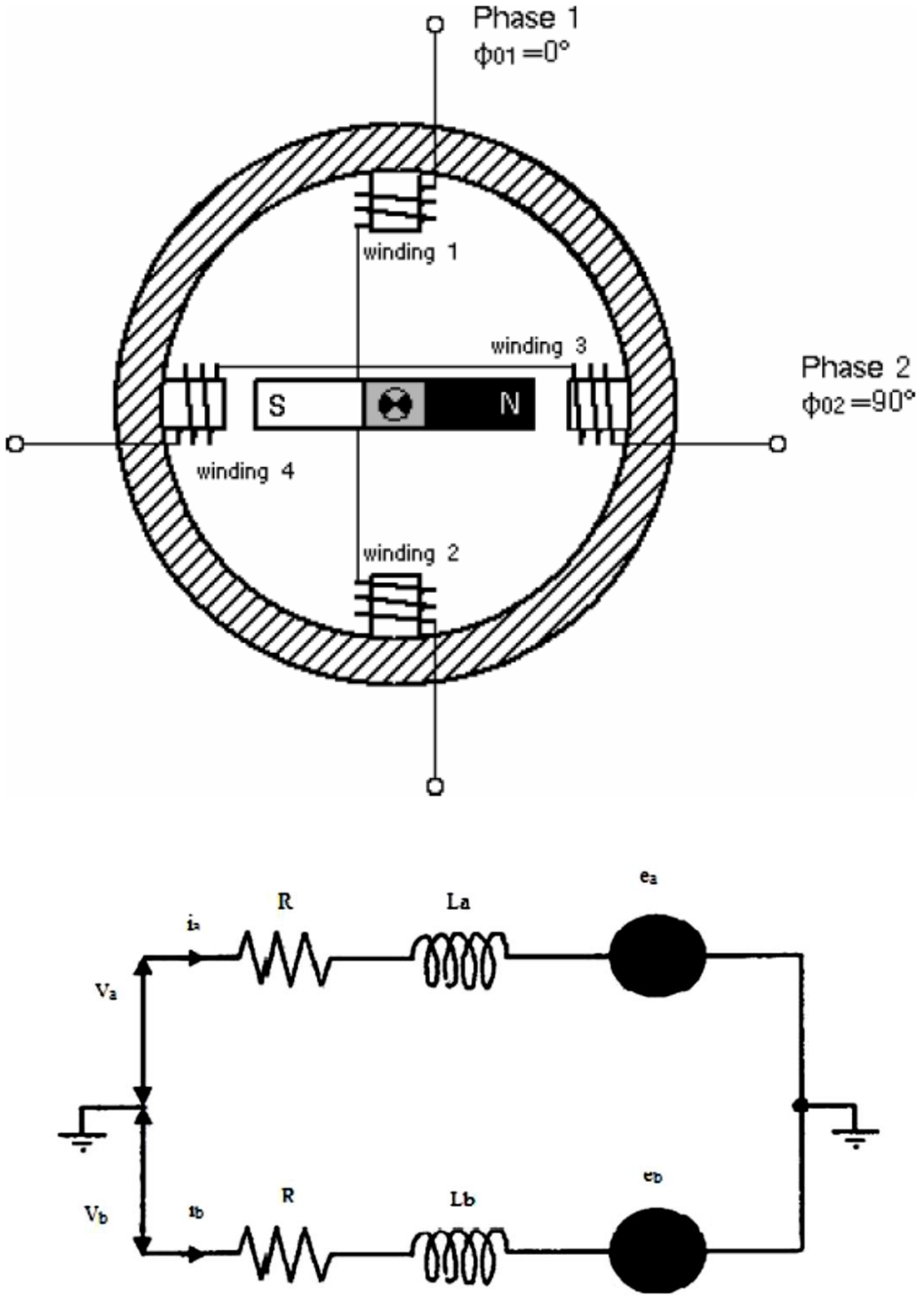


Fig 2.59: (a) A schematic of a two-phase stepper motor [78] and (b) its electric circuit model (taken from [77])

The model of the stepper motor might be presented in more detail later. Looking at the basic Fig, 2.59 (a), a single pulse input moves the rotor 90 degrees. This driving technique is the basic and is named stepping. Hence, the number of steps per revulsion is [78] as per *n* rotor pole pairs and *m* number of phases of the stator

$$S = 2.n.m \tag{81}$$

Thus, the step movement angle, $\Delta\theta$, follows

$$\Delta\theta = \frac{360}{S} \tag{82}$$

Substituting (81) in (82),

$$\Delta\theta = \frac{360}{2nm} \tag{83}$$

For example, for the common two phase motor, a step angle is 90 degrees as mentioned before. Hence, to increase the resolution, several solutions arise such as increasing the number of phases and/or poles or modifying the basic driving technique to generate something called half, quarter, and micro stepping. A brief description about how to accomplish those stepping techniques is discussed in the next section, motor driving fundamentals.

The generation of pulses should be done in a manner to count both directions of any pattern of voltage sequences in the two phases in order to determine the commanded and the current positions of the rotor, θ, such that:

$$\theta_{k+1} = \theta_k \pm \Delta\theta \tag{84}$$

Where *k* is the current count. Therefore, the angular position of the motor is determined from the following:

$$\theta_k = (x - y)\Delta\theta \tag{85}$$

Where *x* and *y* are the number of steps in positive and negative directions respectively. More details about the stepper motor, its dynamics and implementation, will be discussed in the future.

2.6.2. Motor Driving Fundamentals

As perceived, motors do not move at the desired position, speed, and direction unless electrically is driven. To drive different types of motors and their movement, there appear different techniques and fundamentals. Likewise, control signal outputs from controllers remain not able to drive and power up electromechanical devices as motors directly. There should be translators in between by which those signals are interpreted, amplified, and passed to motors in manners that regulate their speed and direction. Those translators are known as drivers. They have two main purposes. First, amplifying control signals and providing compatible power requirements to motors. Second, switching current flowing into motors in ways that determine their states. This subsection mentions essential basic concepts with regards to mechatronic actuation techniques, especially DC and stepper motors: PWM, H-bridge circuits, and other drivers.

2.6.2.1. PWM

In general, PWM is one of those modulation techniques used to encode messages as pulse signals. Therefore, PWM can be used in digital communication and data transmission. However, it is found to be widely used in control applications as regulating power supplied to electrical/electrometrical devices, i.e. motors. PWM is also found applicable in power electronic applications, for example photovoltaic solar battery charging, efficient voltage regulating to maintain maximum power solar cells and panels (MPPT), DC-DC conversion, and others. PWM appears to be fundamental in most motor drivers today as its main function is to control the amount of voltage signals by switching them to different discrete pulse levels as shown in Fig 2.60. These levels are determined by a concept called duty cycles [51], [79] such that the:

$$V_{ave} = D \times V_H = \frac{T_H}{T_H - T_L} V_H = D \cdot V_H + (1 - D) V_L \quad (86)$$

Where T_H and T_L are the time periods of the high and zero or low pulse respectively. Also, V_H and

V_L are the amplitudes of those high and low pulses. Finally, D is the duty cycle.

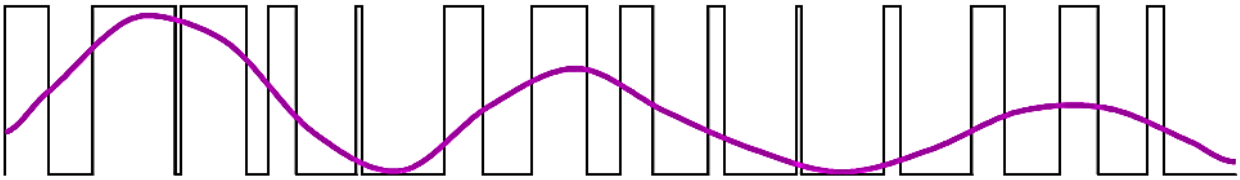


Fig 2.60: PWMs modulate analog signal levels

The duty cycle is then a percentage that factors the modulated analog signal's level from 0 – 100% range. For example, a duty cycle of 100% means the maximum range of the signal is transmitted while 0% for instance interpolates no signal transmission and so on. Fig 2.61 below illustrates various duty cycles of PWM signals. This technique is widely used in both DC as well as stepper motors driving applications. In the first, analog control signals would be sufficient; however, PWM offers more customization and accuracy in regulation where the output approximates desired voltage levels, and the switching noise is smoothed by an RCL filter. For the stepper motor, it is known that it requires pulse inputs only. Those pulses might be constant or regulating as the PWM. It is then mainly used in linking regulatory control algorithms, i.e. the PID to the discrete stepping applications.

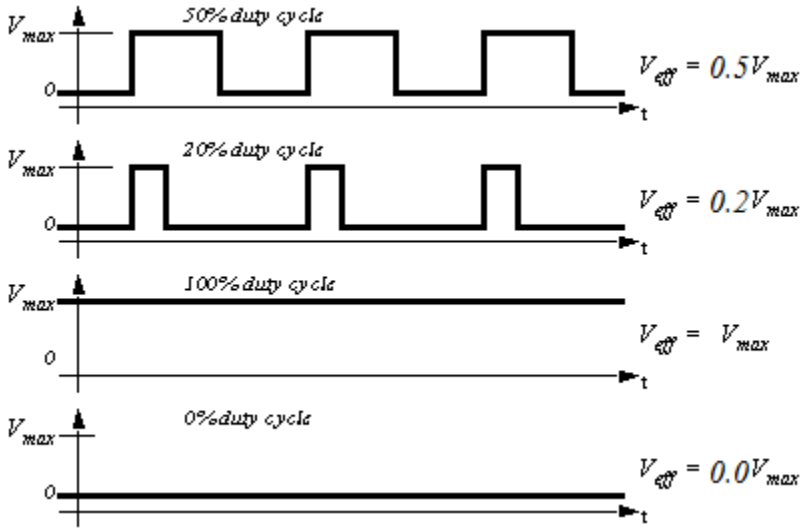


Fig 2.61: Duty cycles and transmitted signals leveling

2.6.2.2. H-bridge and motor drivers' techniques

One of the most common systems used for DC as well as stepper motors driving is the H-Bridge. In general, it is an electronic circuit that has four switches made of BJT or MOSFET technology to enable a voltage to be applied across so that current flows in either direction, forward or reverse. The H bridge circuit is frequently used in motion control and mechatronic applications mainly to drive DC or stepper motors. Also, the H-bridge is found applicable for power electronics applications such as in DC-to-AC, AC/AC, and DC-to-DC converters. A simplified schematic of this circuit is given in Fig 2.62 below.

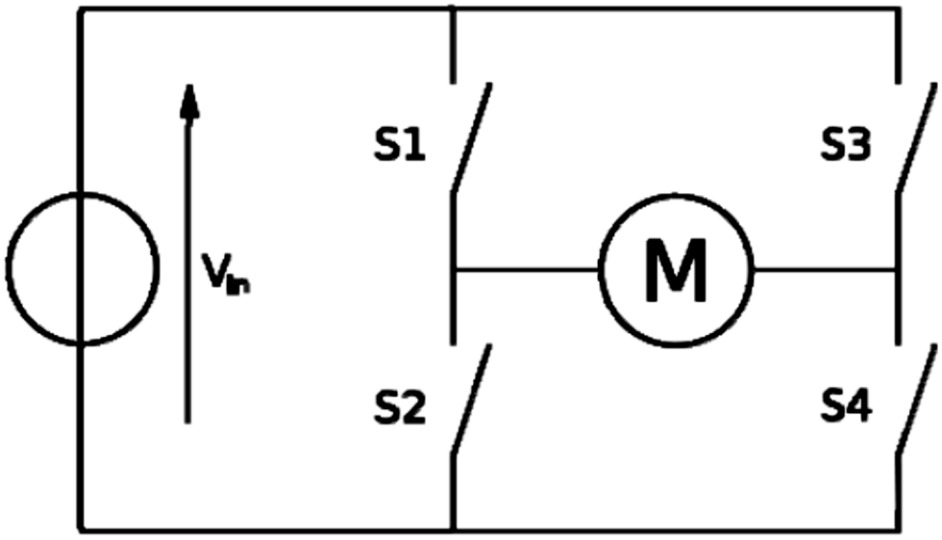


Fig 2.62: The simple schematic of the H-Bridge circuit (taken from [51])

There are various possible techniques and configurations of driving a DC motor in particular using the H-Bridge [80]. For an example of such configurations are a unipolar two-quadrant drive, unipolar 4-Quadrant PWMs, bipolar PWMs, and three phase BLDC drives. Usually, two main signals are to be sent from the controller: the PWM and direction. The traveling state of the motor depends on which single, pair, or group of switches are enabled at an instant. It is then how the two main signals coming from the controller are wired or connected to turn ON or OFF those switches.

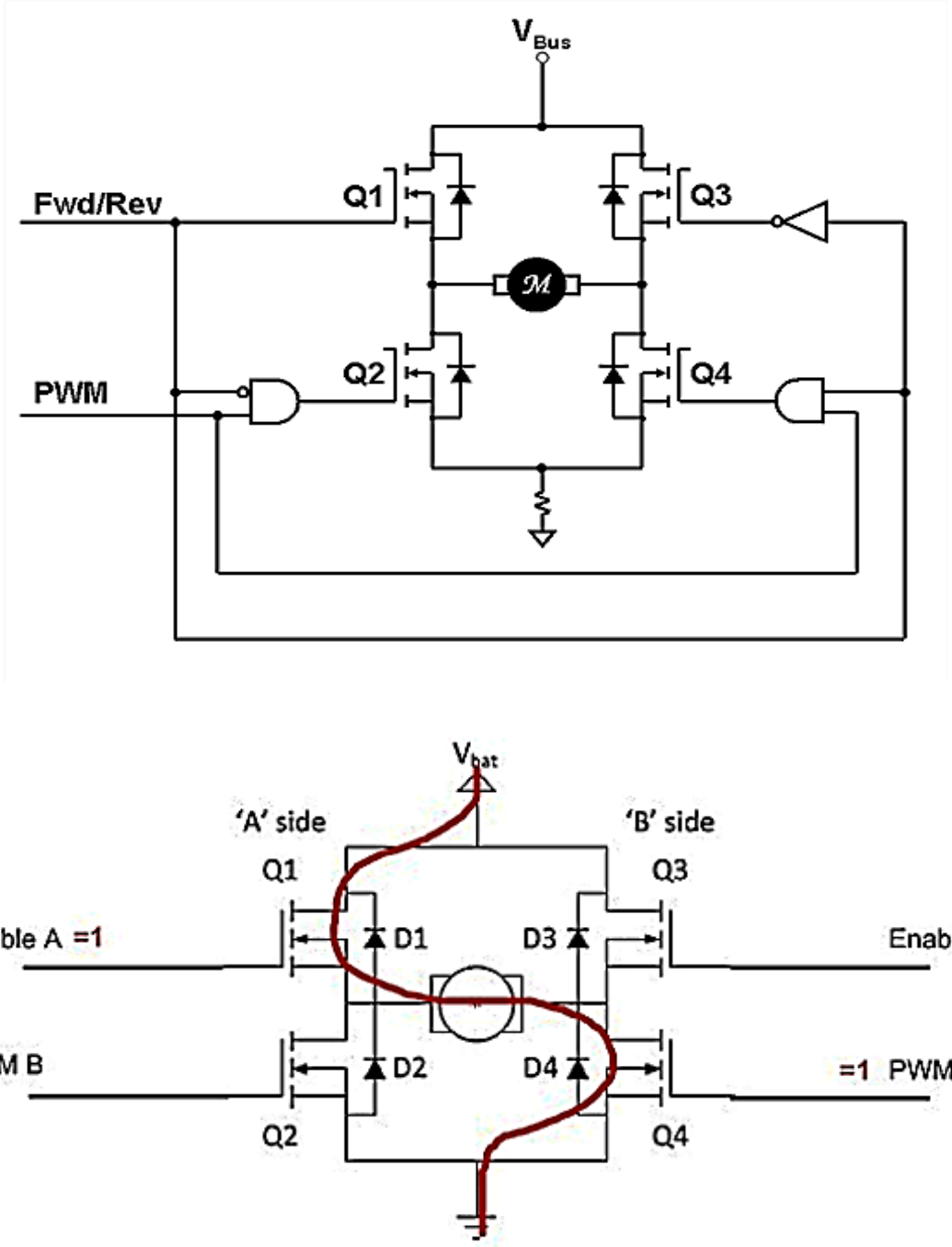


Fig 2.63: (a) A common H-bridge driving technique and (b) one possible example

Fig 2.63 (a) shows a common way of implementing H-bridges in driving DC motors and (b) one possible current flow scenario in which the rotor moves forward. Table 2.4 lists possible scenarios of switching the H bridge circuit and their outcomes on the motor state.

S1	S2	S3	S4	Result
1	0	0	1	**Motor moves right**
0	1	1	0	**Motor moves left**
0	0	0	0	**Motor free runs**
0	1	0	1	**Motor brakes**
1	0	1	0	**Motor brakes**

Table 2.4: Five possible H bridge switching scenarios of driving a DC motor (taken from [51])

Regarding driving a stepper motor, there appear two drive types: unipolar and bipolar drivers. A unipolar driver suit stepper motors that have two windings per phase or one with center tap, one for each direction of magnetic field. On the other hand, bipolar stepper drivers are for motors that carry single winding per phase. The unipolar stepper motor provides only one path of current and power source while the bipolar allows two directions but is more complicated as it needs two H-bridges per phase as shown in Fig 2.64.

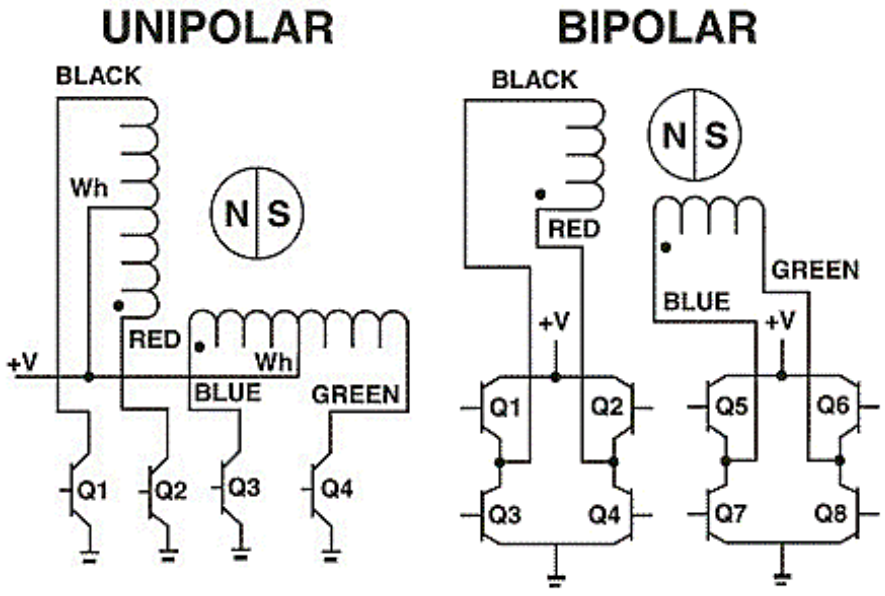


Fig 2.64: Examples of unipolar and bipolar stepper motors and drivers

Now, it actually depends on how pulses inputs are arranged to drive stepper motors. Three main stepper motor driving types are full, half, and quarter (micro-stepping). The full stepping has been introduced as the basic operation of stepper motors. In full stepping, the flowing current in each winding is either maximum positive or negative pulse amplitudes. This is done by activating each phase a at time. In half stepping, however, the current required in each winding is either maximum positive pulse, negative amplitude, or zero that will result in a total of eight steps to complete a full revolution. The half stepping technique is implemented as appears in Fig 2.65 by turning a single phase followed by a Boolean combination of the two of them. The more states added to the current, the more steps and so on until the micro stepping is achieved. In addition to the highest resolution, the micro stepping provides the maximum peak torque out of the limited motor power and size. However, micro stepping, and stepping actuation in general, associates sources of discrete errors such as quantization, rotor poles placement, shape or geometry of poles, and backlash errors. [81] provides detailed information about types of stepper motor drivers and techniques and their internal circuitries and implementation.

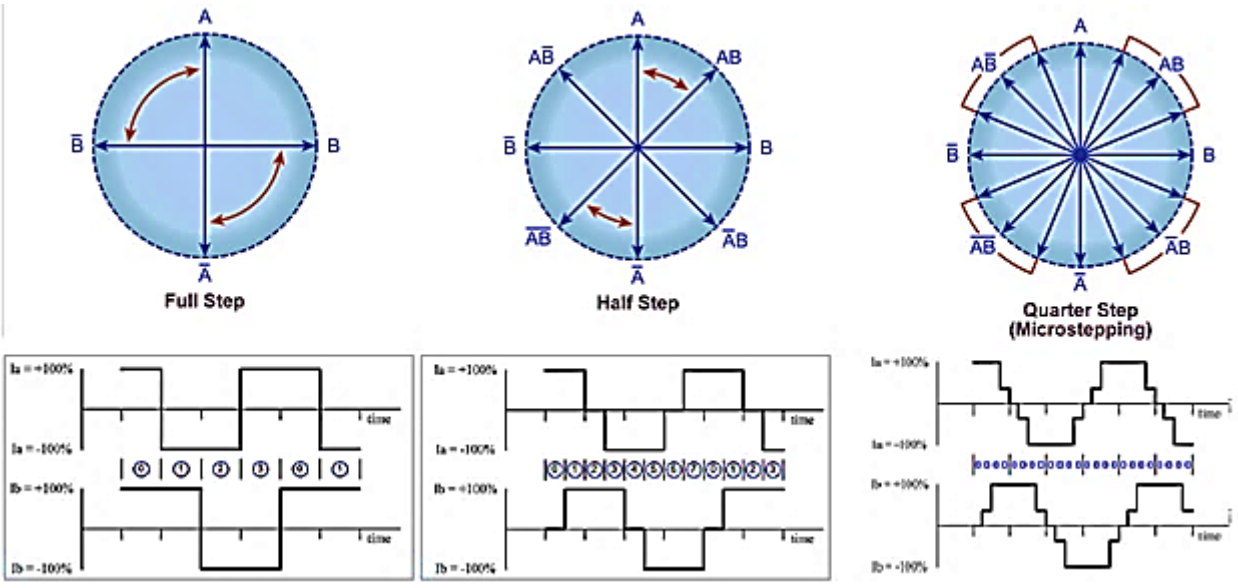


Fig 2.65: Full, half, and micro stepping

2.6.3. Other Instrumentation

This section has mentioned so far the two actuation and final control element modes that are implemented in this work: the DC and stepper motors and drivers. However, there exist other common actuators used in process control that need to be mentioned since this report focuses on industrial automation applications. Some of or all of the below listed actuators types might be implemented and included in the future in the distributed architecture of this work also. They are CVs, heating elements, pumps, and fans.

2.6.3.1. Control valves

CVs represent the most common actuator of all found in industrial process control. Thus, for process lines and other process models as surge tanks, containers, boilers, mixers, distillation columns, etc. [70], that carry fluids or gas, to control common RT variables as flow rates, level, pressure, density, concentration, and even temperate all obligate proportional or discrete CVs as final control elements. This automatic proportional or ON/OFF operation of opening and closing of CVs is either implemented electrically, pneumatically, hydraulically, or mixed. The basic operations of CVs in general depend on principle of volume flow rate, Q, per time (in m^3/s) given in (38) before:

$$Q = Av$$

CVs can be classified into two main categories: proportional (continuous) or discrete (ON/OFF). It seems logical then that the proportional CVs are used for regulatory control applications while the ON/OFF actuates discrete event based processes and subsystems. Communicating different types of signals to/from the control system, i.e. DCS, is covered in Chapter 1. Most current industrial proportional CVs that work on the conventional or HART wiring filed communication systems follow the 3-15psi/4-20mA pressure range standard.

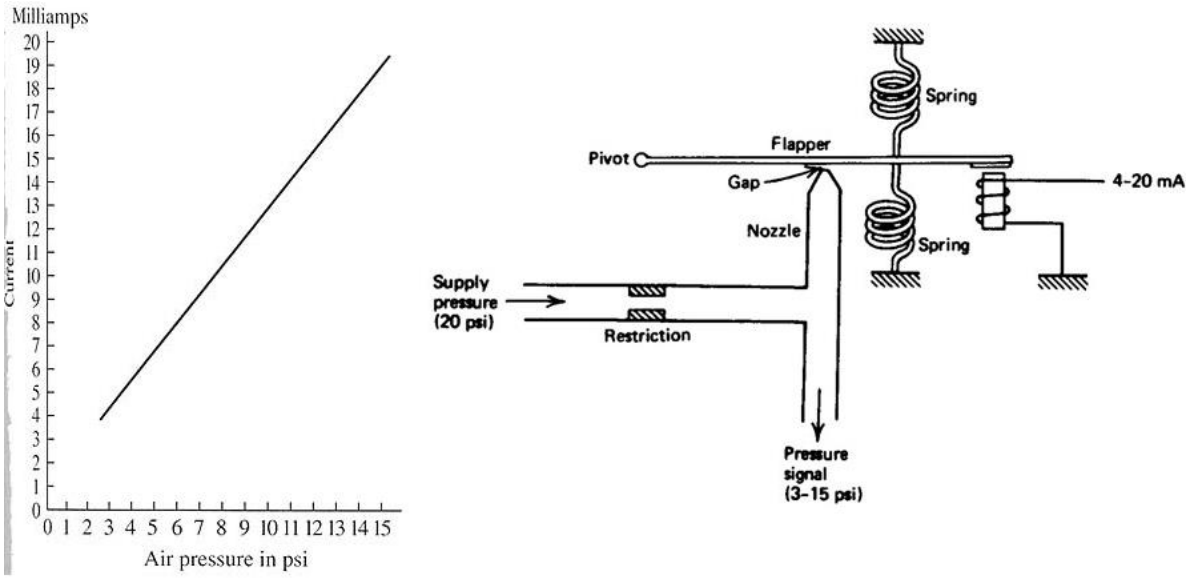


Fig 2.66: I/P convertors used as mode of signal conversion in actuation instrumentation

More specifically, I/P comes in between the AO control signal of the controller and the pneumatic CV. Fig 2.66 above illustrates a simple drawing of the I/P convertor block that is used to convert electrical signals into pneumatic and how the 3 – 15psi pressure output is linearly proportional to current (4 -20mA). A common approach to implement such signal convertor is the nozzle/flapper method as appears. The current flowing through the coil creates a force that pulls the flapper down proportionally, which produces differential pressure in the nozzle. The higher the current supplied, the higher the pressure produced. Adjusting the spring and the position pivot calibrates the device so that perhaps 4-20mA is equivalent to 3 – 15psi [55].

Some other CVs are motor based, so they only require electrical signals and driver amplifiers as discussed. Therefore, aside from being pneumatic, CVs can be electrical, for example solenoids, or hydraulic based [55]. Some other HVAC CVs utilize the conventional analog 0-10V standard while most ON/OFF ones take 0-5V or 1-5V. However, numerous CVs today comply with smart full digital fieldbus devices and protocols, i.e. F.F, Profibus, etc. Thus, CVs in general consist of three main components: actuator, positioner, and body (Fig 2.67 (a)).

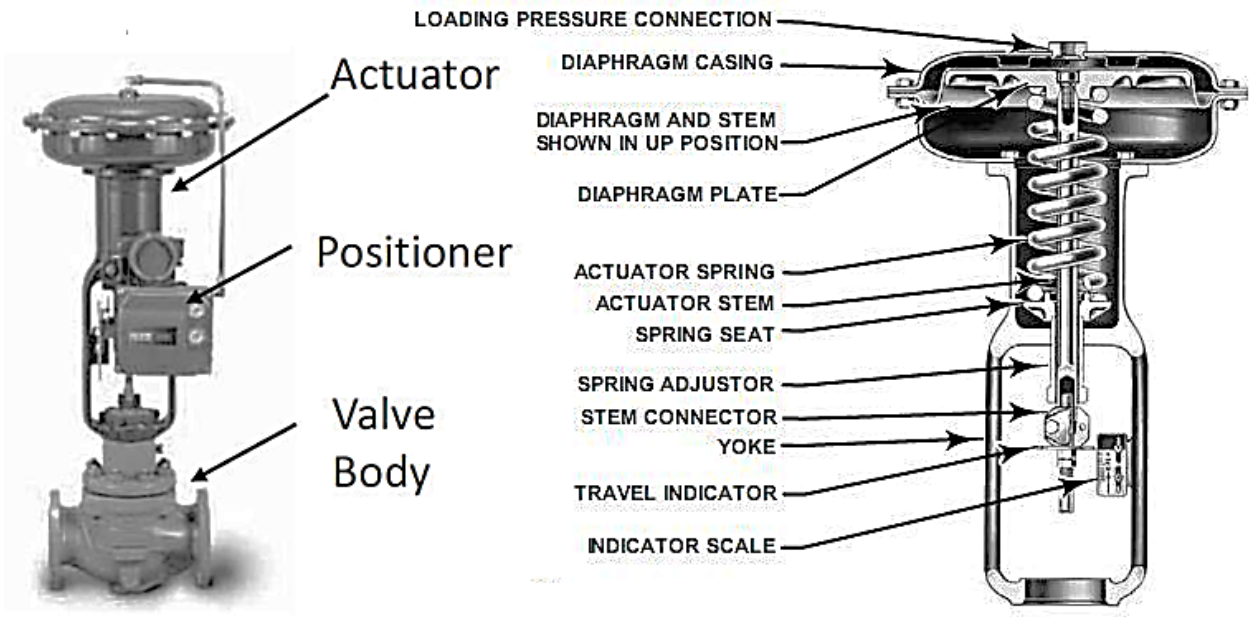


Fig 2.67: (a) The three main parts of CVs and (b) a cross sectional view of a pneumatic CV

The actuator is the part that moves the valve body either as opening or closing. The CV actuator could be electrical or mechanical. Most commonly, a mechanical spring plays the role of a CV actuator as appears in Fig 2.67 (b) above. Pneumatic actuators rely on the basic concept:

$$F = \Delta PA \tag{87}$$

This equation restates the perception that a net pressure difference, ΔP, applied on a diaphragm surface of area A equals the moving force from actuator. The positioners function as measuring instrument that locally displays the position feedback of the valve [55], [70] such that:

$$\Delta x = \frac{A}{k}\Delta P \tag{88}$$

Where Δx, the shaft travel (m), relevant to 0 -100% linear position and, k, the spring constant. Finally, the body is the part that carries flow through and has a valve plug that opens and closes. Some CVs are direct acting while others are considered reverse. In other words, with regard to safety, direct acting CVs are called “fail close” while reverse acting ones are named “fail open” [70]. This naming criteria indicates how the valve should act in case of shut down or power failure.

There exist various available valve bodies as per the application, for example reverse double-ported globe, three-way with balanced plug, flanged angle, valve bodies with cage-style trim, and Soft Seat [55]. As for any sort of actuator, an applied control system designer should select and design the proper control element of his application, and so for CVs. This topic is partially neglected in some control system texts which perhaps assume the correct actuator dynamic tend to be present. In some others that focus on the instrumentation side of the control loop, as [55] and [70], the significant design calculation required is CV sizing such that:

$$Q = C_v\sqrt{\frac{\Delta P}{S_G}} \tag{89}$$

Where Q stands for the flow rate, C_v, the flow coefficient relative to the valve size in inches, $S_{G,}$ is the specific gravity of the liquid, and finally, ΔP stands for the pressure across the valve (3 - 15psi). In other words, in order to control a particular flow range really depends on CV size and the specific gravity of the liquid flowing.

2.6.3.2. Heating elements

Another form of actuation that is used for temperature control and heating applications are heating elements. They convert electrical signals into heat according to the principle of Joule resistive heating. The basic idea here sounds simple; when the current passes through the element, it forms a kind of electrical resistance, which results in heat change [55]. There are metal, ceramic, composite, and combination heating elements.

2.6.3.3. Pumps

Pumps are another common industrial as well as commercial final control element applications. A pump is a device that mechanically transports liquids or gases. There are two major types of pumps: centrifugal and reciprocating pumps. The first operates according to the differential pressure principle while in the second, capacity increases with speed.

2.6.3.4. Fans

Fans are usually motor based actuators that are used for HVAC and other applications as temperature, air pressure, and humidity control. Being motor based, fans are classified under AC, DC, brushed, or brushless. Also, they require the same driving techniques used by motors such as the H-bridge and others. A frequently manipulated variable targeted in order to control other related ultimate RT PV i.e. temperate, pressure, and density, is fan speed.

2.7. Summary and Discussion

This chapter discussed basic theoretical as well as associated practical concepts and definitions to be referred to in this research. A brief summary is given in the next few lines followed by discussions and conclusions.

2.7.1. Summary

To summarize this chapter, a few points of interests can be marked vital to remember:

- Recall the integrated automation pyramid architecture (Fig 2.68 (a)) in which the process, field instruments (sensors and actuator) and communications, CN and interlocks, SCADA, production planning and the plant IT represent 0, 1, 2, 3, 4, and higher levels respectively. Similarly, control strategy functionality should pass though similar hierarchical manner in such large scale projects: single loop indications, single variable regulatory control, interlocks logics, multi-variable control, sequences, then finally optimization as appear in Fig 2.68 (b).
- As automation and systems engineering, the scope begins with level one of the design; however, understanding the dynamics and mechatronic parts of the controlled process is the first step in scheming successful control laws and strategies.

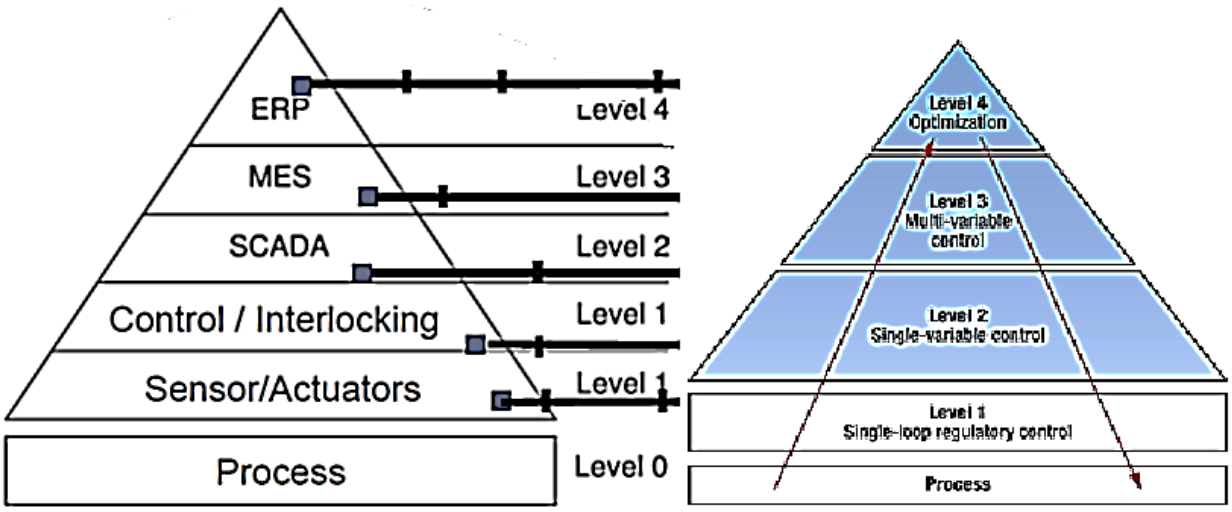
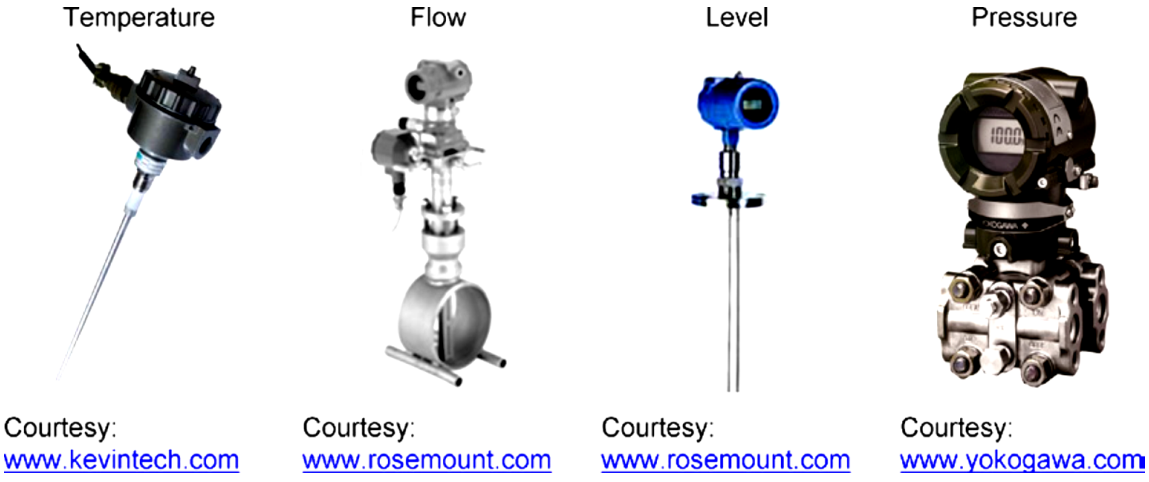


Fig 2.68: (a) Industrial Automation architecture and (b) its corresponding control functionalities

- Understanding the dynamics of the process through modeling and simulation then obtaining control parameters based on theoretical analysis and assumptions might not be sufficient in producing operational RT automation systems. Rather, such implementation should begin with designing field instruments. They would not only be considered as sensors and actuators but as complete measurement and actuation systems where parts similar to sensing and, driving, final elements, signal conditioning and processing, and finally DAQ should be studied and selected properly. The chapter listed main positioning measurement techniques to be used in this research. Besides, common industrial field transmitters and their underlying concepts were briefly described. Fig 2.69 (a) shows several smart industrial measurement systems and transmitters and (b) an example of their internal BD.
- Besides control instrumentation, the chapter pointed out the importance of digital computation controller units. In other words, industry distinguishes between various technologies for different applications. The most common examples are the DCS and PLC. The first delivers smoother and more precise RT (regulatory) process control

functionalities while the latter provides rapid dynamic response that suits ON/OFF and event driven machines, yet certain accuracy levels will be sacrificed. This suggests further empathizing the topic of computer architecture and execution time in control applications on other emerging technologies such as DSP and FPGA. Please notice Fig 2.70 (a)-(c).



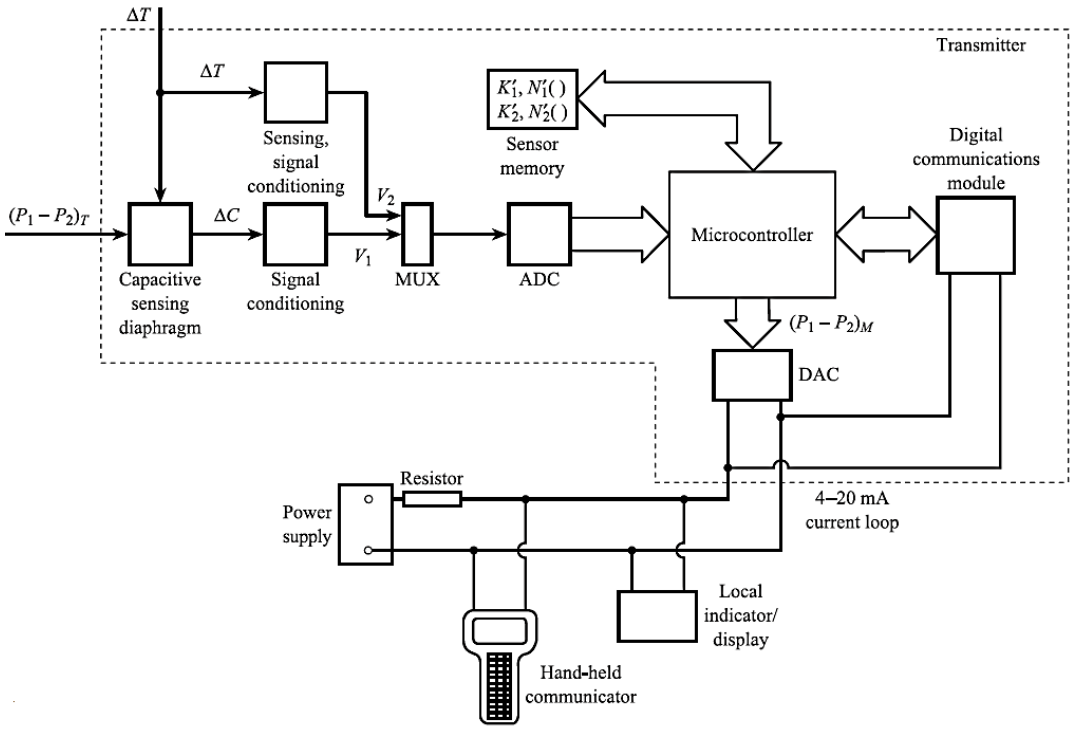


Fig 2.69 (a) Common industrial field transmitters (taken from [23]) and (b) and their internal BD (taken from [27])

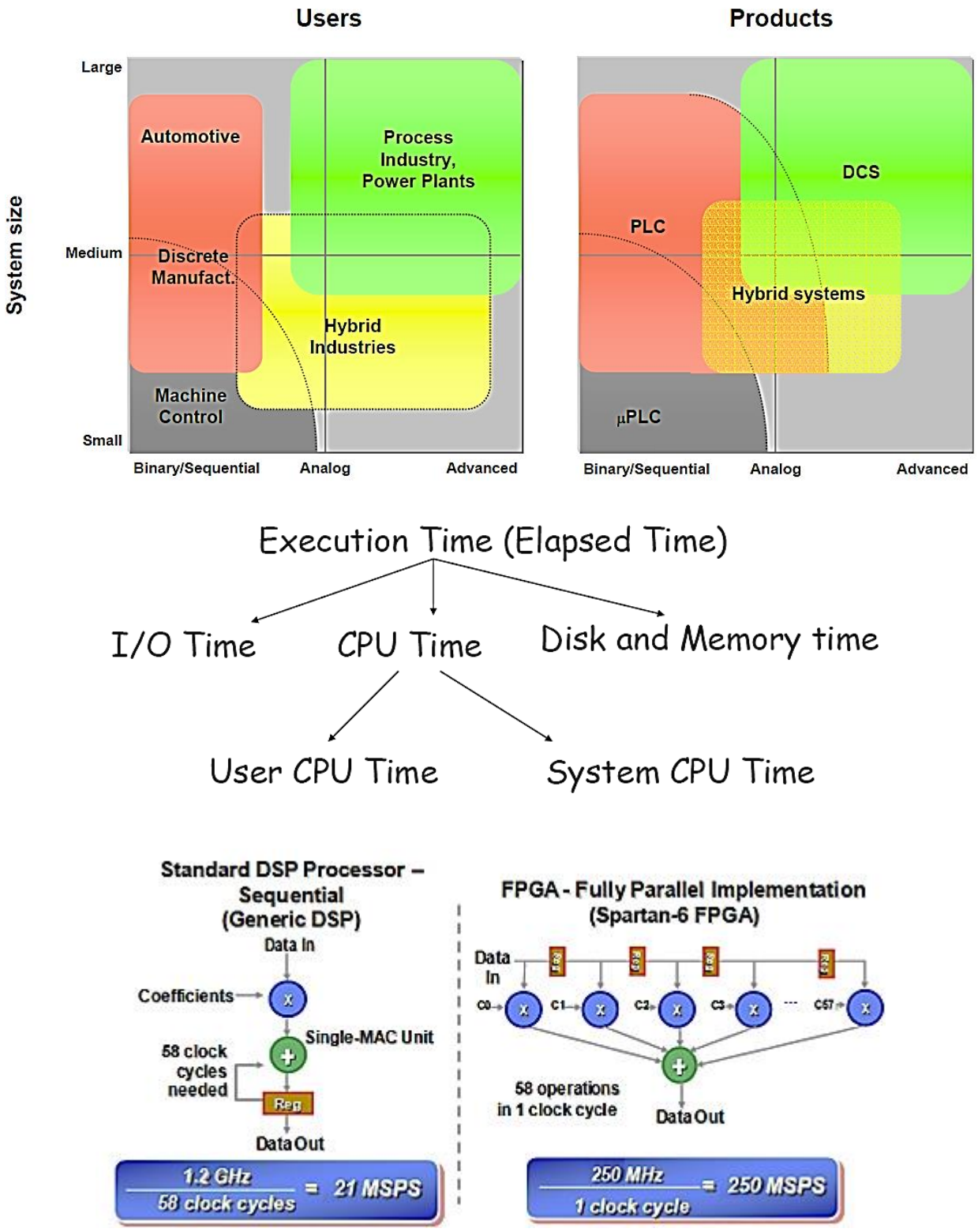


Fig 2.70 (a) DCS and PLC assignment [120], (b) execution time (taken from [57]), and (c) FPGA vs DSP (taken from [59])

- The chapter also called the other side of the design, in addition to instrumentation, control algorithm. Whether the design is model or non-model, the PID is the basic block in industry that can be either reduced to P, PD, or PI or expanded to implement cascade and feed-forward or even included under advanced algorithms. As an example, Fig 2.71 below illustrates a standard configurable loop controller block that is compatible for open, closed, auto, manual, and cascade modes and PID, NL, LPF, and multivariable sequential control functionalities.

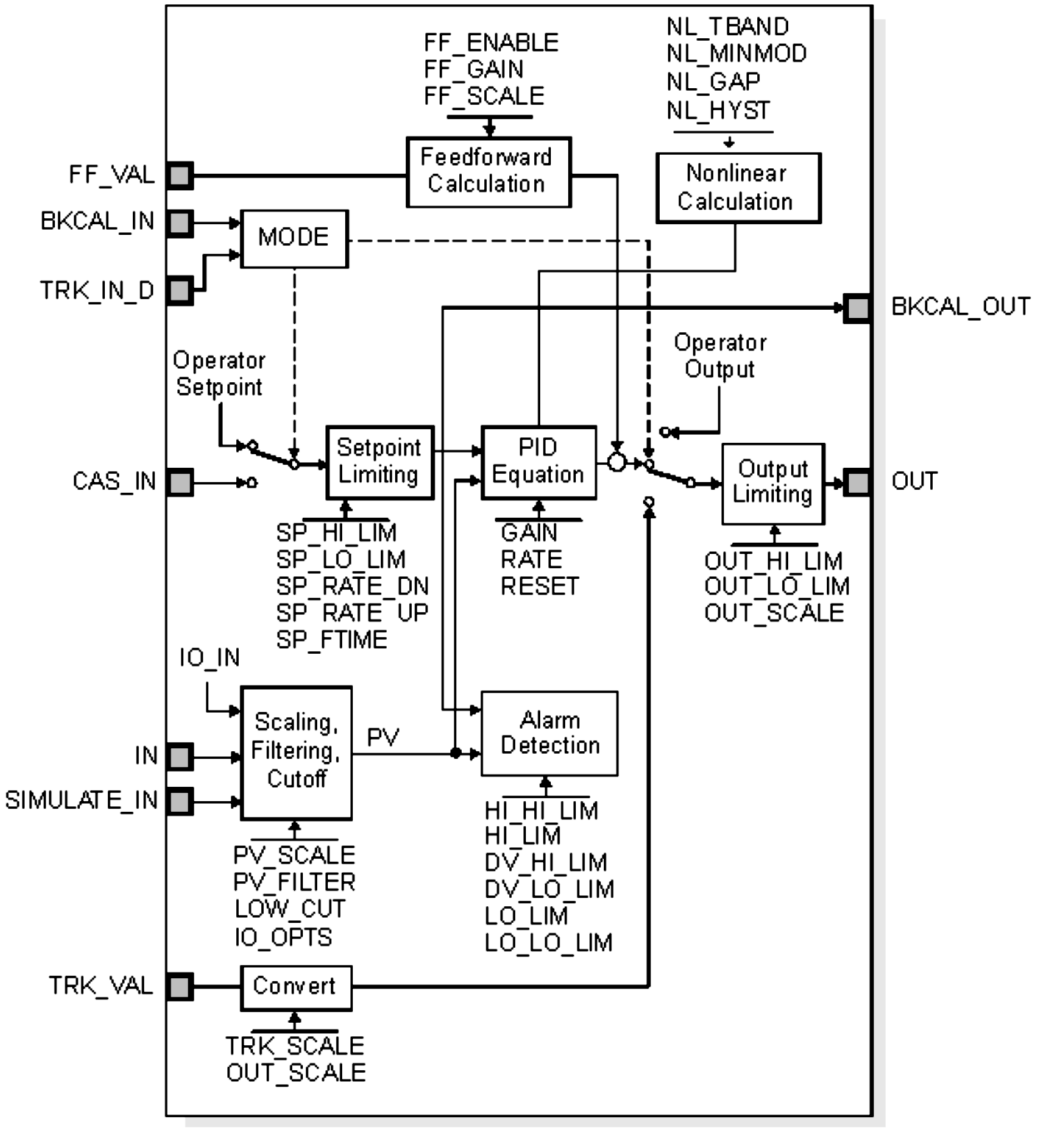


Fig 2.71: A standard industrial soft configurable controller block (taken from [70])

- The chapter overviewed the immense impact of final control elements on RT control with an emphasis on how to choose suitable actuators for RT applications. Selecting proper actuators requires sizing and technology awareness. For instance, two common motion control elements and their driving technique were introduced: DC and stepper motors. However, both along with AC motors can have special configuration, which results in other types. Fig 2.72 below simplifies various classifications of motors.

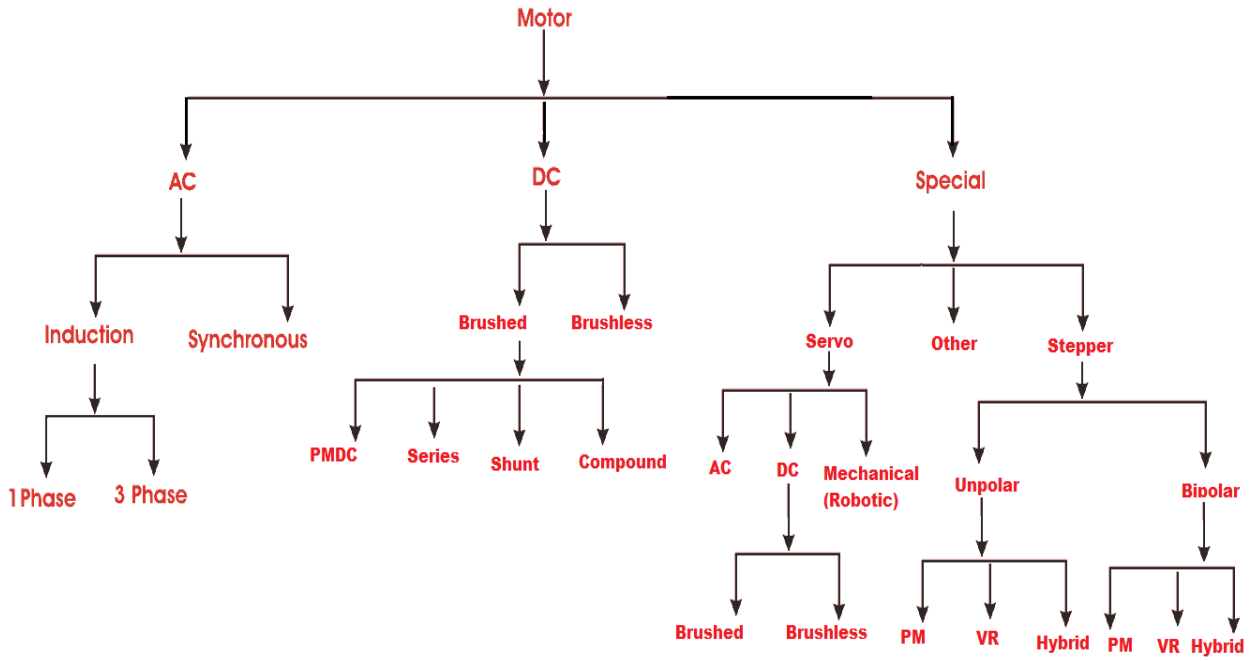


Fig 2.72: Types of electric motors as actuators in control applications

2.7.2. Conclusions and discussions

After going through this chapter, one could conclude the impact of implementation on control system's design. It is not only control algorithm that should be taken care of but other non-ideal RT facts such as communication, processing, and instrumentation technology besides also physical and mechanical setup of the controlled process plant. If any or some of those factors have shortcomings or chosen improperly, eventual RT control will be affected. This mindset composes the philosophy of this research through implementing the ball on beam case study.

CHAPTER 3

A CASE STUDY: THE BALL ON BEAM SYSTEM (THEORY-PRACTICAL ANALYSIS)

One of the most famous and classical plant models that has been used in academia by students as well as researchres to study control systems practically is the ball on beam plant. The requirement in this context is to control the position, and balance, of the ball on the beam using modern and advanced feedback theory and then optimize the behaviour as if the system is a part of a distributed industrial architecture. This continuous controller takes measurement feedback inputs reading of the distance of the ball on the beam, compares the error difference, and then decides or acts accordingly trying to track the desired position SP reference. The control signal, or as other literature refer to as controller or law output or MV, is after that used to actuate a motor that allows the beam to move counter/clockwise to adjust the ball near to the desired position value and track it. Although the ball on beam process or model plant is never found anywhere in industry, it has always been a good example of demonstrating, studying and solving real life's industrial control problems because it simulates numerous similar instances of issues faced in complex dynamics of production plants. In other words, the ball on beam arises as a matching candidate fit for this study due to several reasons such as:

- The plant process is unstable by nature which causes the demand of one of the basic purposes of applying feedback control, stabilizing loop's dynamics, and rejecting disturbance.
- The ball on beam is also an inherntly NL system which opens the door to either applying linearization techniques, to solve the problem using usual linear controllers, or using NL control techniques.

- It is of a higher order dynamic, so it provides a taste of how complex systems behave. In addition, given the fact its dynamic is a higher order allows studying and viewing the problem from both LTI and control systems principles: SISO and MIMO approaches.
- It is either not usually found for sale in markets, on the shelf, or considered to be exessively expensive, and so, in most cases, it needs to be built and implemented from scratch. This fact implies that building a ball on beam physical model and control system may involve different detailed design implementation approaches on several stages such as mechanical and mechatronics, measurement systems and sensing feedback, signal conditioning and calibration, control unit and algorithms, final control and acuation, and finally communications and networks.
- Control delay or dead time is very critical in this application, which increases its difficulty level and makes it challenging. Besides, this requirement of maintaining the ball fast enough on the beam given a dead time introduces whether the applied control technique is robust enough to be able to overcome such a problem smoothly, which eventually contributes to the intended performance evaluation study. Moreover, the need of minimizing delay time is as crucial in evaluating such loop's performance over different communication methods and protocols, especially WSN.
- Accurate steady state values are desirable in such an application, which leads to the demand of for minimizing the SSE by applying several optimal algorithms besides assessing other mechatronics and instrumentation parts of the loop such as feedback quality, actuators resolution and mechanical flexibility and limitations, for example motor coupling slacks, gear backlash, load and friction effects, and others.

- The plant is flexible in terms of ways of implementation as well as accepting disturbance. It then matches the anticipated study of the effect of verity of implementation and scenarios on the eventual control performance.

Selective previous literature that has contributed so far to the ball on beam control problem, along with discussions, appear in the first section below. A theoretical mathematical modeling and analysis of the model follows this literature review. After that, the developed study and simulation takes place. Finally, the chapter ends by demonstrating simulation results and conclusions.

3.1. Literature Review

Many researchers, engineers, and students have contributed to the problem of controlling the ball on beam. Some work has been published while several others are found as Master's thesis dissertations, senior engineering projects, assembled products, or even final term assignments. Below is a list of good selected literature along with their discussion and criticism described. After that, it is logical to mention the new research directions and contribution to the problem.

3.1.1. List of Main Selective Literature

The selection was based on the criteria that the ball on beam should have been implemented and built from scratch meaning papers that show only simulation and theoretical analysis were excluded because they do not match the desired objectives of the study. Also, there are many papers out there that used readymade laboratory ball on beam plant products of manufacturers such as Quanser, TecQuipment, Amira-Elwe, etc. Less concentration was placed on those papers because they also do not fit the exact sense of the study, especially regarding implementation of measurements and instrumentation, mechanical, mechatronics, and actuators selection and design. The clear list of useful and criticized literature appears as follows:

3.1.1.1. EDUMECH, Mechatronic Instructional Systems, ball on beam system

The first chosen implemented ball on beam was developed as a product of Shandor Motion Systems; it was authored by Robert Hirsch, Ph.D. in 1998 [82]. This system is quite old, but it is important to be referred to in this study as it uses the conventional analog controller unit. As appears in Fig 3.1 below, the controller is implemented as an analog circuitry near to the plant. Similar to most others, two main variables are controlled here: ball position and beam angle (motor position). An ultrasonic based distance sensor measures the ball's position on the beam whereas a tachometer analog signal of the motor position is interpreted as the beam angle.

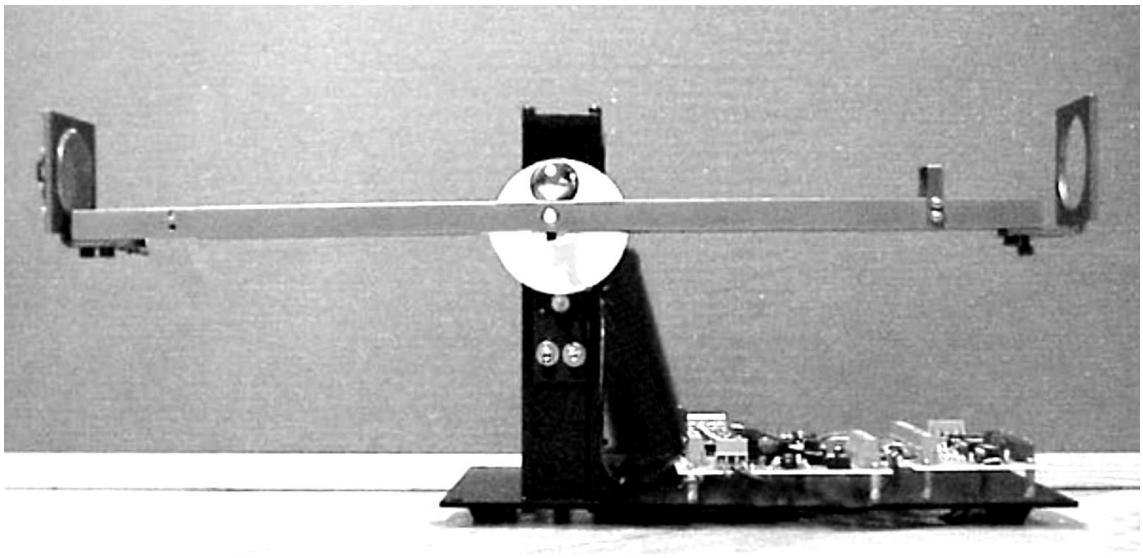


Fig 3.1: The analog ball and beam controller by Robert Hirsch, 1998 (taken from [82])

Being fabricated as a product, this [82] design indicates that the H/W analog control method of such a system tends to be reliable in terms of performance. The design analysis is achieved using LCTI, literally, Laplace domain and root locus method. Nonetheless, despite the performance advantage of analog approach, this system lacks basic benefits of digital control as re-configurability, processing, embedded applications, etc. Also, insufficient information is given regarding the actuator and its driving mechanism; besides, the device misses the importance of HMI, which allows both local and remote supervisory RT operation, tuning, and configuration.

3.1.1.2. A robotic ball balancing beam

The Robotic Ball Balancing Beam by Jeff Lieberman, 2004 [83], has always been referred to as a successfully laboratory implementation of the problem although the basic approach and simplicity. Lieberman nicely built the plant, and he used the concept of Nickel Chromium linear resistive wires to measure the ball position on the beam and an angular potentiometer for the inner loop's beam angle feedback. The control algorithm is performed in a PC desktop by the means of dSpace DAQ hardware and Simulink software tools, and the technique used is the usual theoretical TF pole placement and frequency repose compensation design. The setup uses a DC motor for the actuation, but there are not enough inputs provided about the parameters, torque/inertia calculations, and material/rotor selection. See Fig 3.2 below.

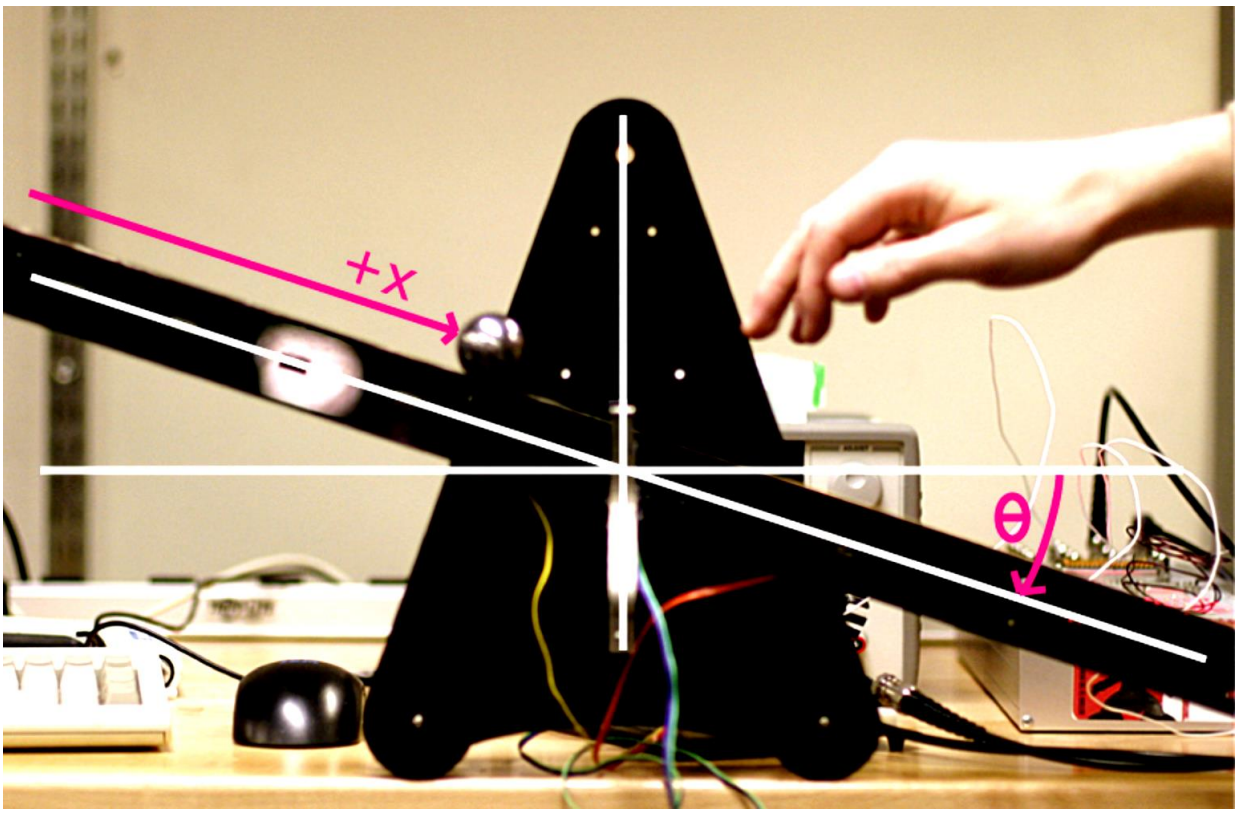


Fig 3.2: The robotic ball-balancing beam of Jeff Lieberman, 2004 (taken from [83])

3.1.1.3. A ball-on-beam system with an embedded controller

Authored by some faculty members in Electrical Engineering Department at the University of North Florida, the Ball on Beam System with an Embedded Controller [84] emerges as the first work in the list that uses an embedded digital MCU to implement the controller. The design approach of this system is quite different from the previous two. The authors put hands on implementation ignoring any theoretical derivation. A positive remark on this paper is that as it is the only one, among other literature, which pointed out signal conditioning of the measured feedback signals as a vital part of the control system. In fact, the paper gives BD as well as circuitry details on how the differential Sharp distance IR sensors' (used to sense the balls position) signals are subtracted, filtered, and amplified using Op Amps to match the MCU AI I/Os.

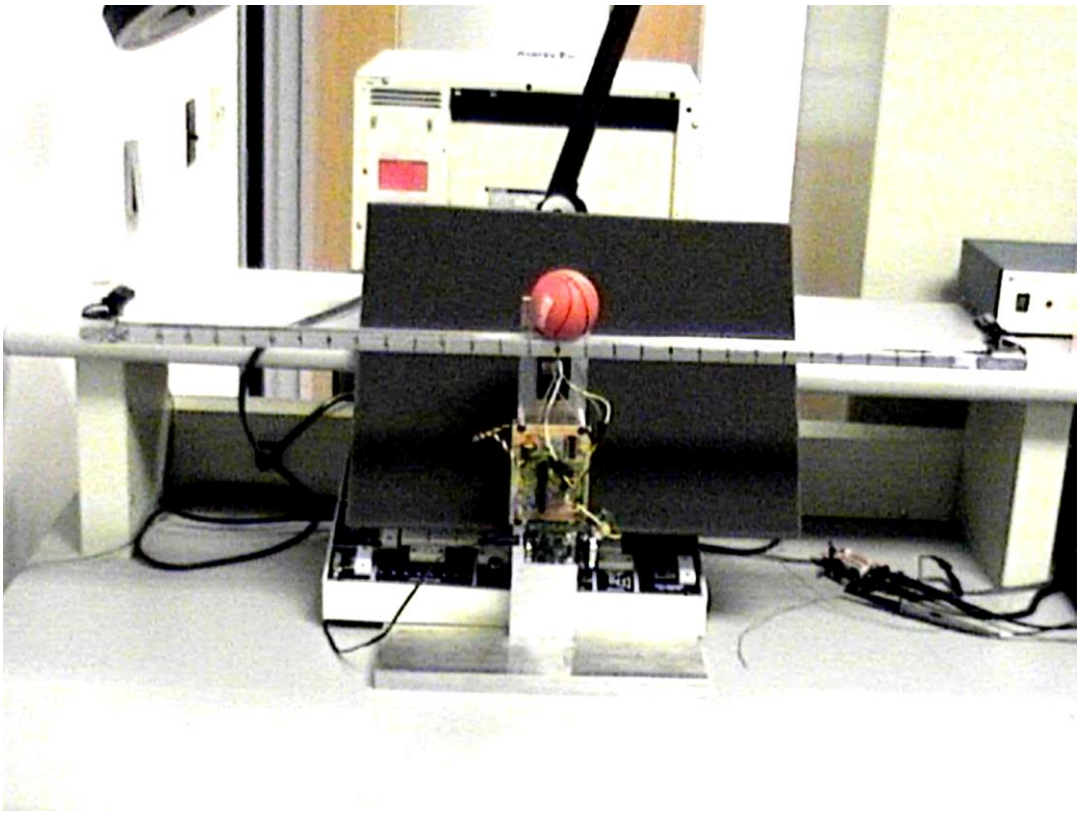


Fig 3.3: A Ball on Beam System with an Embedded Controller in tracking in the lab (taken from [84])

The control and actuation techniques used here and basic yet distinctive. The paper introduces the use of velocity feedback in the ball on beam problem, and it employs one main P loop only, with no inner sensing, because the actuator chosen is a robotic servo that has the mechanism of rotor positioning feedback mechanically internally. Fig 3.3 shows the final working demo. As a shortcoming to this work, although the control is embedded, the system does not provide any online tuning or HMI unit neither locally (LCD) nor supervisory (PC monitor).

3.1.1.4. Application of LabVIEW for real-time control of ball and beam system

"The Application of a LabVIEW for Real-Time Control of Ball and Beam System" by Basil Hamed [51] provides related sort of general implementation to Jeff Lieberman's, 2004. Both systems are computer controlled and not embedded (PC desktop control unit only). The key difference here is that Basil introduces the solution using a different H/W and S/W tool kit, NI's DAQ cards and real time LabVIEW BDs VIs. In terms of study and approach, this system gives more information on the five states SS nonlinear unstable dynamics of the ball on beam. In addition, the paper presents an advanced digital instrumentation tools used currently for ball on beam control, i.e. H-bridge DC motor driving, PWM, and quadrature encoder for inner (beam angle) loop feedback.

The control algorithm applied is the classical PID control tuned by ZN method. However, although the graphical interface represents an advantage of LabVIEW, as per industrial standards, the control and HMI, performed in the same PC unit, remains not satisfactory as per pre-mentioned industrial standards and architectures. Besides, how the inner motor position control loop is being connected to the only main PID, ball position, loop mentioned seems ambiguous in this paper. Fig 3.4 (a) and (b) below shows a snapshot of Hamed's system and the NI DAQ cards used.

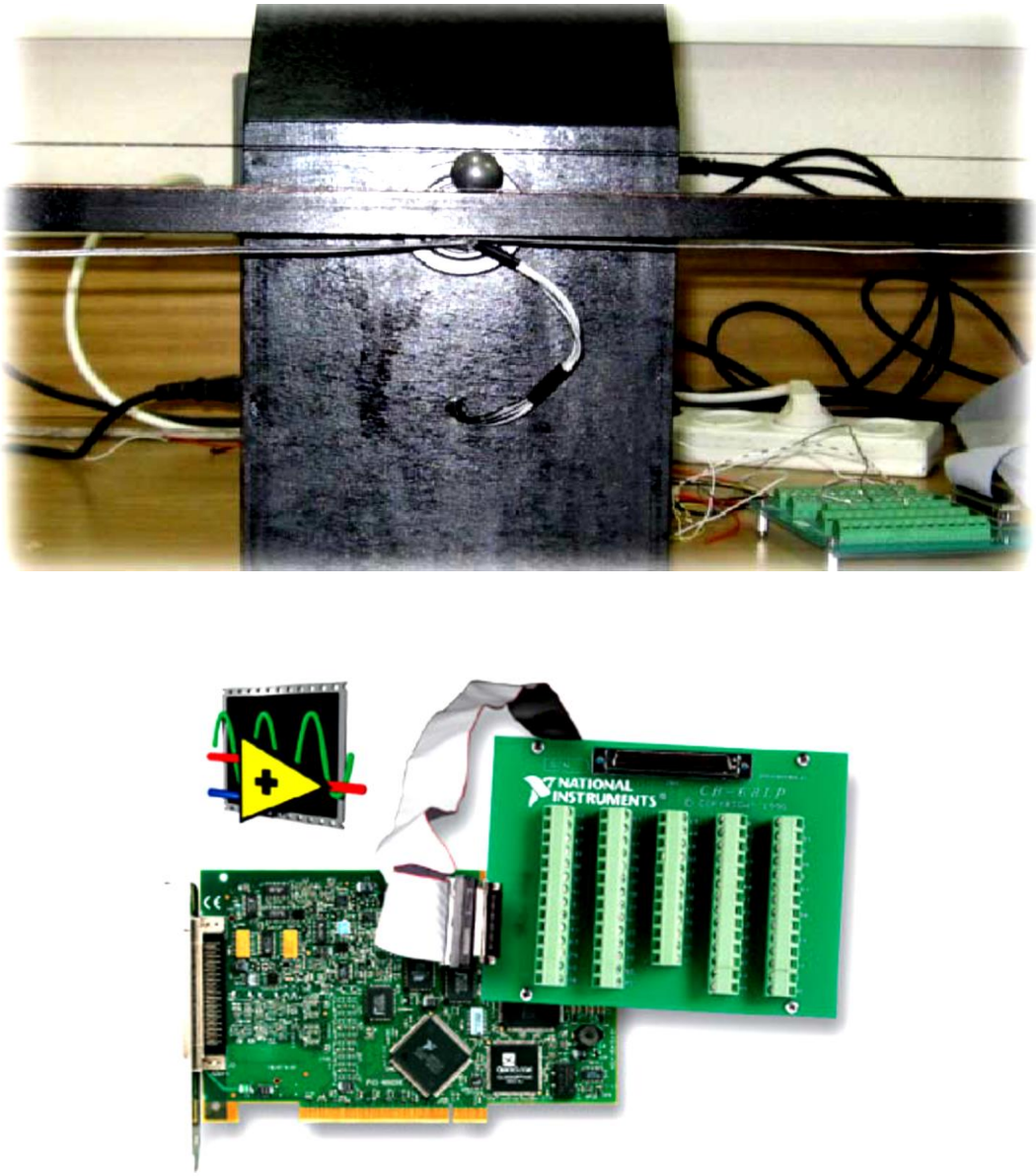


Fig 3.4: (a) Hamed's ball on beam built plant and (b) NI DAQ cards used (taken from [51])

3.1.1.5. Adaptive nonlinear control of the ball on beam system using the centrifugal force term.

As mentioned earlier, the ball on beam dynamics are basically nonlinear; thus, it can be analyzed using advanced NL systems theory. Adaptive nonlinear control of the ball on beam system using the centrifugal force term by Min-Sung Koo1, Ho-Lim Choi, and Jong-Tae Lim, arrives as one of the studies grabbing the opportunity of the flexibility of the ball on beam process to further apply both the advanced adaptive and non-linear control, [85].

Nevertheless, the authors neither assembled the plant nor engineered the automation system. They used the Quanser model instead as shown in Fig 3.5 below. Likewise, as it is for most scholarly papers considering the ball on beam, although experimentally implemented through DAQ and PC control, the paper concentrates on the control algorithm part and ignored other important RT implementation design facts and industrial criteria.

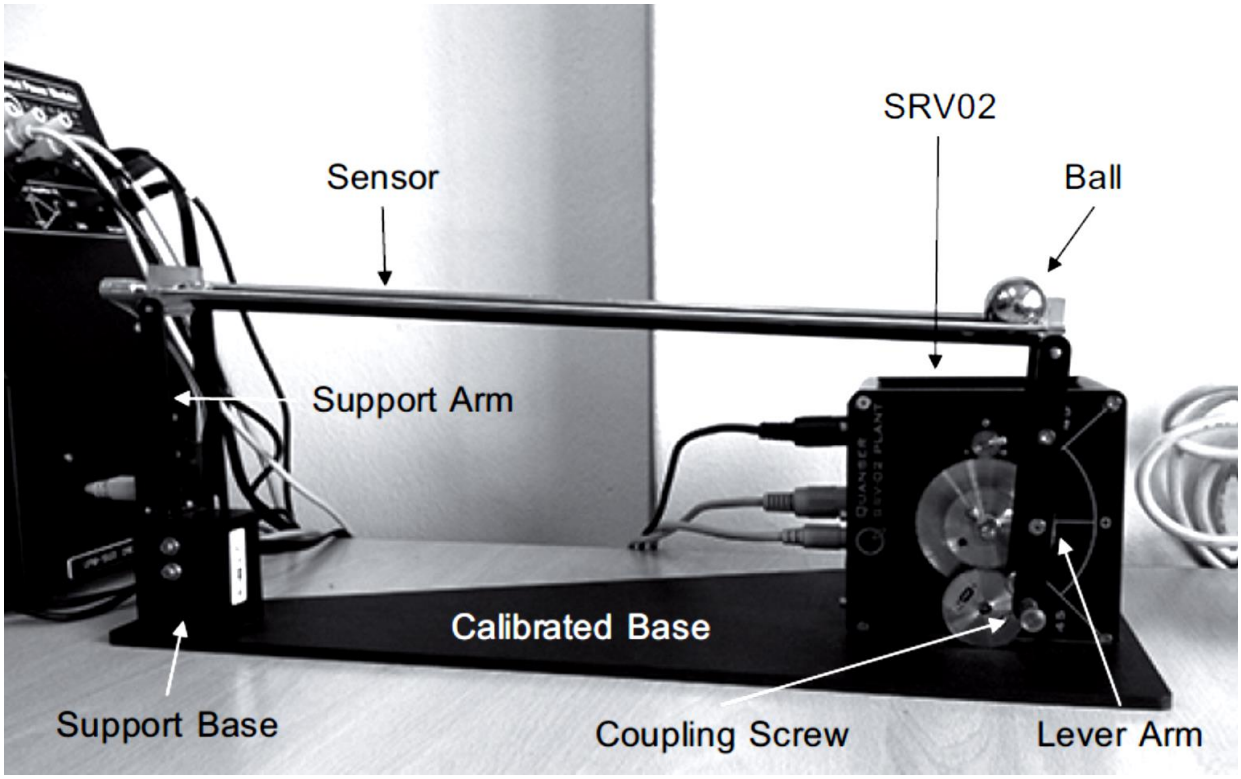


Fig 3.5: The Quanser's ball on beam plant used in the adaptive NL experiment (taken from [85])

3.1.1.6. Performance analysis of MIMO ball and beam system using intelligent controller

Another control implementation of the ball on beam using others' designed plant model, Berkeley Robotics' (Fig 3.6), appears in the published literature, "Performance Analysis of MIMO Ball and Beam System using Intelligent Controller by S.Senthilkumar, P.Suresh, and S.Senthilkumar, [86]. The authors used an implemented model of others lab work [86].

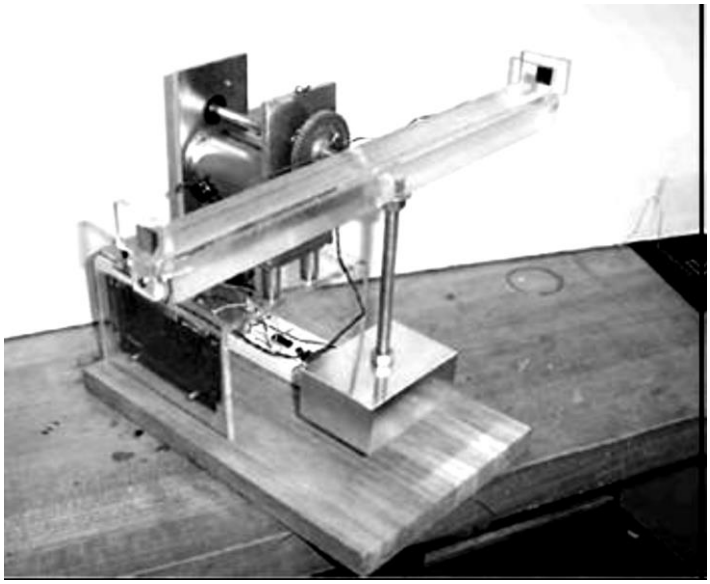


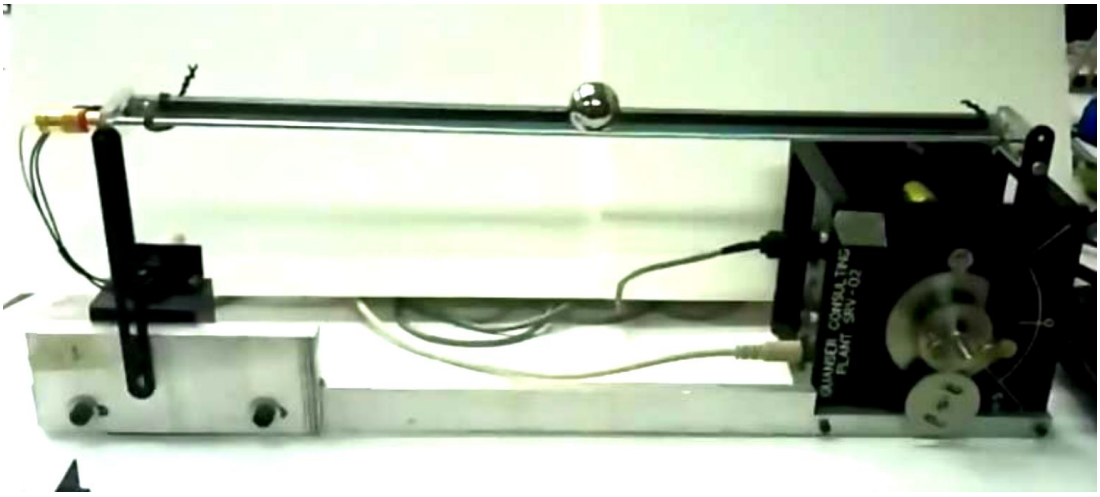
Fig 3.6: The built model of Berkeley Robotics Lab of the ball on beam used in the study (taken from [86])

Even though referred to as MIMO, the paper attacks the problem using the usual cascaded PID servo SISO approach. The only one difference here is that the study adds the useful scheme of fuzzy logic controllers on the top of the two (outer and inner) PID loops, designed by the stability analysis and characteristics equation approach, which boosts the eventual performance. The control implementation, however, follows most others' literature, of missing the point of having two separate units: control station and HMI, although the significance of the ball on beam to training in process industry took place in the introductory motivation section of the paper. The authors also acquired the plant process data to, and then applied the fuzzy PID controller on, a RT MATLAB/SIMULINK tool as both PC control and HMI.

3.1.1.7. Modeling and control of ball and beam using model and non-model based approaches.

This study delivers useful material of using advanced control systems to solve the ball on beam problem though the pre-manufactured Quanser physical model. The paper compares the advanced optimal MIMO control method, represented by the SF LQR controller, with the normal

SISO non-model PID approach. The study develops a third approach also using both techniques combined in the usual cascade control structure where the LQR derives the master and the PID appears as the inner loop [87]. Fig 3.7 below shows (a) the Quanser model deployed in the advanced study and (b) the SF LQR optimal controller realized. Once more, even though the very valuable and successful material this work offers regarding the side of control algorithm, the scholars did instrument automation, and they implemented these advanced controllers on the conventional (non-standardized) PC desktop methodology.



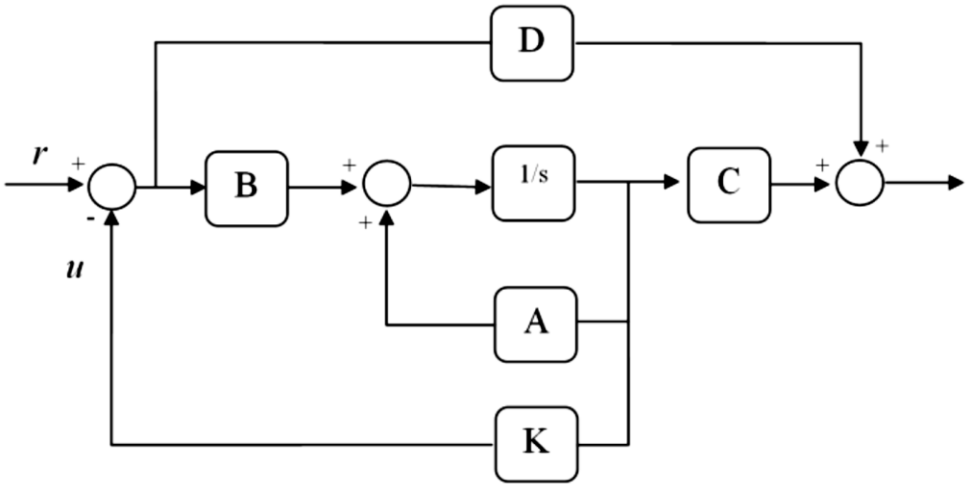
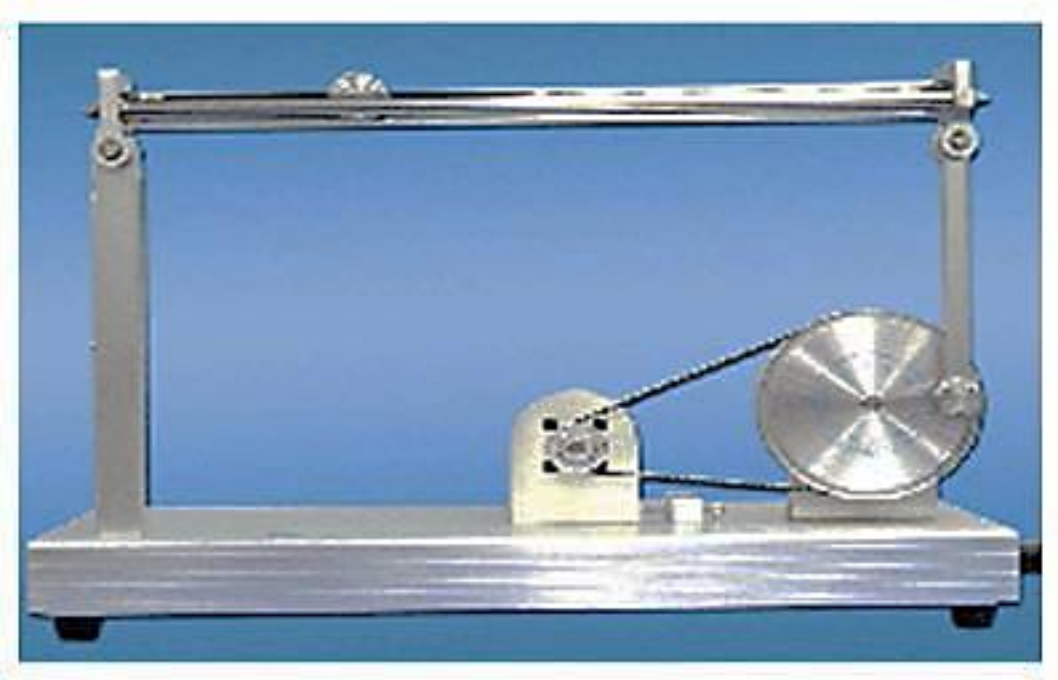


Fig 3.7: (a) the Quanser model deployed in the advanced study and (b) the SF LQR optimal controller realized [87].

3.1.1.8. Design and simulation of gain scheduling PID controller for ball and beam system.

As stated before that the ball on beam plant model is one of the best lab examples that can be realized by various progressive control theories. Similarly, this literature “Design and Simulation of Gain Scheduling PID” by Bipin Krishna, Sagnik Gangopadhyay, and Jim George [79] gives beneficial theoretical and simulation study of adaptive control using the ball on beam. The paper appears on the chosen list this research refers to although no RT implementation is reported, yet the authors relied on real data of an existing physical plant demo, the ball and beam of Google Technology, GBB1004, shown in Fig 3.8 (a) below. It also demonstrates the main contribution provided, adaptive auto-tuning PID using gain scheduling (Fig 3.8 (b)) [79]. However, not only does this work ignore RT implementation, but also is considered to be incomplete as it builds on some modeling assumptions, and it overlooks associated implementation facts. For instance, the paper considers one main PID loop to be realized while in fact the DC motor of the utilized model should have an inner positioning feedback too.



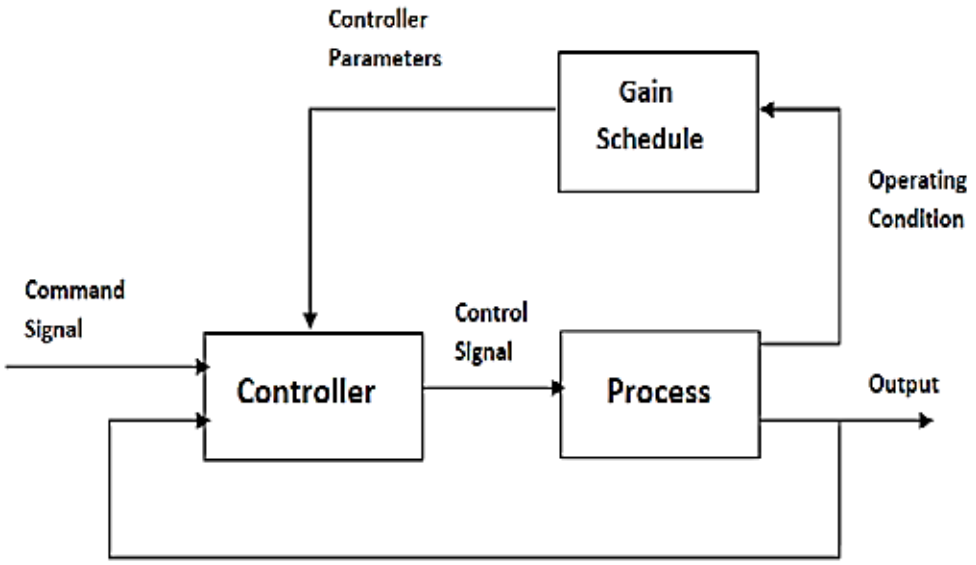


Fig 3.8 (a) The ball and beam model of Google Technology used in the simulation and (b) the gain scheduling adaptive auto-tuning PID controller (taken from [79])

3.1.1.9. Fuzzy controller design for ball and beam with an improved ant colony optimization.

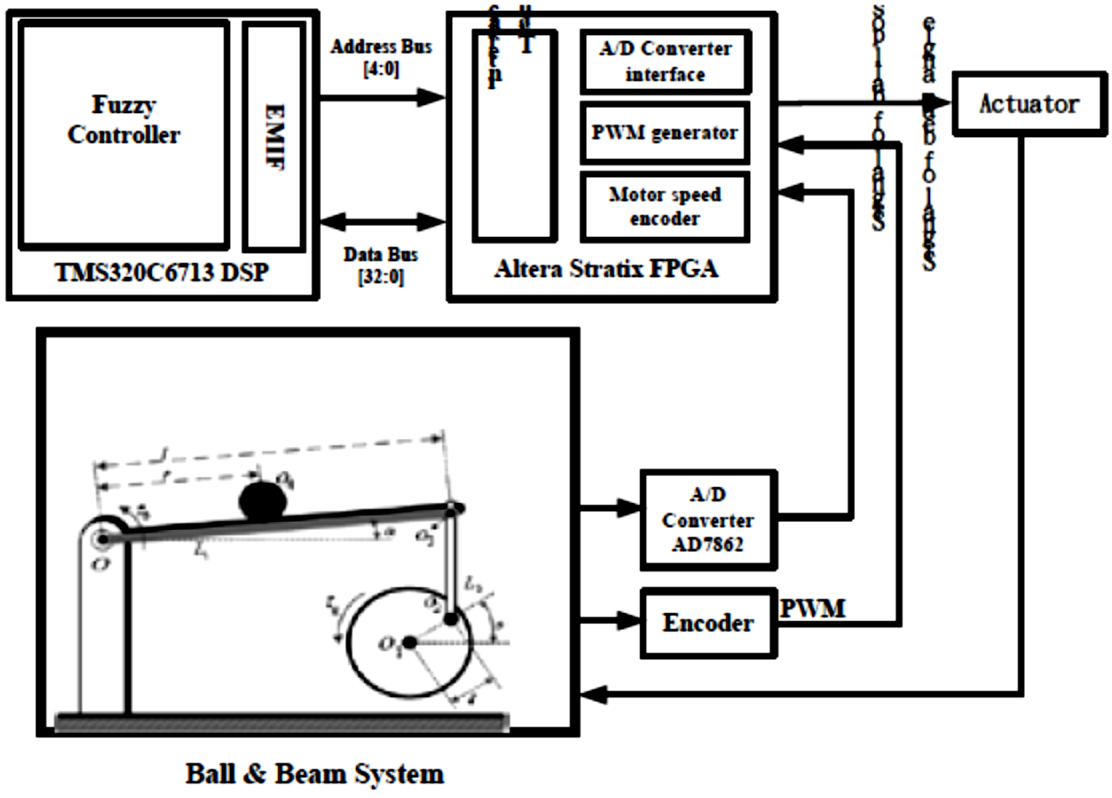


Fig 3.9: (a) The BD of the implemented digital control system and (b) the plant model used with DSP and FPGA boards (taken from [88])

The scholarly paper “Fuzzy Controller Design for Ball and Beam System with an Improved Ant Colony Optimization” of Yeong-Hwa Chang, Chia-Wen Chang, Hung-Wei Lin, and C.W. Tao [88] follows those series of studies that applied advanced linear control using the ball on beam. The authors applied their suggested algorithm, fuzzy controller with improved ant colony optimization, on a commercially manufactured laboratory plant model. Hitherto their work is the only one found in this research that implemented the digital controller by not following the convention of using either MCU or PC desktop as control stations. They introduced using FPGAs and DSP processors to the scene. The BD in Fig 3.9 (a) shows the main components of the digital unit implementing the control system, the DSP and the FPGA of development boards, and part (b) of the figure displays the pre-manufactured ball on beam plant utilized in the experiment.

The digital system is arranged in a way that the DSP executes the advanced fuzzy logic controller while the FPGA interfaces with I/Os. The same encoding and PWM techniques for a DC motor control are also used here. However, how the nice RT performance graphs are continuously acquired is unclear. In other words, it seems that the architecture does not include an HMI station, which indicates that the RT experimentation is for control algorithm validation only rather than both research and engineering implementation purposes. Moreover, the obtained RT experimental results were neither compared with reference SPs nor evaluated by the simulation curves, which does not indicate the actual performance of the suggested algorithm.

3.1.1.10. Design and control of PID-controlled ball and beam system.

Fig 3.10: the PID stepper controlled ball on beam system prototype (taken from [89])

Design and Control of PID Controlled Ball on Beam System, of Sıtkı KOCAOĞLU and Hilmi KUŞÇU [89] emerges as the first work found that uses a stepper motor to actuate the beam angle instead of a DC or servo. The designers implemented a nice looking system plant, executed the control in an embedded MCU controller, and interfaced an LCD based HMI and a control panel (see Fig 3.10). The control algorithm here represents just a normal PID control since the stepper motor does not require an internal feedback for angular position tracking. Yet as per industrial automation standards, this work is considered to be incomplete because the HMI unit emulates only local panels in plants facilities whereas the supervisory PC monitor level is crucial. To sense the ball position in this experiment, a differential distance measurement is deployed by the means of two sharp IR sensors at both ends of the beam. An advantage to this work is that the authors provided a calibration relationship of how the voltage output signals of the IR sensors are interpreted as distance measurement in the microcontroller, which is an important part of the instrumentation process, therefore correct control behavior.

3.1.1.11. Mechatronics design of ball and beam system: education and research

Because of its nature, the ball on beam problem may involve designing, building and implementing mechanical and mechatronics. “Mechatronics Design of Ball and Beam System: Education and Research” by Farhan A. [90] emerges as a recognized work that considers designing the mechanical/mechatronics side of the system. As appears in Fig 3.11 below, not only does the author intend to design the system using embedded PIC MCU and the interfaced the HMI to an LCD display, but he also demonstrates some useful material on the mechatronics part of the ball on beam including: mechanical CAD 3-D design of the plant, H-bridge electronics, and another circuitry layout and wiring associated. The designer adopted PD control, and he used a PMDC motor as an actuator, linear resistive sensors as the ball position feedback, and a V-shaped beam.

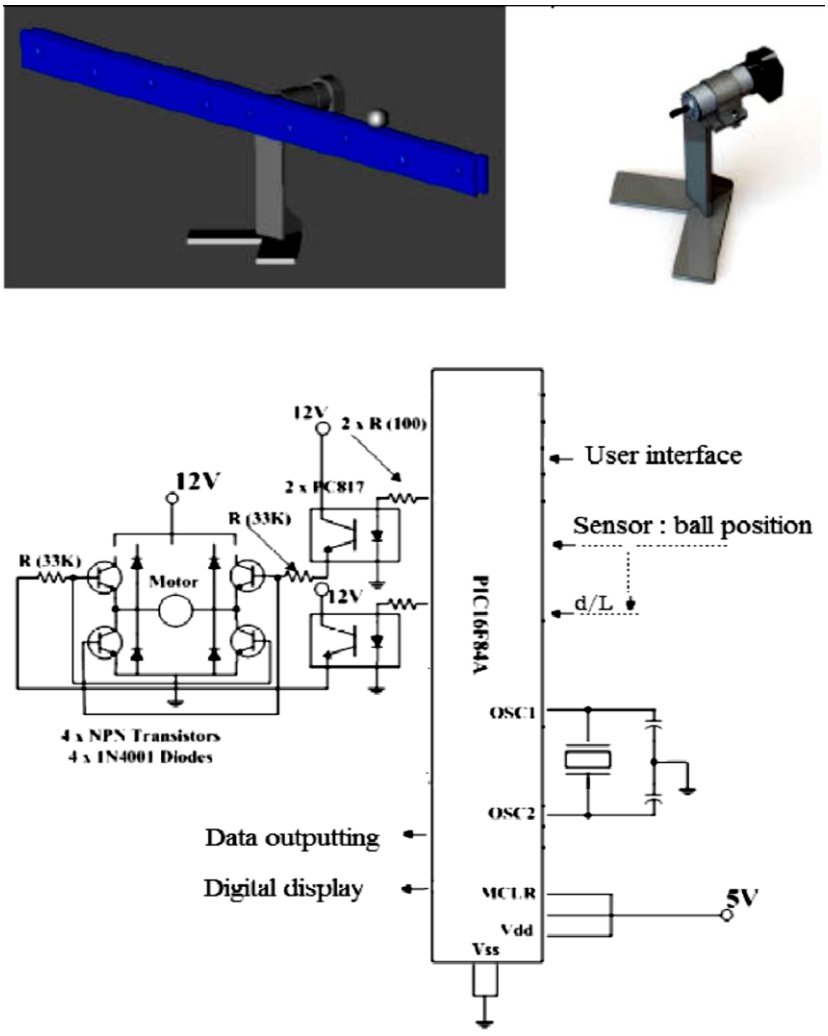


Fig 3.11 (a) The plant mechatronics drawing and (b) the embedded system, PIC MCU (taken from [90])

However, although the author pointed out on important part of the implementation, mechatronics design, the paper lacks details or explanations on why particular material of the beam is chosen, plus actuator sizing, torque, and inertia calculation that contribute to the eventual control loop performance. Moreover, as discussed several times before, the LCD HMI method does not comply with the supervisory monitoring industrial requirements. Also, it is noticed that even though the paper pointed out the mechatronics and RT MCU embedded designs, the displayed analysis and results were Simulink simulation only.

3.1.1.12. Implementation of ball-and-beam control system as an instance of Simulink to 32-bit microcontroller interface

This seems to be the first publication found that tried to consider and satisfy the industrial standard of acquiring two separate units, control and HMI, although it is purposed for educational purposes. Implementation of ball-and-beam control system as an instance of Simulink to 32-bit microcontroller interface [91] implements the ball on beam control system on an ARM MPU while realizing the HMI PC monitor using Matlab/Simulink S/W tool. The communication between the two units is implemented serially by the media of URAT interface as shown in Fig 3.12 (b) below. By the aid of SimMechanics tool, the author nicely built the plant besides designing the control system. He suggested that his final layout of dynamically activating both ends of the beam, rather than the middle, such that one is attached to a static rod while the other is indirectly driven by the motor, accumulates to a better response of the beam manipulation, hence eventual control performance.

This work tends to be complete in terms of implementation as it pointed out selection of physical mechanical dimensions, materials, torque, and the optimal actuator associated. It is the second one found in literature that uses the stepper motor as the actuator; thus, no inner feedback loop is needed for the motor's position tracking. The nice looking ball on beam plant of this literature appears in Fig 3.12 (a) with the components' core numbered and labeled. However, even though the importance of mechanical construction and calculations are pointed out in this work, detailed steps of such scheming are not included. Besides, these mechanical calculations seem to be focused on torque, while it is not the only important design parameter to be considered as will be discussed later. The designer uses IR sensors for sensing the ball position on the beam, and he demonstrates various control strategies as the PID, LQR, and others.

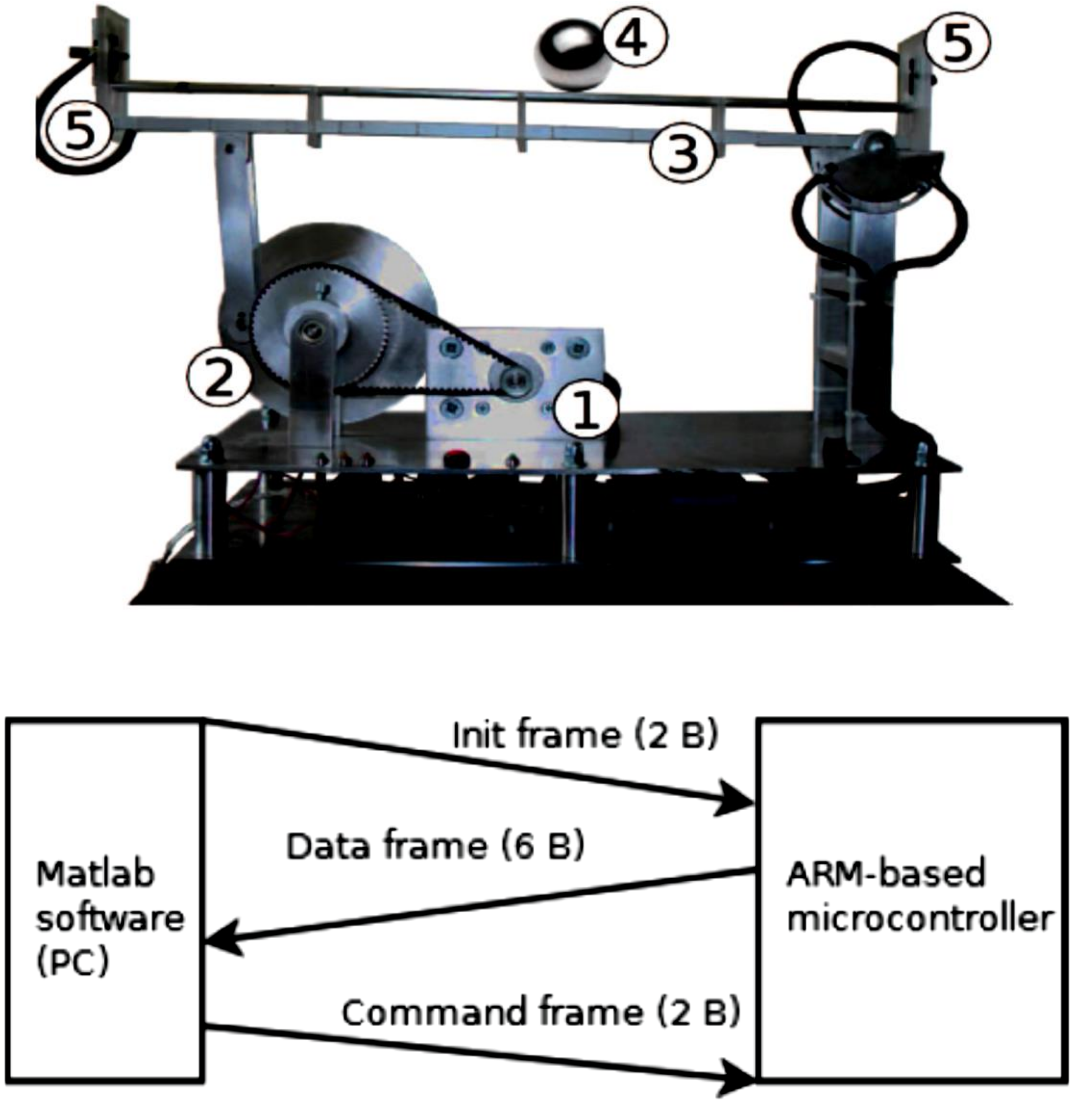


Fig 3.12 (a) the implemented SimMechanics model and (b) communication between HMI and the embedded MCU (taken from [91])

3.1.1.13. A pneumatically actuated ball and beam system

As stated earlier that the ball on beam system puts good opportunities of testing different technologies within control systems automation instrumentation. “A pneumatically actuated ball and beam system” by Z˘eljko S˘ itum and Jos˘ko Petric´ [92] is one of the latest publications in this matter. It uses the conventional pneumatic actuation, instead of the PWM electrical one.

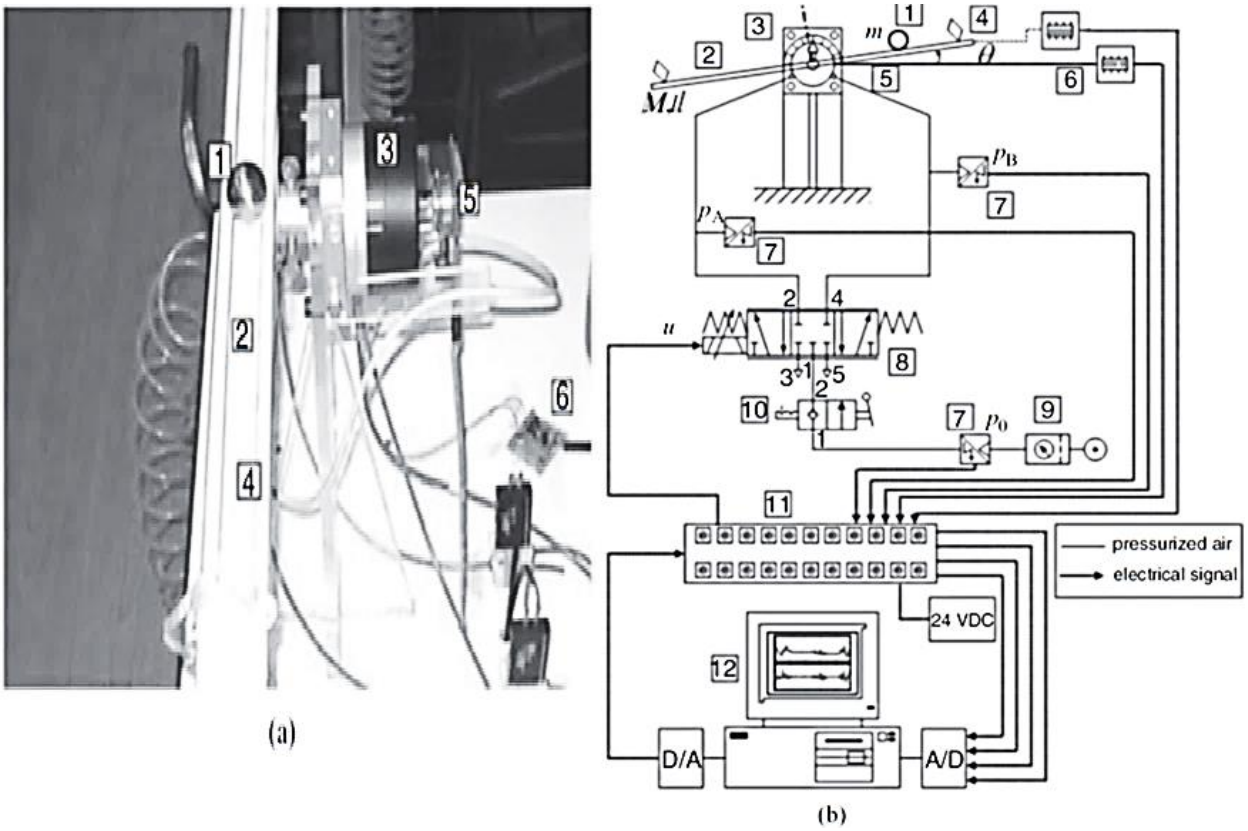


Fig 3.13: The pneumatically actuated ball on beam (a) demo and (b) system drawing

This method requires using AO/DAC and I/P to manipulate the pneumatic actuator which is similar to proportional CVs introduced before [92]. The paper derives a simplified NL model of the ball on beam and a brief illustration of the digital/analog communicated control system (Fig 3.13 below). The engineers use differential ultrasonic sensors at both edges of the beam to track the ball position and potentiometer for the inner (beam angle) feedback. A simple PD and a LQR control are utilized as master controllers consequently to the cascade structure. Although it applies a common industrial actuation, pneumatic CVs, as a disadvantage of this work, the system is not embedded because the function is implemented on a PC based control unit, using MATLAB RT workshop tool. In addition, as the paper suggests a different way of final control element, it did not compare the performance of this approach with other actuation methods. How the mentioned mechanical parameters, materials, and CV sizing were selected is not declared either.

3.1.2. Additional Research Directions and Contribution to the Problem

As explained above, after going through all selective publications and literature, it has been clearly noticed that most of them concentrate mainly on the control algorithm and dynamic response parts ignoring other effective elements of the control loop such as robustness, actuator selection and control unit implementation. Thus, this research paper puts its energized focus on other directions within the same problem. Those new research directions originate on the actuator, control unit, signal conditioning, and on the measurement and control communication (WSN).

3.1.2.1. On the actuator

The first research direction that comes to mind when studying the ball on beam control problem from the implementation point of view is the choice of the actuator. Most previous literature has used DC motors for actuating the control signal and so manipulating the beam angle. Only two papers found have chosen the stepper motor as a final control element of a PID based system [89]. [92] applied a different method of actuation that is similar to proportional CVs used in industry for other chemical, fluids, and thermal processes other than motion control. However, regardless of the CV's one, which might be costly and unfeasible for motion control applications, no one has ever investigated whether the DC brushed or the stepper better applies in such positioning practice. One of the objectives of this setup is to observe the differences of both approaches and try to conclude which one adequately actuate such applications through comparing both control performances. Table 3.1 below illustrates some nominated major differences between DC and stepper motor focusing on position control applications. From control point of view, it has been shown that cascade control has an advantage over single loop control for many applications; therefore, the main point here is to compare the open loop stepper actuation (single closed feedback loop) with the DC servo (cascaded) control performance.

Stepper Motor	DC Motor
Open loop actuation (defined steps)	Needs internal feedback (servo)
Good low speed positioning (high stall and holding torque at low speed but torque reduces with speed)	Capable of good high speed control in general (torque is proportional to speed and high torque over moment of inertia ratio)
Does not require maintenance	Maintenance required (for brushed)
Low speed. Typically, less than 3000 RPM	Higher speed ability
Cost effective in general	More sophisticated motors cost more (requires adds in; gears need to be added to lower the speed, encoder, brakes, etc.)
Accurate position control but unawareness of missing steps may occur (not precise at low speed if no micro-stepping)	Accuracy depends on the encoder feedback resolution and control function (may overshoot and loose position if encoder is out)
Not very smooth motion (discrete steps)	Smoother than stepper (continuous)
Reliable and rugged	Efficient and good dynamic response
Heat problems and noisy	Tends to have electrical noise (brushed) but quieter than stepper in sound and vibration

Table 3.1: Tradeoffs between DC and stepper motors in motion control [93], [94]

3.1.2.2. On the digital controller unit

Besides the actuator, the control performance of this system could be improved or considered if the conventional method or the means of implementing the digital controller is altered. In other words, one can notice that most previous work implementing the real time ball on beam (digital) control system followed the convention of using a processor mode unit where the control calculation is simply executed either inside a PC (desktop) or embedded in an MCU.

Regardless of the disadvantage of just using a PC's CPU to do the job and the standard requirement in industry of installing embedded RT controllers, only one piece of work, [91], found followed this pre-condition where the serially communicated HMI is located in a supervisory PC. Nevertheless, that work does not take the problem to the network level implementation; besides, it does not consider the field communication side as well as both controller and field sensors and actuators networks as parts of the industrial architecture.

Another reference that uses an embedded digital system other than a MCU/MPU, and rather comprises of both a DSP and an FPGA is [88]. However, not only did the authors of this paper not design and build the ball on beam automation, as they just used a manufactured plant including the actuator, but also the concentration was mainly on the advanced control algorithm only as declared several times before. The paper does not even explain the specific programming detailed role of the DSP and the FPGA and how the digital program or control calculation is being executed among both; besides, how the data is acquired and plotted for HMI to get RT graphs is not described rather than considering a network a distributed supervisory level control. What would be the impact of using those DSP and FPGA on final control performance does not take place in that paper either. The onboard integrated FPGA implemented I/O interface only.

This research paper takes into account the industrial requirement of placing embedded networked distributed RT controllers while the function of PCs remains only supervisory monitoring, operation, and HMI, and so the first improvement originates including the ball on beam plant as a part of such architecture. Another important research argument this paper advocates is seeking into answering the question: would replacing a RT processor control mode by an FPGA in certain applications be worth it in industrial environments? In other words, would the suggested DSP advantage of FPGA benefit control applications, or would not be much of a

huge difference when comparing this approach with an advanced RT processing one?

To better elaborate this FPGA environment point in control, it seems logical to mention its advantages in the world of fast computation and digital signal processing. There exist various reasons why FPGAs are thought of as superior over processors in many RT control applications:

- Although newly released desktop CPUs such as Intel's reached clock cycles of GHz frequency, which supposes loop rates up to nano-seconds, many processor-based RT controllers adopted by industrial process control can only make maximum of 10ms execution periods (scan time), i.e. Siemens PLC, Emerson Process Management's DV, and Yokogawa's Centum controllers [95], [96], [97]. Normal FPGAs, on the other hand, can reach a clock cycle of around 40GHz, for example NI cRIO FPGA [98], which is also much faster (in nano-seconds execution time) than the most advanced PLC found in industry. Fig 3.14 demonstrates how fast FPGAs GMACs are developing in comparison with CPU based processors in terms of performance in GFLOPs.

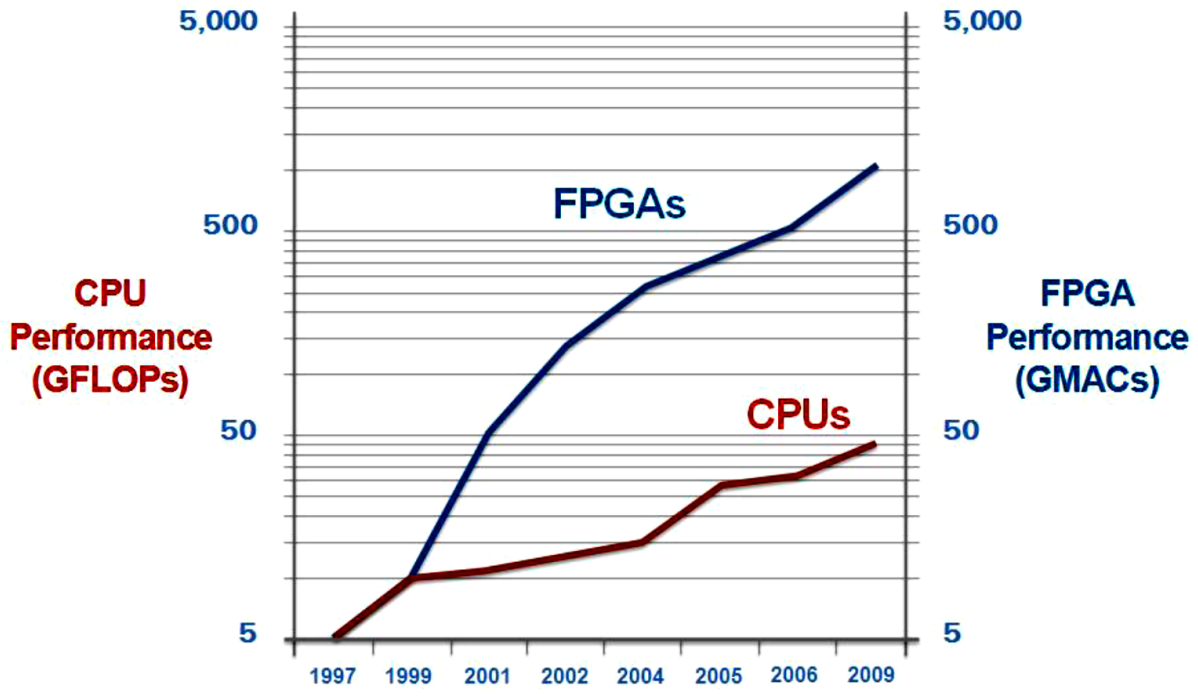


Fig 3.14: Performance measures development of CPUs and FPGAs. (taken from [63])

- The reason why FPGAs' response is fast develops from the fact that they are nothing but hardware mapping implementation equivalence of typical processor computations, which run usually inside a CPU by a lead of a RTOS, using gates level configuration, more specifically RTL, as shown in Fig 3.15 below. This feature makes FPGAs more attached to the real H/W I/O world represented by sensors and control (actuators) signals. As a result, the control feedback delay and response time will be minimized (in n-seconds time delay), [58], [99] as appears in Fig 3.15 below.

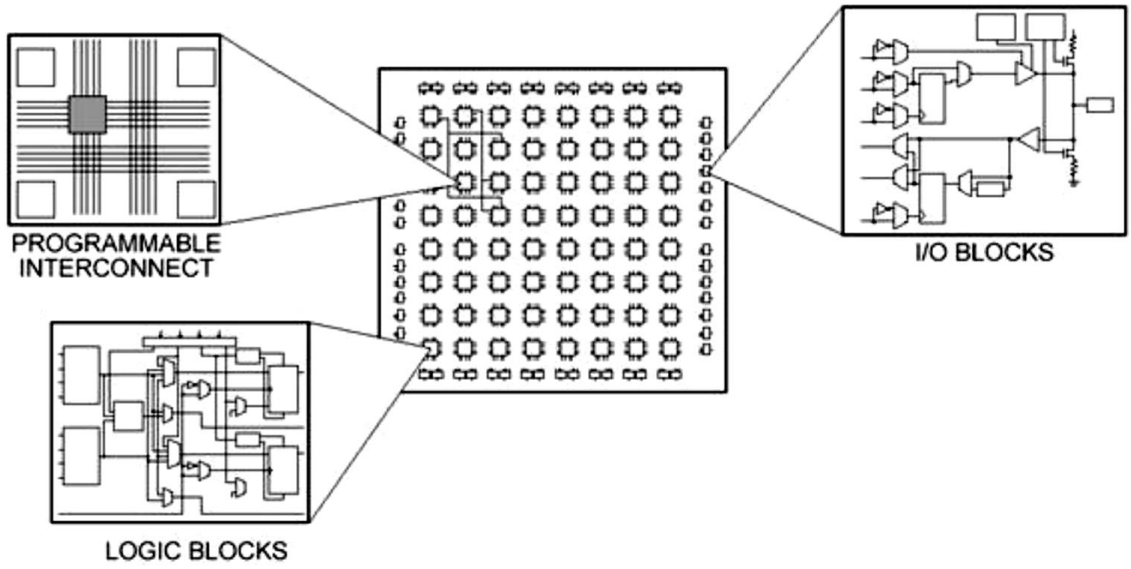


Fig 3.15: FPGA are logic gate level implementation of digital control (taken from [58])

- Another reason why FPGAs are deemed preferable in some applications originate from the fact that their H/W architecture allows direct and faster I/O interface response than other processor method [58]. The reason for this better I/O response is shown in the coming figure, Fig 3.16. For example, an AI block requires DAQ process of three levels: Analog domain RT signal propagation, AI and ADC, and interfacing to the S/W domain to perform the required processing whereas in standard FPGAs, this procedure takes only two levels of signal processing execution excluding the S/W RTOS part, shown in the

figure below. Recall that the S/W implementation exists when dealing with FPGAs, as they are programmable devices, but that soft intervention is due to user HMI interface and programming, which is in a form of host application; hence, it does not extra-propagate control logic calculations time.

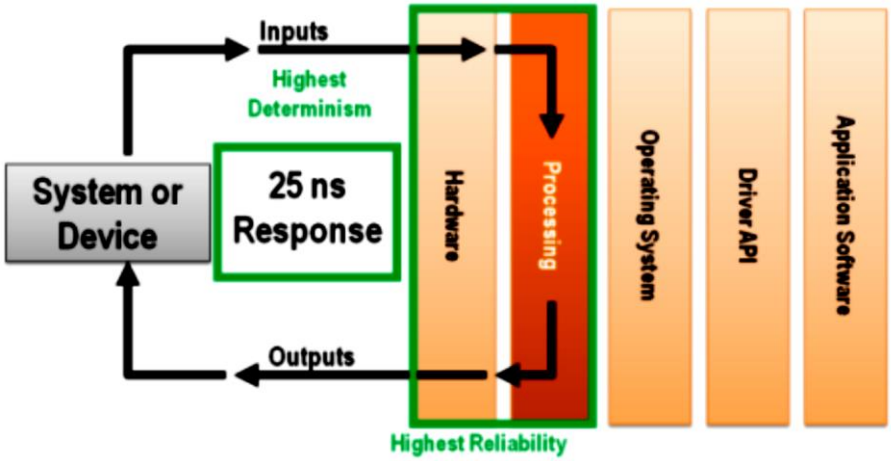


Fig 3.16: The advantage of I/O interface prorogation delay in FPGAs (taken from [85])

- In many cases, FPGAs are thought to be even more powerful in terms of fast computation than standard DSPs. This impression is because of FPGAs' capability and characteristics of implementing digital calculations and logic function blocks in a highly parallel structural manner that minimizes the overall execution time, and consequently control delay. As a result, FPGAs appear suitable for high bandwidth functional demands such as PWM, quadrature encoding, and fast counting, besides the rapid mentioned I/O blocks execution rates [98]. Fig 2.70 in Chapter 2 shows an example of a comparison between a standard serially implemented DSP in contrast with a typical DSP and FPGA in MSPS.
- All of the above motives enable FPGAs to reach the desired H/W response feature of ASICs, yet the advantage of the first relies on the fact that they are programmable, therefore more flexible, productive, and cost effective. As a result, FPGAs provide the

satisfactory solution, located in the middle, possessing both desirable features of CPU processors as well as ASICs [89], [58].

However, on the other hand, there have been observed several reasons why FPGAs have not been yet widely adopted in automation industry, i.e.

- Any type of logic can be theoretically implemented in FPGAs, but they are limited by size (capacity) as there exist a limit on the maximum number of gates that can be implemented; thus, as observed, control functions that require many calculations and logic blocks might not be deployed.
- Even when the logic calculations can get through, their deployment process is time consuming. As an example, it takes an average of 30 minutes to deploy and download a LabVIEW VI. This limitation marks FPGAs as not flexible or suitable with industrial demands that require immediate reconfiguration for editing or changes in control logic programmed, which saves time and cost.
- Another point to be considered is that RT processors have the advantage of being attached with more advanced S/W tools and ready built function blocks, which saves system configuration engineering time and opens doors for more advanced applications. On the other hand, in FPGAs, every piece of logic should be interrupted as digital gates, which involves extended engineering time and limits their use to basic control functions only.
- Most FPGAs support fixed point digital arithmetic while advanced RT controllers can work on floating point representation also. This fact sorts FPGAs more subjected to calculation overflows errors as will be shown in later chapters and sections on this report.

Therefore, by implementing both approaches and comparing noticeable differences in performance, this paper tries to explore whether it is worth it to adopt FPGAs more frequently in

industrial process control automation, especially in some applications where very fast and high performance response are necessary such as the ball on beam, PWM generation, DC motor inner looping, digital encoding, counting, etc.

3.1.2.3. On the measurements and control signals communication

Another important research direction to be considered in implementing the ball on beam control system is evaluating the effect of different communication protocols and technologies on control performance. A lot of the latest research has taken place regarding evaluating process control over various WSN protocols to be trusted in industrial facilities upon their assessed efficiency and reliability, besides wireless advantages. Similarly, this research will focus at some point in the future on assessing WSN on critical process plants as the ball on beam over the ZigBee protocol emphasizing on evaluating the eventual control performance. Two main standard methods are to be compared: the conventional wired point-to-point verses wireless WSN over ZigBee technology.

3.1.2.4. On the control technique and algorithm

Looking at most previous work implementing the ball on beam, one can conclude that the main control strategy involved is the cascaded structure, shown in Fig 3.17. This structure reduces to a simple loop control in case a stepper motor is utilized. No matter what advanced algorithm is applied, most literature applying the SISO approach followed the cascade control strategy intuitively either as per the DC motor requirement, of placing an inner loop, or intentionally. The difference relies on selecting algorithms that fill the ball controller and the actuator controller blocks found in the figure below. For instance, several publications implement the ball controller as a form of a classical P, PI, PD, or PID controller. Some use fuzzy logic as a master ball controller on the top of the PID while others deploy adaptive auto-tuning PID and nonlinear

techniques. The most linear advanced controller realized using MIMO approach is the optimal LQR SF controller considering the LCTI SS model of the ball on beam directly. This LQR algorithm is also used in the cascade control scheme as illustrated in few number of publications.

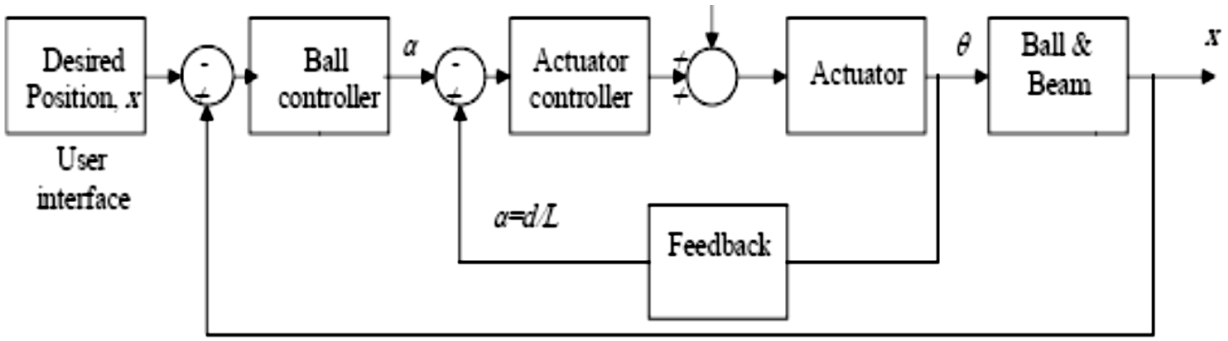


Fig 3.17: The ball on beam cascade structure utilized by literature (taken from [90])

As a contribution, this report proposes two modified control techniques to the problem of the ball on beam. First, the design introduces the use of an advanced modified PIV-Vff-Aff servo controller, released in Fig 3.18 (a), which is somehow similar to the LQR, but it is a component of a number of control principles, feedback, PID, cascade, feed-forward, and SF. Besides the benefit of velocity feedback, this PIV controller gets advantage of Vff and Aff, which have demonstrated vital positive impact over servo position control applications as can be obtained from the studies conducted in [100], [101]. Likewise, as shown earlier in this paper, the introductory Chapter 2 above details major enhancement of general cascade and feed-forward control. Fig 3.18 (b) below is extracted from [101], and it demonstrates an example of a PI servo position control with and without Vff and Aff. In addition to all the advantages above, the suggested controller adds a third level cascaded, current loop, as shown below which increases the level of control customization. Not only does the suggested design use the PIV-Vff-Vff, but adds that to be a PID-PIV-Vff-Vff. The second major suggested control technique is an adaptive and IMC introduced in chapter 2 besides testing other algorithms.

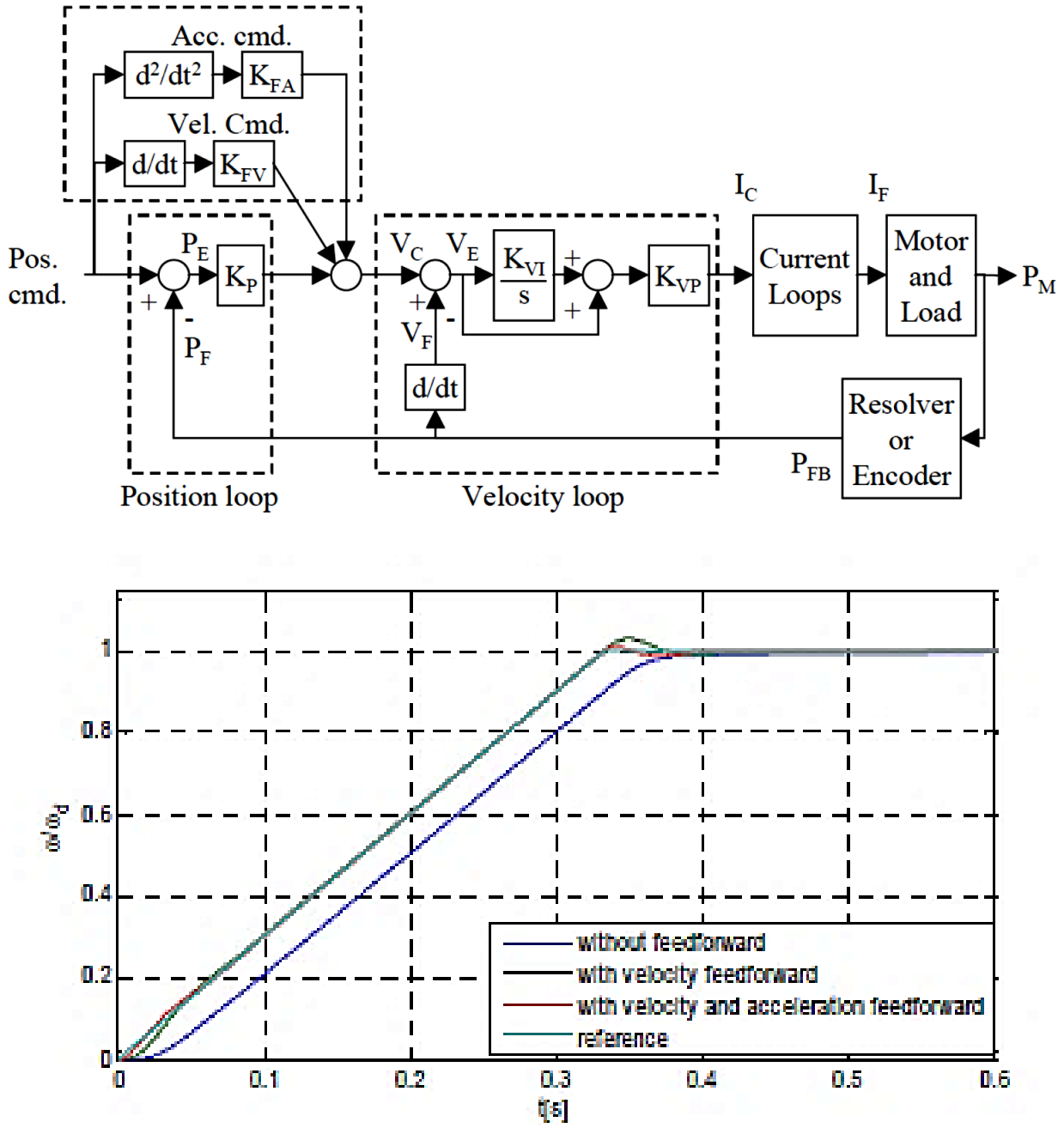


Fig 3.18: (a) The advanced PIV with Vff and Aff controller used in servo applications (taken from [100]) and (b) servo position system with and without Vff and Aff (taken from [101])

In some cases of the research, this adaptive IMC can be placed as the master or slave controller in RT (scan) mode while the previous, PIV-PID one, better fits as a servo (DC motor) inner loop in implemented in FPGA. Different configurations and scenarios will be tested and

compared using various modes: FPGA only, RT controller, and RT controller – FPGA. The main reason behind such reconfigurations is not only presenting the ball on beam control problem using new ideas that have not been published yet, but also trying to well assess the performance of a fast FPGA utilizing fewer of calculations and algorithms versus a RT controller that can accept larger size of such smartness and extra memory for more advanced computations. More design details are explained later. It is important to mention that in case a stepper motor is utilized, a single main controller will be applied only. This fact implies that the first contributed (servo) PID-PIV controller above will fit only for the DC motor configuration. In other words, for the stepper motor system, a simple PID control will be compared with a similar sort for the DC motor in FPGA mode. Other tuning methods and control algorithms can be measured for both stepper and DC motors approaches as well in the RT controller mode.

3.1.2.5. On the mechanical and mechatronics design

An important aspect that should be considered more carefully is the mechanical/ mechatronics detailed design of the ball on beam system plant. This essential factor of the physical dynamic response has been partially neglected in almost all previous work. Even when such mechanical physical parameters are considered, the focus is only on torque requirements. Furthermore, in most cases, it stands obvious that the selected torque requirement is based mainly on intuition, that a chosen actuator and plant material can lead in manipulating the beam quickly enough, and are consequently, able to control the ball on RT. This research paper tries to answer the question, why, regarding selecting those mechanical and mechatronics design parameters by formulating scientific associated calculations extracted from physical characteristics of several parts of the ball on beam plant. Upon answering such question scientifically, better control performance and dynamic behavior can be achieved sensibly.

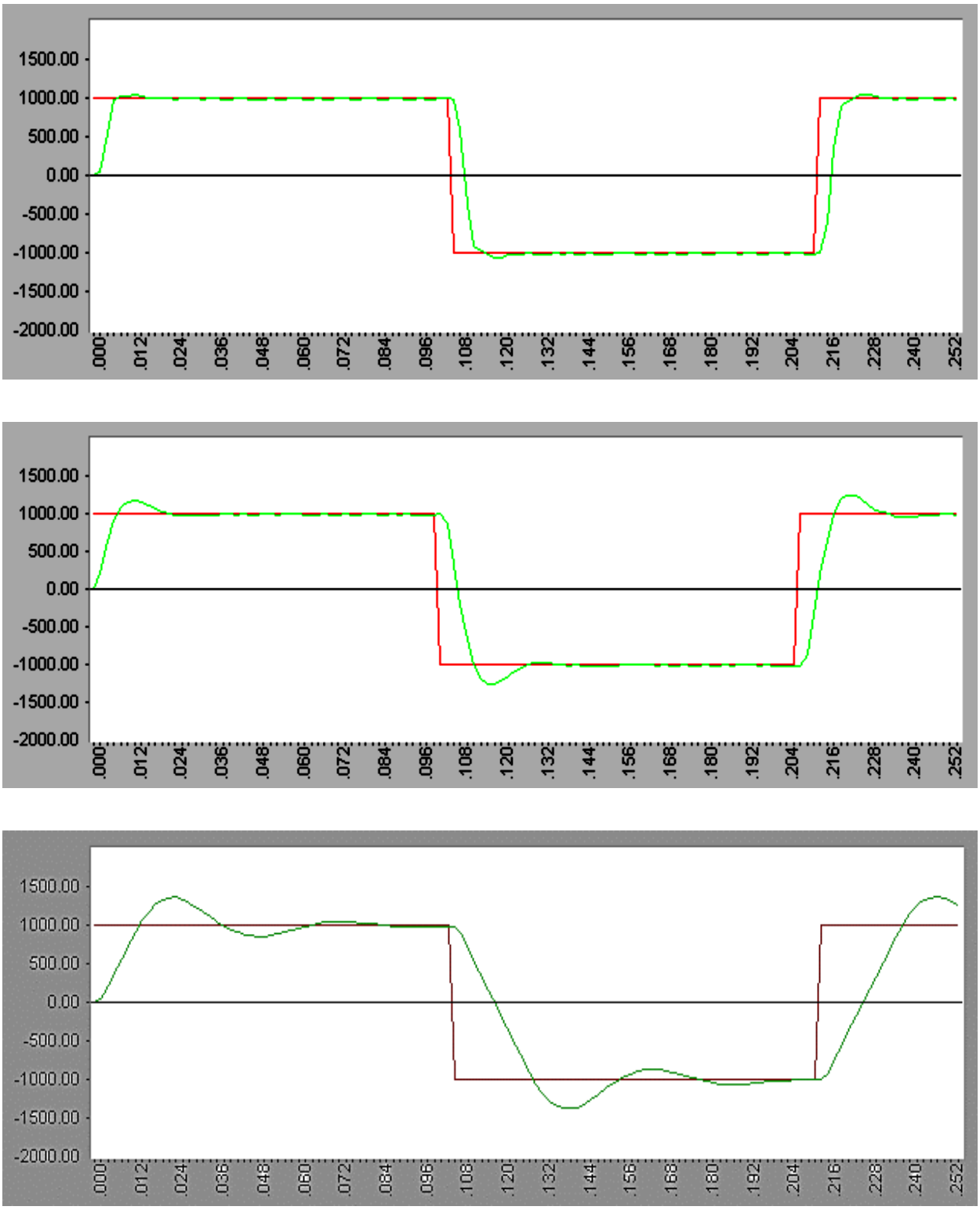


Fig 3.19: A servo control performance with, (a) Jloud/Jmotor = 0, (b) Jloud/Jmotor = 1, and (c) Jloud/Jmotor = 5 (taken from [102])

Moreover, not only do torque requirements seem essential in choosing such mechanical design parameters, but also other very vigorous factors such as moment of inertia, gear ratio calculation (if any), and others. Fig 3.19 (a)-(c) above shows the significant impact of the load to actuator inertia ratio, Jload/Jmotor, and mismatch on control performance as studied and tested in [102]. The above figure clearly shows the significant effect of designing the plant beam inertia with respect to the chosen servo actuator's one. In other words, the figure demonstrates a result of a certain study performed by one famous servo motor manufacturer, and the graphs above correspond to exactly the same actuator with a given motor inertia, Jmotor, and the same control system tuning (stabilized), but the altered variable was the loud inertia, Jload, which represents mainly the beam's, in this case. It can be obtained from the above study that the higher the inertia ratio, the lower the control performance, hypothetically. However, due to numerous facts and causes, an exact match, that is a ratio of 1:1 or less, might not be feasible in practice. Therefore, servo motor manufactures, such as the one that appears in [102], suggest maintaining a ratio within a range of 1:1 to 10:1 which is deemed acceptable.

The same rule should apply to other motion control actuators such as stepper motors, but exact numbers may vary from case to another. As a summary, this research paper considers more detailed design facts of the mechanical/mechatronics phase of the ball on beam. Not only are torque and speed requirements to be calculated in the mechanical and mechatronics design section of this chapter, but also some another important factors as mass moment of inertia ratio of the load (beam) to the actuator (motor). This estimation of torque and inertia requirements of the systems implies taking care of other following parameters and factors, i.e. motor sizing, mass, dimensions, materials selection, gear ratio calculations, efficiency, coupling slack, gear backlash, etc.

3.1.2.6. On the signal conditioning

Another critical part of any control system instrumented is the signal conditioning as shown previously in Chapter 2. This part of the loop seems almost ignored, or less paid attention to, in most other literature utilizing the ball on beam except for a few due to several causes such as using a pre-manufactured plant, although the associated direct effect of the quality and validity of measurement feedbacks on control performance. As seen in Chapter 2, if inaccurate, very small, and noisy signals are fed to the control system, an unstable, less precise, or over damping behavior may occur. Chapter 2 has presented major signal conditioning techniques and principles for process control and signal processioning including attenuation (filtration), amplification (scaling), and ADC using RC/RLC circuits, Op Amps, and others.

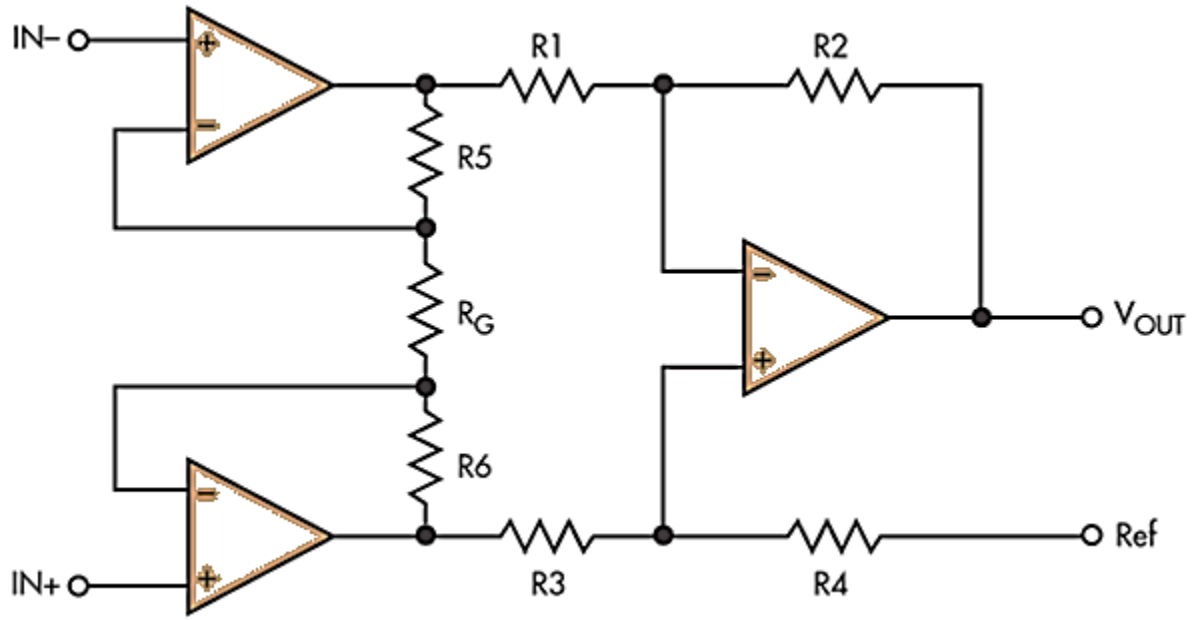


Fig 3.20: Common INA circuit configuration diagram

Similarly, this project exploits the advantage of using INA. For example, the DA, introduced in Chapter 2, is one smart technique reducing noise associated with measurement feedback signals. The INA, on the other hand, shown in Fig 3.20, has the following output voltage using basic circuit analysis:

$$V_{out} = G\Delta V_{in} = (1 + 2R_5/R_G)(R_2/R_3) \times (V_2 - V_1) \quad (90)$$

Where V_2 and V_1 are the two input voltages, G, the is amplification gain, and R_G is the designed regulated desired gain resistor. INA thus is even more beneficial in signal conditioning and instrumentation and includes the following features [55], [23]:

- Very high CMR
- Relatively high input impedance
- Low output impedance
- Lower time and thermal drifting
- Less power consumption
- As can be extracted from equation above (90), gain tuning is much simpler in the case of INA which requires manipulating one design resistance only, R_G
- Therefore, INA is more accurate and less noisy, which satisfies high performance, quality control, and harsh industrial environments requirements

The design approach of this chapter takes into consideration, and adds as contribution, a useful real analog domain signal conditioning circuitry primarily for sensors feedbacks of the ball on beam using the aid of INA to adapt those signals to be compatible with the ADC AIs of the RT controller used in the experiment, cRIO, besides the first smoothing LPF stage. Also, velocity and acceleration can be measured using Op Amp principles as illustrated in Chapter 2.

3.2. System Modeling and Simulation

Any controlled system plant's dynamics should be theoretically modelled and simulated before being implemented. This stage helps to better understand the system, predict possible future behaviors, save reconstruction and engineering time and cost, and may result in better control

system design parameters. As follows, the ball on beam system process plant mathematical model is extracted and formulated, and then identified, verified, validated, and computer simulated using LabVIEW as well as Matlab control systems simulation tool. Fig 3.21 shows an abstract schematic of the system using manipulation of the beam from the middle method (please refer to section 3.3.3, Mechanical and mechatronics design) and including some key variables and parameters of the system such as the beam length and tilt angle, ball position, and the dynamic motor actuator.

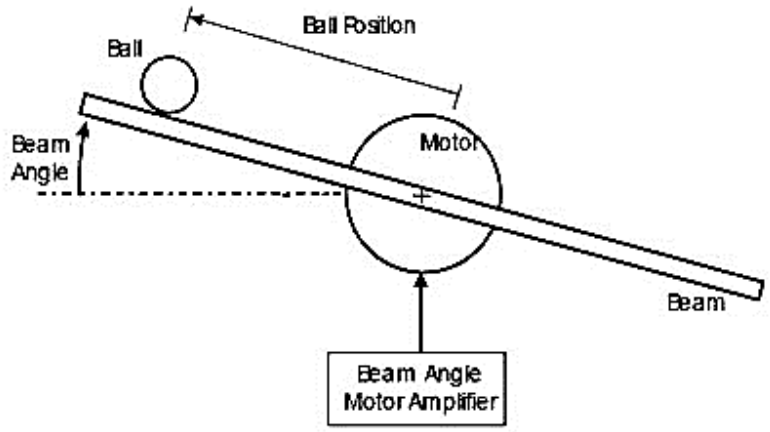


Fig 3.21: The idea and the purpose of the ball on beam process plant (taken from [51])

3.2.1. Theoretical Analysis and Mathematical Model

As per most control systems out there, RT dynamics fit deterministic LCTI models. In spite of the nonlinearity of most real life systems, as well as the ball on beam, linearization techniques can be applied to interpret those dynamics as LTI ones, which are easier to deal with and compatible with many famous controllers blocks, i.e. the PID and the optimal SF LQR. Similarly, this section illustrates detailed derivation of the ball on beam NL model from modeling basic laws, differential equations, and linear algebra. Then a standard linearized MIMO SS LCTI model of the system in addition to its SISO TFs equivalent are in the next subsection. Furthermore, as

the implemented ball on beam is controlled by digital RT or FPGA controllers, it is convenient to extract the discreet time system from the derived LCTI one as follows in the next subsection. The following defines a clear set of the model parameters, variables, and assumptions.

3.2.1.1. Assumptions

Below is the set of main assumptions made prior to constructing the theoretical model:

- The angle of the beam is the same as the output angle of the geared motor (actuator).
- The ball does not slip on beam while rolling and there exists a very small friction effect.
- The mechanical construction of gearbox and the coupling connecting the motor and the beam are ideal such that they exhibit no backlash and slack effects.
- The continuous measurement feedback data is precise and noiseless
- The RT PV values experience no delay (dead) time due to process or communication delay.
- The base of the system is stable and does not vibrate.
- The actuator (gear motor) sizing is selected such that the torque, speed, and inertia, as well as internal feedback precision and update rate requirements are optimum.

As may be noticed, most stated assumptions above do not reflect the actual RT setup that includes non-ideal effects. However, for the sake of modeling simplicity and understanding the underlying dynamics and mathematics, those assumptions should be made, knowing that those effects may tend to be minor even in the RT model upon well construction of the mechanical system and noise and interference clear testing experiment environment.

3.2.1.2. System parameters and variables

Similar to all system modeling problems, a clear set of all the system's parameters, variables, their symbols, value ranges or span, along with their description should be defined as presented in Table 3.2 below.

Parameter / Variable	Name	Engineering Unit	Value/Range	Notes and Description
$x(t)$	Ball position	m	(-0.25-0.25 m)	Designed
$\Theta(t)$	The beam's angular position	rad	(-6π to 6π rad)	Defined for linearity range, and continuously manipulated as an output to motor (beam angle)
Θ_m	The Motor's angle output	rad	(-π to π)	Related to beam angle by the gear ratio
N	Gear Ratio	Constant	65.5	Fixed, given in [114], the selected the DC geared motor's datasheet.
M_{load}	Mass of the load	kg	0.5	(See section 4.3)
M_{ball}	Mass of ball	kg	0.02272	Measured by an electronic laboratory weight measurement device
g	Acceleration due to gravity	m /s^2	9.81	Fixed standard
L_{beam}	Length of the beam	m	0.625	Designed and measured
R_b	Radius of the ball	m	0.015	Measured by a ruler
b	Damping constant of the motor	Nm/(rad/s)	0.053ozin/krpm ×0.00955 = 5.061× $(10)^{-3}$	Given in the motor datasheet (appendix) and converted as per [20]
$\dot{\theta}$(t)	Angular velocity of the beam	Rad/s	d θ/dt	Continuously programmed and S/W estimated.

$\dot{X(t)}$	The linear velocity of the ball	m/s	v = d X/dt	Can be continuously measured in RT using Op Amp differentiation of measured ball position or by S/W counting estimation
$\ddot{\theta}$(t)	The angular acceleration of the beam	Rad/s^2	$\dot{\omega} = \mathrm{d}\dot{\theta}/dt$	Not measured or estimated in RT, but just used in modeling as a state
d	The vertical distance between the center of the ball and the contact point between the ball and the beam	m	0.01	It is the rotational radius of the ball measured from the actual setup
$\ddot{X}$(t)	Linear acceleration of the ball	m/s^2	a = d X/dt	Another state in the model
R	The resistor of the DC motor	Ω	2.49	Given (See [114], DC motor parameters and datasheet)
K	Force Constant of the DC motor	Nm/A	6.49 oz-in/A × 7.06155 × $(10)^{-3}$ = 0.04583 Nm/A	Given and converted (See [114], datasheet and Appendix, common DC motors' parameters' units and conversions).
L	Electrical inductance of the DC motor	H	2.6 ×$(10)^{-3}$	Given in [114].
K_e	Motor constant related to back EMF voltage	V/(rad/s)	4.8 V/krpm × 9.5493 × $(10)^{-3}$=0.0458 4 V/rad/s	Given and converted (See [114] and Appendix 4, common DC motors' parameters' units and conversions.

b_1	Friction constant	Ns/m	0.0227	Estimated later in section 3.2
J	Mass Moment of inertia of the system	Kg. m^2	0.016765	The sum of the load inertia and motor inertia
J_m	Mass Moment of inertia of the motor	Kg. m^2	$7.062 \times (10)^{-6}$	Given and converted See Appendix 3, DC motor parameters)
J_{load}	Mass Moment of inertia of the load	Kg. m^2	0.01	Calculated (the sum of the beam's and ball's inertia see section 4.3)
J_{ball}	Mass Moment of inertia of the ball	Kg. m^2	$2 \times (10)^{-6}$	Referred and calculated (See equation 94 and section 4.3)
$\propto$	Rotational angle of the ball	Rad	changing	See Fig 3.22.
V(t)	Control voltage	V	-10 or 10 V (for FPGA mode) and -10 to 10 to/0 or 5 for RT scan mode	The DAC AO of the controller. An input to the motor driver and specified in NI's doc [19]
E(t)	DC motor's EMF	V	-24 to 24V	$GV(t) - EMF_{back}$
G	Driver's amplification gain	Unit-less	2.4	Calculated as per the output of the controller and the motor driver power requirements
$I(t)$	Current flowing into the motor	A	0 to 2.1 A	Continuously Measured (FPGA) as controlled variable in FPGA mode
$\dot{I}$(t)	Rate of change of motor's current	A/s	dI/dt	One of the states of the system

Table 3.2: The ball on beam dynamic system parameters and variables

3.2.1.3. Continuous time MIMO model

The ball on beam continuous time model of the system can be extracted from, and classified into, three main relations: the vertical movement of the ball on the beam, the beam and motor dynamics, and lastly the driving actuation (electrical part of the motor) dynamics. Regardless of which type of motor is selected, the first two concepts remain the same. Contrary, the actuator dynamics may vary according the mechanical and dynamical characteristics of type of driving realized, either DC servo or stepping applications.

Starting from the first relation, the ball movement on the beam follows the law of motion, but before that the position of the ball is represented by the equation [51]:

$$x = \propto d \tag{91}$$

Then applying D Albert law introduced earlier, the force balance of the ball on the beam is

$$\Sigma F = 0 = M_{ball} x\ddot{(t)} + b_1 x\dot{(t)} - M_{ball}\, g \sin\theta(t) + F_b + F_d \tag{92}$$

Where F_b is the internal applied force on the ball and F_d is considered as an external disturbance input in the experiment. Another important force balancing related to the ball is the angular torque one. It is known as the torque balance of the ball given by the equation [51]:

$$\Sigma\tau_b = 0 = dF_b = J_{ball}\, \ddot{\propto} \tag{93}$$

Where $\ddot{\propto}$ = the rotational acceleration of the ball and J_{ball} is nothing but the mass moment of inertia of a spherical body defined in the table above and calculated according to the following:

$$J_{ball} = {}^{2}/_{5}\, M_{ball}\, {R_b}^2 \tag{94}$$

As a result of substituting (93) and (94) in (92) and assuming $F_d = 0$ yield the following state variable differential equation relating the ball dynamic on the beam:

$$\left[1 + {}^{2}/_{5}\left(\frac{Rb}{d}\right)^2\right] X\ddot{(t)} + \frac{b_1 X\dot{(t)}}{M_{ball}} = g \sin\theta\,(t) \tag{95}$$

Fig 3.22 below illustrates a digram ball on beam concpet modeling explained so far.

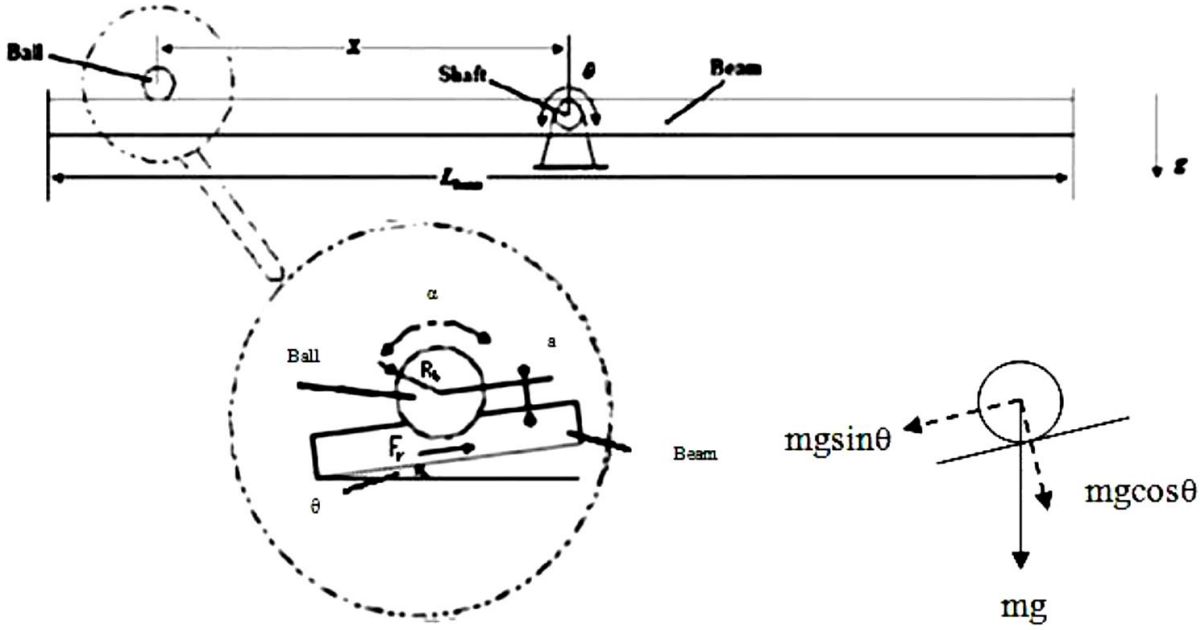


Fig 3.22: The force balance of the horizontal relation of the ball on beam model (taken from [51])

The second main dynamic relation associated with the ball on beam is the motor and beam. To derive such a relation one should understand the physics of attaching a rotating beam to a motor shaft. It follows then that both torque and moment of inertia are important. A typical model of the DC motor has been covered from Chapter 2. However, the beam and other associated mechanicals added other loads. As per modeling basics, the balance torque around the motor is:

$$\tau_{motor} = 0 = KI(t) - J_m\ddot{\theta_m}(t) - b\dot{\theta_m}(t) \tag{96}$$

$J_m$is defined above in Table 3.2 as the moment of inertia of the rotor given as one of its manufactured specifications; the moment of inertia of the ball and beam besides the motor together is simply the sum of all (noting the effect of gear ratio on output inertia of the motor):

$$J = J_{load} + J_m \times N \tag{97}$$

Where $J_{load}$is described and defined in the table and also calculated in Section 4.3, Mechanical and Mechatronics Design. Keeping in mind the motor selected from Section 4.3 (Mechanical/Mechatronics Design) is a geared type of a DC brushed motor; therefore, as per (17)

$$N = \frac{\theta_m}{\theta} = \frac{\omega_m}{\omega}$$

As a result, $\theta = \frac{1}{N}\theta_m$, $\dot{\theta} = \frac{1}{N}\dot{\theta}_m$ and $\ddot{\theta} = \frac{1}{N}\ddot{\theta}_m$. Then (96) can be rewritten as:

$$\tau_{motor} = 0 = KI(t) - JN\ddot{\theta}(t) - bN\dot{\theta}(t) \tag{98}$$

Similarly, the torque of the ball is given by [51]:

$$\tau_{ball} = -X(t)M_{ball}g\cos\theta(t) \tag{99}$$

As a result, the overall torque of the rotating beam is the sum of both,

$$\tau_{beam} = \tau_{motor} + \tau_{ball} \tag{100}$$

Substituting and manipulating (98), (99), and (100) in (96), the following state variable relating the angular acceleration and velocity states to the angular position of the beam yields:

But, we are still interested in θ and $\dot{\theta}$. Then, (100) can be rewritten in state variable form as:

$$\ddot{\theta}(t) = \frac{KI(t) - X(t)M_{ball}g\cos\theta(t) - bN\dot{\theta}(t)}{JN} \tag{101}$$

The third important relation is the electrical part of the motor and driving actuator dynamics. In case a DC motor is realized, it is an LCTI system model that is presented in Chapter 2. The following differential equations can be obtained:

$$L\frac{dI(t)}{dt} + RI(t) = GV(t) - Ke\dot{\theta_m}(t) \tag{102}$$

As a result of considering the gear reduction ratio, the following state varaiable eqation is derived:

$$L\dot{I}(t) = GV(t) - RI(t) - KNe\dot{\theta}(t) \tag{103}$$

To summarize the main five state variables relations of the ball on beam continuous time system, the following equations should be always remembered from which the upcoming NL continuous time invariant SS model develops. The five state variables are the ball position, the beam tilt angle, the ball velocity, angular speed, and electrical current flowing into the motor such as the following:

$$\dot{x} = v \tag{104}$$

$$\ddot{x}(t) = \frac{gsin\theta(t) - \frac{b_1 \dot{x}(t)}{M_{ball}}}{\left[1 + \frac{2}{5} \times \left(\frac{Rb}{d}\right)^2\right]} \tag{105}$$

$$\dot{\theta} = \omega \tag{106}$$

$$\ddot{\theta}(t) = \frac{KI(t) - X(t) M_{ball} g \cos\theta(t) - bN\dot{\theta}(t)}{JN} \tag{107}$$

$$I(\dot{t}) = \frac{GV(t) - RI(t) - KeN\dot{\theta}(t)}{L} \tag{108}$$

As a result, the initial and general five states NL SS model of the ball on beam continuous time dynamics stands the following:

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{I} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{b_1}{\left[1 + {}^2/_5\left(\frac{R_b}{d}\right)^2\right] M_{ball}} & \frac{g}{1 + {}^2/_5\left(\frac{Rb}{d}\right)^2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{-M_{ball} g}{JN} & 0 & 0 & \frac{-b}{J} & \frac{K}{JN} \\ 0 & 0 & 0 & \frac{-NKe}{L} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} X\cos\theta \\ \dot{X} \\ sin\theta \\ \dot{\theta} \\ I \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{G}{L} & 0 \end{bmatrix} [V(t) \quad F_d] \tag{109}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} X(t) \\ \theta(t) \end{matrix}$$

Where the two outputs are the ball position, and the beam angle and the two inputs are force driven from the current to the motor and the disturbance force applied to move the ball on the beam, which has been initially assumed to be zero.

3.2.1.4. Linearization

The above SS is NL due to the nonlinearity components found in the terms $Xcos\ \theta$ and $sin\theta$ that marks the system's dynamics superposition principle incompatible, as introduced in the introduction. The ball on beam system could be approximated to be linear based on the assumption as well as the requirement of maintaining the beam angle very small such that $\theta \approx 0$, so the nonlinear components $X\cos\theta \approx X$ and $sin\theta \approx \theta$. Hence, the following LCTI SS yields:

$$\begin{bmatrix} \dot{X}(t) \\ \ddot{X}(t) \\ \dot{\theta}(t) \\ \ddot{\theta}(t) \\ \dot{I}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{g}{1+{}^{2}/_{5}\left(\frac{Rb}{d}\right)^{2}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{-M_{ball}g}{JN} & 0 & 0 & \frac{-b}{J} & \frac{K}{JN} \\ 0 & 0 & 0 & \frac{-NK_e}{L} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} X(t) \\ \dot{X}(t) \\ \theta(t) \\ \dot{\theta}(t) \\ I(t) \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{G}{L} & 0 \end{bmatrix} [V(t) \quad F_d] \quad (110)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} X(t) \\ \theta(t) \end{matrix}$$

This SS model is based on the assumption that there is no friction as the ball moves on the beam. From this experiment and previous literature, the maximum range of θ that keeps the system approximately linear is known to be within the limits of -6π to 6π rad as shown in Table 3.2.

Placing several equivalent SISO TF in series can represent the above SS model in Laplace and BD form, but before that, the linear forms of the state variable equations in (105), (107), and (108) are:

$$\left[1 + \frac{2}{5} \times \left(\frac{Rb}{d}\right)^2\right] \ddot{X(t)} + \frac{b_1 \dot{x(t)}}{M_{ball}} = g\theta(t)$$

$$\ddot{\theta(t)} = \frac{KI(t) - M_{ball} g X(t) - bN\dot{\theta(t)}}{JN}$$

$$\dot{I(t)} = \frac{GV(t) - RI(t) - K_e N\dot{\theta(t)}}{L}$$

Taking Laplace transform to above three equations produces the relations in the S domain,

$$\left[1 + \tfrac{2}{5} \times \left(\tfrac{Rb}{d}\right)^2\right] s^2 X(s) + \tfrac{b_1 sX(s)}{M_{ball}} = \mathrm{g}\theta(\mathrm{s}) \quad (111)$$

$$s^2\theta(s) = \frac{KI(s) - M_{ball} g X(s) - bNs\theta(s)}{JN} \quad (112)$$

$$sI(s) = \frac{GV(s) - RI(s) - KeNS\theta(s)}{L} \quad (113)$$

3.2.1.5. Derivation and Analysis

It is desired to analyze the system using TF and SISO approach and distinguish between the actuator and the process plant dynamics of the ball on beam. For the sake of the anticipated comparison study between stepping and DC brushed servo actuation on the overall higher order control performance besides matching the industrial as well as designed control strategy, the above LTI SS model can be represented as two TF blocks connected in series. As per the standard feedback BD in Fig 2.56 of Chapter 2, the actuator TF block is the one attached to the controller output (MV). Since this chapter applies the conventional (DC brushed motor) actuator, the well-known TF is nothing but equation (79), given in Chapter 2, from where equations (112) and (113) are derived. Now this relation recalls the desired *I(s)/E(s)* TF as a first stage in the actuation method using DC motor as appears in the figure. Given the EMF inside the motor is the voltage difference between the drivers' amplified input and back voltage,

$$E(t) = GV(t) - EMF_{back} \tag{114}$$

Hence,

$$E(t) = GV(t) - K_e\dot{\theta_m}(t) \tag{115}$$

So the $\frac{I(s)}{\mathrm{E(s)}}$ TF can be obtained from (113) as:

$$\frac{I(s)}{\mathrm{E(s)}} = \frac{1}{Ls+R} \tag{116}$$

Similarly, to formulate $\frac{\theta(s)}{I(s)}$ requires manipulating (112). The resulting DC motor TF is

$$\frac{\theta_m(\mathrm{s})}{I(\mathrm{s})} = \frac{K - M_{ball}gX(s)}{\mathrm{J}s^2+\mathrm{b}s} \tag{117}$$

With assuming a very small ball weight, the term $M_{ball}gX(s)$ can be ignored or eliminated, so

$$\frac{\theta(\mathrm{s})}{I(\mathrm{s})} = \frac{K}{(\mathrm{J}s^2+\mathrm{b}s)N} \tag{118}$$

Now since the output of the actuator is $\theta(s)$, which in turn composes the input to the plant in which the output is $X(s)$, the plant's TF can be extracted by solving (111) with a similar manner. After solving for $\frac{X(s)}{\theta(s)}$, the resulting plant's TF appears to be

$$\frac{X(s)}{\theta(s)} = \frac{g}{\left[1+\frac{2}{5}\left(\frac{Rb}{d}\right)^2\right]s^2+\frac{b_1}{M_{ball}}s} \tag{119}$$

Recall that this TF of the plant is fixed while the above one of the actuator varies according to the situation. Fig 3.23 (a) and (b) below shows the driven Laplace Transform TFs FB SISO model considering inputs coming from the controller as PWM voltage signals and the ultimate controlled output *X* as the ball position on the beam. Note that the driver amplifier gain, *G*, should take place in the model. The driver block arrives after the ADC AO (of +/-10V). Hence, *G* is calculated as per the power requirement of the motor and driver and will be presented in the Mechanical and Mechatronics Design, Section 4.3.

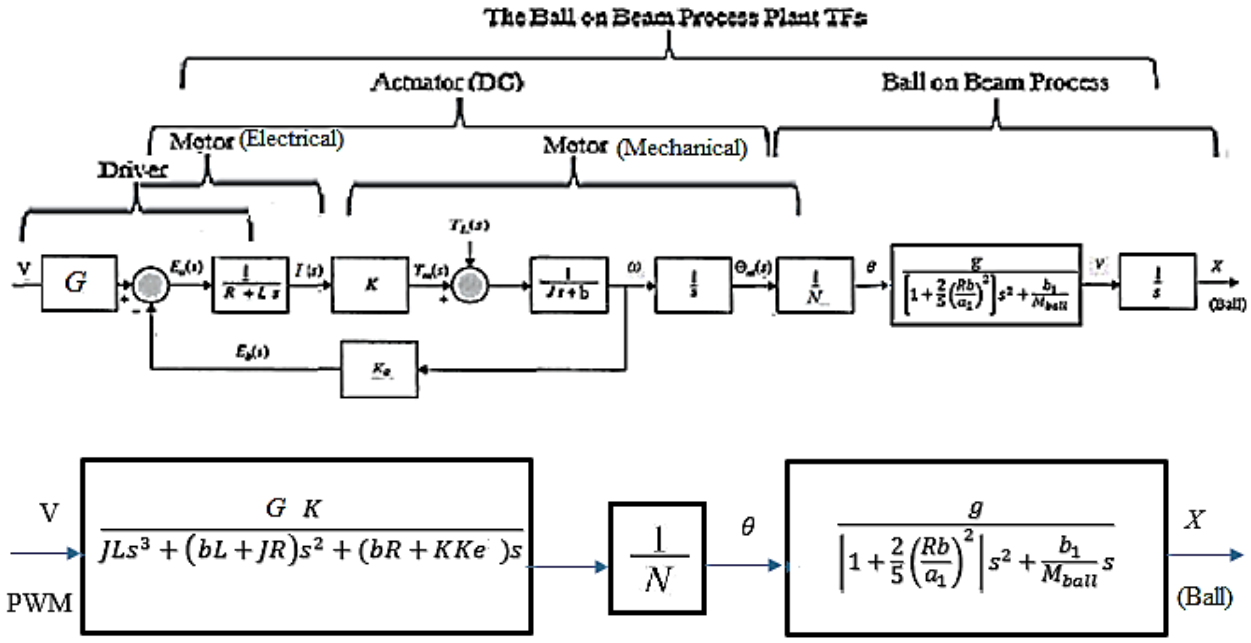


Fig 3.23: (a) The detailed ball on beam TFs process plant open loop system using DC motor actuation and (b) the equivalent two TFs BD of the DC motor derived ball on beam plant.

3.2.1.5. Discretization (digitization)

Converting the plant's dynamics into a LDTI complies with the given fact that the controllers implemented in this experiment are digital. Thus, to obtain the approximated digital equivalence of the previously derived MIMO SS model, first let denote $X(t)$ = X_1, $\dot{X(t)}$ = X_2, $\theta(t)$ = X_3, $\dot{\theta}(t)$ = X_4, and X_5 = $I(t)$ respectively, then the discrete time ball on beam process plant arises as the following using Taylor series expansion presented in Chapter 2:

$$\begin{bmatrix} X1[k+1] \\ X2[k+1] \\ X3[k+1] \\ X4[k+1] \\ X5[k+1] \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{g}{1+{}^{2}/_{5}\left(\frac{Rb}{d}\right)^{2}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{-M_{ball}g}{JN} & 0 & 0 & \frac{-b}{J} & \frac{K}{JN} \\ 0 & 0 & 0 & \frac{-NKe}{L} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} X1[k] \\ X2[k] \\ X3[k] \\ X4[k] \\ X5[k] \end{bmatrix} \Delta T + \begin{bmatrix} 0 & F_d \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{GV[k]}{L} & 0 \end{bmatrix} \Delta T \quad (120)$$

$$Y[k] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} X1[k] \\ X3[k] \end{matrix}$$

Also, the discrete time equivalence of the above LCTI MIMO SS model can be derived after several ways as shown in the introduction before one famous method is the $z = e^{sT}$. Both discrete time systems depend on ΔT defined in Chapter 2. [109] (NI 9215 module) conveys the ADC AI used in the implementation where the sampling rate is given by 100 kilo-sample/seconds/channel. Thus, *fs* = 100 kHz; therefore, ∆ T = 1/100k = 0.00001 seconds. From Fig 3.23 above, let *A(s)* = $\frac{\theta(s)}{V(s)}$ TF and $\frac{X(s)}{\theta(s)}$ = *P(s).* Substituting the values in the table results in

$$A(s) = \frac{(0.04583)(2.4)}{[0.0000436s^3 + 0.042s^2 + 0.0147s]65.5}$$

Thus,

$$A(s) = \frac{0.11}{65.5(s+0.349)(s+962.96)s}$$

Similarly,

$$P(s) = \frac{9.8}{1.9s^2}$$

Assuming no friction, then the approximated corresponding $A(z)$ and $P(z)$ can be obtained by converting all the poles to the z-plan as follows

$$-0.349 \longrightarrow e^{-0.349\Delta T} = 0.999997$$

$$-962.96 \longrightarrow e^{-962.96\Delta T} = 0.99$$

$$0 \longrightarrow e^{0} = 1$$

As a result,

$$A(z) \approx \frac{0.109992z^3}{65.5(z-0.999997)(z-0.99)(z-1)}$$

And

$$P(z) \approx \frac{z}{100000(z-1)(z-1)}$$

As per Z transform theory, since some of the poles of the above combined TFs' are outside unit circle, it is clear then that the system is unstable. The LDTI SS model above definitely tells more information about the internal dynamics of the system. However, SS models involve a defined set of initial conditions for the simulation to start with, and specifically in the ball on beam, it is hard to guess the velocity and acceleration.

3.2.2. Computer Simulation and Model Verification and Identification

After modeling, is simulation, which helps to better understand the time domain response of the dynamics of the system. Usually two sorts of simulation models are applied: open loop triggered response and the closed-loop one, which shows the expected or optimal feedback controlled behavior of the system. Through and between simulations, the theoretical constructed model can be verified and validated by employing tools or comparing stochastic or RT behavioral data of the system. Upon observing simulation results and comparing with the expected RT behavior, the previously constructed model can be modified

3.2.2.1. Open loop plant step response, parameter estimation, and model validation

Substituting the numerical values given and obtained in Table 3.2 into the initially derived LCTI SS model assuming no disturbance, i.e. Fd = 0 and no friction, b_1 = 0, gives

$$\begin{bmatrix} \dot{X}(t) \\ \ddot{X}(t) \\ \dot{\theta}(t) \\ \ddot{\theta}(t) \\ \dot{I}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 5.163 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -0.2 & 0 & 0 & -0.3 & 0.04 \\ 0 & 0 & 0 & -1154.8 & -957.69 \end{bmatrix} \begin{bmatrix} X(t) \\ \dot{X}(t) \\ \theta(t) \\ \dot{\theta}(t) \\ I(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 923 \end{bmatrix} V(t)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} X(t) \\ \theta(t) \end{matrix}$$

Fig 3.24 demonstrates the zero initial condition open loop step response of the whole dynamics of the model using Matlab simulation tool. The graph indicates the sensation of how the system may act if excited by 10V step amplitude with no control. Appendix 1 shows the code.

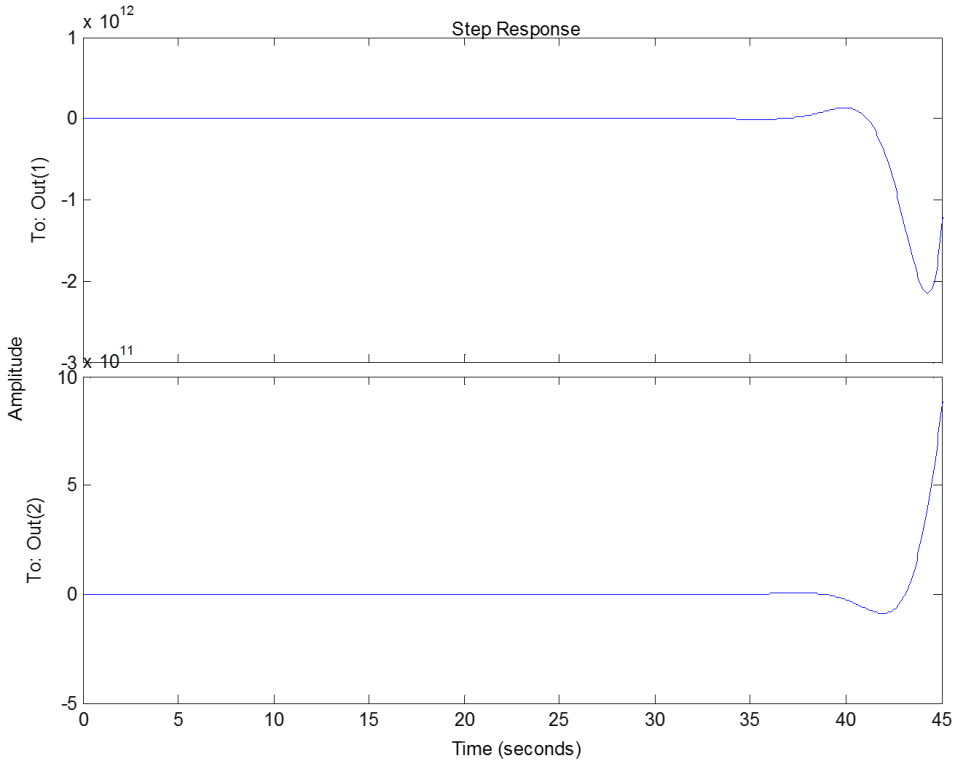


Fig 3.23: Zero initial conditions step response, (a) ball position (upper) and (b) beam angle

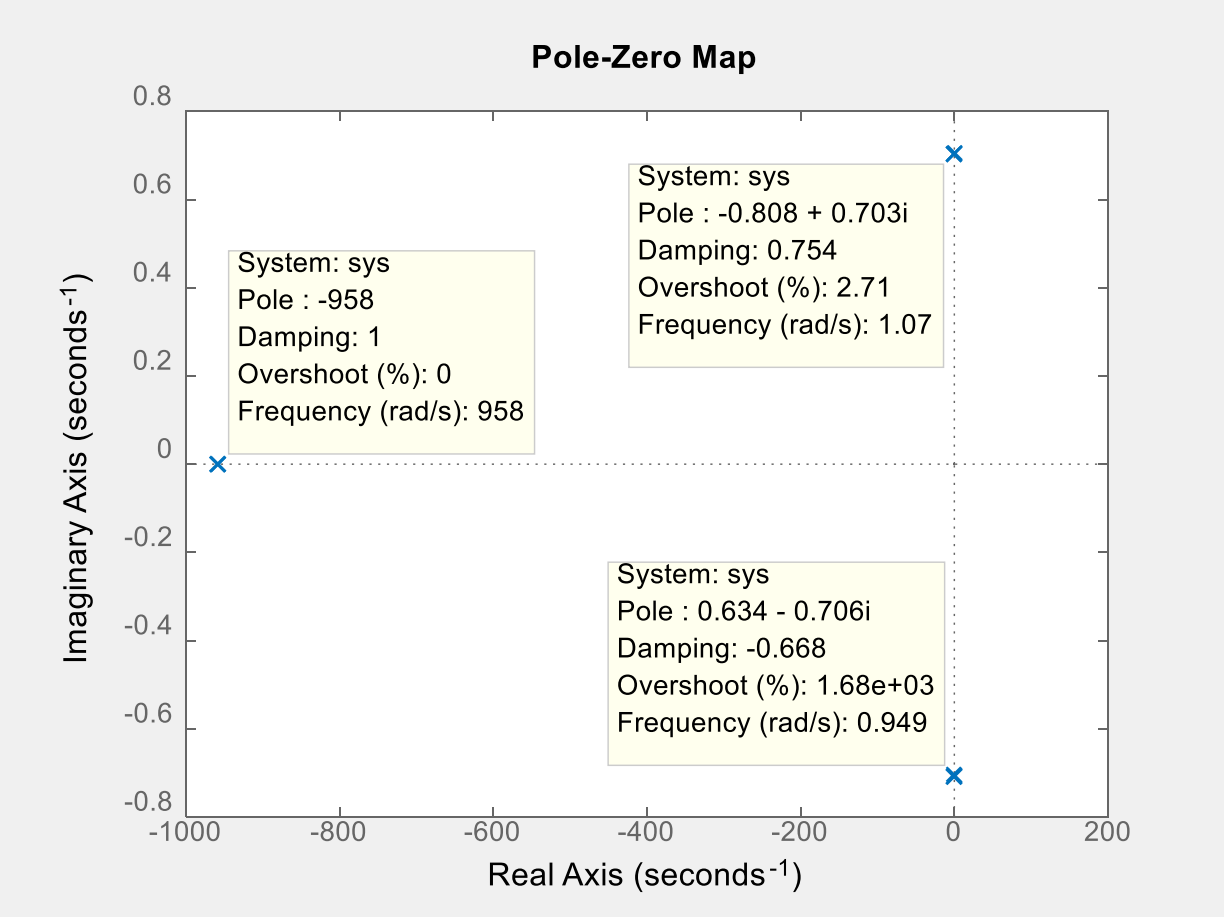


Fig 3.24: Poles-Zero plan of the unstable experimented ball on beam process plant model

Also, Fig 3.24 above displays the poles and zeros s-plan of the initially modeled ball on beam dynamics. The plan shows the five poles of the system and no zeros. As per Eigenvalues stability analysis described earlier in Chapter 2, the system is indeed unstable since two of the poles have their imaginary parts on the positive side of the s plan as appears in the figure. The resulting poles vector is

$$\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\\ \lambda_5\end{bmatrix} = \begin{bmatrix}-958\\ -0.808 + 0.724i\\ -0.808 - 0.724i\\ 0.634 + 0.706i\\ 0.634 - 0.706i\end{bmatrix}$$

Hence, the real parts of λ_4 and λ_5 indicate the instability according to (20).

3.2.2.2. Model reduction

Although the process of the ball moving on the beam appears valid, one can determine from the simulated response above that the overall SS dynamics of the system may be either unobservable or uncontrollable or both. This might be due to several systems' parameter being very small, that it is better to ignore them for the sake of a simpler, observable, controllable, and matching RT practice model. One parameter is the induction of the DC motor, which can be neglected as zero. As a result, the new actuator TF will reduce to

$$\frac{\theta(s)}{V(s)} = \frac{GK}{[JRs^2+(\mathrm{bR+KKe})s]N} \tag{121}$$

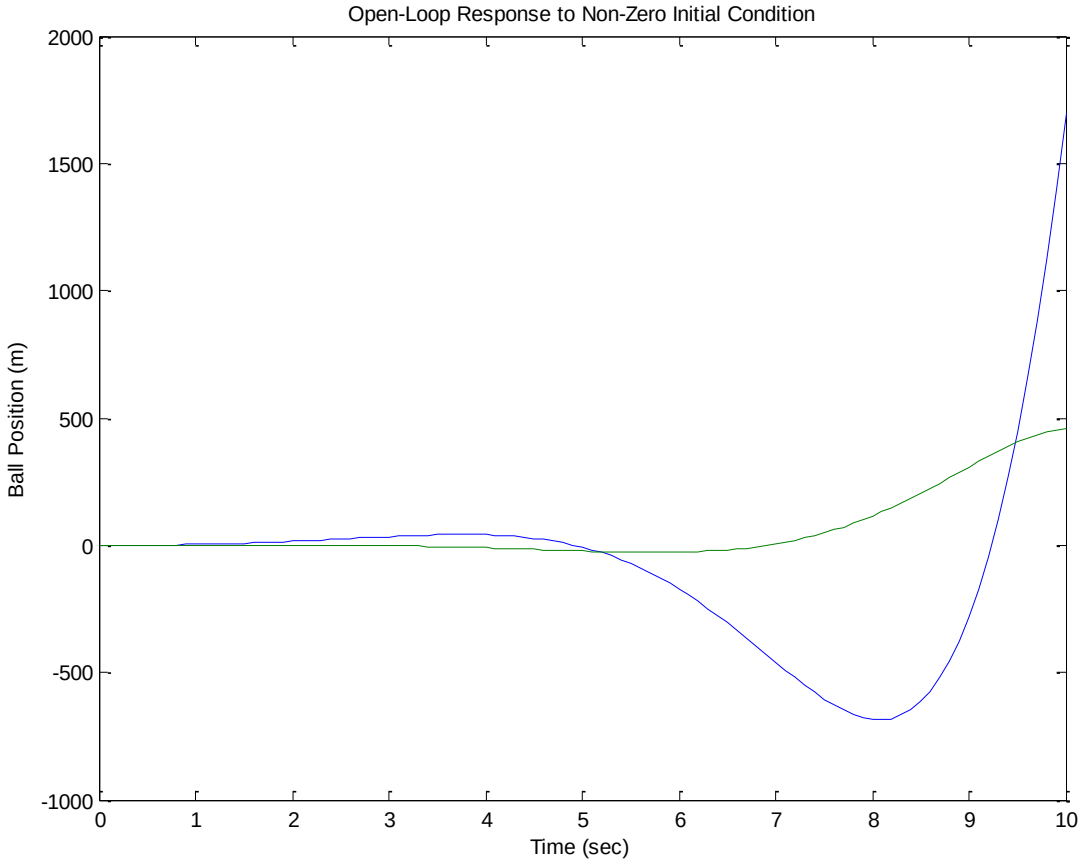


Fig 3.25: The non-zero initial conditions open loop step response of the modeled ball on beam

The unclear zero initial condition response above has triggered curiosity to test the controllability and observability of the system. But, before such investigation, different plant conditions are tested, and the non-zero initial conditions vector of $X_0 = [0\ 1\ 0\ 1\ 1]$ is selected. As a result, a closer to practical response emerges as in Fig 3.25 where the beam angle moves a little bit and the ball position ramps up. Recall the controllability and observability conditions introduced in Chapter 2. Applying the controllability test given in (7) on the modeled system using matrix operations in Matlab tool results in

$$Rank[B\ AB\ A^2B\ A^3B\ A^4B] = 1 \neq 5$$

Hence, the system is theoretically uncontrollable. The number of uncontrolled states is four. Similarly, for observability, apply (8):

$$Rank\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \end{bmatrix} = 1 \neq 5$$

Therefore, the system is also unobservable. As a result of reducing the number of states and (121), following modified SS system model representation gives:

$$\begin{bmatrix} \dot{X}(t) \\ \ddot{X}(t) \\ \dot{\theta}(t) \\ \ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{b_1}{\left[1+{}^2/_5\left(\frac{R_b}{d}\right)^2\right]M_{ball}} & \frac{g}{1+{}^2/_5\left(\frac{Rb}{d}\right)^2} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-M_{ballg}}{JN} & 0 & 0 & -\frac{\frac{KK_e}{R}+b}{J} \end{bmatrix} \begin{bmatrix} X(t) \\ \dot{X}(t) \\ \theta(t) \\ \dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \frac{KG}{RJN} & 0 \end{bmatrix} [V(t) \quad F_d] \quad (122)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} X(t) \\ \theta(t) \end{matrix}$$

After substituting the numerical values, the corresponding Z transform obtains

$$\frac{\theta(\mathrm{z})}{V(z)} \cdot \frac{X(\mathrm{z})}{\theta(\mathrm{z})} = \frac{0.00168z^2}{(z-1)(\mathrm{z}-.9999963)} \cdot \frac{0.00001z}{(z-1)^2} \quad (123)$$

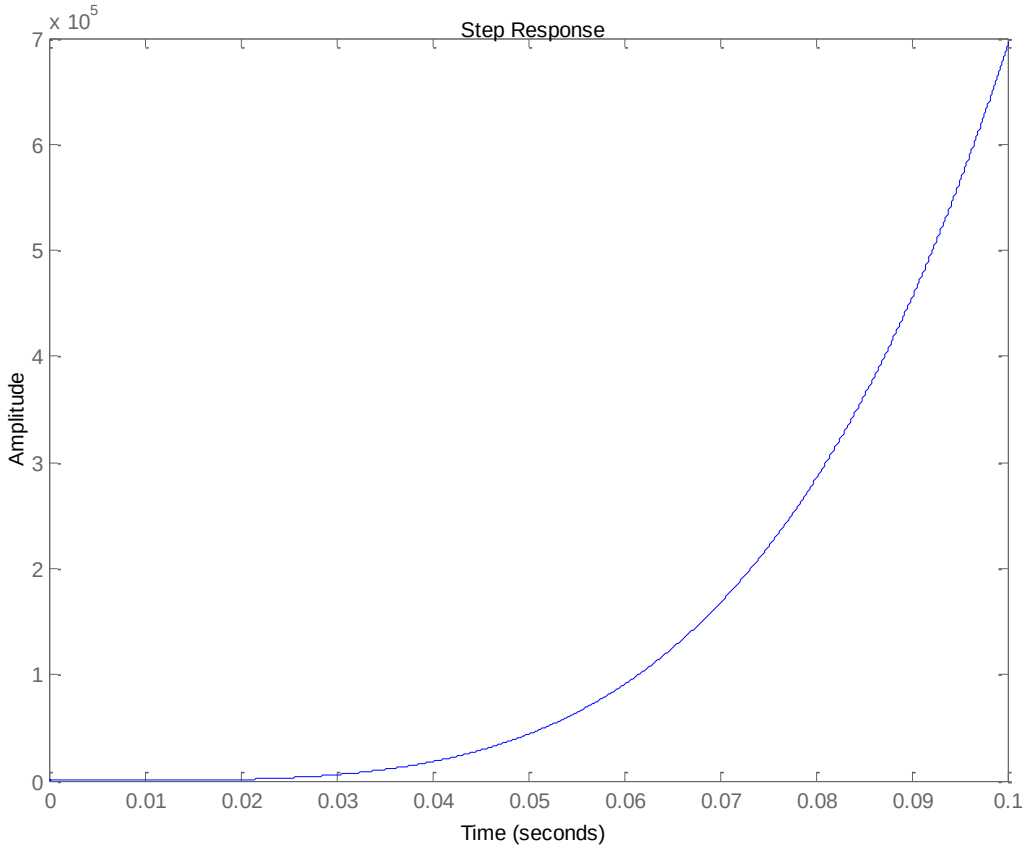


Fig 3.26: The open loop step response of the discrete time ball on beam model with f_s = 100 kHz

Fig 2.26 above displays the discrete time step input behavior of the reduced states modeled open loop ball on beam process when $\Delta T = 0.00001s$ and $b_1 = 0$. Hereby, the response matches the expected behavior of the experimental setup, which supposes a sudden ramp up in the ball position once the system is excited; however, the system is of course unstable and needs control. Not only does the system match the expected open loop experimental action, but also it is controllable as well as observable after going through the same test steps such that

$$Rank[B\ AB\ A^2B\ A^3B\ ] = 4$$

And

$$Rank\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = 4$$

3.2.2.3. Parameter estimation and model validation

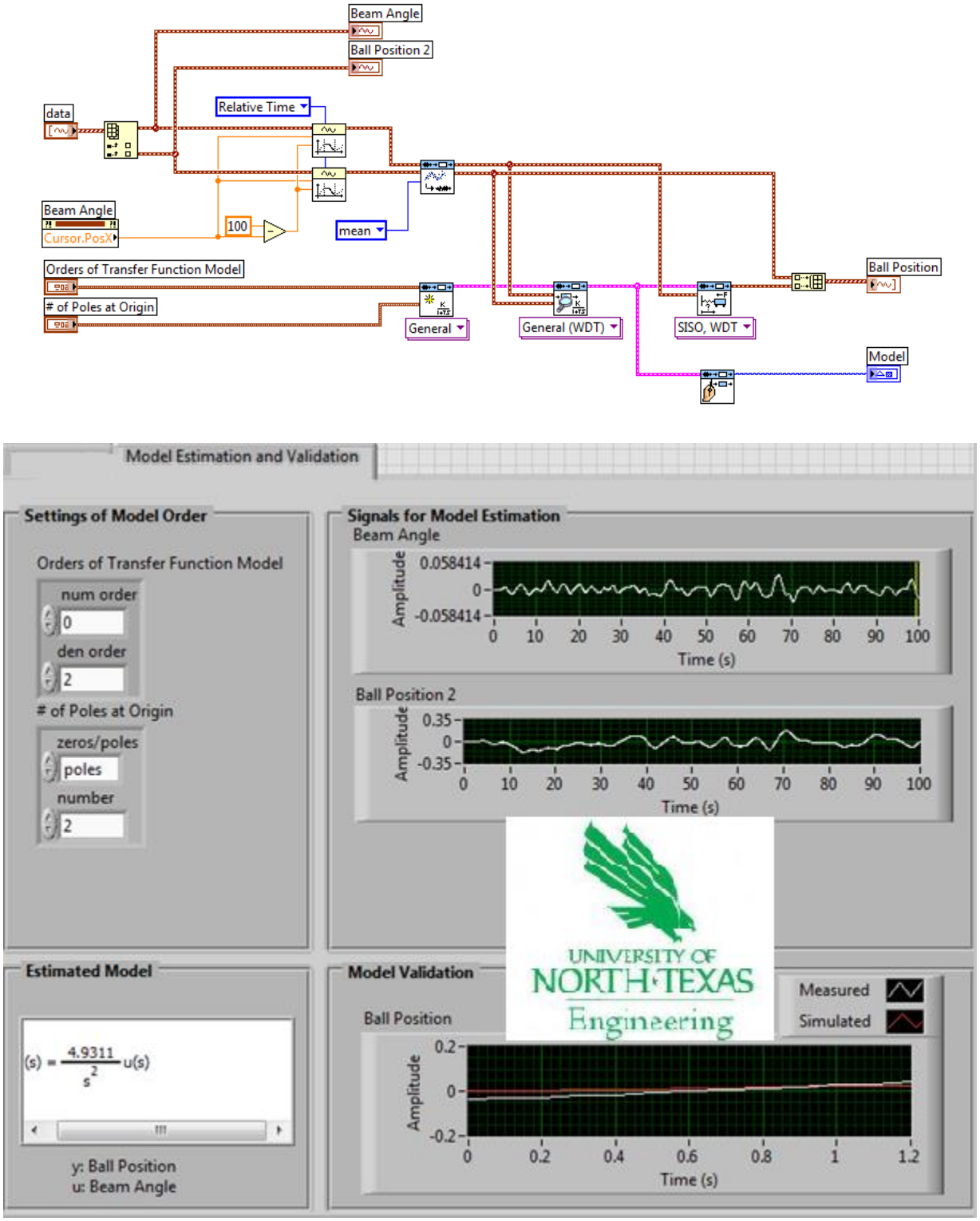


Fig 3.27: (a) The model estimation VI FBD and (b) the graphical front panel when $b_1 = 0$

Since most DC motors (as servo actuator models) are commonly found in the literature, it is rather desired to validate the previously derived theoretical SISO TF model of the ball on beam process through simulating typical RT or random data of a measured model and then estimating the approximated open loop plant. Specifically, the above LabVIEW simulation VI of Fig 3.27 (a) FB diagram and (b) front panel estimates the TF model of ball position plant PV. The program looks for X(t) as an output while considering the beam angle, $\theta(t)$, of the motor actuator as an input to the ball on the beam displacement open loop process. [18] gives thorough details, guidance, and examples on how the useful tool of LabVIEW FB and graphical interface are used in simulating theoretical control systems using both SISO TF and MIMO SS approaches. The cursor indicator in the beam angle data graph is configurable so that it slides along the time axis.

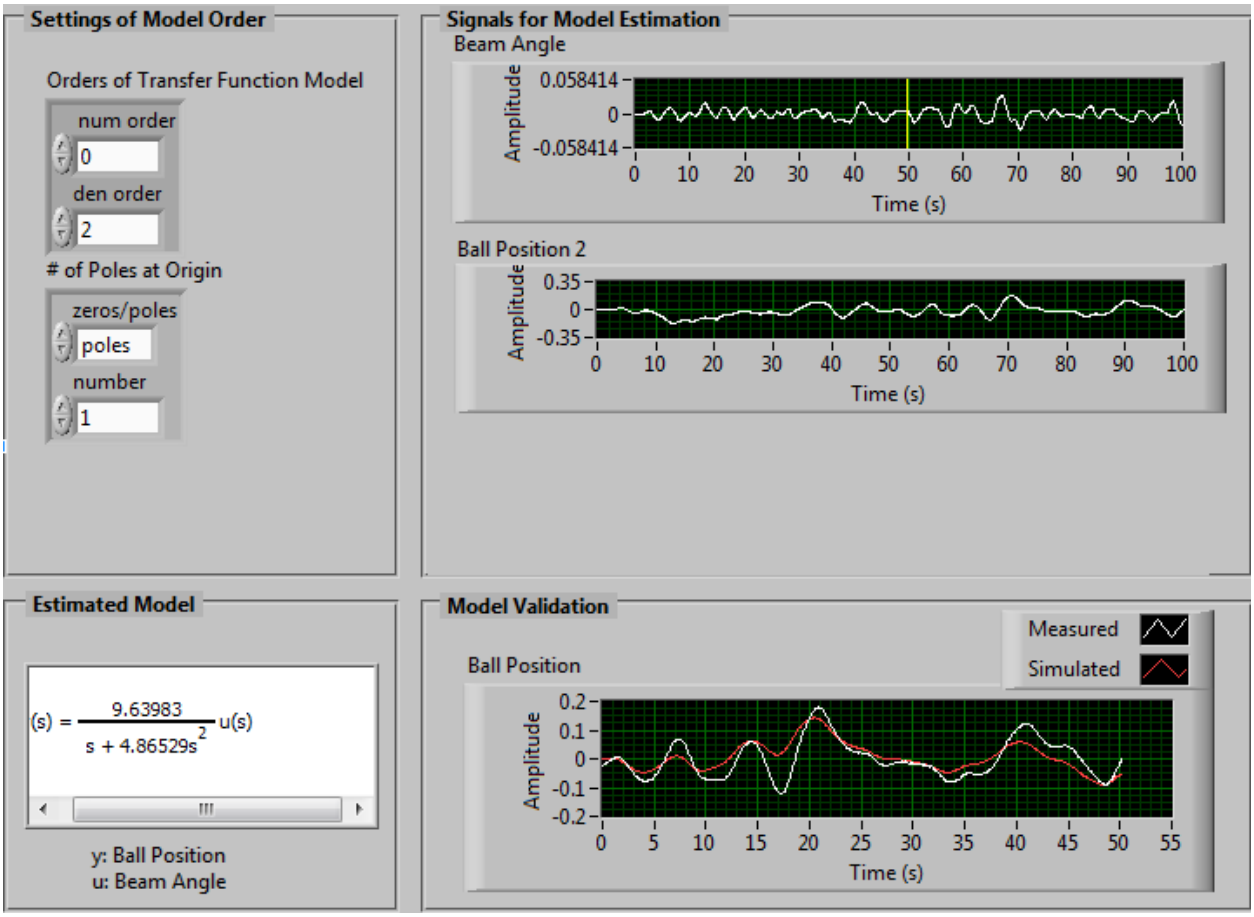


Fig 3.28: A good estimation of X(s)/θ(s) TF with assumed ball friction, i.e. $b_1 > 0$

For this model to better fit the prebuilt one, besides configuring the right number of poles and zeroes expected, as seen in the figure, the cursor should stay at least 30 seconds and beyond across the timeline so that the underground blocks may have some sufficient data accumulation to be able to estimate the proper TF match of the expected model. Accordingly, as appears in the figure the cursor point was chosen to be at around 85 seconds, which results in the emerged triggered TF model. The obtained 2nd order TF can clearly validate (119), especially when assuming that the friction effect is small on the ball movement so that the b_1 term can be canceled, which matches the hypothesis that the above estimated has two real poles very near to or at zero. It is also observed that as the cruiser slides on time timeline, the closer the estimated TF approaches the theoretical one in (119) as appears in Fig 3.27 (b).

As the model seems valid, it becomes possible to estimate the friction constant, b_1, parameter from these data sets and S/W tool. This suggests editing the setup of the desired fitted model to have only one pole at the origin as shown in Fig 3.28. Then b_1 could be estimated as

$$\frac{g}{\left[1+\frac{2}{5}\left(\frac{Rb}{d}\right)^2\right]s^2+\frac{b_1}{M_{ball}}s}=\frac{9.8}{1.9s^2+\frac{b_1}{0.02272}s}\approx\frac{9.64}{4.8s^2+s}$$

It follows that

$$\frac{b_1}{0.02272}\cong 1$$

Hence, $b_1 \cong 0.02272$.

Fig 3.28 demonstrates a good fit or model estimation of the friction constant parameter b_1. The estimated simulated model is based on the setting of expecting only one pole at the origin. If the driver block is viewed as a gain amplifier taking voltage as an input and producing current as an output, then the final generalized, experimental specific LCTI, and LDTI open loop BD of the ball on beam process plant should look like Fig 2.29 (a)-(f).

The underlying functionality of the estimated FBD VI depends on several concepts. First, the plotted estimated model is obtained according to LSE method, which either considers the identified system as a "linear regression", [70] that is a straight line given in (21), or higher order polynomial where the output, y, is a function of inputs, x_i, and the estimated parameters (β_1, β_2, $\beta_{3\ldots}$) such that:

$$y = f(x_1, x_2, x_3, \ldots., \beta_1, \beta_2, \beta_3 \ldots \beta_p) \tag{124}$$

Then, LSE aims to minimize the sum of square errors,

$$min\ S = \sum_{i=1}^{N} (Y_i - \hat{y}_i)^2 \tag{125}$$

Where Y_i is the measured output point and $\hat{y}_i$ is predicted such that

$$\hat{y} = \sum_{j=1}^{p} \beta_j X_j + \epsilon \tag{126}$$

Here X_j are the p functions of x; therefore (125) becomes

$$min\ S = \sum_{i=1}^{N} (Y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 \tag{127}$$

Then for N measured data points, Y_i, β_j, and X_{ij} can be expressed in matrix form as

$$Y = \begin{bmatrix} Y_1 \\ . \\ . \\ Y_N \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_1 \\ . \\ . \\ \beta_p \end{bmatrix} \qquad X = \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{N1} & \cdots & X_{Np} \end{bmatrix}$$

Therefore, estimating linearly dependent set of parameters requires matrix operation so that:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{128}$$

Considering the 1st order TF model in (23), the corresponding differential equation is

$$\tau \frac{dy(t)}{dt} + y(t) = Kx(t - L) \tag{129}$$

To remind, τ, K, and L are the time constant, steady state gain, and dead time, respectively. Using the predefined concept of LDTI approximation of LCTI systems, a randomly distributed or RT data of measured inputs and outputs that fits (129) can be written in difference equations forms,

$$y(k) = \left(1 - \frac{\Delta T}{\tau}\right) y(k-1) + \frac{K\Delta T}{\tau} x(k-L) \tag{130}$$

After manipulation and finite series approximation, (130) can be rewritten as:

$$y(k) = e^{-\Delta T/\tau} y(k-1) + K(1 - e^{-\frac{\Delta T}{\tau}}) x(k-L) \tag{131}$$

In other words, (131) appears as a linear regression polynomial of two parameters *a* and *b*,

$$\hat{Y}_i = aY_i + bX_{(i-\frac{L}{\Delta T})} \tag{132}$$

Where $a = e^{-\Delta T/\tau}$ and $b = K(1 - e^{-\frac{\Delta T}{\tau}})$, which are extracted from (131). Thus, using LSE described above or any other curve fitting method that employs known statistical data of the process would estimate the numerical values of *a* and *b* from which τ and K can be calculated.

Similarly, for a 2nd order TF given in (24) in a more generalized form is represented as

$$G(s) = \frac{K(\tau_a s+1)}{(\tau_1 s+1)(\tau_2 s+1)} \tag{133}$$

The corresponding difference equation is

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b_1 x(k-1) + b_2 x(k-2) \tag{134}$$

Now this is also a polynomial of a form given in (124). Where

$\beta^T = [a_1\ a_2\ b_1\ b_2]$, $X_1 = y(k-1)$, $X_2 = y(k-2)$, $X_3 = x(k-1)$, and $X_4 = x(k-2)$; then,

$$a_1 = e^{-\Delta T/\tau_1} + e^{-\Delta T/\tau_2} \tag{135}$$

$$a_2 = -e^{-\Delta T/\tau_1} e^{-\Delta T/\tau_2} \tag{136}$$

$$b_1 = K(1 + \frac{\tau_a-\tau_1}{\tau_1-\tau_2} e^{-\Delta T/\tau_1} + \frac{\tau_2-\tau_a}{\tau_1-\tau_2} e^{-\Delta T/\tau_2}) \tag{137}$$

$$b_2 = K(e^{-\Delta T(\frac{1}{\tau_1}+\frac{1}{\tau_2})} + \frac{\tau_a-\tau_1}{\tau_1-\tau_2} e^{-\Delta T/\tau_2} + \frac{\tau_2-\tau_a}{\tau_1-\tau_2} e^{-\Delta T/\tau_1}) \tag{138}$$

And the gain *K* is obtained from (134) at steady state

$$K = \frac{\overline{y}}{\overline{x}} = \frac{b_1+b_2}{1-a_1-a_2} \tag{139}$$

Note, at $\tau_2 = \tau_a = 0$, the system will result in to the 1st order form.

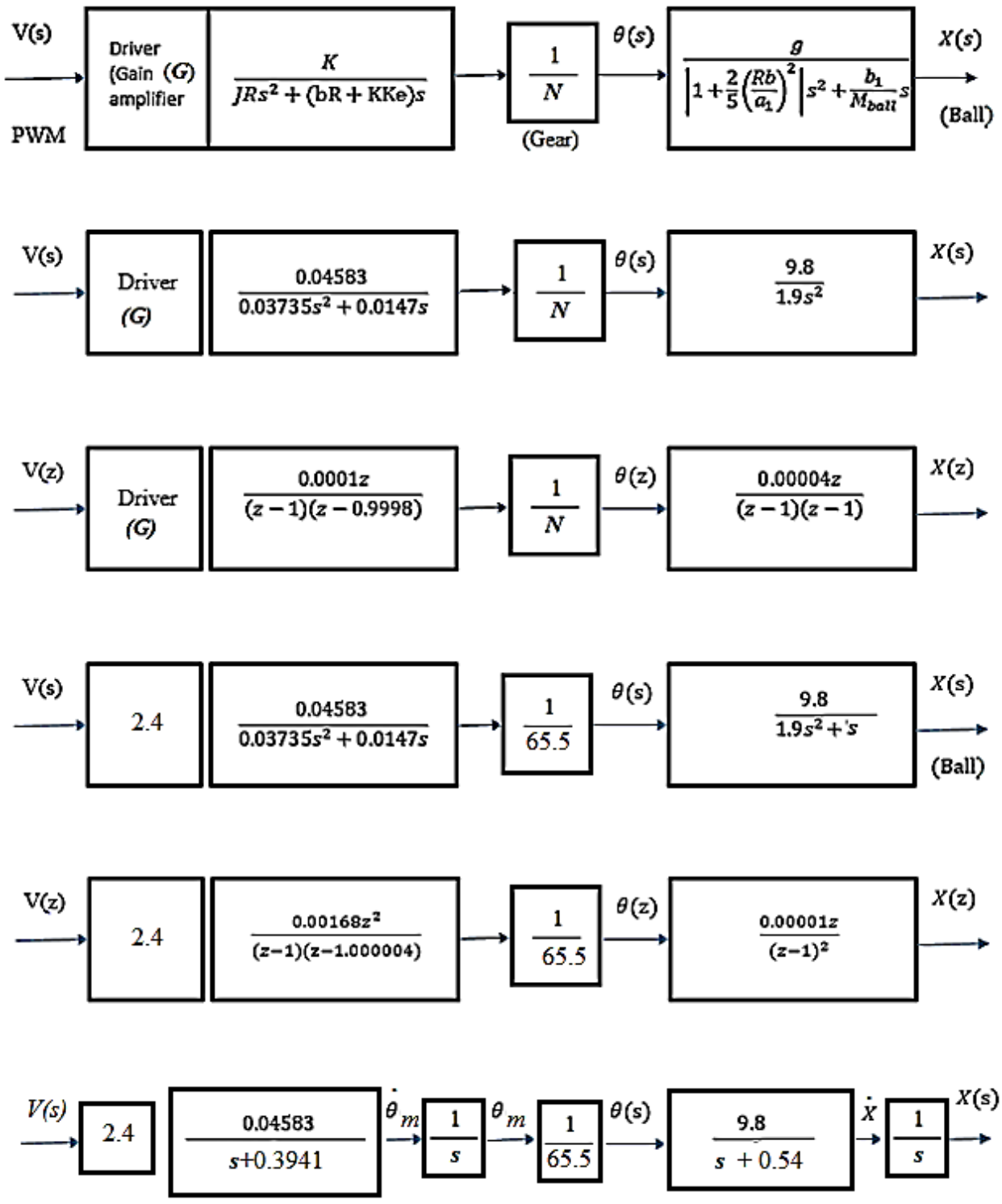


Fig 3.29 (a) The final general ball on beam process plant TFs, (b) the experimental one with no friction, (c) the equivalent z-domain, (d) the numerical model with the estimated $b_{1,}$ (e) the discrete Z domain of the system with friction, and (f) the detailed designed FB TF ball on beam

3.2.2.4. Optimal closed loop simulation response

After estimating the remaining parameter, b_1, the system is now ready for feedback control law design. To recall, the final version of the studied ball on beam model using a DC geared motor is a four states unstable model as appears in the LabVIEW VI simulation model of Fig 3.30 (a) below. This section aims to simulate the desired feedback control response of the ball on beam before designing the actual implemented RT controller. Plugging in the numerical values in Table 3.2 yields the following SS. The SS configuration model in LabVIEW is used where the number I/Os, states, their numerical values, and the set of initial conditions can be defined and configured as follows in the figure. This page appearance is actually an internal, double click view of the LabVIEW VI control and simulation toolkit emerges in the next part of the figure, (b).

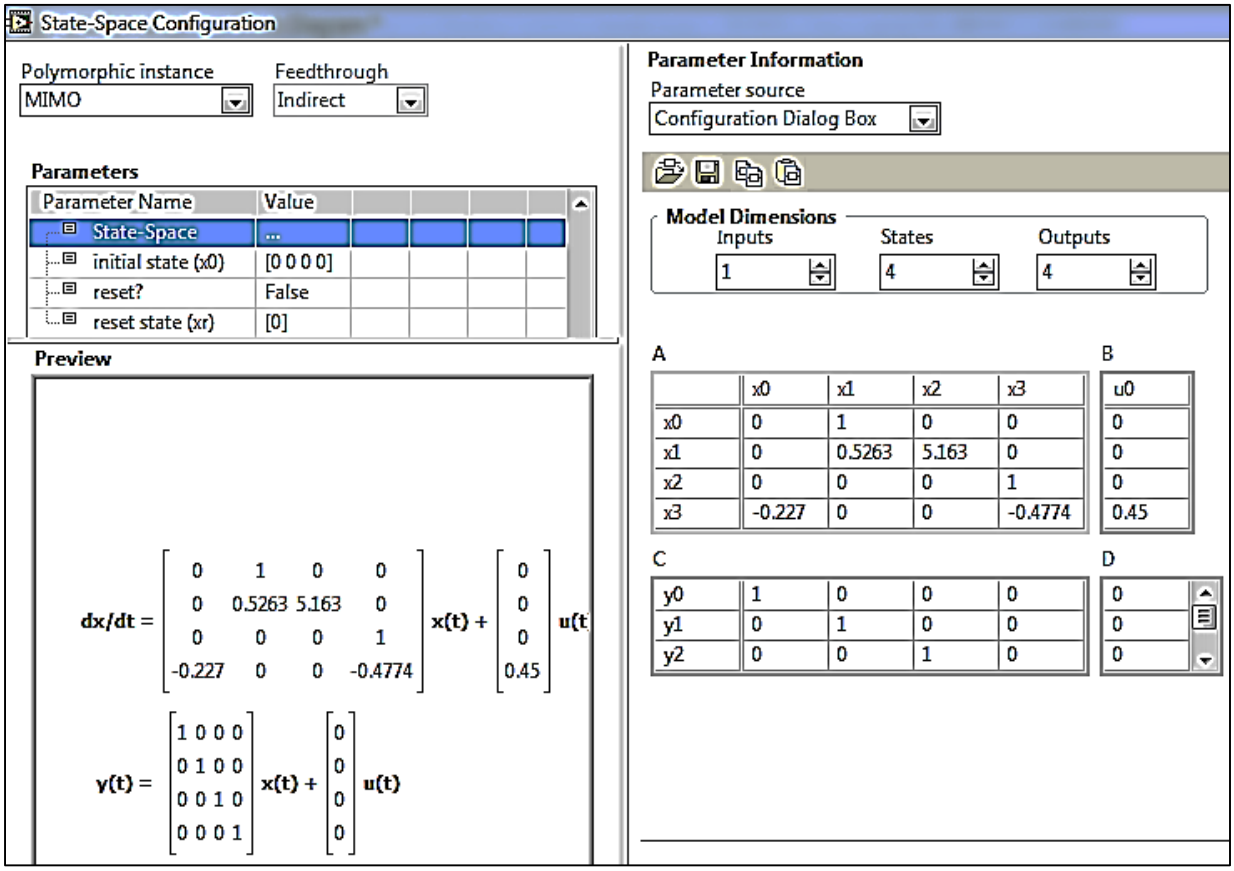


Fig 3.30: (a) The configured SS model after substituting the numerical values of the designed system

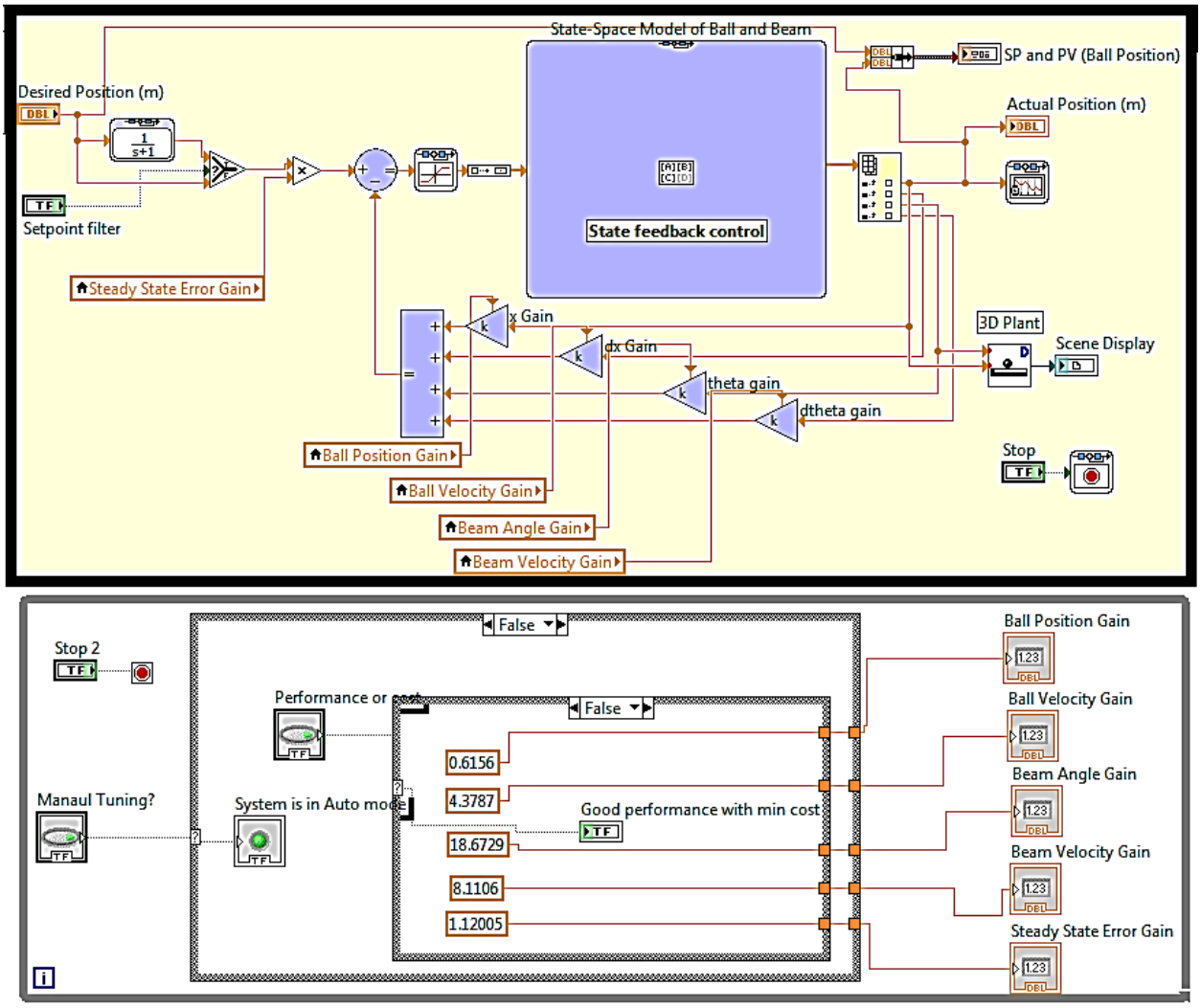


Fig 3.30: (b) the overall FBD program of the simulation model with added performance logic

The selected controller for simulation is the optimal LQR as can be seen from the FBD of the closed loop feedback control model of the studied experimented ball on beam process plant problem. The reason behind selecting LQR originates from the fact that:

- This controller realizes the LCTI SS model directly, which matches the derived MIMO ball on beam process plant.
- The LQR controller approach has been proving its validity and effectiveness in controlling such a plant efficiently; therefore, the result of this method could be considered as a reference for other suggested designed RT controllers.

- The optimal LQR helps in measuring both the control cost and transient dynamic performance of a suggested designed RT policy. Though, the LQR may not aim for robustness, but this problem is resolved by adding a SP filter as a robust control design method introduced in [18] for example.

Using Matlab simulation tool, the feedback control responses are obtained where the ball position and beam angle actions of the controlled experimented SS model. Recall the LQR targets for minimizing the cost function given in (62). This cost function is also called a "performance index." To design the optimal LQR gains requires mainly matrix operation after defining *A, B, C,* and *D* matrices of the SS model obtained in Fig 3.30. The full Matlab code from which the following optimal LQR gains are attained besides the next maximized performance ones is attached as Appendix 1. To remind, we aim to design the full SF controller upon the LQR optimal method such that:

$$u = Kx$$

The four state variables to be controlled here are the ball position, X, its velocity, $\dot{X}$, beam angle, θ, and the angular speed, $\dot{\theta}$. Therefore, the four designed optimal minimized cost SF gains vector that correspond to those variables respectively appear as:

$$[K] = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} = \begin{bmatrix} 0.61558 \\ 4.37869 \\ 18.67287 \\ 8.1106 \end{bmatrix}$$

The other designed set of gains that maximizes the dynamic response performance, such that the settling time, rise time, and time constant are minimized but with cost as well as overshoot tradeoffs, as a result of assigning more weights on R and Q matrices, emerges as:

$$[K] = \left[k_{1,} k_2, k_3, k_4\right] = [223.103, 164.707, 242.364, 31.7766]$$

The result of this set of gains as well as the previous ones are shown in Fig 3.31 (a) and (b).

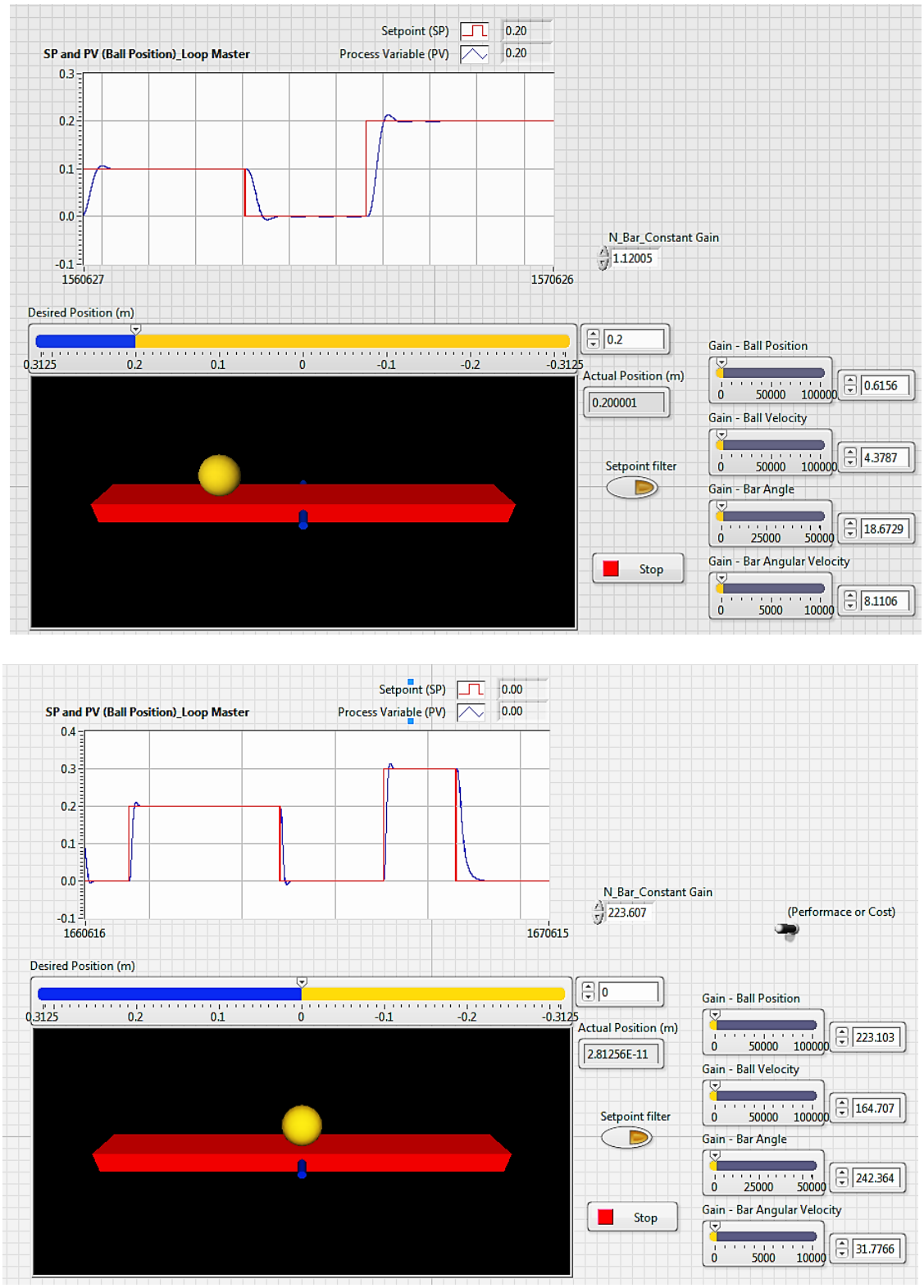


Fig 3.31 (a) A good controller with minimum cost (top) and (b) a maximized performance (weight) one

The figure above demonstrates the front panel of the FBD page shown in Fig 3.30. (a) shows the SP tracking or the closed loop simulated control of the first set of designed gains. (b), on the other hand, illustrates the maximized performance portion of the simulated design by adding more control weight on the matrix Q, which increases the value of gains. The control performance difference between the designs appears to be clear, but each has its advantages and disadvantages while both might not be very robust unless turning the SP filter button on. The graphical environment of LabVIEW nicely simulates the ball movement on the beam and the rotation. However, those four design parameters are not the only ones to tune in the full SF LQR controller. There exists also a gain factor design parameter, $\overline{N}$, which deals with the SSE while the above four gains tune the transient characteristics of the system. For the first and second design respectively, $\overline{N} = 1.12005$ and 223.603.

3.4. Summary, remarks, and discussion

In summary, this chapter analyzed the control problem of the ball on beam process plant from a theoretical perspective and reviewed a couple of previous trials of implementing this automation system. Namely, modeling and simulation of the ball on beam dynamics based on RT data besides estimated parameters were obtained. Viewing the problem from the MIMO perspective and then analyzing it in detailed TF SISO were crucial to understand the interconnections between inner and outer inputs and outputs of this process plant. It has been confirmed that the ball on beam is indeed an unstable, NL, uncontrollable, and unobservable process or system after constructing the SS model and testing it. The five-state MIMO SS model in (109) suggests this nonlinearity, uncontrollability, and not being observable. For this reason, model reduction as well as linearization approximation are important to adapt the system with control designs. Hence, the model in (122) of the four states is controllable and observable.

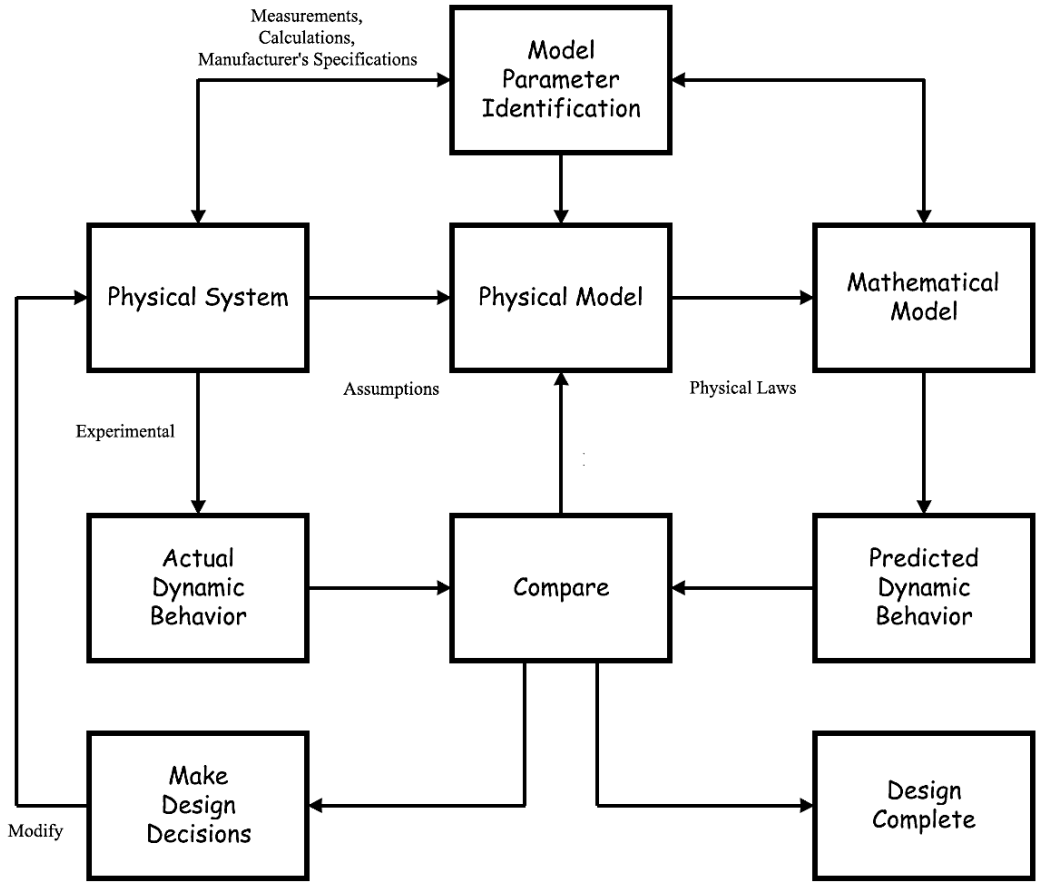


Fig 3.32: Modeling and simualtion process

This chapter contributed to the problem of the ball on beam by estimating the friction constant, *b*, using LSE parameter estimation, system identification concept, and S/W tools, such as LabVIEW modeling and simulation from which also the constructed model was validated and verified. This result and the whole process suggest analyzing dynamic systems for the purpose of control through a similar manner shown in Fig 3.32 where theoretical modeling construction based on understanding the physical model arrives first. After that, analysis of this constructed model should take place which will help decide whether reduction is needed or not. Then, model validation occurs after estimating unmeasured parameters to compare theoretical results with predicted ones. Finally, this chapter simulated an optimal control response with either performance or cost objectives.

CHAPTER 4

THE BALL ON BEAM PROCESS AS A PART OF INDUSTRIAL ARCHITECTURE

(AUTOMATION DESIGN AND INSTRUMENTATION IMPLEMENTATION)

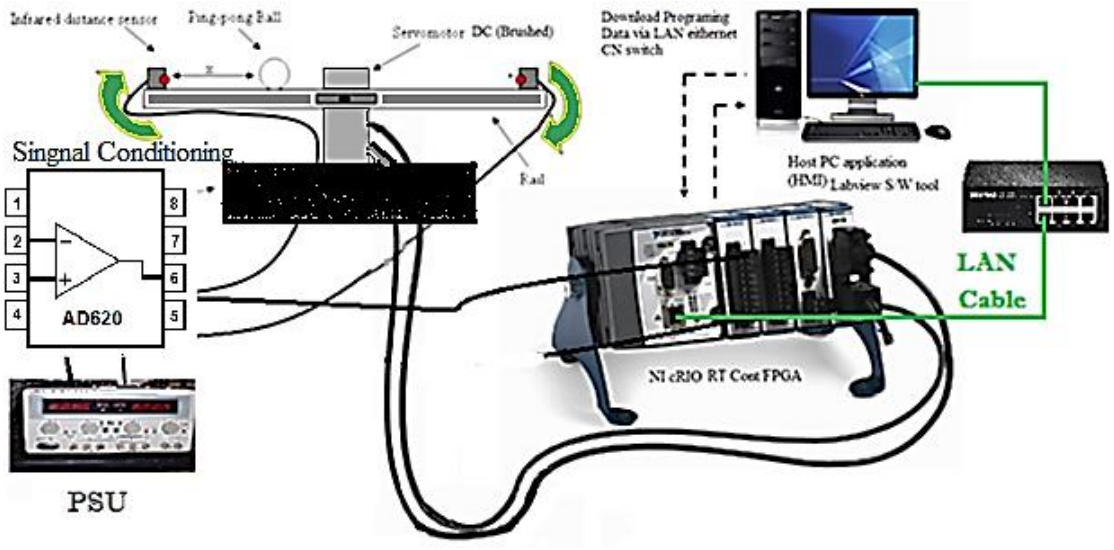


Fig 4.1: The ball on beam process plant as a part of an industrial automation architecture

In this chapter, the final RT automation design of the ball on beam process plant is described in detail. This work is distinguishable from all others in the sense that it considers the process plant as in the introduced industrial architecture. Fig 4.1 exemplifies the intended automation architecture and design. The figure shows level-0, the ball on beam process plant, level-1, measurement systems including sensors, actuators, and their signal conditioning used, level-2, the DCS CN containing a minimum of one RT-controller (NI's cRIO), a level-3 SCADA monitor workstation operating on LabVIEW S/W, and a network switch. Not only does this chapter implement the industrial automation of the ball on beam plant, but it also tries to evaluate the control performance of such a process under various instrumentation technologies and algorithms described earlier. The automation of the ball on beam using DC motor actuation is detailed by presenting the H/W platform and system parts, calibration and signals conditioning, mechanical and mechatronics design, the control strategies, and finally experiment results.

4.1. Hardware Platform and System Parts

The designed system components are either H/W or S/W parts. Overall, the automation system of the ball on beam process is here divided into: the physical plant, sensors and measurement feedbacks, NI cRIO components and a PC monitor, the actuators, other analog signal conditioning circuitry, and lastly S/W programs and requirements of the system.

4.1.1. Physical ball on beam plant

The physical ball on beam plant was designed and built in the UNT Engineering Labs. The mechanical part of the plant is described in Section 3. Fig 4.2 (a) and (b) below demonstrates a snapshot of the front and rear view of the built physical plant that includes:

- A metal base
- A very light aluminum beam and a motor stand
- The DC geared motor
- Other mechanical components, i.e. the motor coupling, the beam base, and others.

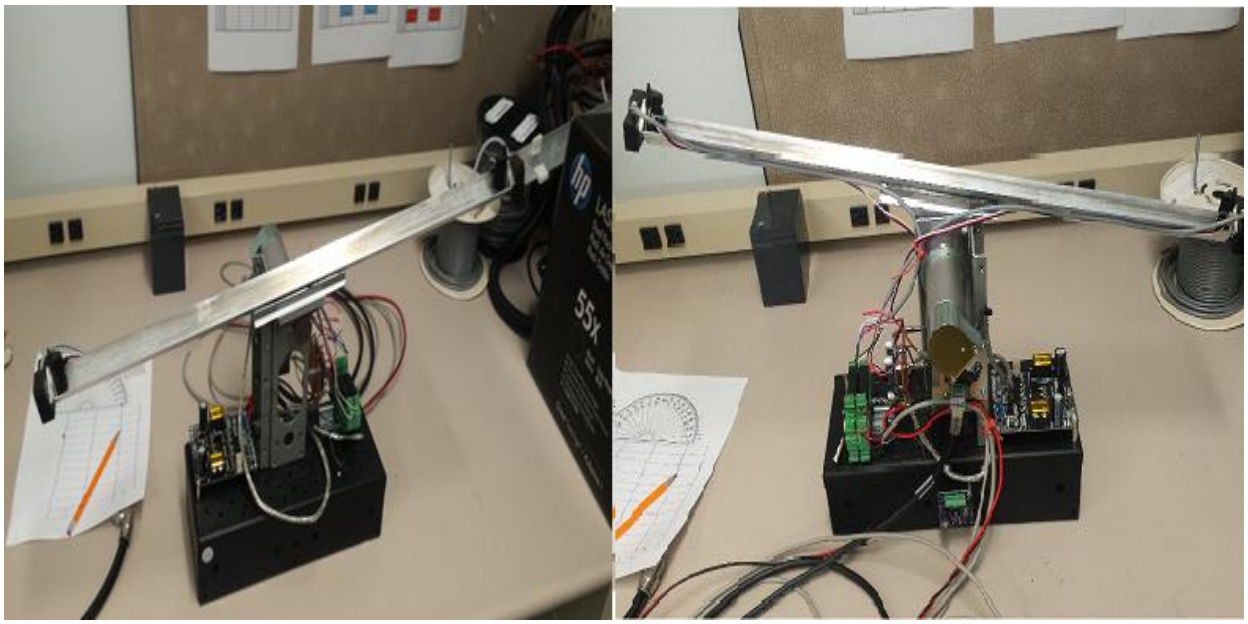


Fig 4.2: The built plant model's (a) front and (b) rear views

4.1.2. Measurement Sensors and Feedbacks

The measurement feedback loops of the typical ball on beam control problem contain two main sensing stages, ball position and beam angle, from where other states are extracted. However, the FPGA version of this proposed design has four to five cascaded loops: ball distance and velocity, beam angular position and speed, in addition to the current feedback. The current sense is a part of the NI DC motor driver used in the FPGA approach, and both velocity and motor speed are soft estimated at this stage. Velocity, and acceleration of a measured signal in the RT can be continuously acquired in the analog domain too using the concept of Op Amps differentiation reviewed in Chapter 2. This stage suggests two main RT variable measurements: the ball displacement using distance sensors and beam angular position.

4.1.2.1. Ball position measurement (IR distance sensors)

To measure the ball position on the beam, the Sharp IR distance sensors, shown in Fig 4.3 (a), are selected due to their low cost, availability, and direct analog outputs. This design suggests differential measurement, placing two IR sensors at the sides of the beam, which adds more sensitivity as well as static linearity and noise immunity. More details about the output signal of the differential measurement appear later in the signal conditioning design and calibration experiment. The Sharp IR sensor used follows the concept of optical displacement position measurement introduced in Chapter 2, namely an LED as a source and a PIN diode as a position sensitive detector. The model selected, GP2Y0A21YK0F, covers a measurement range of 10 – 80 cm and produces analog voltage signals around the range of 2.5 (min) to 0.5 (max). The update rate of the sensed signal is 38.3ms±9.6ms, and maximum delay is 5ms. Such a sensor requires 5 volts of power and consumes an average of 30mA (low power). Part (b) of the figure shows an internal BD of the IC. Also, [103] views the manufacturer datasheet for more specifications.

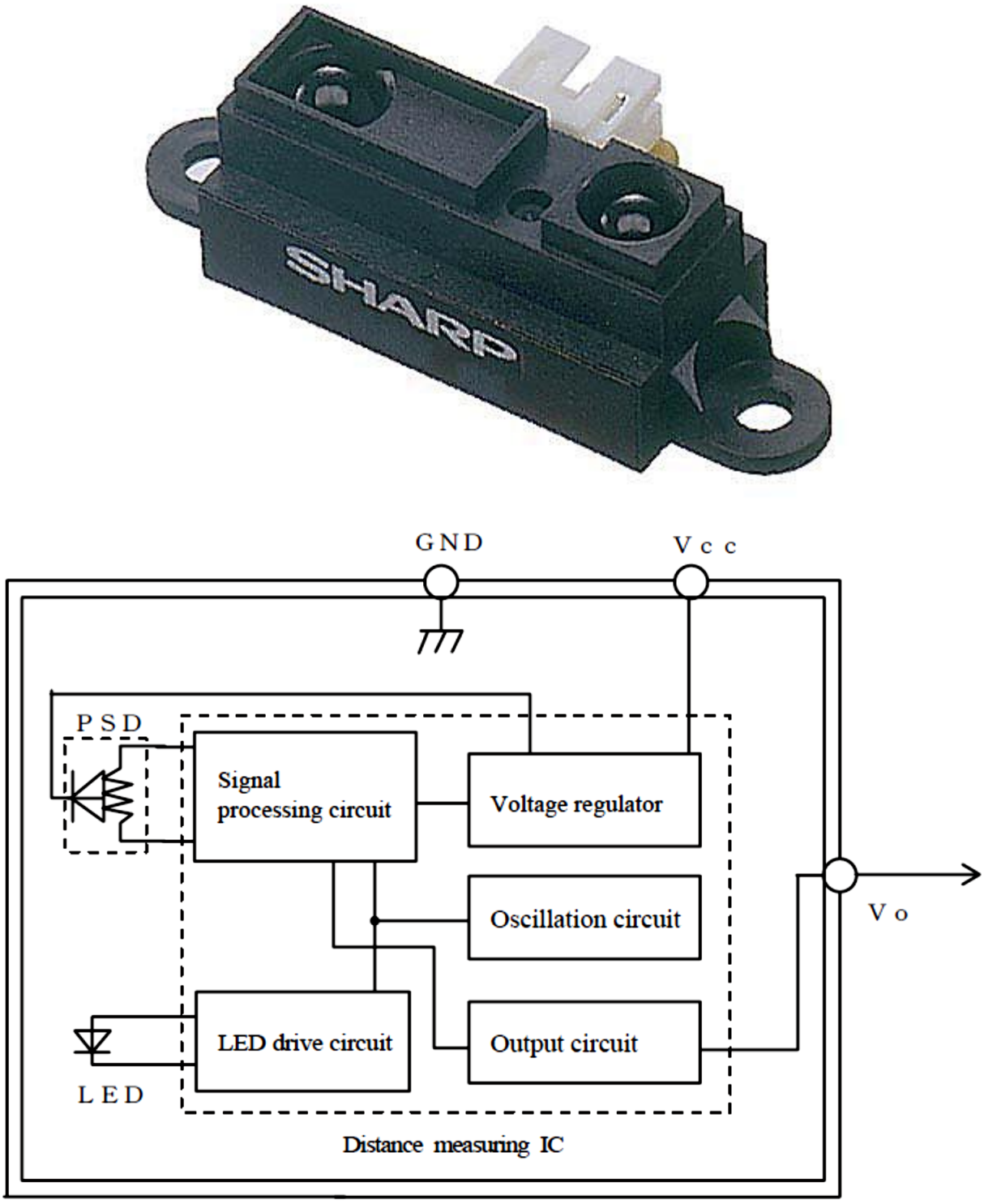


Fig 4.3: (a) The Sharp distance sensor used in the implementation, (b) a BD of the internal IC including the source, detector, power and signal processing circuitry, and others [103]

4.1.2.2. Beam angle measurement

Other literature utilizes either one of two modes to measure the beam angular position: encoder or analog (potentiometers). As a contribution of this design, the FPGA version of this work allows the user to select between both modes on the SCADA HMI workstation. This is because of the noise effect on digital rotary encoders due to several root causes such as wireless and power signal interference, dust and harsh environment, and lack of proper and isolated assembly cables and transmission mediums associated with such devices. Now the user can switch between both beam tilt's measurement modes whenever either one is not working efficiently.

The encoder used is the Ametek Pittman's, E30B Incremental Optical Encoder. It comes intergraded into the brushed motor used, DC Gearmotor GM9236S027-R1-SP. All specifications, drawing, dimensions, and pin connections are given in [104], the datasheet of this encoder. It is an incremental type that uses optical TX/RX as introduced in Chapter 2. This encoder provides a resolution of 500 PPR, outputs 3-channel quadrature, X_4 type, pulses signals, A, B, and Z (index) at a bandwidth of 100kHZ, and it is TTL compatible. However, the resolution measurement range of the beam's angle increases with the gear ratio, N = 65.5. i.e. and (30) becomes:

$$\theta = \frac{Counts}{x \cdot PPR \cdot N} . 360° \tag{140}$$

Recall, x is the encoding type, x = 4 in this case, and *Counts* is the encoder counter measurement output. The power requirement of the encoder is 5VDC, where the high pulse appears as 2.5 and above where the low (zero) one accepts 0.4 and lower. Fig 4.4 shows the five connection pins.

Pin	5-pin Single-ended
1	Ground
2	Index
3	A channel
4	+5VDC power
5	B channel

Fig 4.4: The three output signal pulses of the single-ended encoder adopted [53]

The same signal measured as position in counts can be used to estimate velocity and acceleration such as [105]:

$$\omega = \frac{\frac{Counts}{PPR}\times\frac{60sec}{1min}}{Fixed\ Time\ Interval\ (\text{sec})}(RPM) \tag{141}$$

$$\dot{\omega} = \frac{(\frac{Counts_k - Counts_{k-1}}{PPR})}{(Fixed\ Time\ Interval)^2(sec^2)} \times \left(\frac{60sec}{1min}\right)^2 (RPM^2) \tag{142}$$

The other method implemented to measure the angle of the beam applies the mean of analog measurement using the same optical sensing explained above. In this approach, the beam angle can be obtained from the distance measurement at the bottom of the beam and the nearest edge surface by substituting $\emptyset$ in (27), given in Chapter 2, instead of θ since the last denotes the desired tilt to be sensed while the first is related to appear in Fig 4.5. Then as per (27), $\emptyset$ equals to the following. Note that differential sensing is used here too where its calibration will be given later.

$$\emptyset = \cos^{-1}(\frac{1}{c}\cdot X^2 V_{out}) \tag{143}$$

So θ can be determined from the known relation:

$$\emptyset + \theta + 90 = 180\ (Deg) \tag{144}$$

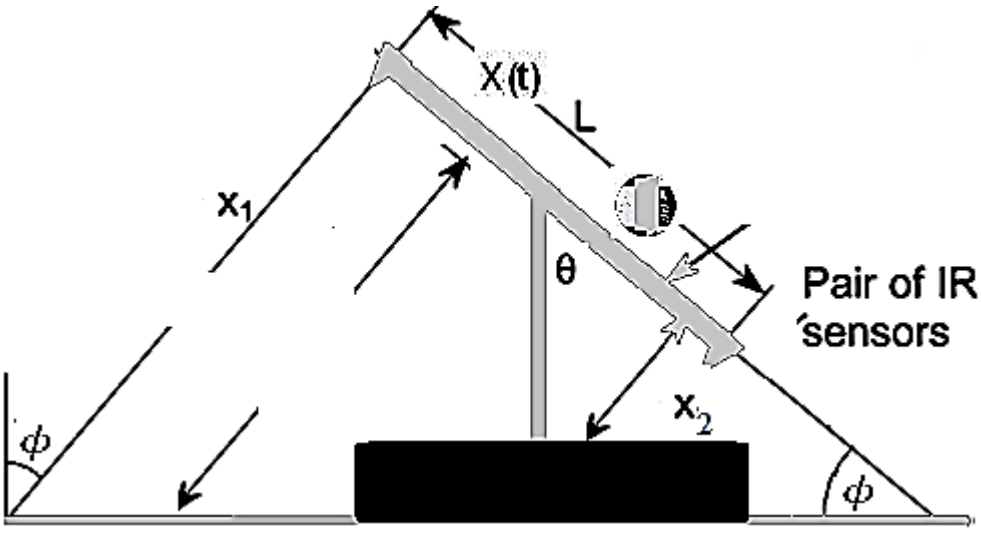


Fig 4.5: Analog measurement of the beam angle using differential pairs of IR distance sensors

4.1.3. Control Unit/System: NI cRIO RT-Controller/FPGA

The master control unit of the system is the NI cRIO RT-controller(s) shown in Fig 4.6 (a) and (b). Together with HMI workstations through LAN networking, a complete set of DCS can be formed. This H/W platform emulates DCS architecture and consists of the following [106], [107]: the RT controller scan engine, cRIO FPGA, I/O Chasses, host application SCADA PC monitor, and CN LAN network switch. The whole H/W architecture is made in a way that FPGA can be accessed through the CPU that runs on RTOS mode while the host application remains the LabVIEW S/W installed in the supervisory PC [108]. The system accepts two/three modes of I/O interface and S/W configuration: RT-controller directly, FPGA only, or both (Fig 4.6 (c)).

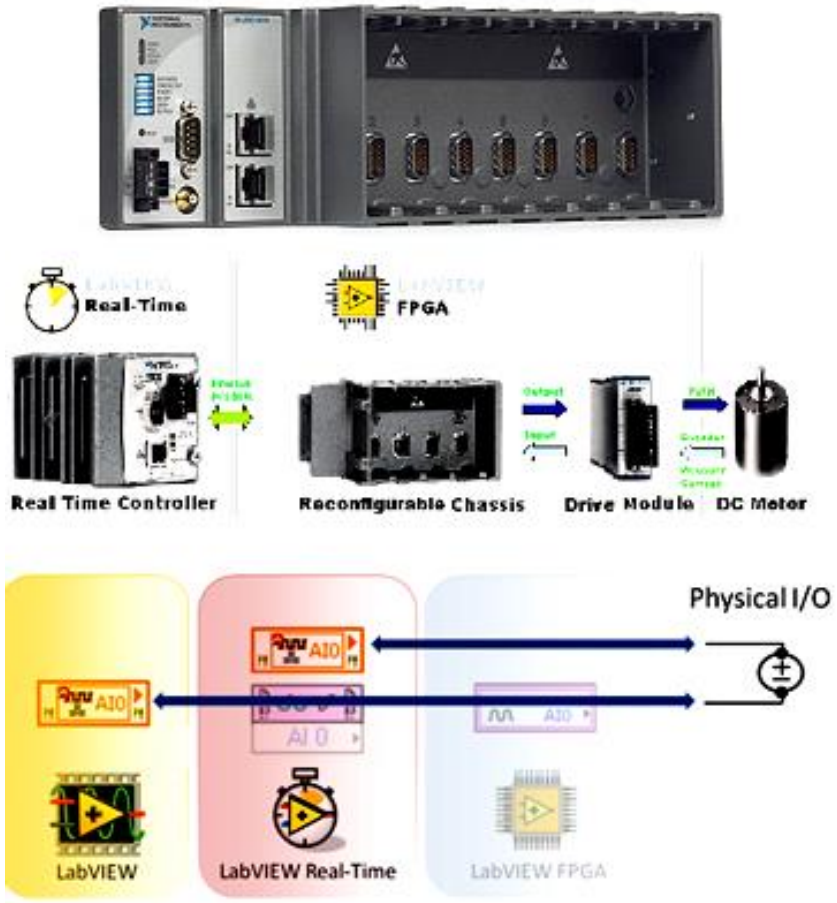


Fig 4.6: (a) NI cRIO platform and its I/O chasses (upper), (b) the integration of CPUs, FPGA and motor drivers, (c) the communication between the HOST application, RT Scan, FPGA, and I/Os

4.1.3.1. RT-controller scan engine (CPU processor)

The NI cRIO's RT scan processor is a MPU based architecture. The model used here is NI cRIO-9073 [106]. It has a processor speed of 266 MHz, dynamic RAM of 64 MB, and internal nonvolatile ROM of 128 MB. Besides the different types of field communications (including WSN) and I/Os it provides, the cRIO supports Ethernet as well as serial CN. The scan (update) engine of the main processor has a default execution period of 10ms. The CPU brand is PowerPC while VxWorks is the RTOS that is running.

4.1.3.2. The integrated c-RIO FPGA

The FPGA integrated into the H/W architecture is a product of Xilinx, Spartan-3 2M. It has a RAM capacity of 720 Kbits and carries a number of 46080 logic cells and 40 multipliers [106]. This FPGA produces an onboard clock cycle frequency of 40 MHz [63]. The FPGA is accessed and programmed through higher level S/W programing tool of LabVIEW RT on main processor.

4.1.3.3. Reconfigurable Chasses and I/O modules

As in industrial RT-controller architectures, the I/O chasses represent the carrier that accepts different types of AI, AO, DI, DO, serial, and other cards. Similarly, here, the cRIO H/W platform includes "plug and play" reconfigurable chasses that can carry 8 slots (I/O cards) and compatible is with expansion. The I/O modules used for this and other experiments are:

- NI 9215, a 16-bit 4 channels AI of range ±10 V and ADC sampling rate, 100 kHz [109].
- NI 9505: a 24V DC brushed servo motor driver that interfaces with the FPGA mode [110].
- NI 9219, a 4-channels, 24-bit, universal AI that accepts various signals types/ranges [111].
- NI 9403, a 32-channel 5 V/TTL, bidirectional sourcing/sinking DI/O of duration 7μs [112].
- NI 9474, a high speed 5-30 V sourcing DO of 8 channels and a switching time, 1 μs [113].
- NI 9263: a 4 channels, 16 bit, ±10 V range AO card with a sampling rate of 100 kS/s, [116].

4.1.3.4. PC monitor(s) (host application)

As emphasized several times before, to make the system industrially compatible, the control unit should be embedded and architecturally separate from the HMI, where the first performs the RT process control algorithms and logics while the latter functions as SCADA monitoring, operation as well as engineering, programming, system and network configuration access. In this case, any computer machine configured to the CN and has all S/W requirements and drivers installed such as LabVIEW, NI RIO, etc. can access any of the RT-controllers on the LAN. The master machine PC workstation that has been used so far to configure, program, and graphically operate and monitor the system is a Dell desktop that carries an Intel inside core i7 pro x86 MPU and runs on Windows OS. The workstation communicates with the rest of the nodes on the network through Ethernet TCP/IP LAN, and the scan engine period is 1ms.

4.1.3.5. CN LAN and a network switch

The cRIO controller supports 10BaseT and 100BaseTX Ethernet network interface, more specifically, is IEEE 802.3 protocols compatible. In other words, the CN of cRIO RT-controllers can take a form of TCP/IP, UDP, Modbus/TCP, and serial protocols. Furthermore, the architecture is IT intranet network and web friendly, for example HTTP and HTML [106]. The network switch(es) used is the Allied Telesis 8-port Ethernet switch shown in Fig 4.7 below. The communication rates of such a network are 10 Mbps or 100 Mbps, and the maximum LAN cable distance is 100m/segment [106].

Fig 4.7: The CN LAN switch used

4.1.4. The Actuator(s)

As may be known by now, the actuators used for this experimental implementation study are a servo DC (brushed) geared motor and a geared stepper (open loop) one. The stepper approach might be described in detail in the future from both theoretical and practical aspects, however, here, only the motor selected is briefly mentioned.

4.1.4.1. The DC motor

The geared DC motor selected, as per mechanical and mechatronics section of the design, is the Ametek Pittman Brush DC Gear motor of a model number GM9236S027-R1-SP seen in Fig 4.8. The motor is selected according to calculated and designed performance requirement of the system, primarily torque and inertia ratio. The datasheet, in which all the motor's manufacturer's design and specifications and from where a portion of the ball on beam system parameters presented in Table 3.2, is given in [114]. Also, Appendix 2 and 3 respectively express the torque, speed, current, and power performance curves and the mechanical and dimensional drawings of this motor. This motor is geared, and it has feedback rotary encoder E30B, described above attached at its back. The whole product forms an integrated unit of all the above.

Fig 4.8: The brushed DC geared motor (taken from [114])

4.1.4.2. The stepper motor

Complete design and comparison study of the stepper motor approach will follow in future work. The chosen motor is Anaheim Automation 17YPG001S-LW4-R27 [115].

4.1.5. Other Electronic and Signal Conditioning Circuitry and Apparatuses

The system parts include also other additional electronic circuitry at different stages. Those circuits emerge mainly in the analog domain whether as measurement signal conditioning or actuation, motor driving circuitries, or power supply and distribution. The blocks are classified into: PSU(s), the INA, external motor drivers, and other circuitry.

4.1.5.1. PSU(s)

Two to three power supplies are used in the system because of different devices, requirements. Most instruments on board demand either 5, 12, or 24 VDC to power up. Some supplies are combined in one source, 24VDC, and then distributed or divided to 12 or 5V through placing voltage dividers or regulators in between. The cRIO controller has its own PSU of 24V.

4.1.5.2. INA

As revealed, this design adds the advantage of using INAs in the analog signal conditioning phase of the system. The INA applied is the AD620 [117] as appears in Fig 4.9. The IC has eight input pins where one resistor, R_G, should be designed to produce the desired gain. Other pins connect to inputs, outputs, and power according to the application while REF denotes the zero level signal and usually tight to GND, because of differential inputs. The full design is given later.

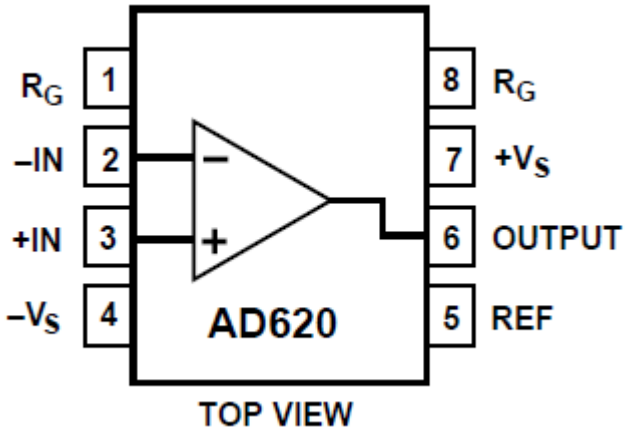


Fig 4.9: The AD620 INA used in the design (taken from [117])

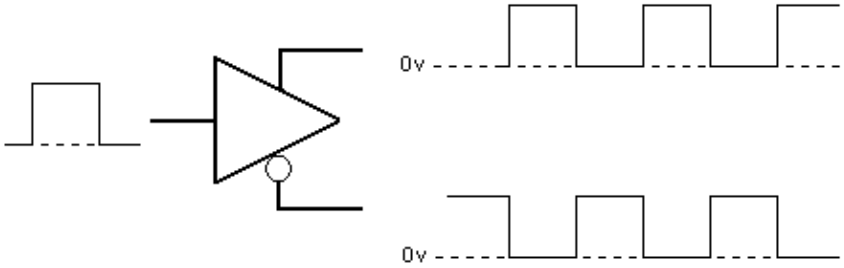
4.1.5.3. External DC motor drivers

The NI 9505 brush servo drive module introduced above operates in the FPGA mode only, so to configure the system in the RT scan processor environment necessitates external H-bridges boards. Here the external DC motor driver is a 24V, MD30C 30A DC [121], an H-bridge based and takes two inputs, PWM and direction.

4.1.5.4. Others circuitry and components.

Other components and circuits would be:

- A Differential line driver,



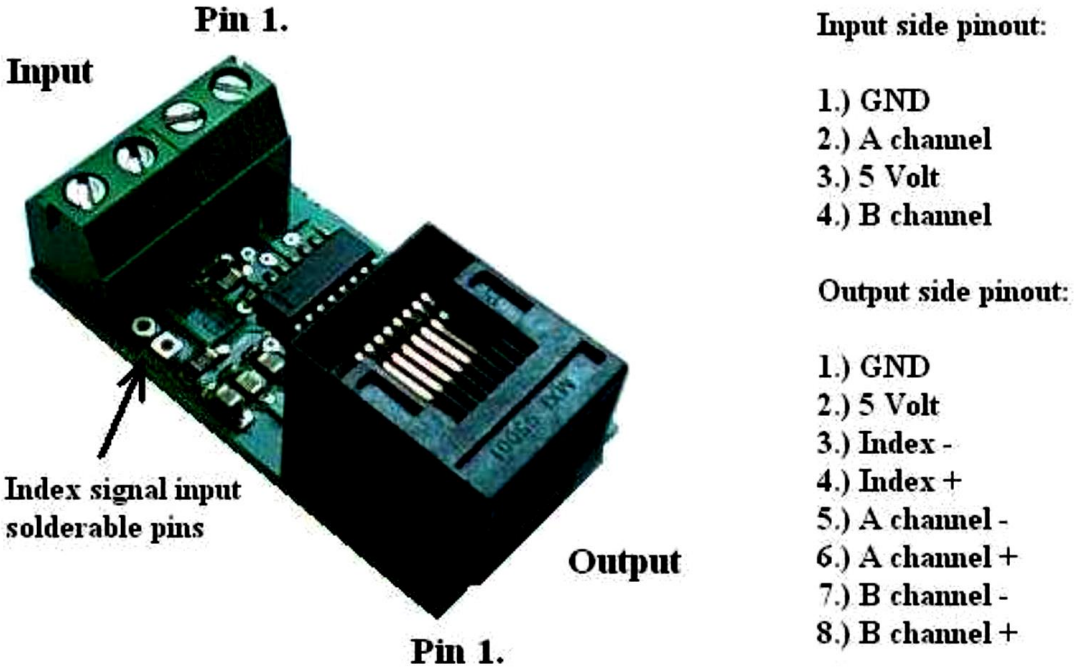


Fig 4.10 (a) The concept of differential line driver (b) The CNC differential line driver used

A differential line driver is a device that converts single ended pulse signal and provides two outputs, a buffered and an inverted version of the input as appears in Fig 4.9 (a). The differential line driver is used here as a conditioning phase for the incremental encoder signals. Thus, encoder signals tend to be noisy, especially single ended ones, which may produce wrong high/low directional readings that affect the overall control system performance and correctness. For this reason, a differential line driver is designed to convert single-ended to differential encoder signals. In other words, the five signals I/Os of the used encoder, E30B, A, B, Z and (index) pulses, besides GND and power will expand to eight, A^+, A^-, B^+, B^-, Z^+, Z^-, 5V and GND. The differential line driver circuit selected is a product of CNC shown in Fig 4.10 (b). It is a TTL dual differential line driver designed to meet serial connection specifications such as RS-422/232 as introduced in Chapter 1. For better noise immunity, the output of the circuit should connect to a shielded twisted pair cables.

- Voltage regulator(s) and buck convertors:

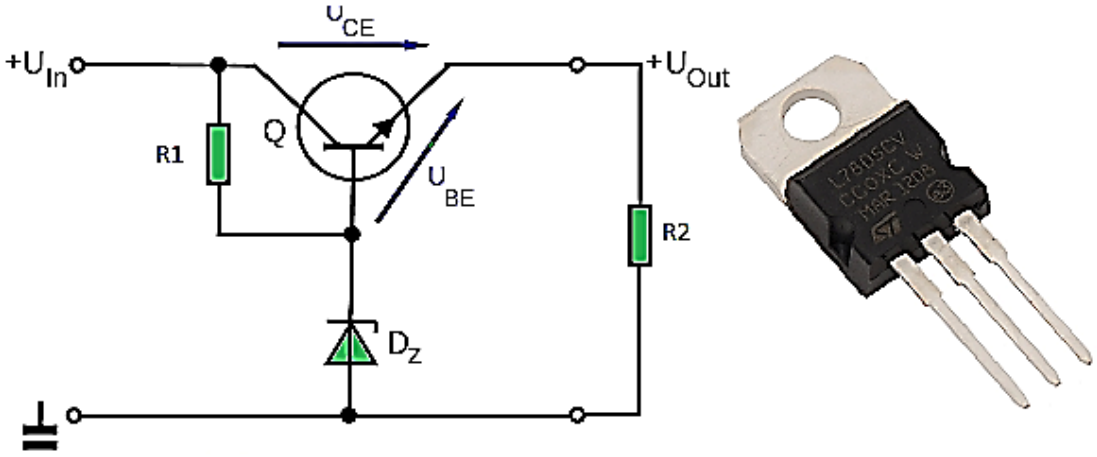


Fig 4.11: The BJT transistor based voltage regulator applied

Voltage regulators similar to Fig 4.11 above are implemented so that a common power source can be used. Two voltage conversion present from a 24V supply, 12V and 5V.

- Cables and connectors: other tools and components of the implementation are wires, special cables, and connectors. To overcome the noise and interference issues in the UNT engineering building, a twisted pair shielded cables and wires are recommended, which help sustain less near wireless and electromagnetic waves' and fields' effects [26].

4.1.6. Software Requirements

After listing the H/W parts, the system also consists of S/W side of the components. The S/W program, module, and driver requirements of the whole system can be either generally categorized under the host application PC monitor or the control unit (cRIO RT-controller).

4.1.6.1 Under host PC applications (SCADA monitoring HMIs)

As a recap, the host application PC machine in this architecture (TCP/IP LAN) functions as a master node in which it should possess all the S/W drivers and modules to be installed in targeted RIO devices (RT-controller), slaves. In addition to modules and drivers to be downloaded into RT targets, host machines should acquire the following minimum S/W requirements:

- Windows OS
- LabVIEW: LabVIEW 8.20 or later developer suite
- LabVIEW FPGA module
- LabVIEW professional development system
- LabVIEW-RT module
- NI-RIO driver, NI-RIO version 2.1 (or later) for series and cRIO embedded targets
- NI distributed manager and NI max programs
- NI soft-motion development module for LabVIEW
- Networking compatibility: Ethernet IEEE 802.3 (TCP/IP, UDP, or Modbus/TCP) [106]

Fig 4.11 on the next page exemplifies the S/W relation between host stations and cRIOs.

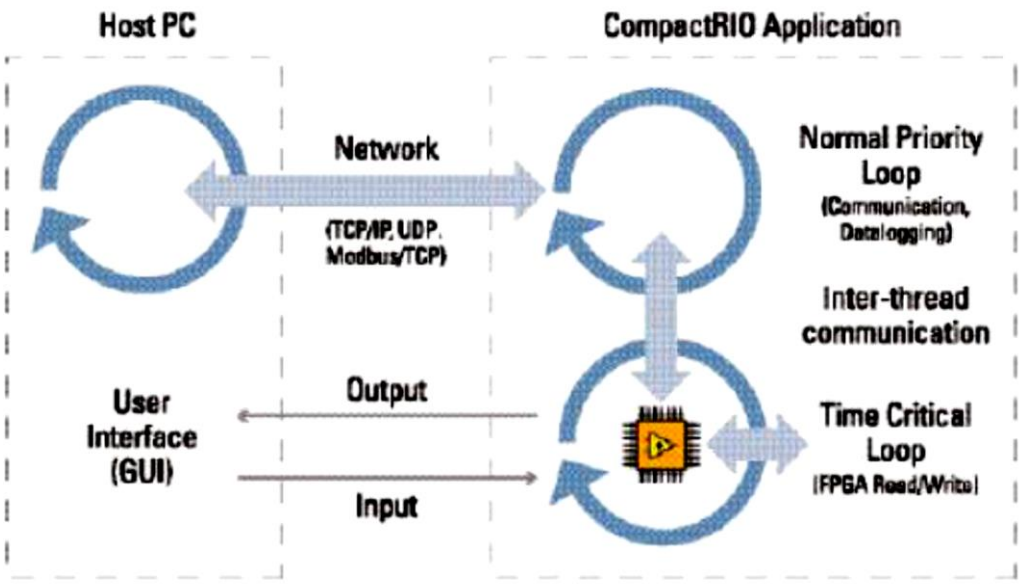


Fig 4.12: Communicating host PCs to cRIO RT targets over the CN LAN (taken from [106])

4.1.6.2 Under RT-controller targets

Since the RT-controller is adopted to implement the DCS, besides the RTOS and other adds it should contain the following S/W modules as well to function in LabVIEW RT environment:

- VxWorks RTOS
- NI-RIO driver
- LabVIEW
- LabVIEW FPGA module
- LabVIEW professional development system
- LabVIEW RT module
- Industrial monitoring option for NI developer suite
- NI Soft-Motion development module for LabVIEW
- IEEE 802.3 networking compatibility (TCP/IP, UDP, or Modbus/TCP)
- Other modules and drivers, i.e. LabVIEW Control and Simulation, signal express, etc.

Fig 4.12 (a) shows an example comparison of the higher level powerful LabVIEW VI FB and graphical environment in comparison with a typical VHDL FPGA programing. Similarly, (b) clarifies a BD of the whole S/W architecture and communication between different parts.

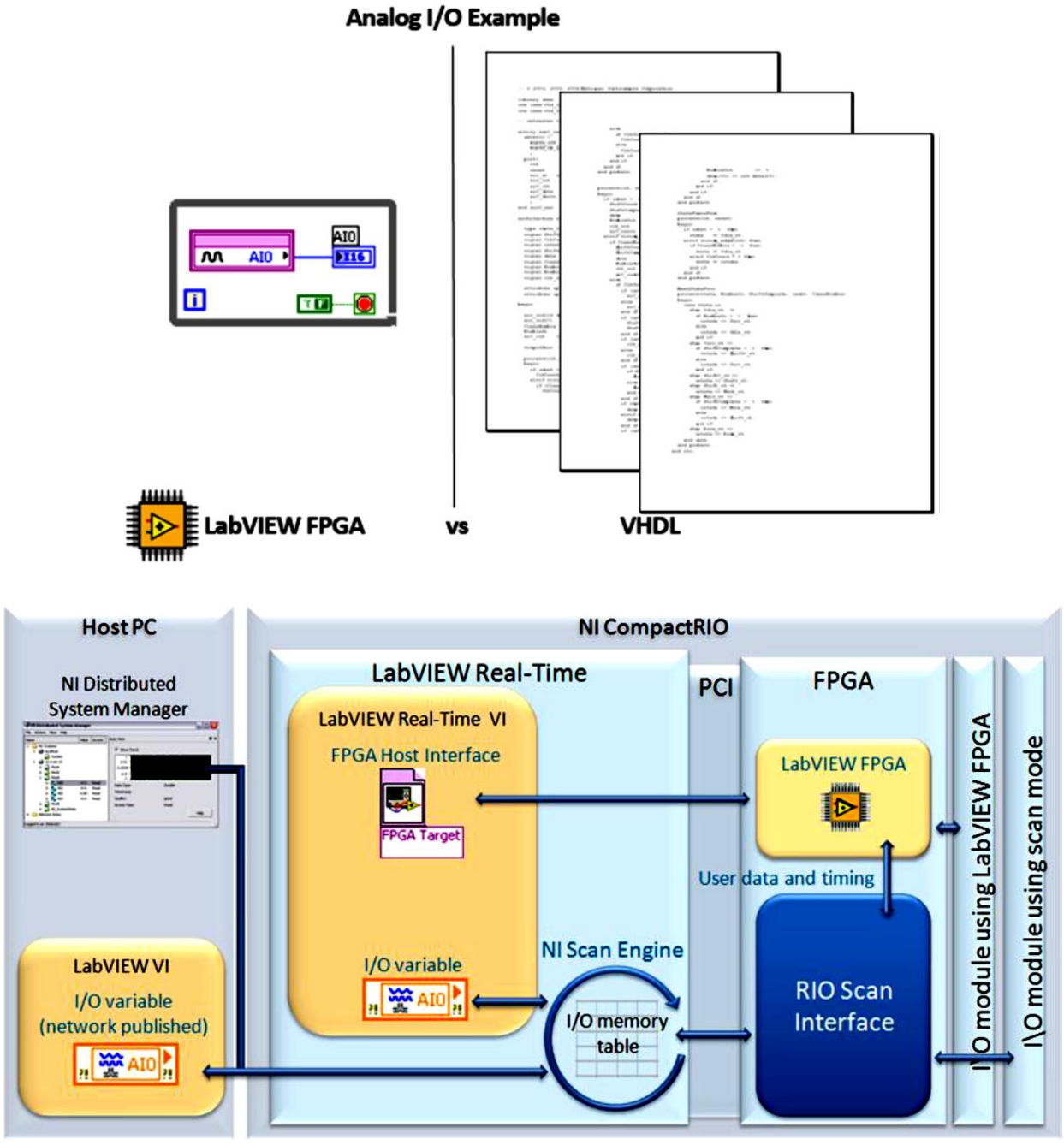


Fig 4.13 (a) VI vs VHDL programing [98] and (b) the distributed inter-communication (taken from [108])

4.2. Signals Conditioning and Calibration

This section explains the RT measurement instrumentation phase, namely signal conditioning and calibration. But, before getting into detailed designs, it is convenient to model the feedback sensors available. The encoder measurement relation has been displayed. Still, for the analog ball and beam's angel position sensing, the IR sensor is used as discussed. The static part of this sensing element is introduced in Chapter 2. This section exhibits the dynamic response models of the designed differential measurement system of both ball distance and motor position. Fig 4.13 shows the manufacturer's specified IR sensor static behavior.

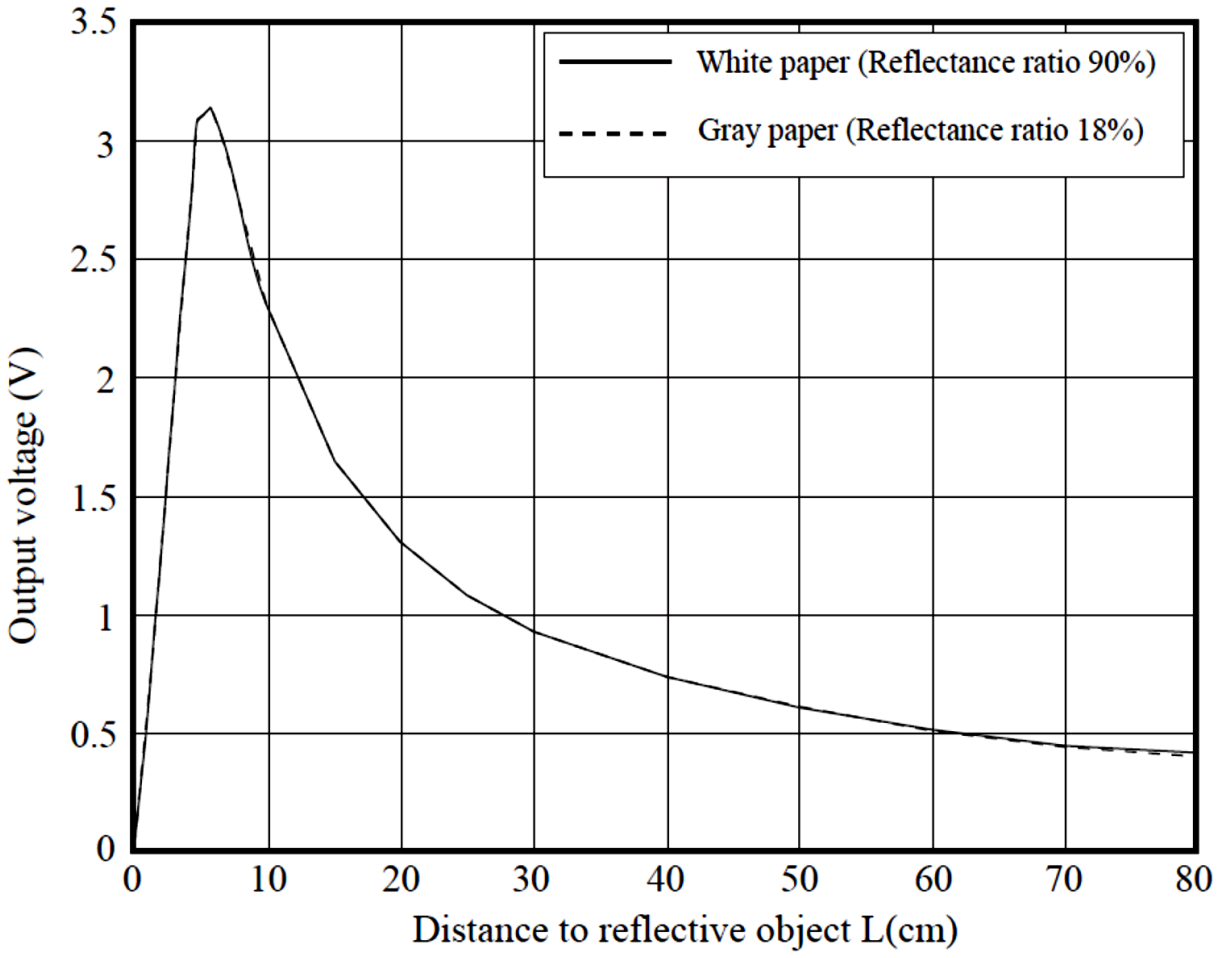


Fig 4.14: The static NL Sharp IR proximity sensor V(X) calibration graph (taken from [103])

4.2.1. Signal Conditioning

As presented before, there are at least two continuous RT feedback measurements in the ball on beam automation system: the ball's linear displacement and beam's angular position. As any measurement system of a feedback loop, a signal conditioning phase is crucial. Therefore, the signal conditioning design of this system is divided into two parts as per the number and types of outer and inner feedbacks implemented: distance and angle measurements.

4.2.1.1 Distance measurement signal conditioning

The problem statement is formulated as: it is required to convert and transmit the two noisy 0.5 to 2.5 V output signals of the IR distance sensor, used to measure the ball position, to be a smooth standard, and cover the full ADC resolution of the NI cRIO AI I/O module of range, 0-10V or -10 - 10V (differential). A logical way to approach this common problem is analyzing analog signal conditioning from the algebraic view first, i.e. developing output equations in terms of inputs. Therefore, for scaling field sensors signals, the equation is nothing but a straight line equation such as:

$$V_{out} = GV_{in} + V_o \tag{145}$$

Where *G* is the slope of the line and it here represents the scaling or amplification gain, and V_o stands for the intercept in which $V_{in} = 0$ (the zero trimming). However, before amplifying the sensor signals, it is critical to filter out unwanted noise associated with them. Thus, the raw sensed signal should pass through filtering noise through implementing a LPF and then amplifying those smoothed measurements to satisfy AI ADC ranges.

- LPF: for simplicity and availability, a passive RC LPF circuit as the on in Fig 4.14 is chosen to implement a LPF of a cutoff frequency, *f*c, around 1600 Hz to filter higher frequencies and noise components from the signal. To restate, the 1st order LPF has the TF,

$$V_0(s) = \frac{1/RC}{s+1/RC} V_{in} \quad (146)$$

Where RC is the time constant, τ, and it is a product of selected resistors and capacitor values so that the designed *f*c is met, i.e.

$$\omega_c = 2\pi f_c = \frac{1}{\tau} = \frac{1}{RC} \quad (147)$$

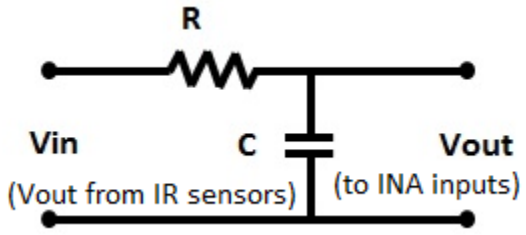


Fig 4.15: The 1st order RC LPF implemented where R =10Ω and C = 10μF

For the desired *f*c band of around1600 Hz and after some experimentation, it was found that R = 10 Ω was available and C = 10 uF was selected. The output of the LPF passes to the INA. This yields the following TF:

$$LPF(s) = \frac{10000}{s + 10000}$$

- INA: back to the scaling problem started above, the IR sensor output span is approximately from 0.5 to 2.5V while the required AI range is -10 - 10V. A general solution to the problem would be subtitling in (128) above, which yields the following system of equations:

$$10 = G2.5 + V_o$$

$$0 = G0.5 + V_o$$

Solving for *G* and V_o produces G = 5 and Vo = -2.5V. This can be implemented by placing a non-inverting Op Amp with and sum the output with a constant -2.5V by summing amplifier. However, since this project proposes the differential measurement, and, the use of the INA contributes to the problem, a different amplification design emerges where:

$$V_{out} = G(V_{in1} - V_{in2}) \tag{148}$$

This would turn the problem to look for amplifying the differential output of the two IR sensors placed at the opposite sides of the beam; therefore, the maximum range(s) would be 2.5 – 0.5V or 0.5 – 2.5V = +/- 2V, hence G = 5. As introduced earlier, for designing an INA requires selecting one resistor only, $R_{G.}$ As per the utilized INA's, AD620 displayed in Fig 4.15, datasheet [117], R_G can be designed from the relation,

$$G = \frac{49.4\ k\Omega}{R_G} + 1 \tag{149}$$

So that

$$R_G = \frac{49.9\ k\Omega}{G-1} \tag{150}$$

Now for G = 5, R_G = 12.35 kΩ.

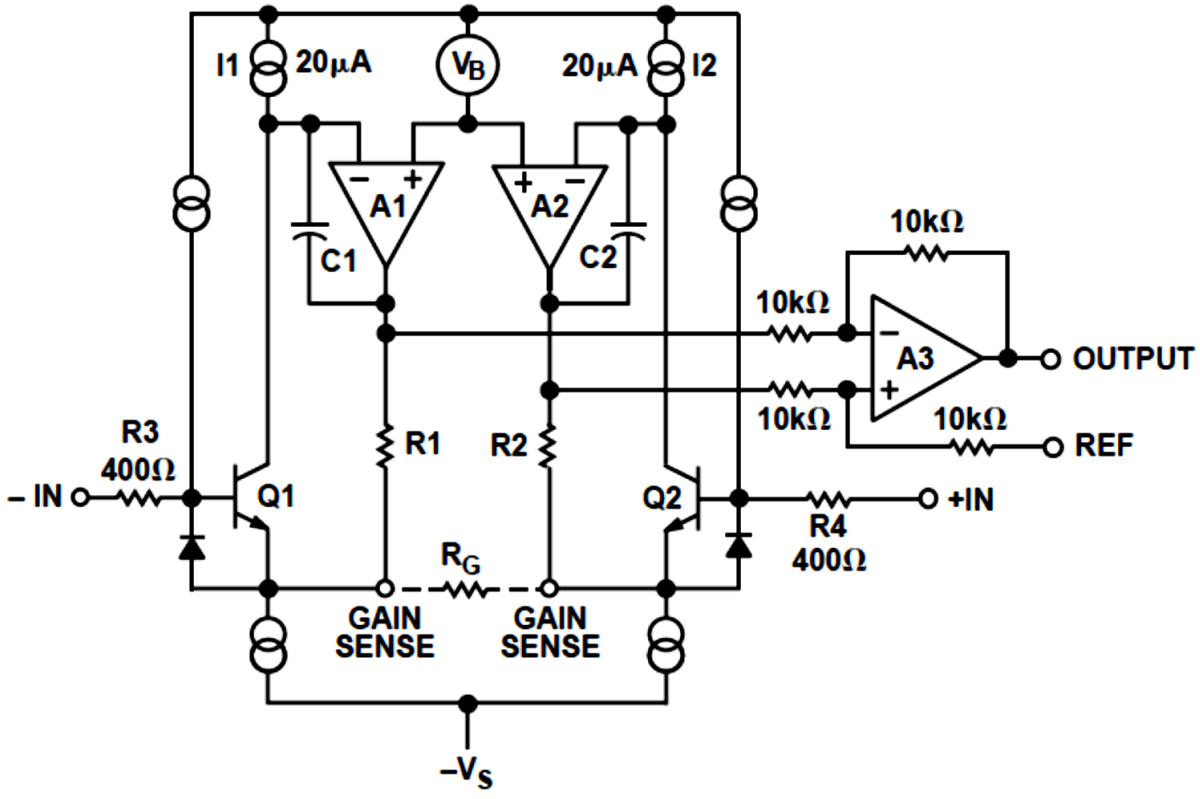


Fig 4.16: The internal IC of the AD620 INA (taken from [117])

4.2.1.2. Beam angle measurement signal conditioning

Depending on the mode, the designed signal conditioning of the beam angle varies. For the analog mode explained earlier, the same LPF and INA method displayed above will be applied. Nevertheless, for the encoder type of measurement, a couple of issues have been observed to be associated with such devices. First, encoders tend to be noisy, especially low resolution ones, due to quantization error and environmental effects such as electromagnetic interference. Second, the pulse outputs of digital encoders should be dealt with as in serially communicated bits where certain standards are necessary for healthy data transmission; thus, the field communication must embrace at least a proper PHY layering. For example, for RS-485 standard introduced in Chapter 1, digitally transmitted bits pass through a shielded twisted pair(s) cabling. Not only does shielding appear to be important in wiring encoder lines [110], but also pull-up resistors as recommended in [104] and [110], to ensure strong zeros and ones are being transmitted. Therefore, shielding should take place in encoding measurement systems, otherwise erratic reading will occur, especially in highly noisy environments, i.e. the UNT Engineering building.

The third issue, and the most critical, is bandwidths of the pulse signals, and how fast our digital controllers' rates should be able to construct and control such frequency adequately. After a little research in this matter, it has been found that a sort of a digital filter should be present after acquiring raw encoder signals and before passing them for control algorithms execution. This S/W signal conditioning step aids in two ways: first, filtering out erratic readings caused by the AWGN in the channel, and second, adding a delay time period in which it reduces and conditions the bandwidth with digital loop rates. The fourth practice that contributes in measuring closer to accurate rotary encoders' signals is the differential encoding technique presented earlier. Thus, if one line provides uncertain PVs, the other adjusts.

4.2.2. Static Sensing Elements and Calibration Experiments

Fig 4.13 above displays a single IR sensor static calibration graph, but the proposed design suggests placing two sensors at the opposite sides of the beam. The differential calibration graph experiment was constructed at the lab using simple tools: a white Ping-Pong ball, a ruler, and multi-meter, and the sensors outputs as functions of inputs graphs were extracted using Matlab curve fitting tool. The readings were taken at two stages, the output of the INA's voltage and at the AI module. Two calibration experiments are required, the ball distance measurement element and the angle of the beam. Recall the distance range is about (-25 to 25 cm) while the beam's tilt should not exceed $(-\pi/6 \text{ to } \pi/6)$ at maximum. For the encoder measurement of position, velocity, and acceleration, equations (123), (124), and (125) are applied directly.

4.2.2.1. The ball position measurement calibration

Distance (Inch)	Instrumentation Amplifier Differential Reading (V)
-8.5	-9.9
-7.5	-9.2
-6.5	-8.5
-5.5	-7.8
-4.5	-6.6
-3.5	-5.1
-2.5	-3.7
-1.5	-2.2
-0.5	-0.7
0	0
0.5	0.8
1.5	2.2
2.5	3.7
3.5	5.1
4.5	6.7
5.5	7.8
6.5	8.5
7.5	9.2
8.5	9.9

Table 4.1: The ball position differential measurement calibration experiment

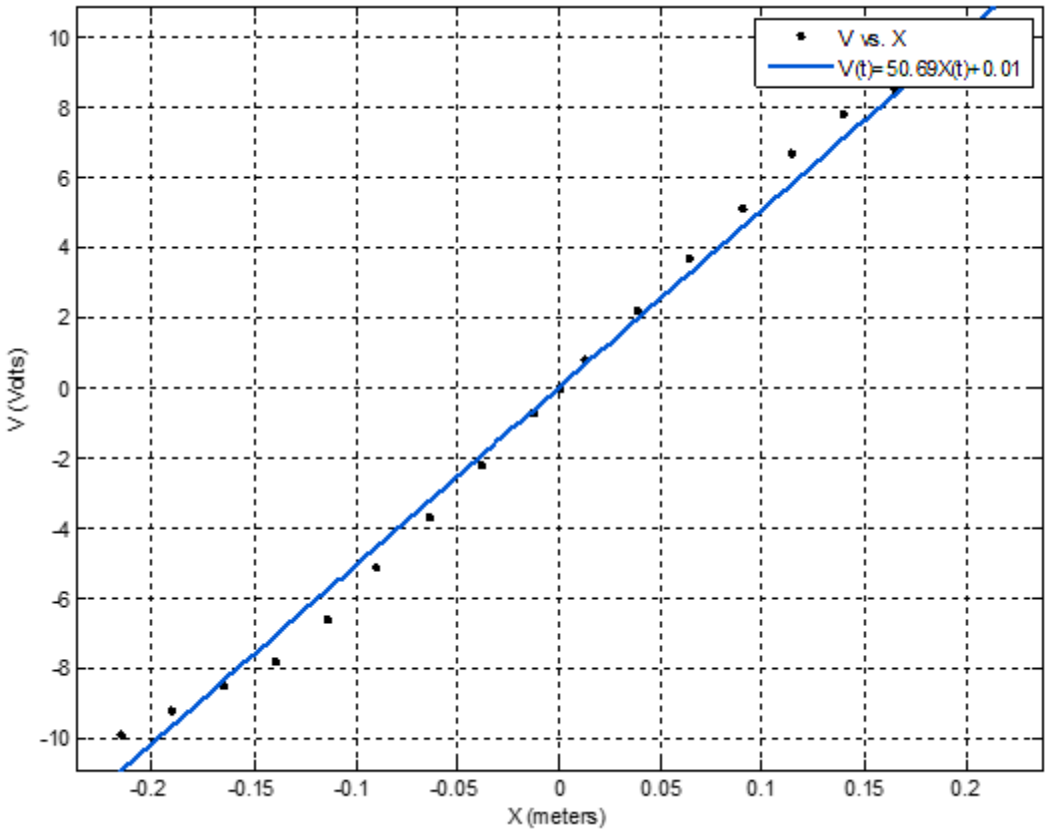


Fig 4.17 (a): A good linear fit for calibrating the ball measurement

Table 4.1 above displays the calibration experiment for the differential ball position measurement in volts. To perform the experiment, the ball was placed on a number of readable points on the beam (in inches), and readings were recorded. As a result, Fig 4.16 (a) above and the following differential sensor(s) output, V, relation to the disablement, X, produce:

$$V(x,t) = \ 10.24 * sin(0.1524 * x + 0.001533) \qquad (151)$$

From which the distance equation as a function of the measured voltage can be typed or translated in the main program, RT (scan) host VI, for displaying the distance in inches or centimeters. Please notice the huge difference between the graph in Fig 4.13 and the differential one obtained recently. This is one advantage of signal conditioning. This calibration graph can be also simplified to be modeled as a straight line, which also displays the benefit of utilizing INA.

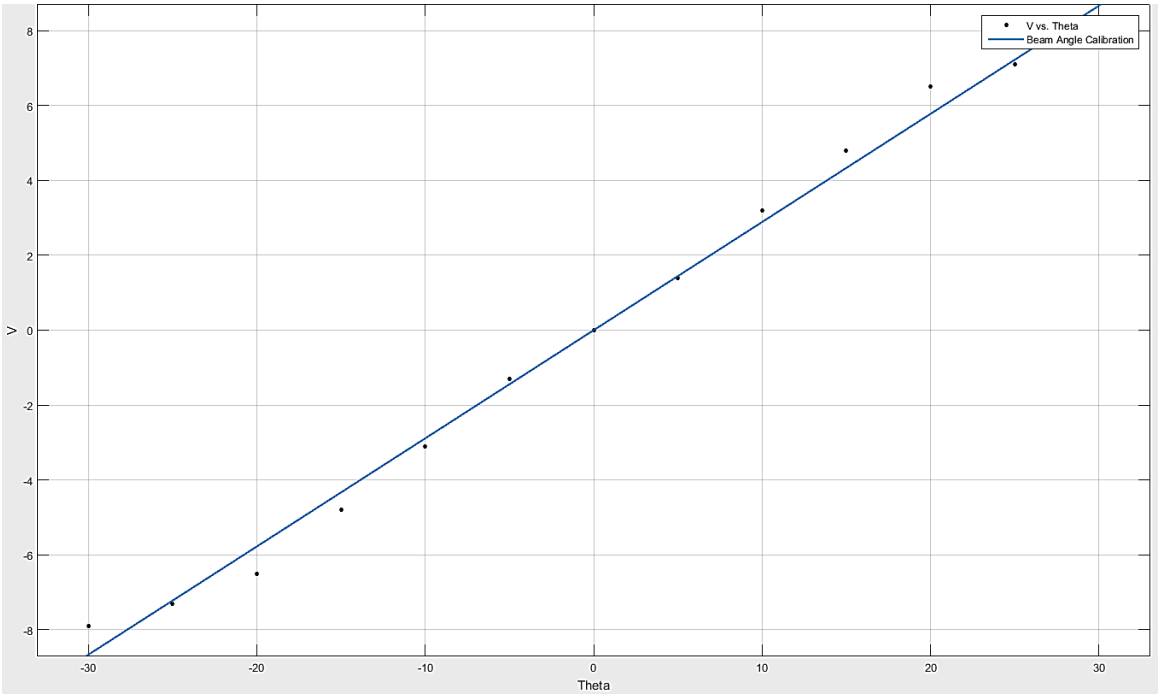


Fig 4.18: The differential analog beam angle linear calibration graph in voltage as a function of θ

The final linear static model or calibration graph of such a system is written in the equation,

$$V(x,t) = 50.9X(t) + 0.01 \quad (152)$$

Where X here is in meters and V in Volts.

4.2.2.1. The beam's angle measurement calibration

The beam's angle calibration differs whether the analog (differential) or digital encoder measurements is applied. For the analog method, similar concepts and tools executed measuring the distance. As an exception here, a third variable should be measured which is θ which depends on both distance and voltage readings as discussed several times earlier. Note that for proper cascade control design a faster response optical set of sensors is used here, FADK 14U4470/IO Photo electric sensors [118]. The operating range of such sensors is (4 cm – 40 cm).

Fig 4.17 (b) above shows analog beam angle measurement system's calibration with respect to changes in θ. The following linear two-dimensional equation was obtained:

$$V(\theta) = 0.2902\theta(t) - 0.0004959 \quad (153)$$

ΔX (cm)	ΔV	θ (degrees)
34.5	7.9	30
27.5	7.1	25
17.5	6.5	20
7	4.8	15
4.5	3.2	10
1.5	1.4	5
0	0	0
-1.5	-1.3	-5
-4.5	-3.1	-10
-7	-4.8	-15
-17.5	-6.5	-20
-27.5	-7.3	-25
-34.5	-7.9	-30

Table 4.2: The beam angle position differential measurement calibration experiment

Considering the 3-D model of the proposed analog θ sensing results in the following (Fig 4.18):

$$V(X,\theta) = 0.02597 + 0.2792X + 0.18830 + 0.005892X^2 - 0.007738X\theta + 0.002357\theta^2 + 0.008875X^3 - 0.01606X^2\theta + 0.008875X\theta^2 - 0.0015390^3 \quad (145)$$

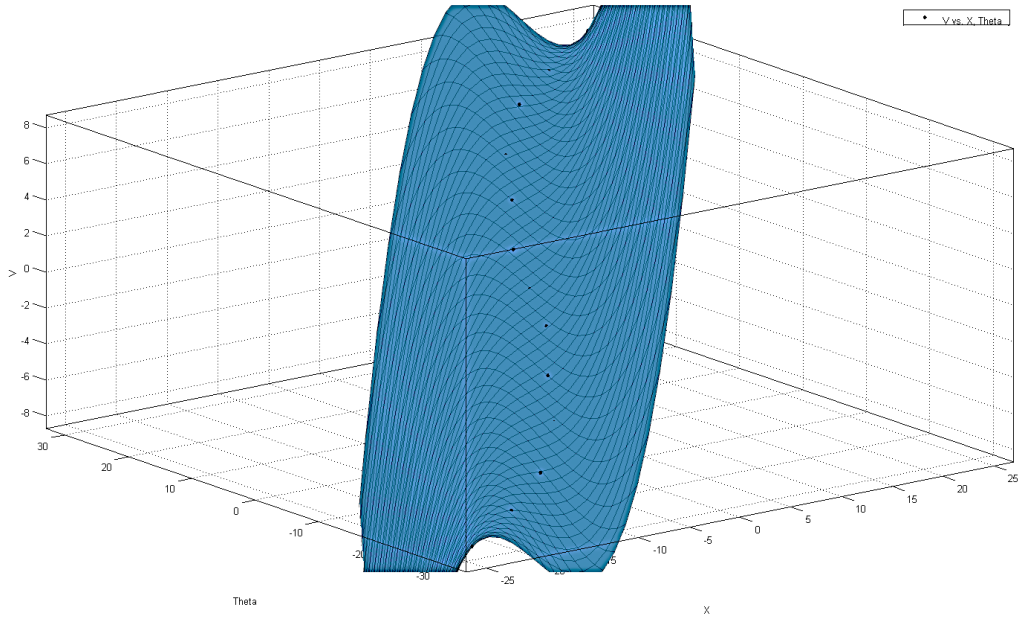


Fig 4.19: The analog beam angle measurement voltage output relation as a function of θ and X

For the digital encoding position measurement, the output is counter incremental ticks in both polarities, so after rewriting (140) and substituting the given and designed numerical values:

$$Counts = \frac{x \cdot PPR \cdot N \cdot \theta}{360°}$$

Therefore, for $\theta_{max} = \pi/6 = 30°$,

$$Counts/30° = \frac{(4) \cdot (500) \cdot (65.5) \cdot (30)}{360} \approx 10000\ counts$$

This calibration result is crucial in designing the control strategy, especially in the FPGA mode where both position and velocity loops are scaled in count units of controllers' gains. However, this calibration factor may alter after applying the digital time-stamp filter, and so the resolution will degrade in favor of reducing noise and bandwidth to be compatible with the loop rate.

4.2.3. Sensors dynamic response and transducers' models

The IR sensor can be modeled dynamically as a 2nd order response, but the proposed design suggests placing two sensors at the opposite sides of the beam and adds signal conditioning. The dynamic response of such a design for both the ball distance and beam angle emerge here. Since no known literature found has done this experiment, the differential measurement designed for the ball position was modeled by applying the concept of a step response and then using empirical data modeling [70] to construct the estimated TF. For the ball position feedback, it is found that it follows a 1st order step response (Fig 4.18 (a)), which can be model-identified as the following TF. A similar manner models the beam's angle dynamics:

$$M_d(s) = \frac{\Delta V_{IR}(s)}{\Delta X(s)} = \frac{Ke^{-Ls}}{\tau s+1} \cong \frac{29.4e^{-0.005s}}{0.012s+1} \approx \frac{29.4(1-0.005s)}{0.012s+1} \quad (155)$$

To eliminate the exponential term, the TF can be further approximated [70] as above. Similarly

$$M_p(s) = \frac{\Delta V_{LR}(s)}{\Delta \theta(s)} \cong \frac{10e^{-0.003s}}{0.0045s+1} \approx \frac{10(1-0.003s)}{0.0045s+1} \quad (156)$$

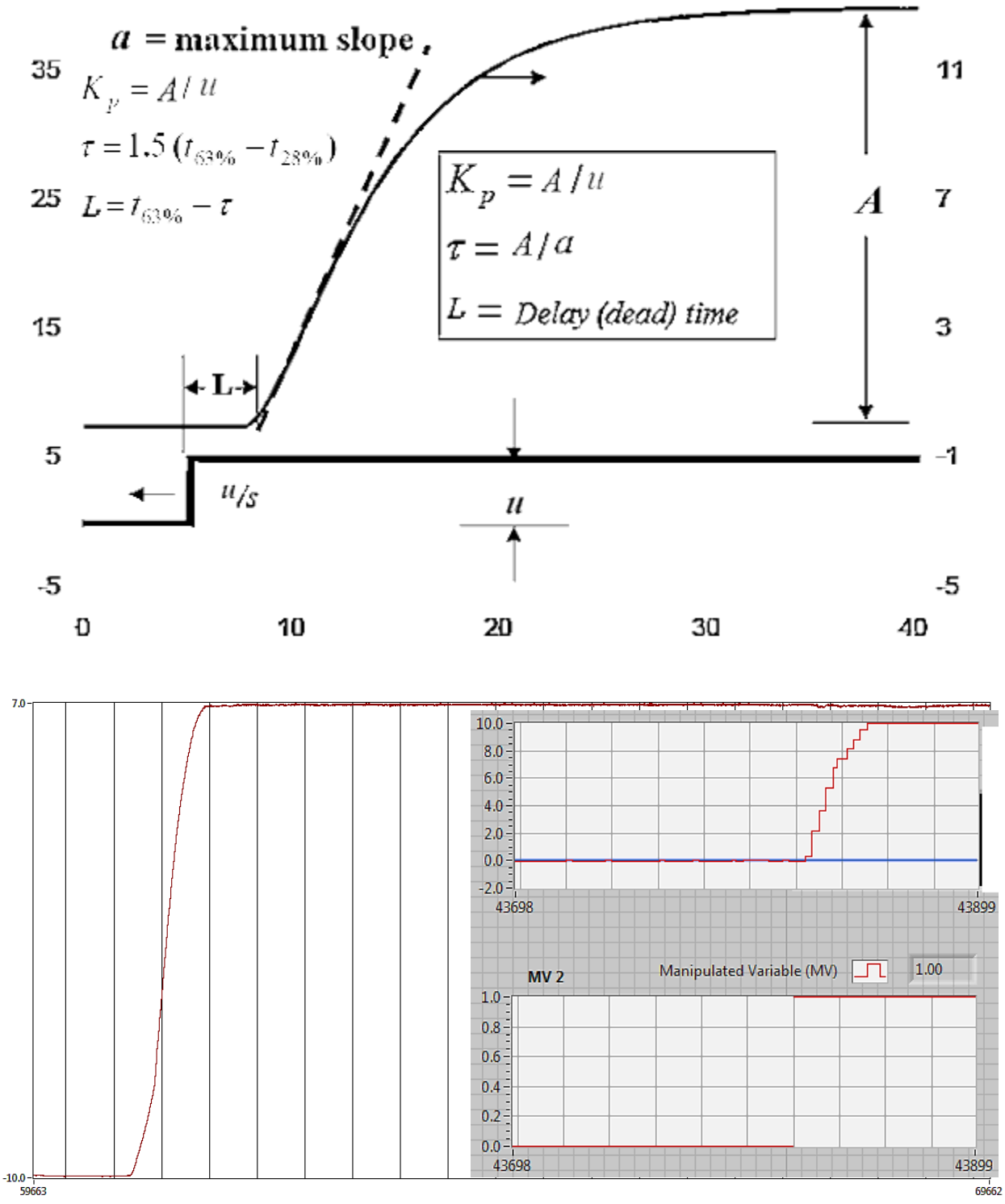


Fig 4.20 (a) Empirical RT step response modeling, the 1st order step dynamic response of the IR distance sensor, (b) bottom left, and (c) the analog beam angel's (after signal conditioning and differential amplification)

4.3. Mechanical and Mechatronics Design and Calculations

This section illustrates the basic calculations in which the mechanical/mechatronics parts of the built RT ball on beam plant takes place. Two main queries are to be answered here: the actuator (sizing) and materials selection. Actually, the two matters correlate with each other as both depend on several design parameters: torque/speed requirements of the system and inertia ratio, along with available voltage, motion profiles, power, resolution, and cost constraints.

Starting from the most important mechatronics requirement of all, torque. It is essential in the ball on beam applications because basically in order to be able to control the ball quicker than, or as rapid as, its speed in RT requires maintaining minimum amount of rotating acceleration. Recall that as introduced in Chapter 2, as in (13), the torque of a rotating actuator is a function of its angular acceleration and mass moment of inertia. In other words, torque is directly proportional to acceleration, which implies that increasing one would lead to the increase of the other. Usually, mass moment of inertia is a fixed parameter where both acceleration and torque vary as seen before in Chapter 2.

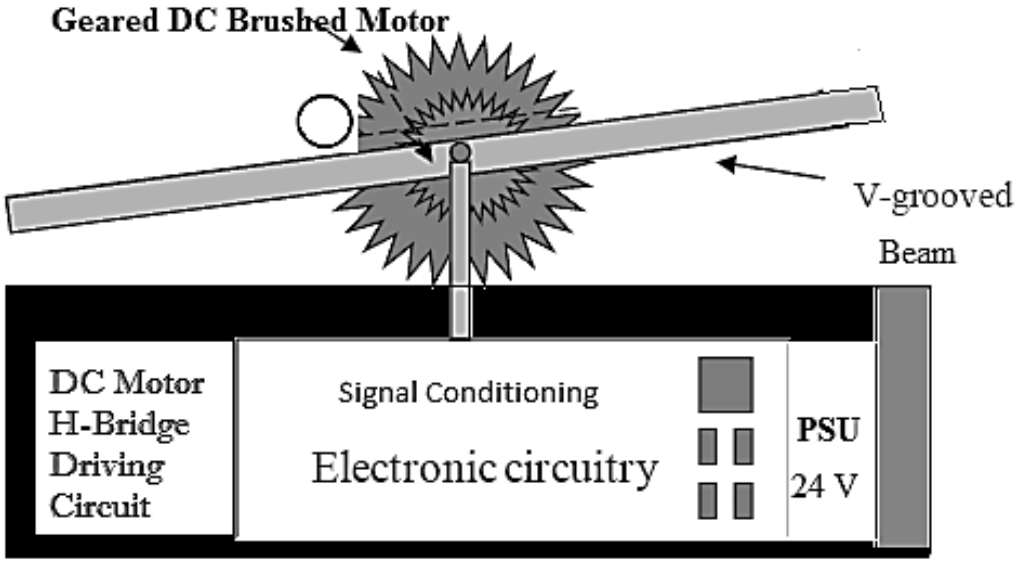


Fig 4.21: Mechanical and mechatronics parts of the ball on beam (taken from [90])

This problem is then formulated initially as to select a suitable DC motor available that could satisfy the acceleration, therefore torque desired. Fig 4.19 exemplifies the initial guessed actuator that could be thought of in such applications, a DC brushed servo geared motor. Intuitively, selecting a geared DC servo motor benefits increasing the torque range of the actuator while not sacrificing cost. The advantages of attaching gears to outputs of DC motors are represented in the following relations:

$$Output\ \omega\ = Motor\ Speed/N \tag{155}$$

$$Outout\ \tau\ = Motor\ torque \times N \tag{156}$$

$$Output\ J = \ Motor\ Inertia\ \times N \tag{157}$$

Since the regular behavior of the ball on beam system rotating practice is not spinning continuously, then speed is less important to be paid attention to, and so the torque arises as the most vital parameter in the ball on beam application. In fact, reducing speed as a result of adding a gearbox to the motor's shaft output may guarantee better speed control and manipulation on the continuous jumps and stops in such an application. In other words, a gearbox attached to a DC brushed is as an economical solution adding more output torque to the actuator. Consequently, the problem statement initially derives as searching for an economical DC brushed gear motor that satisfies the torque requirement.

A rough estimation of how much average torque is needed can be calculated by using the simple equation (13) targeting for the desired acceleration. Since this desired acceleration is difficult to be measured accurately, then it appears logical to convert an angular rate of change of velocity that is at least equivalent to gravity, g, in linear motion, after imagining a ball with mass, M_{ball}, falling. Therefore, converting g to an angular form would provide a good guess on how much desired angular acceleration, $\dot{\omega}$, is needed. It is known from basic physics background, a

linear acceleration is equivalent to a lever arm, r, times the angular acceleration as in the following equation and Fig 4.20,

$$a = r\dot{\omega} \tag{158}$$

Therefore,

$$\dot{\omega} = \frac{a}{r} \tag{159}$$

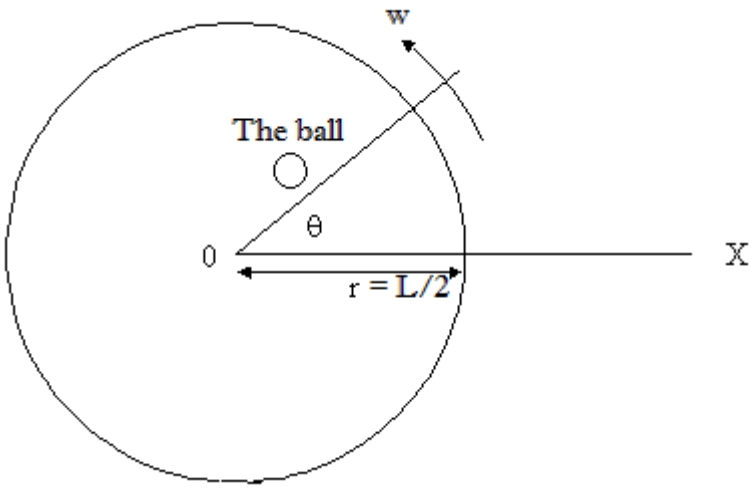


Fig 4.22: The relation between angular and linear acceleration

In this ball on beam plant, r equals to half the beam length,L_{beam}, so,

$$r = \frac{L_{beam}}{2} \tag{160}$$

Hence, to calculate the desired acceleration, let $a = g$ and substitute (160) in (159), then

$$\dot{\omega} = \frac{2g}{L_{beam}} \tag{161}$$

Where the numerical values of g and L_{beam} are given in Table 3.2. As a result of calculating those values in the equation, $\dot{\omega} = 31.36 \times 2\pi$ = 196.95 Rad/s^2which is the initial desired acceleration for the application. The second parameter to be plugged in (13) to roughly calculate the average torque required is the mass moment of inertia of the load on the actuator (motor) which is introduced in Chapter 3 as the summation of both the beam's and ball's inertias such as

$$J_{load} = J_{ball} + J_{beam} \tag{162}$$

Recall that J_{ball} can be calculated from (94) presented in Chapter 3 as

$$J_{ball} = {}^{2}/_{5}\ M_{ball}\, {R_b}^{2}$$

Where several configurable values M_{ball} and Rb can be thought of according to various types of materials available in the lab and that could be used for this experimental purpose; meanwhile, one standard ball parameter is used initially and given in Table 3.2. The values in the formula are plugged in as follows

$$\mathrm{J_{ball}} = {}^{2}/_{5}\ (0.02272\ \mathrm{kg})(0.015\ \mathrm{m})^{2} = 2 \times (10)^{-6}\ \mathrm{Kg.m^2}$$

Correspondingly, J_{beam} can be considered a regular rectangular beam of length, L_{beam}, subsequently calculated from the given equation [51]

$$J_{beam} = {}^{1}/_{12}\ M_{beam}\, {L_{beam}}^{2} \tag{163}$$

Note that in practice M_{beam} has some weight additives such as sensor mounted, motor couplings, and the mechanical piece known as the transmission between the beam and the couplings. The mass of the beam plus its additives are all measured using very precise weight measuring instruments available in UNT Engineering Technology and Material Science labs. As a result, the average observed M_{beam} is 0.5 kg. Please notice Fig 4.21.

The calculation appears as follows

$$\mathrm{J_{beam}} = {}^{1}/_{12}\ (0.5\ \mathrm{kg})(0.625)^{2} \approx 0.0163\ \mathrm{Kg.m^2}$$

Substituting the two calculated values J_{beam} and J_{ball} above in (163) produces

$$= 0.015\ \mathrm{Kg.m^2} + 2 \times (10)^{-6}\ \mathrm{Kg.m^2}$$

$$J_{load} \approx 0.0165\ \mathrm{Kg.m^2}$$

Hence, the roughly calculated average amount of torque required can be obtained after substituting the values in (13),

$$T = I\dot{\omega}$$

$$T = \mathrm{J}_{load}\dot{\omega}$$

$$\mathrm{T} = (0.0165\ \mathrm{Kg.\,m^2})(196.95\,\frac{\mathrm{Rad}}{\mathrm{s^2}}) \approx 3.25\ \mathrm{Nm} \approx 460\ \mathrm{oz.in}$$

As a result of the first output of such investigating calculation is to look for a cost effective DC brushed geared servo motor that can generate ≥ 460 oz.in torque. However, as declared before on several occasions, torque is not the only key parameter a designer should take account when selecting a proper actuator for his motion control application, especially as the ball on beam. This research discerns that, aside from torque, inertia ratio emerges as a critical factor that determines the performance of the dynamics of the mechanical system, but is often overlooked by the designers as also seen in all previous literature implementing the ball on beam control problem.

Inertia ratio is defined as the ratio of the load's inertia divided by the motor's rotor inertia divided by the square of the gear reduction [102],

$$Inertia\ ratio\ = \frac{J\ (load)/J\ (motor)}{(N)^2} \tag{164}$$

Where $N = Gear\ ratio$ = 65.5:1 is a given motor's specification [114], the moment of inertia of the motor is nothing but J_m given in Table 3.2, and finally J_{load} is calculated above as maximum of 0.015 Kg.m^2; consequently, the inertia ratio of the designed mechatronics system is

$$Inertia\ ratio\ = \frac{0.0165\ \mathrm{Kg.m^2}/7.062\ \mathrm{x}\ (10)^{-6}\ \mathrm{Kg.m^2}}{(65.5)^2}$$

$$= \frac{2336.5}{4290.25}\ = 0.545 \approx 1$$

Which is closer to the ideal ratio 1:1 discussed earlier in Section 3.1. Calculating inertia ratio allows selecting suitable materials for the physical plant to be built. So, the objective relies on minimizing the plants, mass as much as possible. The beam and motor coupling, as well as stabilizing base materials were chosen carefully for that purpose. As a result, an aluminum V-

shaped beam, steel couplings, and heavy stable metal are selected for the design.

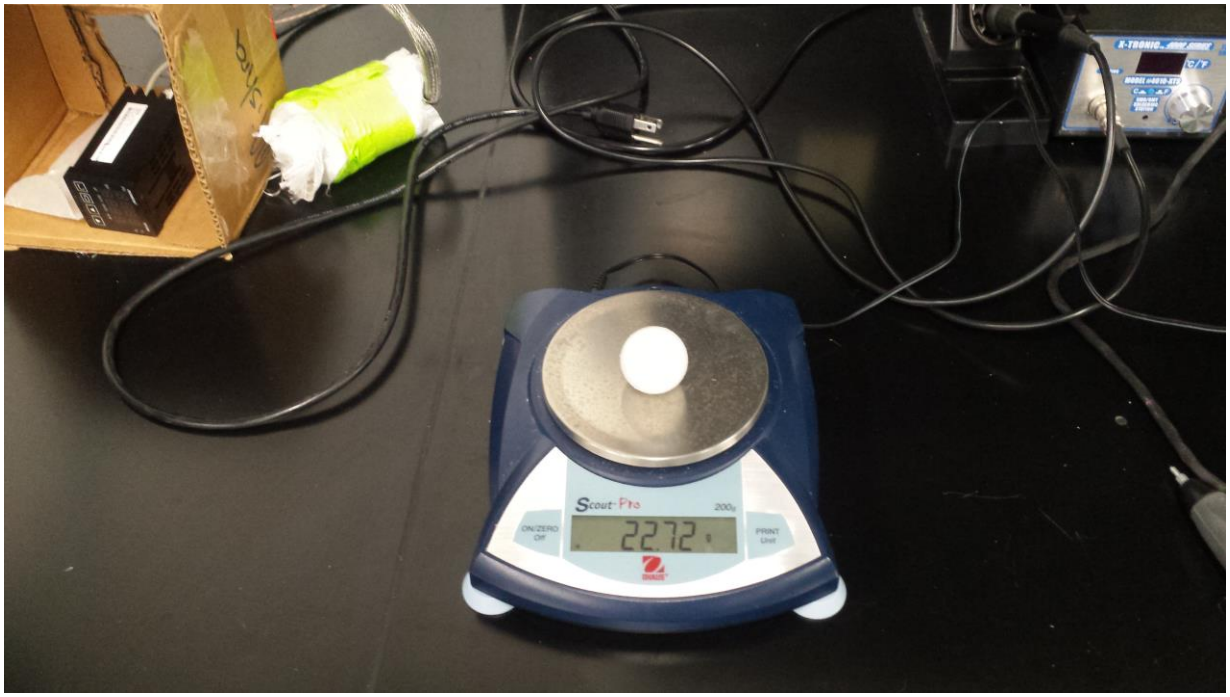


Fig 4.23: The ball's weight being measured at UNT Engineering labs

Satisfying the two most vital factors of servo selection, torque and inertia requirements, and complying with other requisite elements such as power, together forms the image of the final decision. Power requirements of the system suggests discussing further the drivers chosen in the design. As introduced in Chapter 2, the core driver functionality emerges as taking control voltage signals as inputs and then amplifies and produces current as output driven to the motor coils in addition to H bridging switching. Usually power of motors varies with varying currents given that voltage is always fixed as a supply value. It implies that the selected servo motor should match the supply voltage of the driver chosen as well. Several scenarios are configured in this design. One of them is the built in NI brush servo driver that can be plugged into the cRIO RT controller as seen when discussing the system's components. This driver accepts 24 V, so the selected motor should be of the same amount of power. Thereby, the magnification gain

factor, *G*, used in the model before and found in Table 3.1, is calculated as per:

$$\pm 24\ \text{V} = G(\pm 10\ \text{V})$$

Hence, $G = 2.4$, where the ± 10 is the controller AO's range.

Assuming all the above is feasible design parameters and conforming with cost constants of minimizing the price as much as possible results in finally selecting the DC brushed servo geared motor actuator, Ametek Pitman DC Gearmotor**,** GM9236S027-R1-SP appears in Fig 4.8, which has an average continuous output torque of 490 oz-in. This motor emerges as the best, if not the optimal, solution available; however, there appear several non-ideal effects cause the RT response to not exactly match the theoretically expected from such an actuator:

- Because of the cost constraints, the attached resolution of encoder (at the back of the motor) is 500 CPR, which is below average amongst other available values, i.e. 1000, 1500, and 1520 CPR.
- Adding a gearbox to the servo system aids in a number of advantages declared above, yet, gear motor usually encounters two major accuracy and performance drawbacks: first, mechanical backlash due to transmission between motors' shafts and gears, and second, gear efficiency, ε, reduces torque, and therefore power such that:

$$Output\ Torque\ = J_m \times N \times \varepsilon \tag{165}$$

The signal conditioning part of the system is covered in a previous separate section. Note that all theoretical designs and calculation, as well as materials selection and collection were autonomously performed. However, due to lack of amenities, tools, and resources, the mechanical and other handiwork necessary for attaching physical plant's parts together such as welding, cutting, edging, etc. were mostly performed by the help of Engineering Technology Department's mechanical workshop and technicians referred to the acknowledgment.

4.4. Control Strategy

As declared before, one of the objectives of this work is to study and compare the effects of the different technologies and algorithms on control. Here, the ball on beam automation and control system is implemented by two computing approaches: FPGA and RT controller-scan mode. Fig 4.22 below describes the general control strategy employed in BD form. The BD shows all stages of the instrumented control system as well including the plant (the ball on the beam), differential feedback measurement of the ball position, signal conditioning and conversions, ADC, cascaded and feed-forward controllers, and finally DC motor servo system as an actuator and an inner loop.

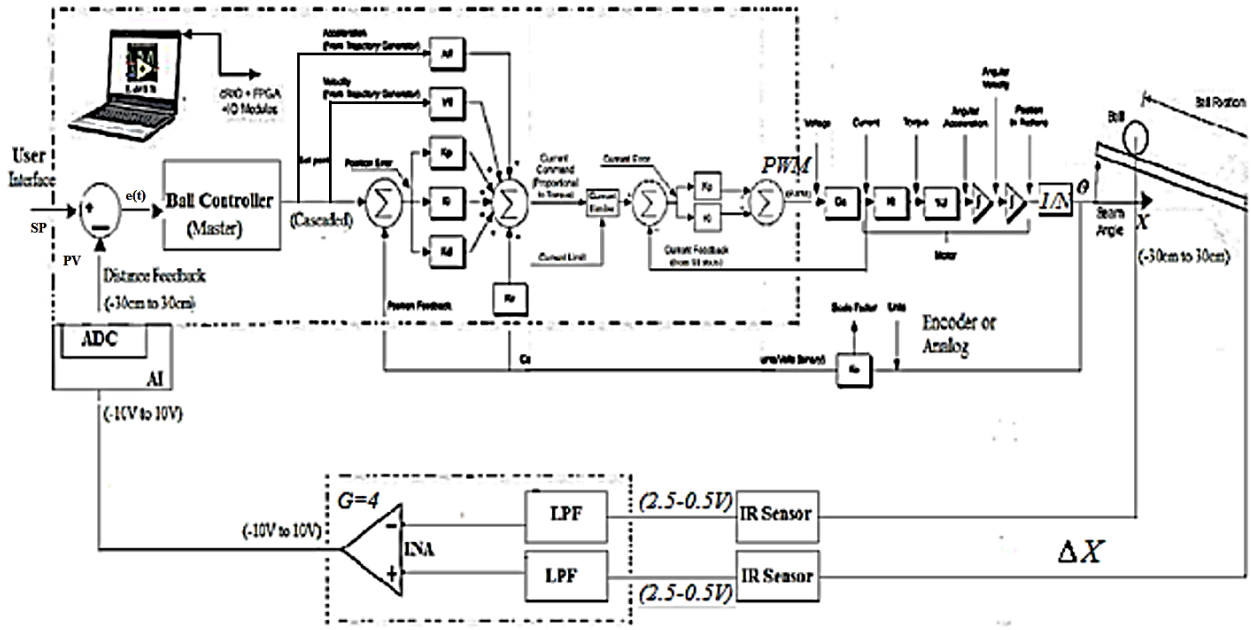


Fig 4.24: The general BD of the design and implemented system

4.4.1. FPGA mode

The NI c-RIO controller is built in a way that allows interfacing the main CPU with the FPGA on board so that each can execute suitable functions. This research introduces the ball on beam control problem using a complete FPGA control, unlike [88], which used the technology to

interface I/Os only and implemented the main control on a DSP chip. Fig 4.23 below illustrates the BD of the design implementing the complete cascaded system as a cRIO FPGA map while the supervisory monitoring and HMI interface VI is kept as the main CPU (scan engine) for RT displaying, trending, and SP changes. The BD below shows all the details of the system in TF form including the process plant, measurement feedback and signal conditioning I/O and DAQ, actuator (DC motor) and driver containing its internal current feedback, loop, and finally the series and combination of cascaded and feed-forward controllers as a robust design.

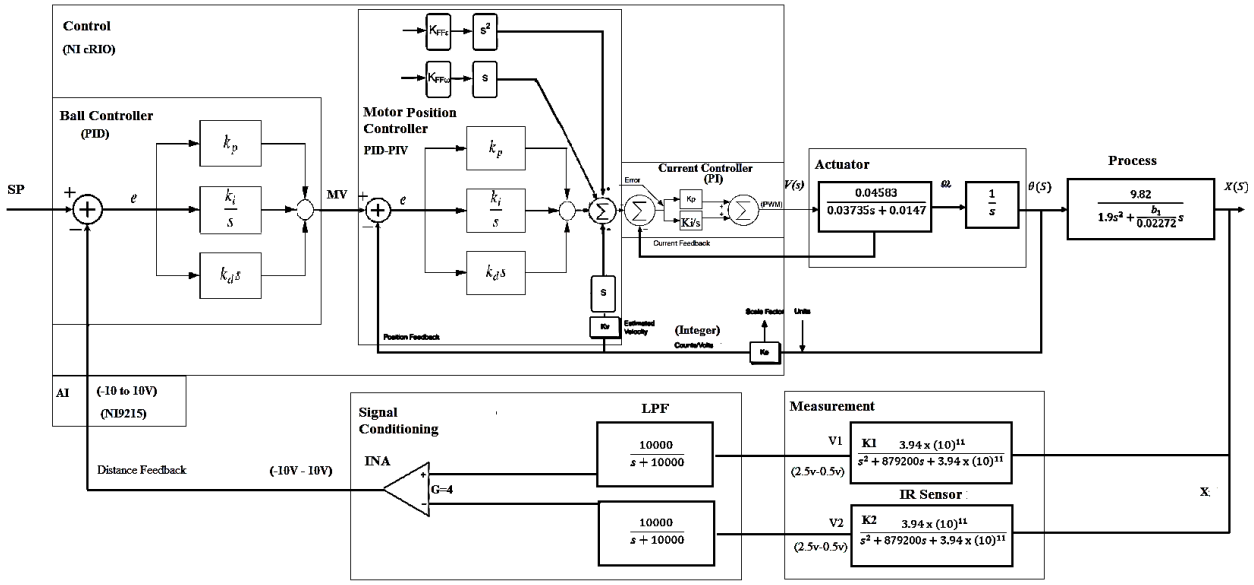


Fig 4.25: The BD of the proposed ball on beam control system design on an FPGA

Since the NI soft motion module is the software tool used in this control program, most logic in the designed VI should be converted in the 16.16 fixed point format [19], which represents precision in a 32-bit integer covering the range -65536 to 65535 counts. The upper 16 bits represent the part of the number before the decimal point while the lower 16 denote the part after. This is because the main unit in such BD environment is encoder counts explained and calibrated earlier. Remember the motor position loop's SP range should not exceed 30 degrees, which corresponds to about +/-10000 counts for direct or reverse directions.

Therefore, scaling and synchronization should take place in all other connected logic blocks and pieces. For example, the master PID loop's (ball distance) output is scalable and configured to cover the -/+ 10000 counts range, whereas the measurement input, +/- 10V is represented as a fixed point number, as a result of the AI block acquisition, then converted into a 32-bit integer data type to be compatible with 16.16 format mentioned above. After that, for higher resolution range out of the 10 V represented as an integer, this feedback input is multiplied by a factor of 1000 before being wired to the outer PID block (distance loop).

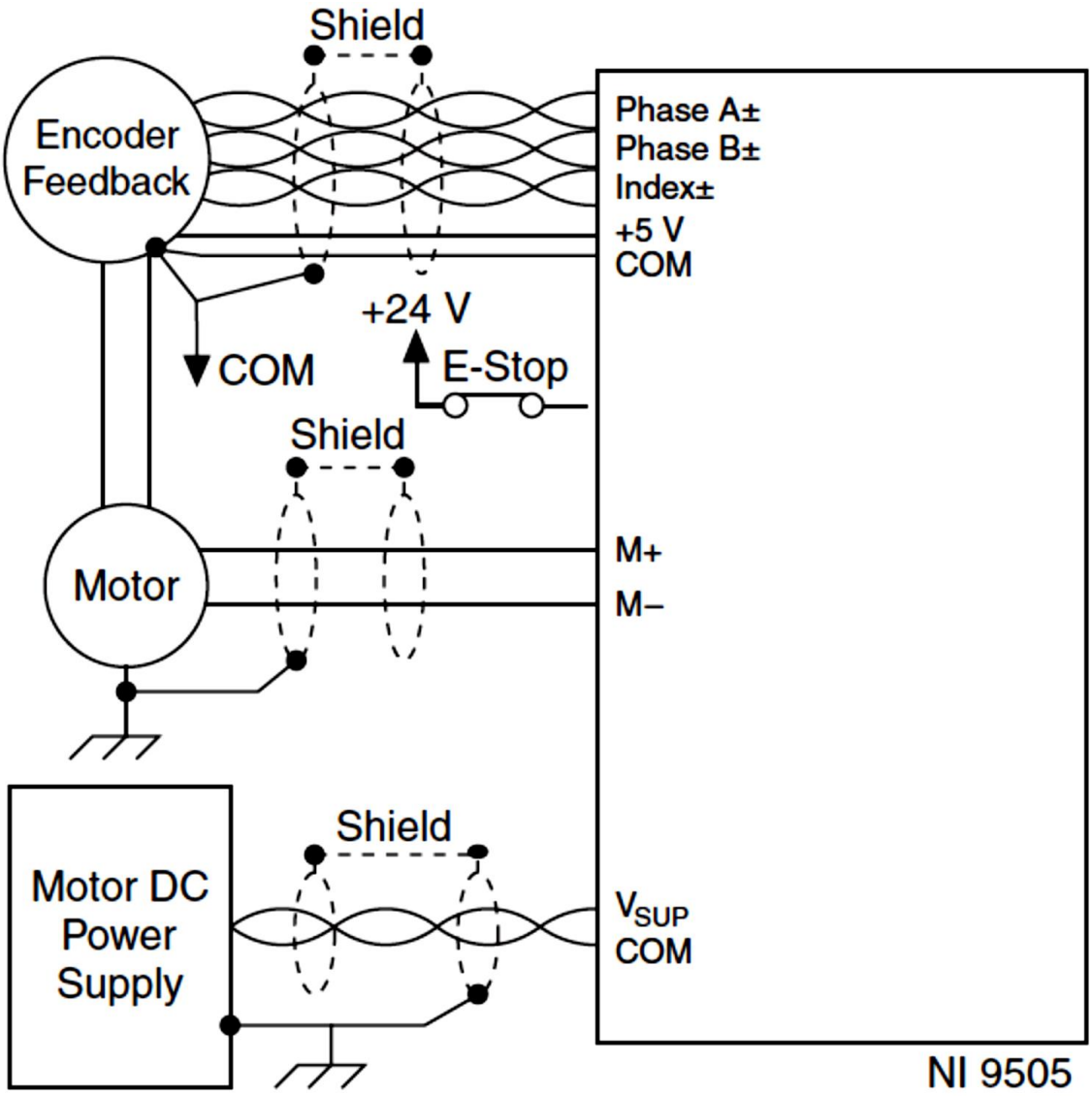


Fig 4.26: Wiring and connections between the DC servo motor and driver, NI9505 [110]

Fig 4.25 next shows the designed VI BD of the FPGA control portion of the system. But before describing this VI, the hardware loop is shown in Fig 4.24 above containing the main wiring and connections between the actuator (DC servo motor) and the driver, NI 9505 besides the PSU, 24V. As stated several times before, one advantage of the FPGA technology is the possibility of optimizing the control logic program in a highly parallel manner, meaning multiple RT tasks could be executed all at once. Fig 4.25 indicates using 9 while loops, 4 main and inner single cycle loops, and 3 rate configured loops to implement the control strategy and BD shown in Fig 4.23. A single cycle while loop executes in one FPGA clock rate, 40MHz, click or sample period of 25 nano-seconds! Most of the while loops take as little execution time as possible; some of them require one FPGA clock cycle to execute while others, containing more calculations and logic, may need more time.

The two while loops at the upper right corner of the VI contain the two AI blocks acquiring the ball distance and beam angle (in analog mode) feedback measurements (a). Next to (a) is (b), FPGA implemented incremental quadrature counter including A and B encoder DIs besides the index input. The counter logic functions as resettable up/down counter utilizing the two phases and the direction. This counting value is sent to the master PID block, NI FPGA discrete, distance loop as ball position feedback in a 32-bit integer data type. (c), (d), and (e) in the middle show the three main cascaded controllers, ball distance (PID), motor position (PID-PIV), and finally the current PI loop as the most inner. For the NI cRIO's FPGA PMW duty cycle specifications of 2 KHz, the output range of the beam's angle position PID-PIV is scaled to 2000 within the 16-bit integer (-32,768 to +32,767) PV/SP range. All single timed, while, and controller loops interconnect by local variables to avoid long wiring and optimize the mapping by minimizing time delay, as per [122]. Appendix 7 displays the entire ball on beam VI under FPGA mode.

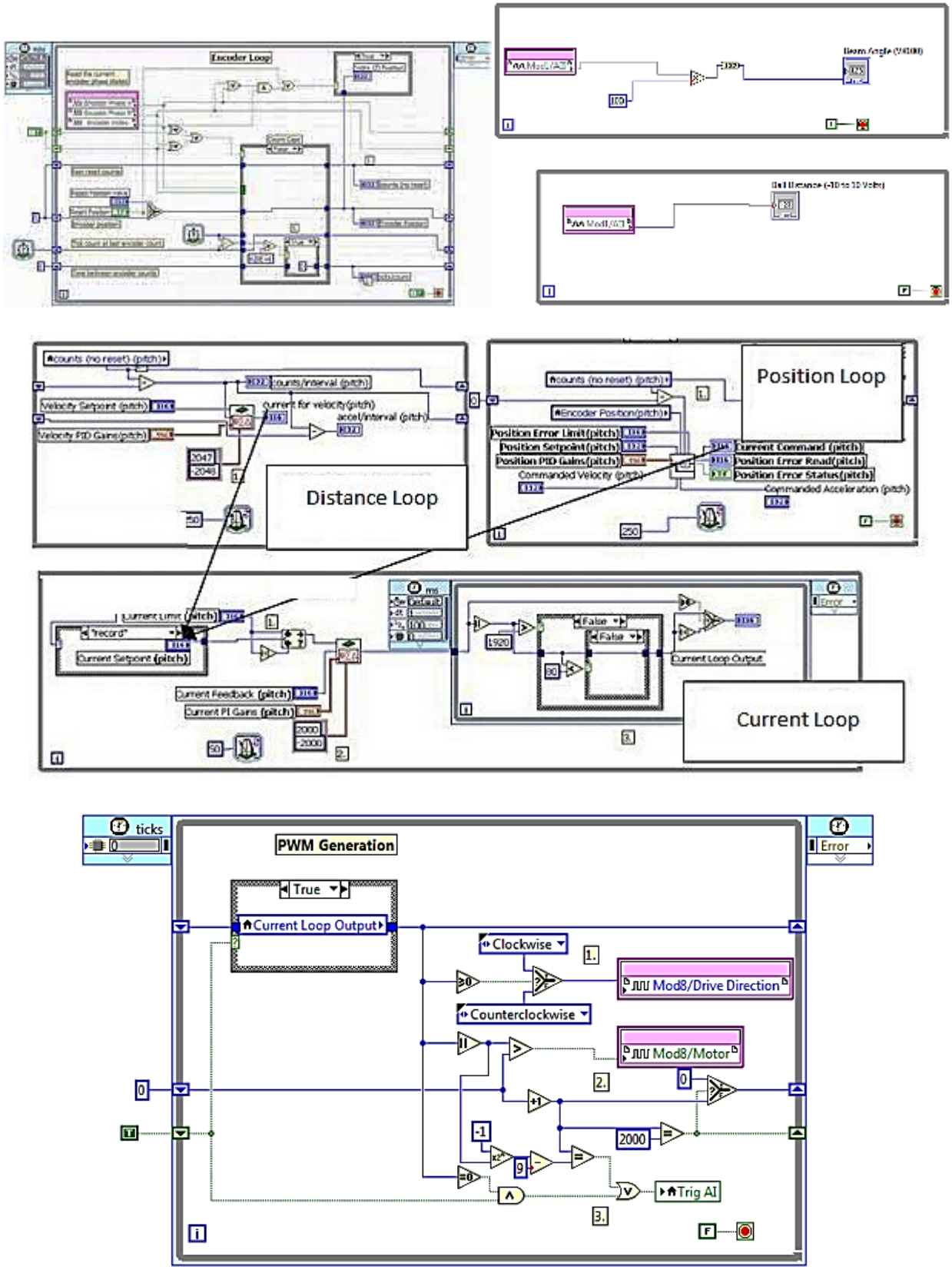


Fig 4.27: The main Ball on Beam's LabVIEW cRIO FPGA environment program and loops

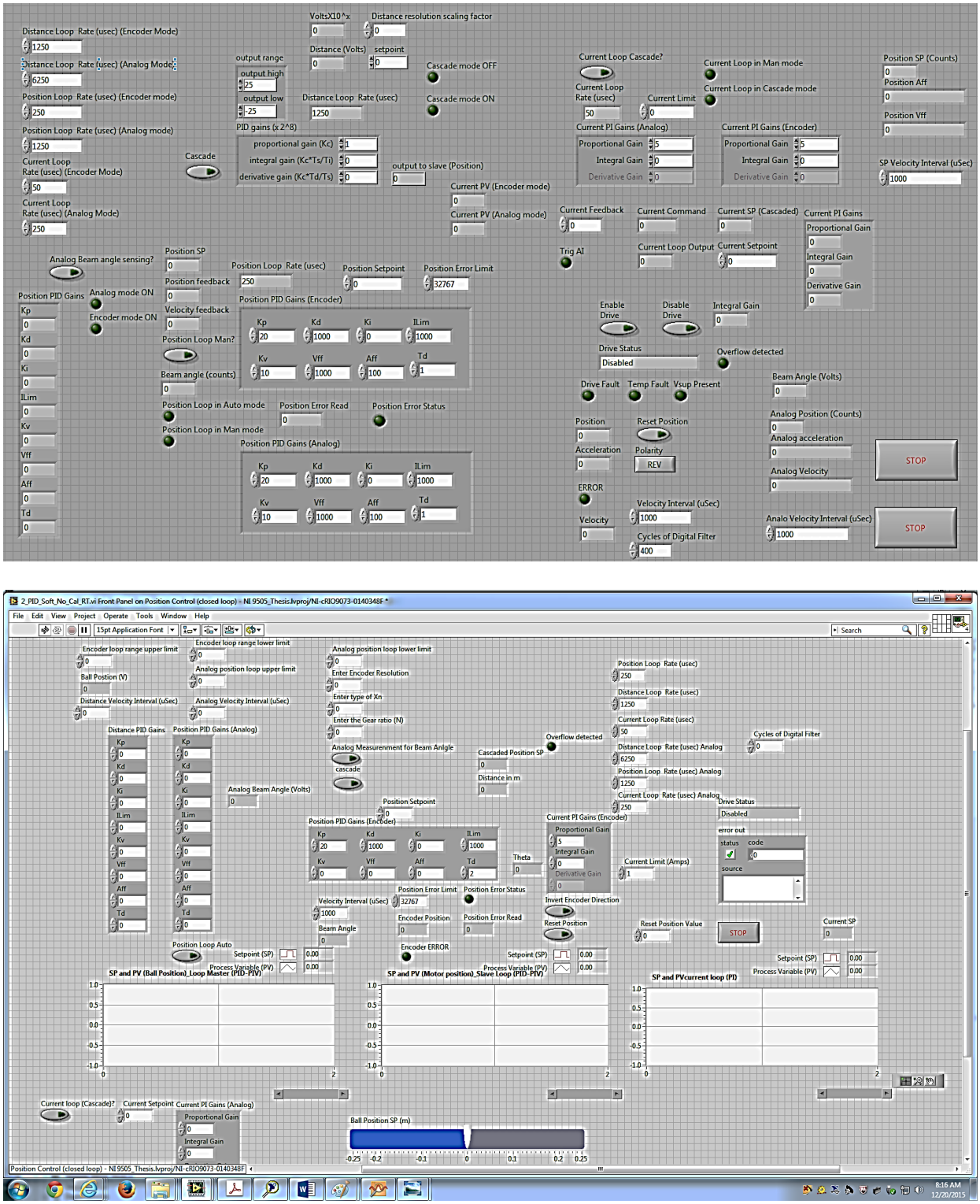


Fig 4.28: (a) The ball on beam FPGA VI front panel (upper) and (b) its corresponding supervisory host application (HMI) VI in real scan time

Fig 4.26 (a)/(b) above shows the front panels interface of the system. The upper portion represents the direct FPGA interface front panel, which is crucial for tuning control loops and testing the functionality of the mapped program, especially after deployment. (b) underneath shows the supervisory host application user interface VI (HMI), which represents the SCADA level of the system. Unlike the FPGA VI that executes the actual control logic, this one is downloaded in the main process (RT-controller) board. In other words, the host VI reads from and writes to the FPGA program as a master entity which is needed to display readings and trends in RT, and scan engine's sampling period of 10 ms, rather than FPGA clock.

4.4.2. RT scan (processor) mode

Besides the introduced FPGA mode, this paper implements the ball on beam using standard MCU based RT-controller design as regular industrial application for the sake of assessing new technologies through comparison study. Several advantages of utilizing processor mode design in such a complex system is the flexibility, simplicity, and memory capacity to occupy more advanced calculations and blocks. LabVIEW RT environment offers a large set of function blocks libraries, which enable applying a variety of algorithms. In addition to comparing FPGA with RT-controller outcomes, it is intended to experiment different advanced control algorithms, introduced earlier, in higher order systems as the ball on beam.

Neither current sensing and control nor encoder based tilt measurement appear to be present in this approach since an external motor driver is required besides relatively slower computational speed, 100 Hz sampling. Hence, only the developed analog beam angel measurement and regular main ball on beam double cascaded control strategy are used here. Fig 4.27 illustrates the main program within the VI and the front panel of this design. The VI comprises of two while loops, one contains the two PIDs while the other includes the selected auto-tuning method.

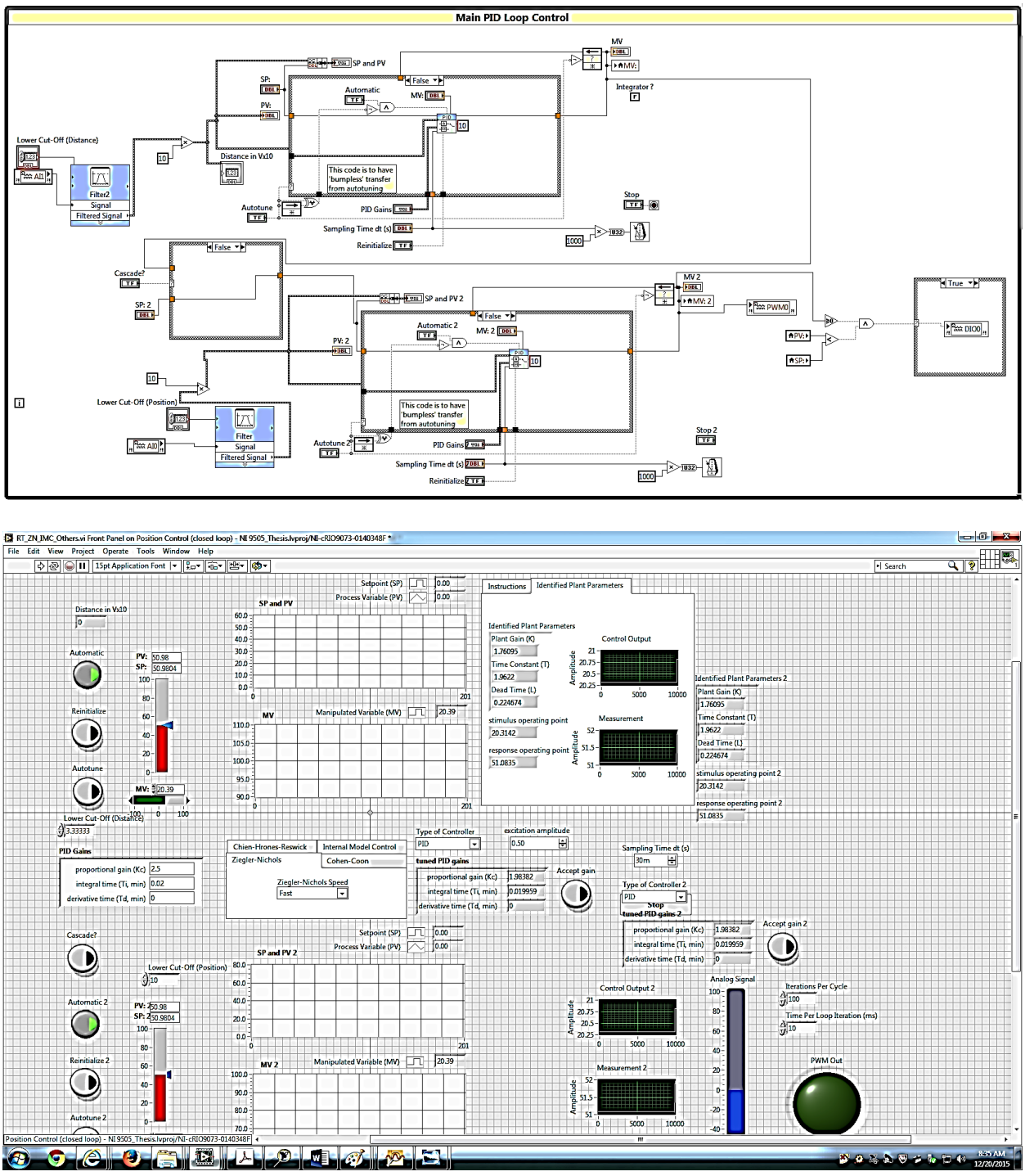


Fig 4.29: (a) The main FBD of the cascade loop and (b) front panel in RT scan mode

The program allows selecting between various algorithms on the top of PID and switching between modes as automatic, manual, cascade, and auto-tune. If for example an auto mode is selected, manually tuning parameters will be activated. The VI encloses PWM output to drivers.

4.5. Experimental Results

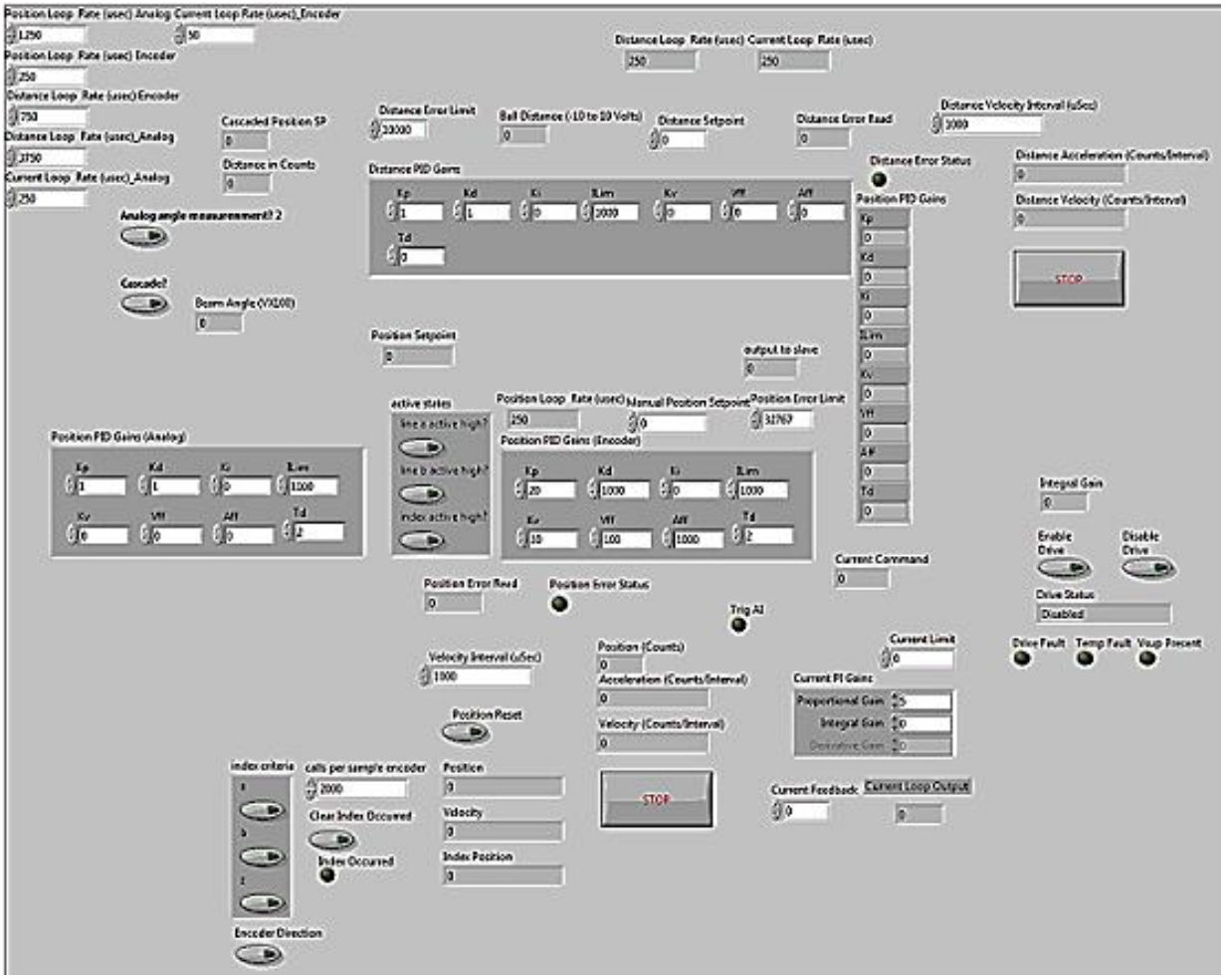


Fig 4.30: The FPGA VI in run mode

Fig 4.28 shows the FPGA VI in run mode. The user should make sure permissive conditions hold true before running the program by enabling the driver first, and he can set tunable parameters. However, once the host application VI is run, it will override all the parameters in the FPGA one. The following displays RT control results of the two outputs of the system, beam angel and ball position, with respect to their reference SP inputs using the two approaches: FPGA and RT-controller (processor) modes.

4.5.1. FPGA mode

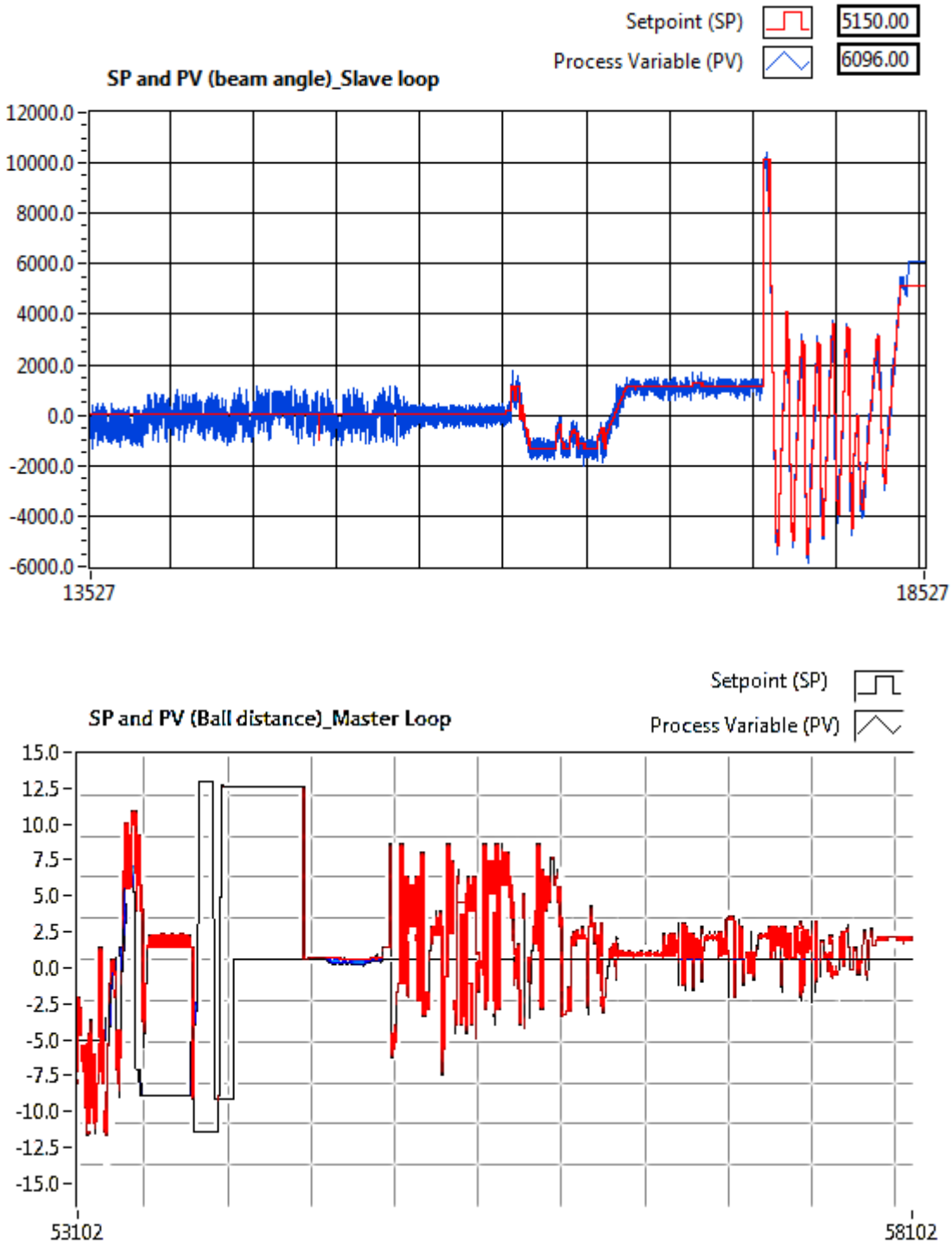


Fig 4.31: (a) Beam angle (motor position) and (b) Ball distance (lower) PV/SP RT control trends

The two graphs above display the RT control results with respect to the SPs of the beam angle at the top and the ball position. The motor position loop runs on 250µs timeline and

represents quadrature count units from -10000 to 10000 counts (equivalent to -30 to 30-degrees angel) while the distance loop executes in 750 µs, and its reading denotes the voltage PV of the ball position within (-15 to 15 V). Both graphs were extracted using the encoder feedback method. The current loop, which is the innermost, is controlled by a PI controller of gains 5 and 1 of the limiter (- 200 to 200 integer values of the 16.16 fixed point, which is equivalent to around 1 A). The master controller (ball position) follows PID algorithm of gains P = 250, I = 10, and D = 750 of multiples of $10\,V \times 10^3$ counts gain units. The inner motor positions (beam angle loops) follow the PID-PIV-Vff-Aff strategy control parameter, which are P = 20, I = 1, D = 1200, V = 10, Vff = 100, and Aff = 100. Unless otherwise limited, all the parameters accept the range of 16-bit integer representation of 16.16 fixed point arithmetic of (-32,768 to +32,767).

4.5.2. RT scan mode

The alternative developed design is the RT-controller (processor) approach, which is based on the scan engine sampling period, 10ms, and follows sequential program execution in general. LabVIEW RT and control and simulation modules offer a large library that contains a variety of classical and advanced blocks. This experiment examined several control algorithms and tuning methods under the PID-PID cascade control strategy such as open loop step response, ZN, Cohen-Coon (PID), smith predictor, and IMC, and observations were noted. although, few attained successful RT control.

The RT graphs (a) and (b) of the following figure, Fig 4.30, demonstrate the two outputs of the system, beam angel in volts (-10 to 10 V), and the ball position PV/SP within the same range. ZN of PID parameters, P = 0.463919, I = 0.23881, and D = 0.05972, and IMC's 0.257151, 0.003607, 0.000721 respectively, are the two most successful algorithms for motor control. The master distance loop was tuned using simple step response method.

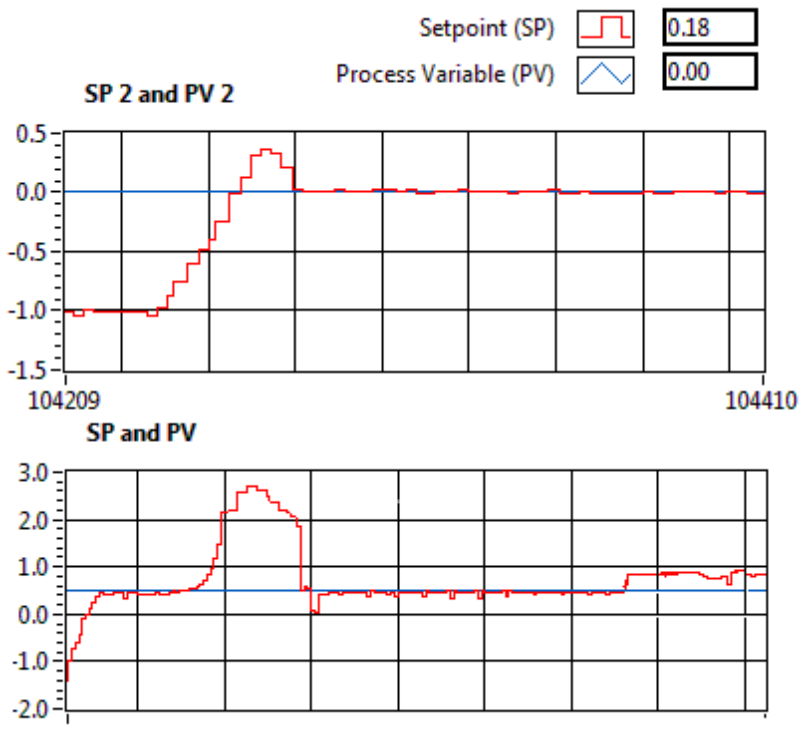


Fig 4.32: The PV/SP RT trends of (a) the beam's angle (above) and (b) ball position (below)

4.6. Conclusions and Summary

This chapter proposed the ball on beam control system problem in the context of this research, representing level 0 (process plant) of the industrial automation pyramid architecture. The design also included level 1 (sensors and actuators), which is IR distance sensors, digital quadrature angular positioning encoders, optical sensors (for analog beam measurement), and DC (servo) motor. Level 2 comprises of a distributed CN of NI cRIO RT-controllers while the supervisory HMI (SCADA) is PC desktops, and monitors represent level 3. The system can be expanded in the future to cover more processes to derive hybrid by including subsystems PLCs and WSN as added field communication technology. With regard to this research, one remarkable advantage of NI cRIO platform is that it offers offering two environments, RT scan mode utilizing regular MPU based control managed by an RTOS and newly the emerged technology, FPGA.

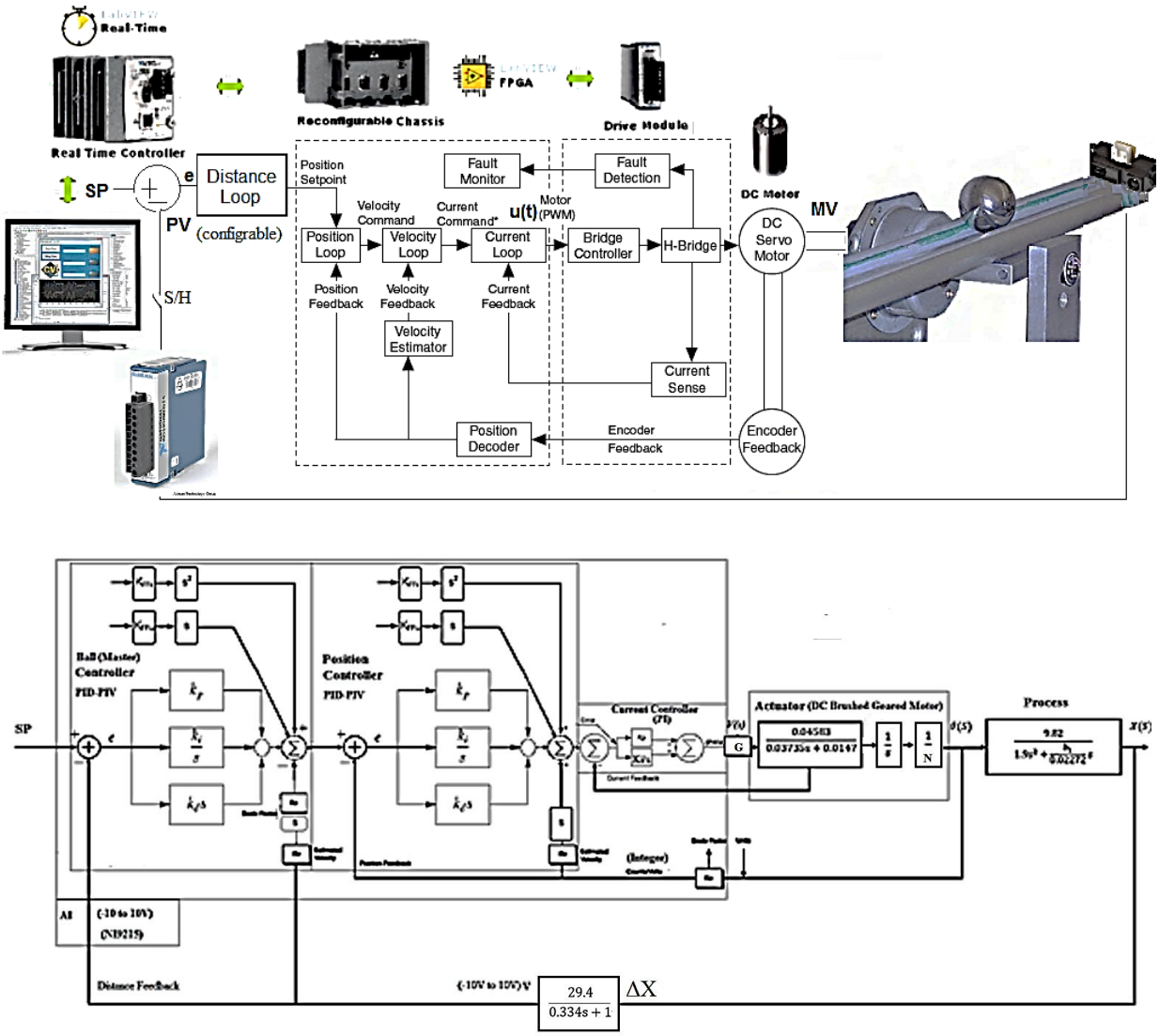


Fig 4.33: (a) The overall FBD of the proposed design and (b) its equivalent in Laplace domain

Fig 4.31 above summarizes the suggested design in FB form showing the main devices and components as well (a), and the detailed TF blocks, (b), including the process, feedback dynamic, controllers, and actuator (driver plus DC motor). (b) also suggests a possible improvement of adding PIV-Vff-Aff control method to the master controller as the current setting utilizes this type in the inner loop only from which results have been recorded so far. This design relied on several previous reports discussing the control and implementation problem of the ball on beam, yet the paper added several contributions such as:

- Considering the ball on beam control problem as in actual RT distributed industrial automation architecture which will not only assist in training and introducing students to such practical application but also permit a rich observed related R&D environment.
- This design has implemented the ball on beam control system as a full FPGA program as a first known trial. The FPGA technology provides a couple of advantages noted before.
- In addition to the cascaded PID-PID structure presented in previous literature, this design also introduced advanced robust techniques in both modes, FPGA and processor, used in similar motor control applications such as PID-PIV with Vff and Aff, which its robustness and performance effect over the system has been noticed and recorded, and IMC.
- In addition to regulatory level algorithms, the design included a sequential scheme for multiple selectable modes, safety, and device protection purposes signified in Fig 4.34.
- For the first time, the control structure also included an innermost current sensing and control loop, which has allowed better customization and manipulation over the proportional torque, which determines the amount of acceleration the DC motor travels.
- The chapter recognized the significance of mechanical and mechatronics part of the hardware design over the RT control performance by considering inertia ratio besides torque and adheres to the 1:1 rule recommended by motor manufacturers; hence, detailed calculations leading to selecting proper material and actuator were demonstrated, which have been partially ignored in previous work.
- The work studied pros and cons of two different computing technologies in control, RT-controller (processor), and FPGA for the sake of assessing the latter, which comparably offers a number of features, to be adopted in industrial applications. Results were noted and conclusions could be drawn in this matter, which will arise later.

- Because of several issues associated with encoder measurement, the design presented allows users to switch or choose between two beam angel measurement modes: encoder feedback and analog as appears in Fig 4.32(a) below as a unique feature this work adds.
- This design introduced the use of INA in the signal conditioning phase of the ball on beam instrumentation, which is a smart, simple, and cost effective way of amplifying.

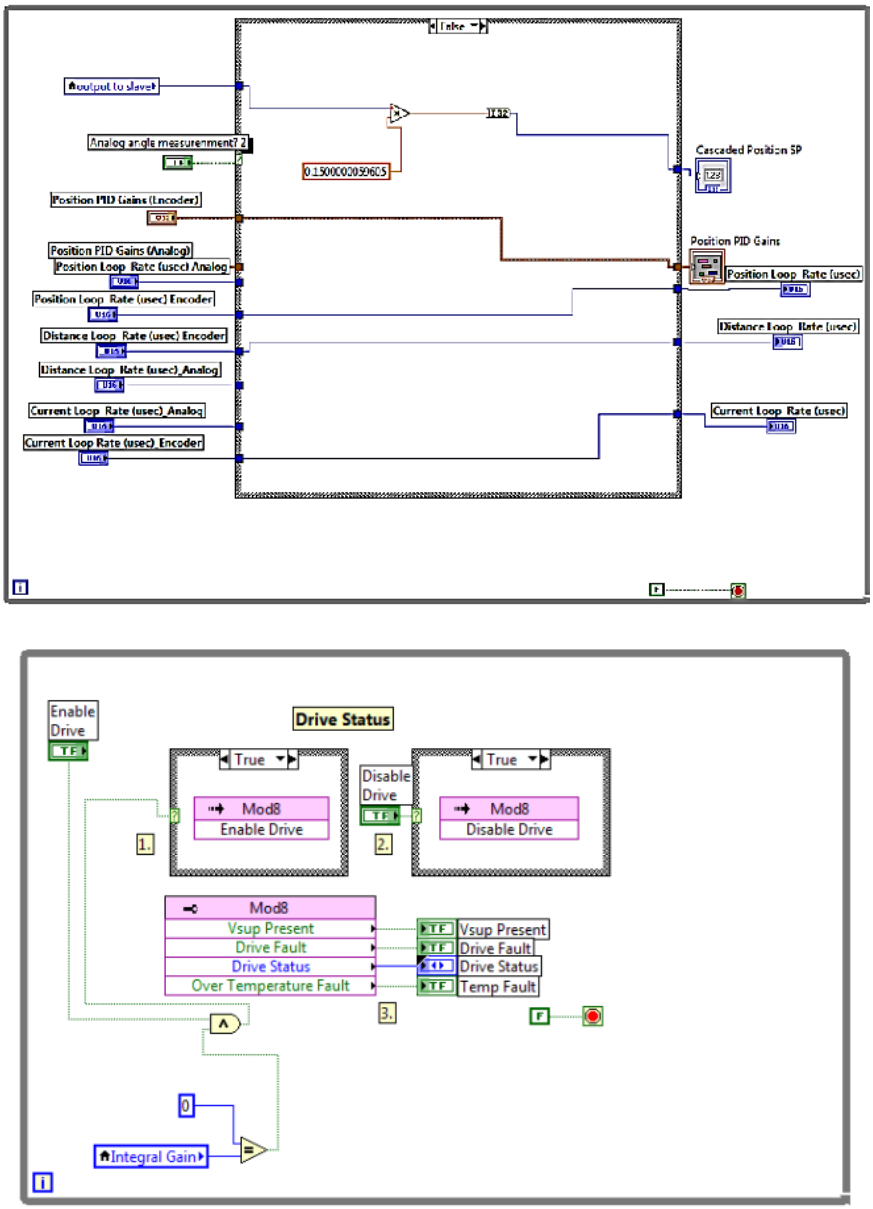


Fig 4.34: (a) The beam angle measurement mode selection and (b) overflow detection logic

On the other hand, a couple of problems have been encountered during the implementation:

- The mechanical setup of the plant has posed an immense difficulty preventing smooth progress of the system development. A small mechanical slack of the motor coupling led the system to be unstable countless times. This was resolved by rebuilding a customized plant in UNT Engineering Technology workshop. Also, the mechanical backlash due to gear transmission has degraded the control accuracy, especially in steady states. This problem could be addressed by replacing the current DC motor with a relatively larger (ungeared) actuator, which may stratify torque and inertia specifications; however, this will result in much higher cost.
- Several issues associated with FPGA control method were noted. First, the memory size of the Linux FPGA is limited and does not allow extended calculations, hence various advanced strategies cannot be realized unless this issue will be considered by manufacturers for future improvements. Second, the fixed point arithmetic of this FPGA is less precise than floating point system; as a result, comparatively less accurate RT control was expected. This fixed point precision lead to an additional problem, overflow, which was fixed at certain detectable segments of the control program. For example, Fig 4.33 (b) displays developed logic to indicate overflow status through monitoring the integral gain of the current loop, which has been observed to cause sudden motor jump up, consequently damage if it exceeds zero before enabling the motor driver. Finally, because of the high calculation bandwidth FPGA operates, final RT control accompanied some vibration and jitter compared to regular MPU outputs.
- Another issue is encoder's feedback noise and incorrect reading problems because of interference and higher bandwidth than the controller's. This issue was fixed by replacing shielded twisted pair cables, differential line driver, and a digital skip filter (Fig 4.33).

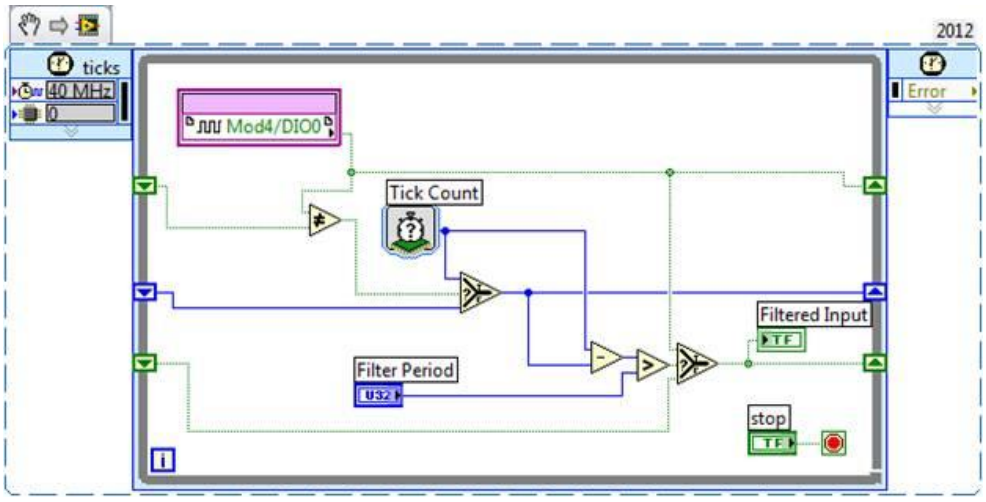
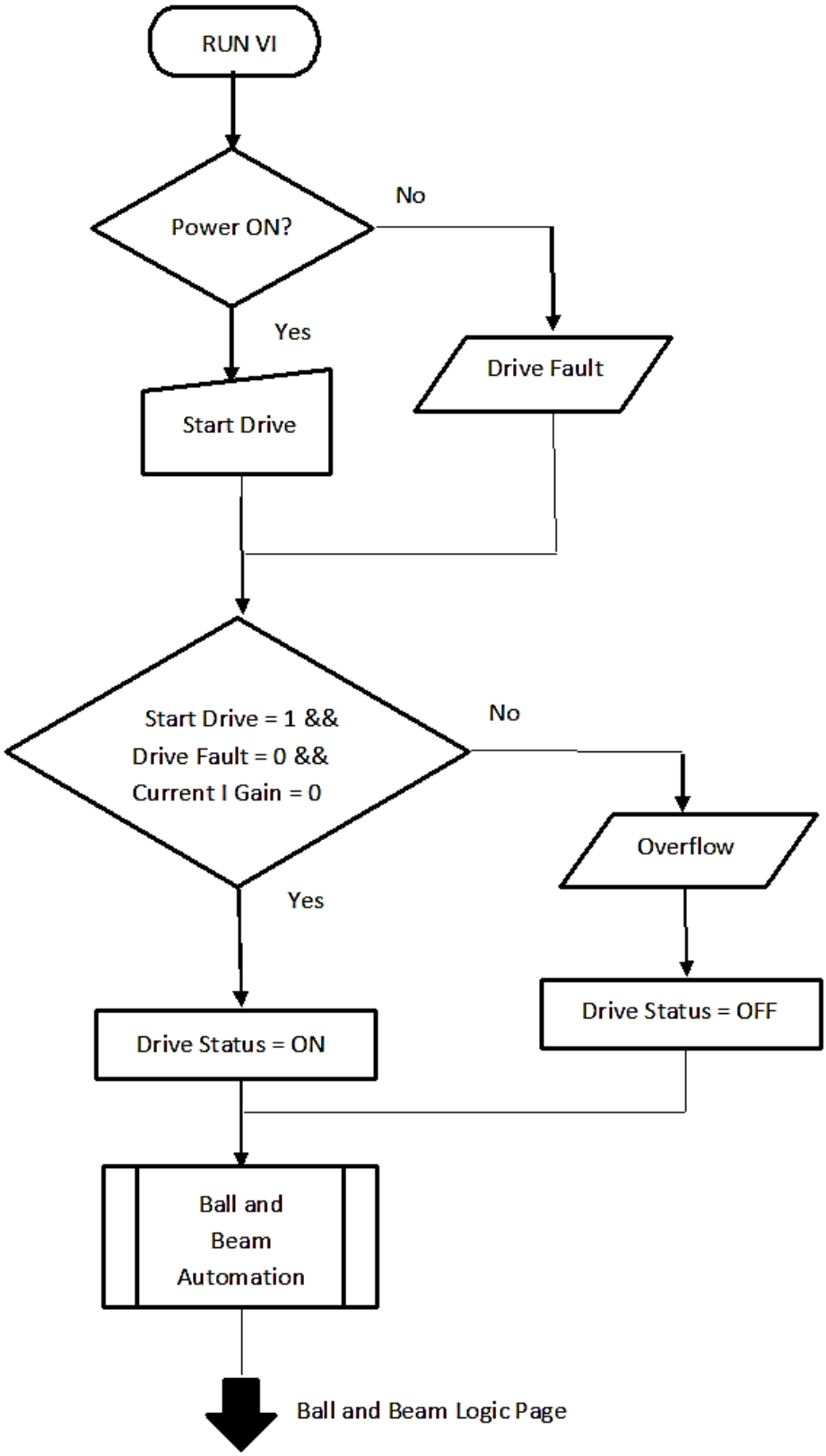


Fig 4.35: A cRIO FPGA digital time stamp filter logic

RT results obtained and observed during this experiment infer a number of conclusions:

- Instrumentation/mechatronics implementation is equally important to algorithm design to attain achievable and satisfactory RT feedback control performance and robustness curves. The ball on beam as in industrial environment work exemplified an experimental proof of this hypothesis.
- Indeed, adding PIV, Vff, and Aff gains to regular PID controller aids in increased performance and robust loop design and response as recorded.
- The FPGA technology in control is promising and, indeed, increases eventual RT control performance (transient response) in such complex systems. On the other hand, the processor mode indicated smoother and more accurate control results. Therefore, if program mapping flexibility, memory size, and fixed point precession challenges were to be improved in the near future by FPGA manufacturers, this paper would recommend industrial suppliers as well as operators to start adopting this technology, or at least include it within hybrid RT-controller stations' architectures, which will enable users to choose between performance and accuracy/smoothness upon specific desired applications.


RUN VI
Power ON?
No
Yes
Drive Fault
Start Drive
Start Drive = 1 &&
Drive Fault = 0 &&
Current I Gain = 0
No
Overflow
Yes
Drive Status = OFF
Drive Status = ON
Ball and
Beam
Automation
Ball and Beam Logic Page

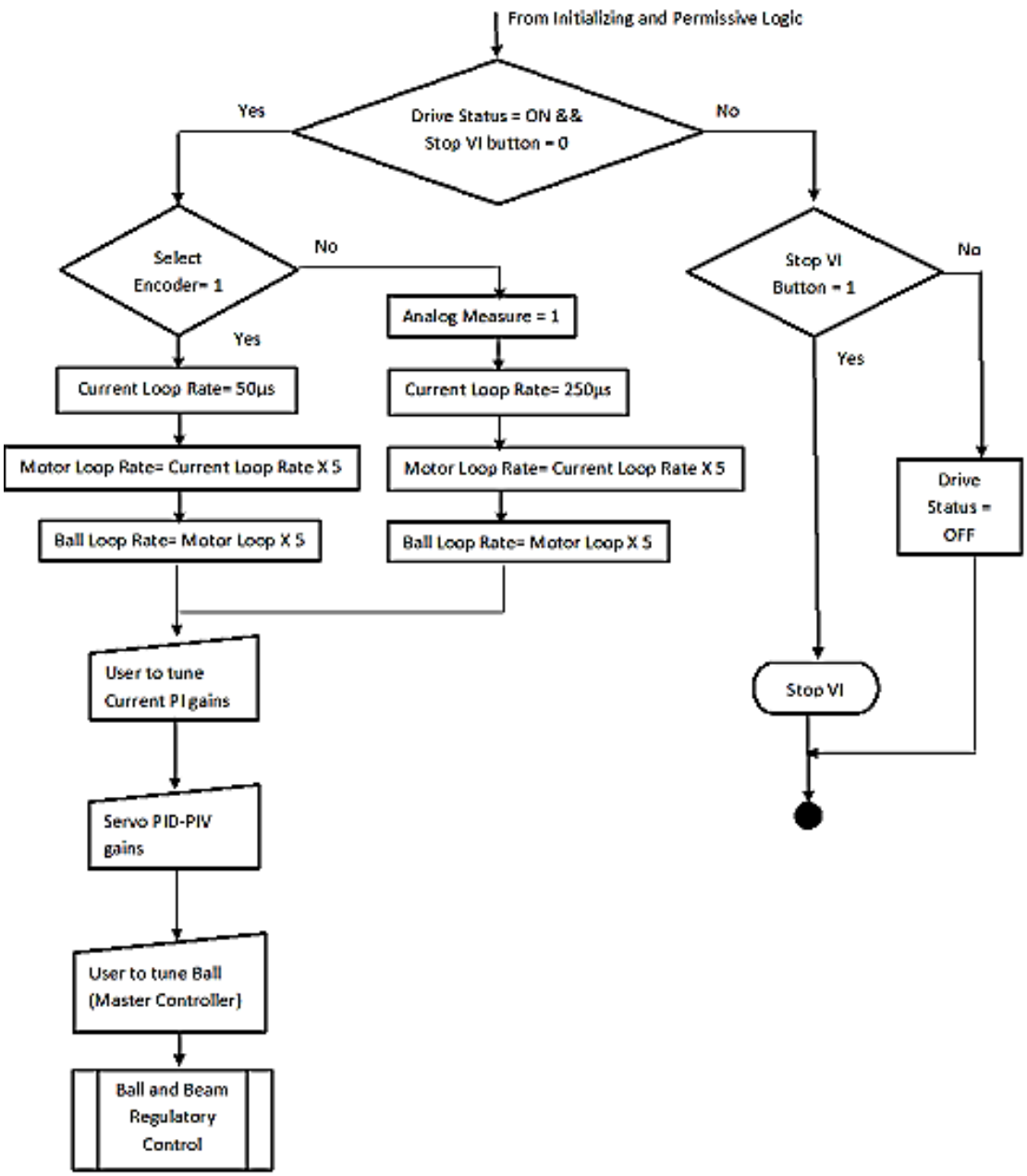


Fig 4.36: (a) The developed overflow detection and permissive logic and (b) the rest of the ball on beam program execution and supervisory/sequential control

- This experiment also infers several lessons and thoughts concerning designing structured multiple (complex) loops as the series of cascaded controllers arose in this system, the ball on beam control and automation. In other words, improper tuning of cascade loops may

result in the following. First, loops will fight each other which will result in large SSE. Second, this phenomenon may create oscillations, therefore instability. As a result, the RT variable, distance, position, velocity, and current will not be properly controlled.

- Not only do tuning parameters appear vital in determining satisfactory control design but also digital controller's sampling rates values. Most pervious literature points out the ADC's as the only sampling in the system, which is discussed here and also in earlier chapters denoted by ΔT obtained by AIs' S/H step. However, in theory as well as practice, each digital controller block, for example PID, should encompass an additional key tunable parameter known as loop sampling (execution) period. This experiment has exposed the clear relation between controller sampling rate and eventual RT control response performance. The higher the sampling rate, the higher the performance; however, through this experiment, this statement is not always true as there should exist reasonable upper and lower pounds of such parameter limits because as the sampling rate increases, control parameters feasibility interval shorten, which may contrarily degrade the performance due to fractional precision limitations. Unlike Nyquist's sampling theory, $f_s \geq 2B$, no known scientific proof or rule of thumb has determined the exact best rate selection practice of digital controller rate. Hence, this experiment suggests that values within the interval, $2B \leq f_c \leq 10B \leftrightarrow \omega_s \geq \Delta T_c \geq 0.1\tau$, along with properly tuned control parameters would result in acceptable results. Yet, for best performance practice, experts recommend regular PID sampling period values such that, $\Delta T_{PID} \leq \frac{\tau}{10}$.
- One essential remark in this experiment is sampling rates of cascaded loops. It is known that innermost loops should operate faster than outers (masters). Specifically, experts have suggested that inner loop rates should be multiples of 5 to 3 of their masters [72].

CHAPTER 5

CONCLUSIONS

This work intended to relate educational and experimental setups to one of the most common and important everyday life applications of control systems: industrial automation and instrumentation. Hence, the paper tried to close the gap between industry and academia by conducting historical study first to technological eras on applied control and automation. Then associated basic concepts and theories were revised. After that, a combination of both aspects posed the identity of the research philosophy on the case study, the ball on beam system. Numerous points have been learned from analyzing and implementing the ball on beam system as a reference for other complex RT processes. However, this work is considered to be incomplete since a number of the research directions and previously set targets have not been met yet. The following lines state this pending research besides current and future work. Also, there exist many difficulties and limitations faced which prevented segments of the setup besides some results to occur. Finally, this work can be ended by mentioning final remakes and thoughts, in addition to the future vision of this research.

5.1. Lessons Learned and Deductions Concluded

A couple of bottom line points have been learnt from this research such as:

- State-of- the-art technologies in control and instrumentation might not necessarily provide better performance or cost saving results. Rather, a clear understanding and right selection of each technology including pros and cons that fit particular purpose may lead to optimal solutions with regard to RT control irrespective of the application.
- No matter how robust or optimal the control design is if the mechatronics and instrumentations phases of the system are not implemented and selected properly, RT

control cannot be achieved. This statement has been proven several times during this report. For example, the clear effect of inertia ratio of load to motor to determine the control performance was shown. Also, in the ball on beam example, a larger slack and backlash of the motor coupling and transmission will affect the stability of the system. Another example would be the encoder feedback measurement status, which has determined the overall performance and accuracy of the control system.

- In addition to instrumentation/mechatronics implementation, a control engineer should take care of the soft digital controller design. In other words, besides programing skills and awareness of S/W tools available to implement control algorithms, understanding the computation architecture environment and its features such as parallelism, throughput, capacity, memory, RTOS, clock cycles, and loop rates aid in attaining desired results and selecting proper processing units for the application. For instance, the ADC sampling rate satisfying Nyquist limit in order to fully reconstruct acquired signals from analog domain would not be enough knowledge to keep in mind. Rather, the digital control loop rate should surpass the bandwidth of the measured signal by multiples that exceed that limit. In other words, if the time constant of the process is greater than or even equal to the PID sampling rate, for example, control error will occur or performance will degrade.
- There is no control design that can satisfy all requirements such as performance, robustness, and cost. Hence, there should always be design tradeoffs between robustness, performance, and cost in control. For this reason, a control engineer should select proper algorithms and technology that suit the design specification/requirements.
- Table 5.1 summarizes observations of various algorithms and experiments applied on the ball on beam, some of which are implemented while others were computer simulated:

Implementation Technology /Simulation	Algorithm	RT Control	Performance	Robustness	Notes/drawbacks
RT Scan (MPU)	Open loop step response	Achievable	Good	Low	Subjected to human estimation errors
RT Scan (MPU)	ZN (PID)	Achievable	High	Low	Requires instability which may damage setup
RT Scan (MPU)	Cohen-Coon (PID)	Failure	N/A	N/A	An off-line tuning method and assumes 1st order model
Hybrid	Smith Predictor [74]	Failure	N/A	N/A	Assumes 1st order model and suits processes with long delays
RT Scan (MPU)	IMC	Achievable	Good	Good	Offers middle points between robustness and performance
FPGA	PID-PID-PIV-Vff-Aff plus current feedback	Achievable	High	High	FPGA memory limitation and fixed-point precession issues
LabVIEW Simulation	Optimal Control (LQR)	SS must be linear, controllable/observable	either High or degraded but cost effective	Low	Robustness not considered and simulation results might not be realizable

Table 5.1: Various algorithms and technologies applied in this research and their observations

5.2. Ongoing Work, Possible Improvements, and Current Research

As revealed earlier, several research directions and experiments are still to be conducted:

- The intended higher order control dynamics response comparison study using DC servo and stepping actuation is still to be answered. In other words, it is a technology (stepper) versus algorithm (cascading servo) problem with respect to control performance and robustness knowing that the first may offer less cost and control effort. The two plant models were both built in UNT Engineering labs and workshops; however, the comparison has not taken place yet. Fig 5.1 below shows the two built plant models.

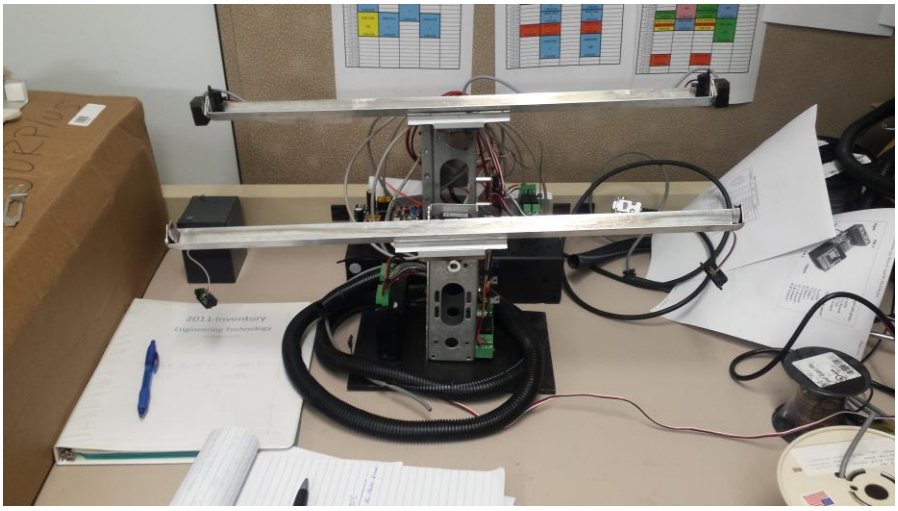


Fig 5.1: The ball on beam plant model by the two approaches DC and stepper motors

- The other query to be answered is investigating how efficient the standard WSN protocol, ZigBee would be performing RT control, measurement, and actuation. In other words, evaluating ZigBee as WSAN would give us a sense of how other industrial protocols such as W-HART and ISA 100 may behave in complex processes and loops. Fig 5.2 on the next page exemplifies the proposed WSN node and architecture compatible with the distributed NI cRIO setting obtained. The design suggests using NI 3202 (a) as a WSN node(s) and NI 9795 as c-series wireless gateway module, which appears below, (b).

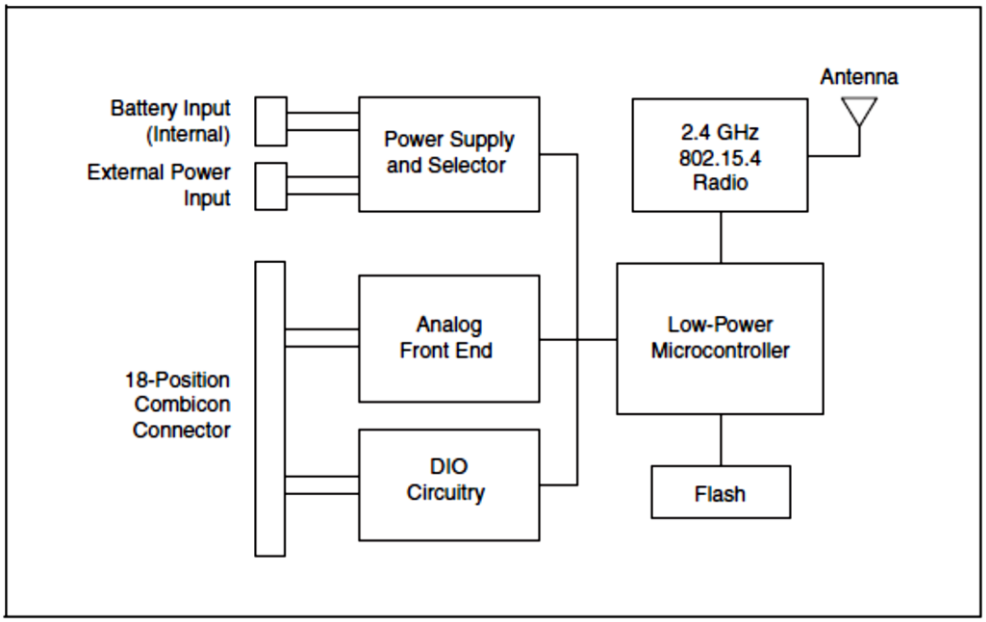


Fig 5.2: (a) NI WSN node BD [124] and (b) the suggested WSN in the system architecture [123]

- The ball on beam as a part of distributed industrial architecture setting also opened doors for enhanced studies through implementation of advanced control algorithms doing several stages and their applications on complex systems as the ball on beam such as optimal, adaptive, MPC, robust, NL, and statistical models and stochastic based control.
- It has been observed that estimating velocity and acceleration, which are the states of the ball on beam dynamic, in S/W digital domain after acquiring position is associated with several issues and limitations such as noise, accuracy, encoder resolution, etc. Therefore, possible improvement may occur in either one of the directions. First, RT measurement of those variables can be implemented in the analog domain using Op-Amps differentiators. Second, applying advanced algorithms in which more efficient estimation is possible such as Kalman filters might be implemented also.
- Most textbooks discussing digital control focus on ADC sampling rate which is an important phase in loop and DAQ; however, almost no or little literate found has clearly pointed out the methodology of selecting optimal controller sampling period, which its direct effect over performance has been clearly shown in this research. Hence, a more systematic and scientific approach would be crucial regarding selecting optimal digital controllers' sampling rates with respect to signal bandwidth, tuning parameters, ADCs, clock, or scan time. This implies further research in this topic to suggest loop rates, especially in complex structures as multiple cascade loops. At this point, it can be concluded that higher controllers' sampling rates do not necessarily outcome in better results, yet the optimization problem should take into account upper and lower pounds besides best control design parameters satisfying certain control strategies.
- Upon resolving time synchronization issue, a hybrid (CPU-FPGA) structure can be tested.

5.3. Barriers, Limitations, and Difficulties Faced

During work on this research, a couple of difficulties and limitations have posed hurdles and barriers preventing smooth and healthy progressing of this work. Those problems and other technical issues resolved are summarized in the table below, 5.2.

Problem and Difficulty Encountered	Solution
Lack of basic and necessary amenities, resources, tools, and proper communications.	Ordering online, purchasing from local electronics shops, and working in other departments besides self-learning and autonomous approach.
Noisy encoder feedback signals and their bandwidth exceeds the controller's loop rate.	Installing shielded twisted pair wires, a differential line driver, and a digital (time stamp or skip) filters to cutoff the bandwidth.
Fixed point arithmetic overflow and quantization errors in the ball on beam experiment implemented by the FPGA control mode.	Implementing an overflow detection logic to switch off the brushed driver in order to prevent sudden motor jumps that damages the mechanical part. (See Fig 4).
Persistent mechanical slack and backlash on the experimental ball on beam setup due to internal or external forces and interference.	Re-coupling the motor with the beam and trying to protect the setup from further external interferences.

Table 5.2: Major difficulties faced and their solutions

5.4. Final Remarks, Thoughts, and Future Vision (Towards the Hybrid Integrated Architecture)

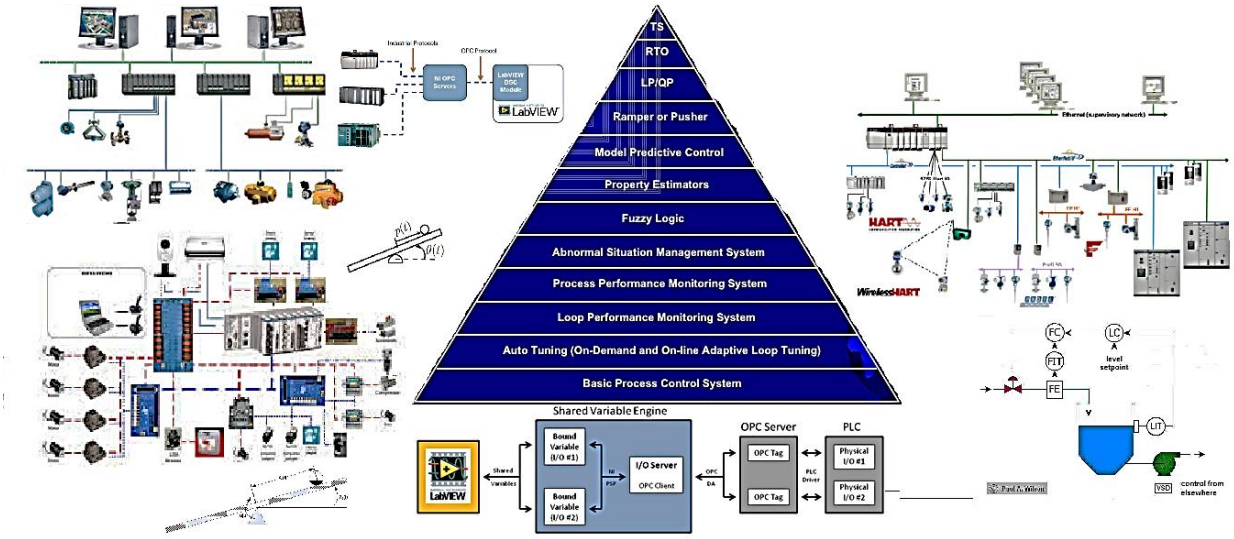


Fig 5.3: The future lab research vision considering higher level applications, power, and cost

The current setup would be expanded to enclose more subsystems communicated via OPC or serial (Modbus) communications such as PLCs, advanced technologies, e.g. WSN, WSAN, APC and web, and higher level applications. [124] demonstrates how to connect LabVIEW RT applications with any subsystem PLC. A complete setup would mimic the industrial integrated architectures and systems in order to develop existing technologies through closing the gap between academic research and corporate engineering problems. Such hybrid integrated and distributed laboratory control environment could benefit in not only training and introducing college students to industrial systems but also creating rich experimental research atmosphere. The current available industrial hybrid architecture should include multiple computing technologies within controller units to allow higher availability and flexibility. Also, applying WSN and WSAN on a complex process such as the ball on beam would further study, assess availability and capability of protocols as ZigBee-Pro, W-HART, and ISA 100 to take over field industrial RT control and other asset management applications. In this context, [126] shows the effectiveness of

the newly developed algorithm, PID-PLUS, in overcoming wireless communication package drop or loss in performing RT feedback control, WSAN.

This research has considered RT feedback control performance and accuracy, robustness over various wired communicated implementation technologies, and algorithm. However, as introduced at the beginning, to provide solutions to industry requires deeper further studies of those techniques and approaches. For example, we have confirmed that FPGA control enhances performance, but would this technology save or consume more power compared to conventional CPU? Also if developing WSN protocols proves robustness and capability of RT process control over distributed complex systems, which cost advantage would this technology offer, other than wiring's, concerning power consumption? These questions and more should be answered upon viewing the whole distributed control problem from more comprehensive perspectives in addition to feedback theory and design, instrumentation, mechatronics, and system architecture and applications. The topic of configurable computing and computer architecture should be integrated to such hybrid industrial automation architecture. For instance, it is known from [57] and [64] that dynamic power of a digital computing hardware is a function of a number of variables as activity (switching) factor, α, circuit size or memory capacity, *C*, power supply, $V_{DD,}$ and clock frequency, *f*, such that:

$$P_{switching} = \alpha C V_{DD}^2 f \tag{166}$$

This equation indicates that the dynamic power, $P_{switching,}$ of a given system increases with increasing speed, clock, and sampling rates, besides loading the controller. Together with previous considerations, performance, accuracy, robustness, and cost, the topic of power consumption should be included in the top of the SCADA level in a complex set of LP optimization problems subject to constraints that vary according to the application and requirements.

APPENDIX A

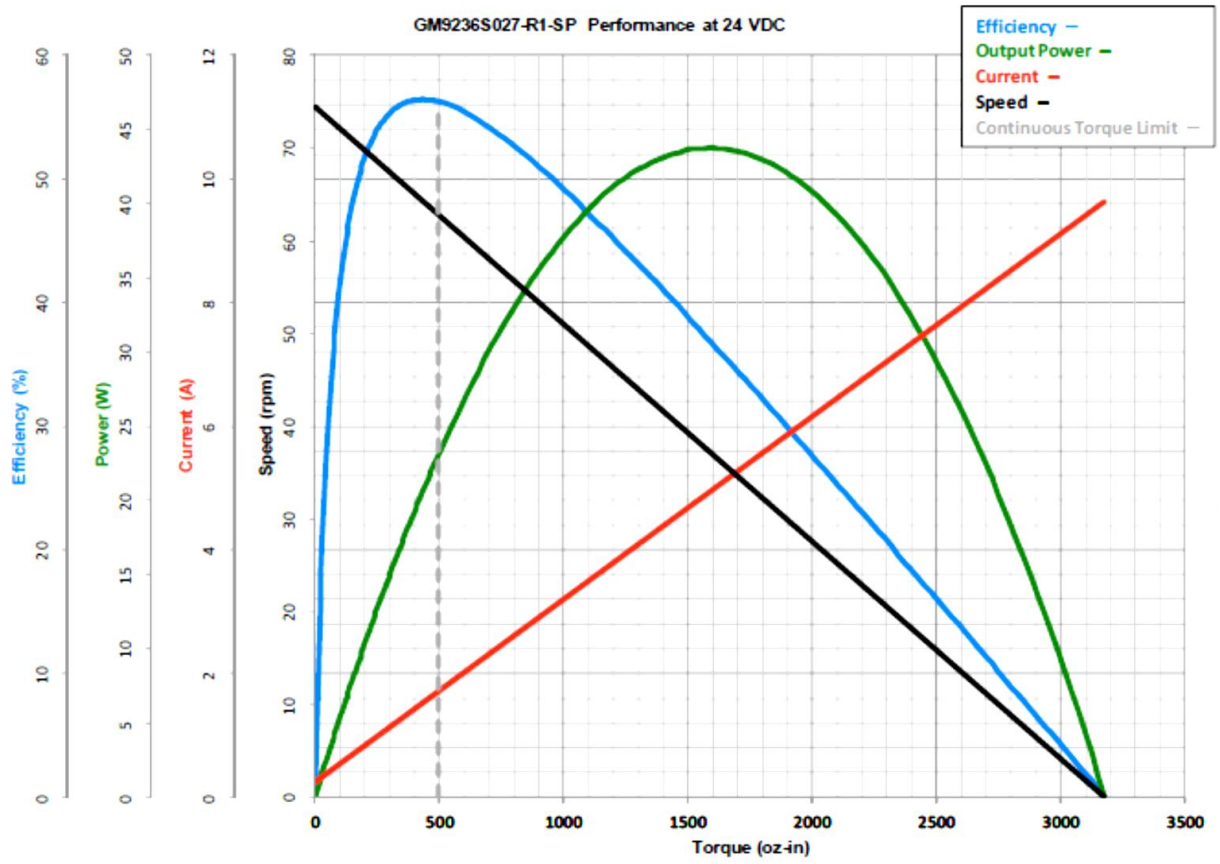
MATLAB CODE FOR OPTIMAL CONTROL DESIGN SIMULATION

```
%this is without friction
%a = [0, 1, 0, 0, 0; 0, 0, 5.163, 0, 0; 0, 0, 0, 1, 0; -0.2, 0, 0, -0.3,
0.04; 0, 0, 0, -1154.8, -957.69];
%b = [0; 0; 0; 0; 923];
%c = [1, 0, 0, 0, 0; 0, 0, 1, 0, 0];

%this is the reduced model with friction
a = [0, 1, 0, 0; 0, 0.5263, 5.163, 0; 0, 0, 0, 1; -0.227, 0, 0, -0.4774];
b = [0; 0; 0; 0.045];
c = [1, 0, 0, 0; 0, 0, 1, 0];


sys = ss(a,b,c,0)
figure
step(sys)
Co = ctrb(a,b);
k = rank(Co);
unco=length(a)-rank(Co);
poles = eig(a);
grid on;
figure
pzmap(sys);

obsv(a,c)
Ob = obsv(sys);
unob = length(a)-rank(Ob);
I = rank(Co);

t = 0:0.1:10;
u = zeros(size(t));
%x0 = [-2 -3 1 1 1];
x0 = [1 1 1 1];

figure
[y,t,x] = lsim(sys,u,t,x0);
plot(t,y)
title('Open-Loop Response to Non-Zero Initial Condition')
xlabel('Time (sec)')
ylabel('Ball Position (m)')

H1 = tf(1,[-1,1],0.00001);
H2 = tf([0,0.000168],[-0.999996,1],0.00001);
H3 = tf([0,0.00001],[-1,1],0.00001);
H4 = tf(1,[-1,1],0.00001);

H=H1*H2*H3*H4;
```

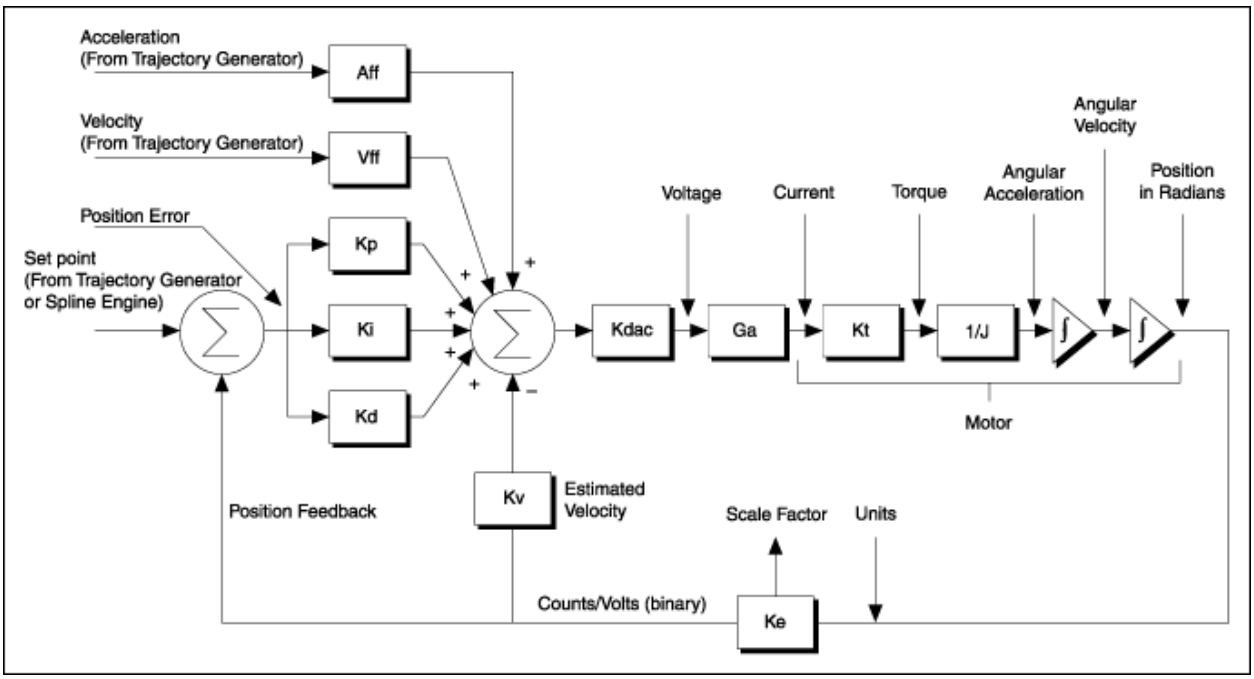
```
figure('Color', 'w')
title('The overall ball on Beam open-loop Process Plant step response')
hold on
stepplot(H)
[y,t] = step(H,5:0.00001:1);

%optimal control algorithm
Q = c'*c;
Q(1,1) = 50000;
Q(3,3) = 10000;
R = 1;
K = lqr(a,b,Q,R);

Ac = [(a-b*K)];
Bc = [b];
Cc = [c];
Dc = [0];

states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'r'};
outputs = {'x'; 'phi'};

sys_cl =
ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,'outputname',outputs);

t = 0:0.01:5;
r =0.2*ones(size(t));
[y,t,x]=lsim(sys_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','Ball position (m)')
set(get(AX(2),'Ylabel'),'String','Beam angle (radians)')
title('Step Response with LQR Control')

Cn = [1 0 0 0];
sys_ss = ss(a,b,Cn,0)
Nbar = rscale(sys_ss,K);

sys_cl =
ss(Ac,Bc*Nbar,Cc,Dc,'statename',states,'inputname',inputs,'outputname',output
s);
t = 0:0.01:5;
r =0.2*ones(size(t));
[y,t,x]=lsim(sys_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','ball position (m)')
set(get(AX(2),'Ylabel'),'String','beam angle (radians)')
title('Step Response with LQR Control')
```

APPENDIX B

THE DC MOTORS PERFORMANCE CURVES AND PID-PIV-Vff-Aff WITH CURRENT FEEDBACK AND DRIVING BD

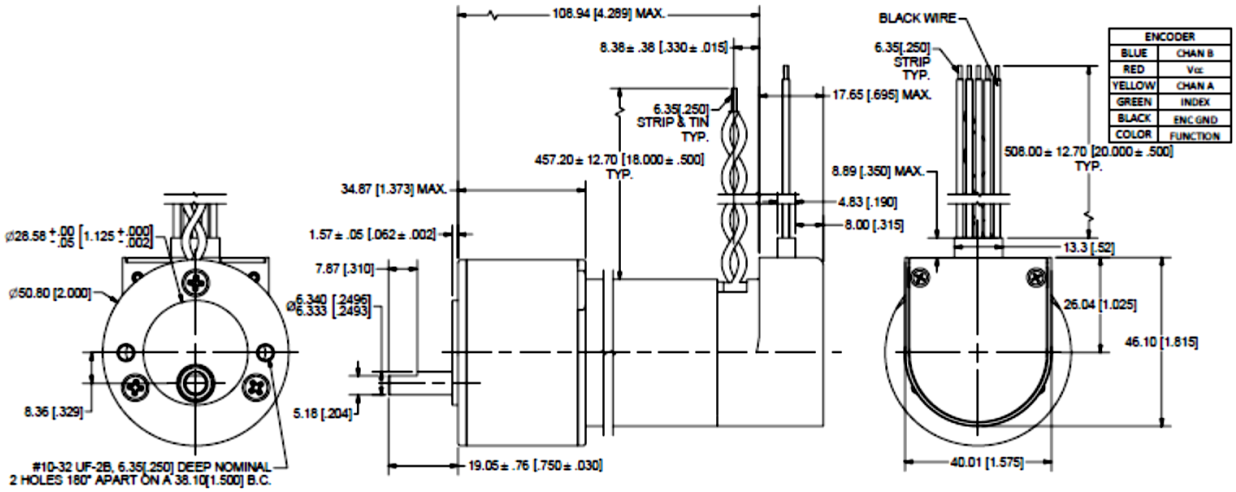


Appendix 3: PID-PIV-Vff-Aff with current feedback and driving BD



APPENDIX C

DC MOTORS DRAWINGS AND CONVERSIONS



KT – Torque Constant

1 Nm / amp = 141.612 oz-in / amp

1 oz-in / amp = 7.06155 x 10-3 Nm / amp

KE – Back EMF Constant

1 v / krpm = 9.5493 x 10-3 volt per rad / s

1 volt per rad / s = 104.72 v / krpm

KM – Motor Constant

1 oz-in / √w = 7.0615 x 10-3 Nm / √w

1 Nm / √w = 141.612 oz-in / √w

JR – Rotor Inertia

1 oz-in-s2 = 7.0615 x 104 gm-cm2

1 gm-cm2 = 1.14 x 10-5 oz-in-s2

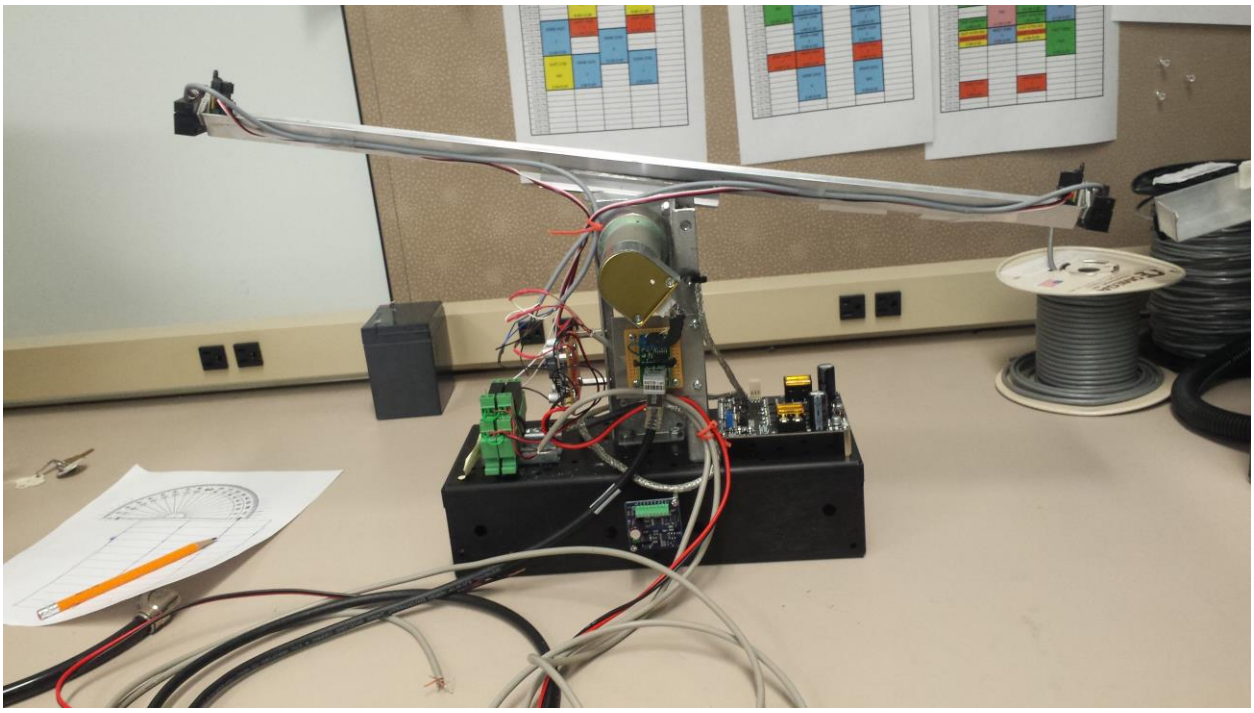
APPENDIX D

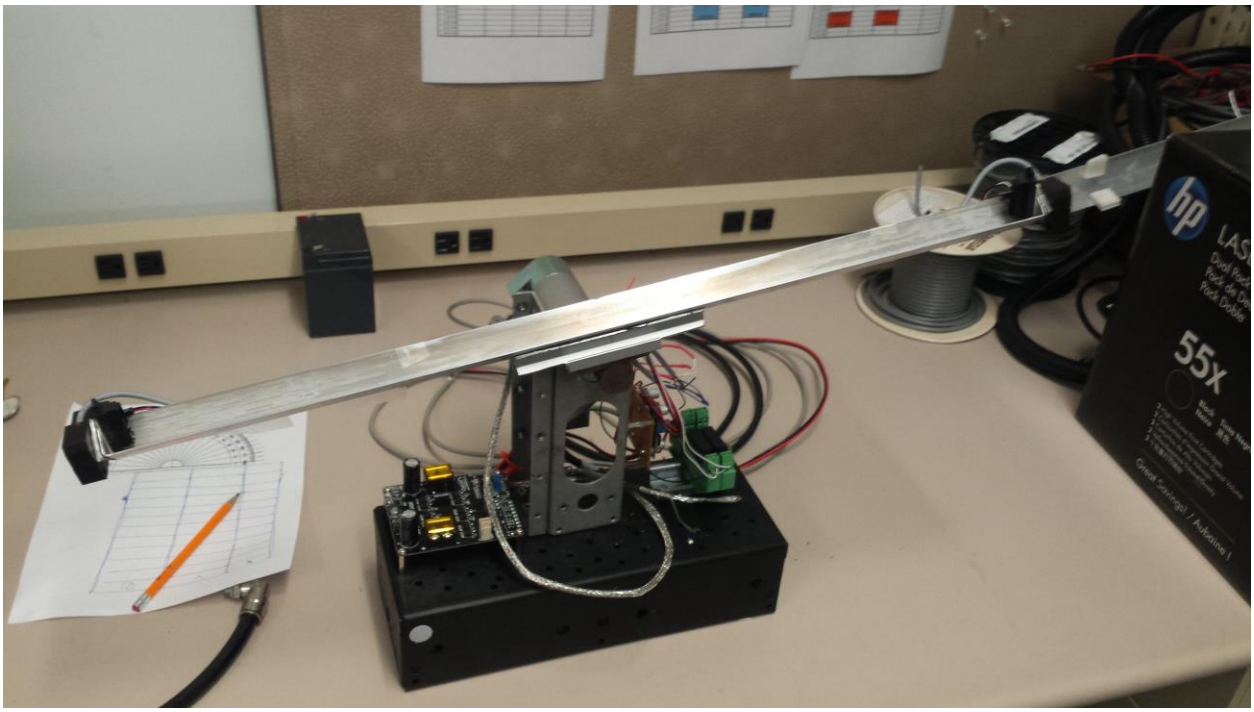
THE BALL ON BEAM AUTOMATION SYSTEM ALGORITHM (PSEUDO CODE)

```
Run VI;
User to start drive button
IF (Start Drive = 1 && Power = ON Current Loop Integral Gain = 0)
  Drive status = ON;
Else
  Overflow message is detected
  Drive status = OFF;
End IF
While Drive status = ON && Stop button = OFF
  If (Encoder Selection = 1) THEN
    Current Loop rate = 150 Micro second;
    Motor loop rate = 150X5;
    Ball Loop Rate = Motor loop rateX5
  Else
    Analog Measure = 1;
    Current Loop rate = 250 Micro second;
    Motor loop rate = Current Loop rate X5;
    Ball Loop Rate = Motor loop rate X5;
  End IF
  User to tune current PI gains;
  User to tune Motor PID-PIV gains;
  If Ball Controller = PID
    User to tune PID gains
  Else if ball controller = PID-PIV
    User to tune PID-PIV
End While
```

APPENDIX E

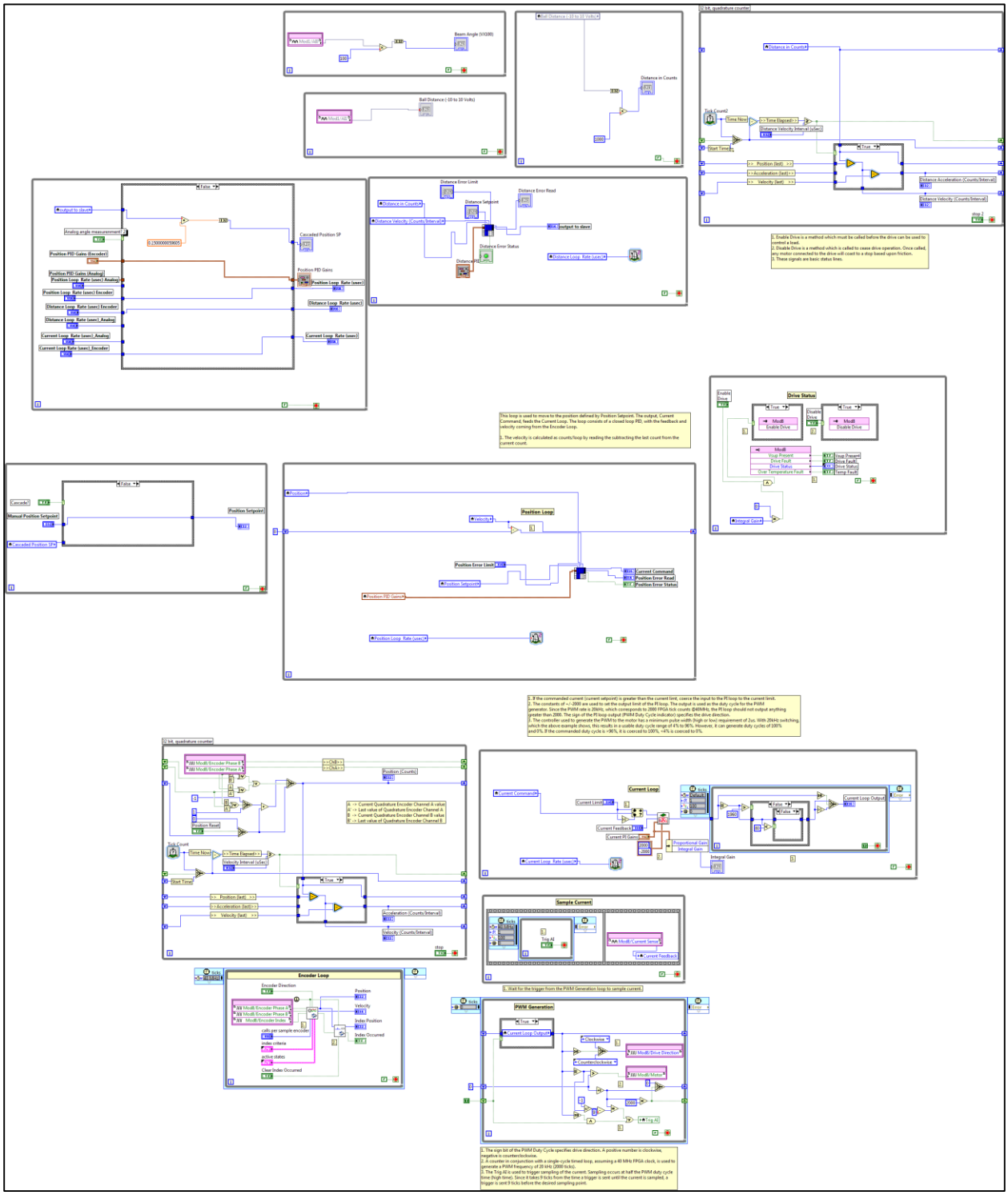
REAR AND FRONT VIEWS OF THE BUILT BALL ON BEAM PLANT (CLOSER VIEWS)




hp
55x

APPENDIX F

THE BALL ON BEAM cRIO FPGA VI



APPENDIX G

SOME COMMON LAPLACE TO Z-TRANSFORM PAIRS

Entry	Laplace Domain	Time Domain	Z Domain (t=kT)
unit impulse	1	$\delta(t)$ unit impulse	1
unit step	$\Gamma(s)=\frac{1}{s}$	$\gamma(t)$	$\frac{z}{z-1}$
ramp	$\frac{1}{s^2}$	t	$T\frac{z}{(z-1)^2}$
parabola	$\frac{2}{s^3}$	t^2	$T^2\frac{z(z+1)}{(z-1)^3}$
t^n (n is integer)	$\frac{n!}{s^{(n+1)}}$	t^n	
exponential	$\frac{1}{s+a}$	e^{-at}	$\frac{z}{z-e^{-aT}}$
power		$b^k \quad (b=e^{-aT})$	$\frac{z}{z-b}$
time multiplied exponential	$\frac{1}{(s+a)^2}$	te^{-at}	$T\frac{ze^{-aT}}{(z-e^{-aT})^2}$
Asymptotic exponential	$\frac{1}{s(s+a)}$	$\frac{1}{a}(1-e^{-at})$	$\frac{z(1-e^{-aT})}{a(z-1)(z-e^{-aT})}$
double exponential	$\frac{1}{(s+a)(s+b)}$	$\frac{e^{-at}-e^{-bt}}{(b-a)}$	$\frac{z(e^{-aT}-e^{-bT})}{(b-a)(z-e^{-aT})(z-e^{-bT})}$
asymptotic double exponential	$\frac{1}{s(s+a)(s+b)}$	$\frac{1}{ab}\left(1-\frac{be^{-at}-ae^{-bt}}{(b-a)}\right)$	
asymptotic critically damped	$\frac{1}{s(s+a)^2}$	$\frac{1}{a^2}(1-e^{-at}-ate^{-at})$	$\frac{(1-e^{-Ta}(1+Ta))z^2+e^{-Ta}(Ta-1+e^{-Ta})z}{a^2(z-e^{-Ta})^2(z-1)}$
differentiated critically damped	$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}$	$\frac{z(z-(Ta+1)e^{-Ta})}{(z-e^{-Ta})^2}$

BIBLIOGRAPHY

[1] Vanessa Romero Segovia and Alfred Theorin, *History of Control, History of PLC and DCS*, 2012-06-15. [online]. Available: www.control.lth.se/media/Education/DoctorateProgram/2012/HistoryOfControl/Vanessa_Albert-PLCDCS.pdf [Accessed: Sep 31 2014]

[2] Bela Liptak, *Tuning Interacting Controllers: Is the Famous Ziegler-Nichols (ZN) Open-Loop Tuning/Closed-Loop Tuning Parameter Calculation for Interacting or Non-Interacting PID*? Control Journal, Tuning Interacting Controllers, May 14, 2012.

[3] International site for Spirax Sarco, *Temperature Control for Steam Applications*, Spirax Sarco, home, resources online, steam engineering Tutorials, 2015. [Online]. Available: http://www2.spiraxsarco.com [Accessed: Aug 16 2015].

[4] Johan A kerberg, Mikael Gidlund, Mats Bj¨orkman, and M¨alardalen, *Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation*, 2011 9th IEEE International Conference on Industrial Informatics, Caparica, Lisbon, 26-29 July 2011, pp 410 – 415.

[5] Detlef Zuehlke, *SmartFactory—Towards a factory-of-things*, Annual Reviews in Control 34 (2010) 129–138, www.elsevier.com/locate/arcontrol.

[6] Reza Abrishambaf and Majid Hashemipour & Mert Bal, *Structural modeling of industrial wireless sensor and actuator networks for reconfigurable mechatronic systems*, The International Journal of Advanced Manufacturing Technology, February 2013, Volume 64, Issue 5, pp 793-811.

[7] Pedram Radmand, Alex Talevski, Stig Petersen and Simon Carlsen, *Comparison of Industrial WSN Standards*, 4th IEEE International Conference on Digital Ecosystems and

Technologies (IEEE DEST 2010), 2010, pp 632-637.

[9] Stig Petersen, Simon Carlsen and Amund Skavhaug, *Layered Software Challenge of Wireless Technology in the Oil & Gas Industry*, 19th Australian Conference on Software Engineering, 26-28 March 2008, IEEE 2008, Perth, WA, pp 37 - 46.

[10] Jianping Song, Aloysius K. Mok, Deji Chen, and Mark Nixon! *Challenges of Wireless Control in Process Industry,* [online]. Available: http://moss.csc.ncsu.edu/~mueller/crtes06/papers/009-final.pdf. [Accessed: Sep 2014].

[11] Xiuming Zhu, Thomas Lin,Song Han, Aloysius Mok, Deji Chen, Mark Nixon, and Eric Rotvold, *Measuring WirelessHART against Wired Fieldbus for Control*, Industrial Informatics (INDIN), IEEE 10th International Conference on Industrial Informatics, 2012 IEEE 25-27 July 2012, pp 270 - 275.

[12] _, *Difference between Discrete and Process Manufacturing*, October 11th, 2010. [Online]. Available: www.batchmaster.co.in/blog/difference-between-discrete-and-process manufacturing/.

[13] Miri Kesner, *Batch Process or Continuous Process?* Weizmann Institute of Science University. [Online]. Available: http://stwww.weizmann.ac.il.

[14] Tang Zhong, Zeng Peng, and Wang Hong, *Analysis and Design of Real-time and Reliable Industrial Wireless Control Communication Network and Protocol*, International Journal of Proceedings of the 29th Chinese Control Conference, July 29-31, 2010, Beijing, China, pp. 4157- 4163.

[15] Delaney, M. C., ProSys, Inc., Baton Rouge, Louisiana, *Advanced process control: A historical perspective,* Hydrocarbon Processing Magazine, 03.01.2012.

[16] Stuart Bennett, *A Brief History of Automatic Control,* IEEE Control Systems, June 1996,

pp 17-24.

[17] Jim Strothman, *M&C technology history: More than a century of Measuring*, ISA, *In*Tech June 1995.

[18] Robert H. Bishop, *Modern Control Systems with LabVIEW*, Marquette University, National Technology and Science Press, 2012.

[19] National Instruments' technical manuals, *Control Loop (Fixed Point PID)*, NI Home, Support, Manuals, NI SoftMotion, Module Help, 371093G-01, June 2010.

[20] Yong Zhou, *DC Motors, Speed Controls, Servo Systems: An Engineering Handbook*, 1st ed. Reading, Elmsford: Pragamon Press Inc. 1977. [E-book] Available: Technology & Engineering.

[21] Christer Rameback, Vice President, ABB Process Automation, *Process Automation -History and Future*, ABB Automation Technology, 2003-09-16. [Online]. Available: http://www.iestcfa.org [Accessed: Aug 18, 2015].

[22] Jerker Delsing, Jens Eliasson, Rumen Kyusakov, Armando W. Colombo, Francois Jammes, Johan Nessaether, Stamatis Karnouskos, and Christian Diedrich, *A Migration Approach towards a SOA-based Next*, IEEE, 2011.

[23] Sharma, K.L.S, *Overview of Industrial Process Automation*, 1st ed. Reading, Saint Louis, MO, USA, An Elsevier Title, [E-book] Available: UNT electronic library.

[24] Siemens online magazine, *Holcim upgrades the Simatic DCS in one of its Lebanese operations in record time by teaming up with Siemens Solutions Partner Esprocessing*, Siemens process news magazine. [Online], available: http://www.industry.siemens.com. [Accessed Aug 18, 2015].

[25] Yokogawa Electric Corporation, *Connect Field Instruments and Upstream Devices with*

HART Communication compatible Signal Conditioners, Yokogawa, 1994 – 2015. [Online]. Available: http://www.yokogawa.com. [Accessed: Aug 20 2015].

[26] Brendan Galloway and Gerhard P. Hancke, *Introduction to Industrial Control Networks*, IEEE Communications Surveys & Tutorials, 25 July 2012, pp 860 - 880.

[27] John P. Bentley, *Principles of Measurement Systems*, 4th ed. Pearson Education Limited, Edinburgh Gate, Harlow, England, 2005.

[28] FieldComm Group, HART communication protocol, *HART Protocol Specifications*, HART communication foundation, 2014. [Online]. Available: http://en.hartcomm.org [Accessed: Aug 30, 2015].

[29] Fieldbus Foundation, technical resources and manuals, *Guide to Implementing Foundation H1 Filed Devices white paper, and wiring and installation Fieldbus manual*. Fieldbus Foundation white papers and technical guides, 2006. [Online]. Available: http://www.fieldbus.org [Accessed: Aug 30, 2015].

[30] National Instruments, *What is a Wireless Sensor Network?*, NI white papers, Publish Date: Feb 28, 2011. [Online]. Available: http://www.ni.com [Accessed: Sep 5, 2015].

[31] Vaibhav V.Ukarande, Shaikh F.I, *WirelessHART a Right Choice over ZigBee*, National Conference on Advances in Computing, Networking and Security, NCACNS, 2013. PP, 81-84.

[32] Vehbi C. Gungor, and Gerhard P. Hancke, *Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches*, National Conference on Advances in Computing, IEEE Transaction on Industrial Electronics, vol. 56, No. 10, Oct 10 2009, pp. 4258-4265.

[33] Raymond S. Wagner, *Standards-Based Wireless Sensor Networking Protocols for*

Spaceflight Applications, Aerospace Conference, 2010 IEEE, 6-13 March 2010, pp 1 - 7.

[34] Hisanori Hayashi, Toshi Hasegawa and Koji Demachi, *Wireless Technology for Process Automation*, ICROS-SICE International Joint Conference, August 18-21, 2009, Fukuoka International Congress center Japan. PP, 4591-4594.

[35] Tomas Lennvall and Fredrik Hekland, *A Comparison of WirelessHART and ZigBee for Industrial Applications*, IEEE, 2008.

[36] Mohammad reza Akhondi, Alex Talevskiand, Simon Carlsen, and Stig Petersen, *Applications of Wireless Sensor Networks in the Oil, Gas and Resources Industries*, IEEE Computer Society, 24th IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 941-948.

[37] Simon Carlsen, Amund Skavhaug, Stig Petersen, and Paula Doyle, *Using Wireless Sensor Networks to Enable Increased Oil Recovery*, 2008 IEEE, pp. 1039- 1048.

[38] Miroslav Pajic, Shreyas Sundaram, George J. Pappas, and Rahul Mangharam, *The Wireless Control Network: A New Approach for Control Over Networks*, IEEE Transaction on Automatic Control, VOL. 56, NO. 10, OCT 2011, pp. 2305-2318.

[39] Modbus Organization, *MODBUS over serial line specification and implementation guide*, MODBUS.ORG, Vol 1. 12/02/02. [Online]. Available: http://www.modbus.org [Accessed: Sep 12, 2015].

[40] George Thomas, Contemporary controls, *Introduction to Modbus Serial and Modbus TCP*, the extension: A technical Supplement to Control Networks, Vol 9, no 5, pp. 9-10, 2008.

[41] OPC Foundation, The Interoperability Standard for Industrial Automation, *What is OPC?* OPC Foundation Group, 2005. [Online]. Available: https://opcfoundation.org [Accessed: Sep 15, 2015].

[42] Darek Kominek, *OPC: The Ins and Outs to What It's About: The Every Man's Guide to OPC*, Matrikon OPC, 2009. [Online]. Available: http://www.matrikonopc.com [Accessed: Sep 15, 2015].

[43] National Instruments, *Introduction to OPC*, NI White Papers, pp, Sep 07, 2012. [Online]. Available: www.ni.com [Accessed: Sep 15, 2015].

[44] Digital Bond, British Columbia Institute of Technology, Byres Research, *Understanding OPC and How it is Deployed*, OPC Security WP 1 (Version 1-3b).doc, July 27, 2007.

[45] Thilo Sauter, *The Three Generations of Field-Level Networks—Evolution and Compatibility Issues*, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 57, NO. 11, NOV 2010, pp. 3585- 3595.

[46] TANG Zhong, ZENG Peng, WANG Hong, *Analysis and Design of Real-time and Reliable Industrial Wireless Control Communication Network and Protocol*, Proceedings of the 29th Chinese Control Conference July 29-31, 2010, Beijing, China, pp. 4157- 4163.

[47] IDC Technologies, *Process Control: Automation, Instrumentation, and SCADA*. Bookboon, Houston, Texas, USA, 2012.

[48] Stig Petersen and Simon Carlsen, *WirelessHART Verses ISA100.11.a: The Format War Hits the Factory Floor*. IEEE Industrial Electronics Magazine, Dec 2011, pp. 23-34.

[49] National Instruments, *Understanding Servo Tune*, NI White Papers, pp, Jul 09, 2012. [Online]. Available: www.ni.com [Accessed: Sep 15, 2015].

[50] Charles M. Close, Dean H. Frederick, and Jonathan C. Newell, *Modeling and Analysis of Dynamics Systems*, 3rd ed. John Wily & Sons, Inc, New York, NY, USA, 2002.

[51] Basil Hamed, *Application of a LabVIEW for Real-Time Control of Ball and Beam System*, IACSIT International Journal of Engineering and Technology, Vol.2, No.4, August 2010

ISSN: 1793-8236, pp. 401- 407.

[52] G. Benet, F. Blanes, J.E. Simó, P. Pérez, *Using infrared sensors for distance measurement in mobile robots*, Robotics and Autonomous Systems 1006 (2002) 1–12, March 2002.

[53] National Instruments, *Encoder Measurements: How-To Guide*, NI White Papers, pp, Aug 09, 2013. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[54] National Instruments, *Op Amp Filters*, NI White Papers, Publish Date: May 02, 2012. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[55] Curtis D. Johnson, *Process Control Instrumentation Technology*, 7th ed. Pearson Education, Inc, New Jersey, USA, 2003.

[56] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing*, 3rd ed. Prentice-Hall, Inc, New Jersey, USA, 1996.

[57] John L. Hennesy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th ed. Elsevier, Inc, Waltham, MA, USA.

[58] National Instruments, *FPGA Fundamentals*, NI White Papers, Publish Date: May 03, 2012. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[59] Altera Corporation, *FPGA vs. DSP Design Reliability and Maintenance*, Altera White Papers, May 2007, ver. 1.1. [Online]. Available: www.altera.com [Accessed: Oct 7, 2015].

[60] Bruno Paillard, *An Introduction To Digital Signal Processors*, Bruno Paillard Ing, January 27 2002.

[61] Lee Eng Kean, Intel White Papers, *Microcontroller to Intel Architecture Conversion Programmable Logic Controller (PLC) Using Intel Atom Processor*, Intel Corporation, January, 2010. [Online]. Available: www.intel.com [Accessed: Oct 7, 2015].

[62] Jennifer Eyre and Jeff Bier, *DSP Processors Hit the Mainstream*, Berkeley Design

Technology Inc. (BDTI), August 1998.

[63] National Instruments, *Real-Time High-Performance Computing with NI LabVIEW*, NI White Papers, Publish Date: Feb 03, 2012. [Online]. Available: www.ni.com [Accessed: Oct 08, 2015].

[64] Neil H. E. Weste and David Money Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Parson Education Inc, Boston, Massachusetts, USA, 2011.

[65] Victor P. Nelson, H. Troy Nagle, Bill D. Carroll, and J. David Irwin, *Digital Logic Circuit Analysis and Design*. Prentice Hall, NJ, USA, 1995.

[66] Bill Mostia Jr., PE, *The Safety Instrumented function: An S-Word Worth Knowing*, Control for the Process Industries, D351102X012 / 5K AQ / 8-03, Aug 2003.

[67] Honeywell, *Safety Instrumented Systems (SIS), Safety Integrity Levels (SIL), IEC61508, and Honeywell Field Instruments*, Honeywell International Inc, Industrial Measurement and Control. [Online]. Available: http://www.honeywell.com/imc [Accessed: Sep 28, 2015].

[68] ___, PE, *Controls Field communications*, Process Engineering, Controls, Tuesday - 17 October 2006.

[69] Charles L.Phillips and H.Troy Nagle, *Digital Control System Analysis and Design*, 3rd ed. Prentice Hall, NJ, USA, 1998.

[70] Dale E. Seborg, Thomas F. Edgar and Duncan A. Mellichamp, *Process Dynamics and Control*, 2nd ed. John Wiley & Sons, NJ, USA, 2004.

[71] Instrumentation and control. (2006). *Process Control Fundaments*. [On-line]. Available: www.PAControl.com [Oct 10, 2015].

[72] Joseph Casler, Andry Haryanto, Seth Kahle and Weiyin Xu. (Oct 31, 2006). *Cascade Control in the Michigan Chemical Engineering Process Dynamics and Controls Open*

Textbook. [On-line]. Available: https://controls.engin.umich.edu/wiki/ [Oct 10, 2015].

[73] Ming T. Tham. Part of a set of lecture notes, Introduction to Robust Control, Topic: *Internal Model Control*. Chemical and Process Engineering, University of Newcastle upon Tyne, UK, 2002.

[74] N.Abe' and K.Yamanaka, *Smith Predictor Control and Internal Model Control - A Tutorial*, SICE Annual Conference in Fukui, Fukui University, Japan August 4-6,2003, pp. 1383-1387.

[75] EJ Moyer. Lecture notes, Topic: *Basics on electric motors*. U. Chicago, USA, April 29, 2010.

[76] Center of Innovation and Product Development. Part of a set lecture notes, Designing with D.C. Motors, Topic: *Understanding D.C. Motor Characteristics*. Mechanical Engineering, MIT, 1999.

[77] Aniket B. Kabde and A. Dominic Savio, *Position Control of Stepping Motor*, International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering (IJAREEIE), ISSN (Online): 2278 – 8875, Vol. 3, Issue 4, April 2014, pp. 8974- 8981.

[78] Alexandru Morar, *Stepper motor model for dynamic simulation*, ACTA Electrotehnica, Vol. 44, Number 2, 2003, pp. 117- 122.

[79] Bipin Krishna, Sagnik Gangopadhyay , and Jim George, *Design and Simulation of Gain Scheduling PID Controller for Ball and Beam System*, International Conference on Systems, Signal Processing and Electronics Engineering (ICSSEE'2012) December 26-27, 2012 Dubai (UAE), pp. 199- 203.

[80] Dave Wilson, *So, which PWM technique is best?* Motion Products Evangelist, Texas Instruments, Fri, 04/20/2012. [Online]. Available: www.ti.com/mcblog-ecn [Accessed: Oct 23, 2015].

[81] SGS-Thomson Microelectronics, Appl. Note 235/0788, Stepper Motor Driving, pp.1-17.

[82] Robert Hirsch, *Ball on Beam System*, EDUMECH. Product of Shandor Motion Systems, Mechatronic Instructional Systems, 1998-9.

[83] Jeff Lieberman, *A Robotic Ball Balancing Beam,* [Online]. Available: http://www.bea.st/sight/rbbb/rbbb.pdf, February 10, 2004.

[84] David Evanko, Arend Dorsett, Chiu Choi, *A Ball-on-Beam System with an Embedded Controller,* Department of Electrical Engineering, University of North Florida, American Society for Engineering Education, ASEE, [Online]. Available: http://www.asee.org, 2008.

[85] Min-Sung Koo, Ho-Lim Choi, and Jong-Tae Lim, *Adaptive Nonlinear Control of the Ball on Beam System Using the Centrifugal Force Term*, International Journal of Innovative Computing, Information and Control ICIC International, 2012 ISSN 1349-4198, Volume 8, Number 9 - September 2012, pp. 5999-6009.

[86] S.Senthilkumar, P.Suresh, and S.Senthilkumar, *Performance Analysis of MIMO Ball and Beam System using Intelligent Controller*, International Conference & Workshop on Recent Trends in Technology, (TCET) 2012. Proceedings published in International Journal of Computer Applications, (IJCA), September 2012, pp. 19-25.

[87] S.Senthilkumar, P.Suresh, and S.Senthilkumar, *Modeling and Control of Ball and Beam System Using Model-based and Non-model Based Approaches*, International Journal On Smart Sensing and Intelligent Systems, VOL. 5, NO. 1, March 2012, pp. 14-35.

[88] Yeong-Hwa Chang, Chia-Wen Chang, Hung-Wei Lin, and C.W. Tao, *Fuzzy Controller Design for Ball and Beam System with an Improved Ant Colony Optimization*, World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:3, No:8, 2009, pp.

1584-1589.

[89] Sıtkı KOCAOĞLU and Hilmi KUŞÇU, *Design and Control of PID-Controlled Ball and Beam System*, International Scientific Conference, 22 – 23 November 2013, GABROVO, pp. III-41-III-46.

[90] Farhan A. Salem, *Mechatronics Design of Ball and Beam System*, Control Theory and Informatics, ISSN 2224-5774, pp, ISSN 2225-0492, Vol.3, No.4, 2013. (Online) available: www.iiste.org, [Accessed: Oct 23, 2015].

[91] Krzysztof NOWOPOLSKI, *Implementation of Ball-and-Beam Control System as an Instance of Simulink to 32-Bit Microcontroller Interface*, Pozan University of Technology Academic Journals, No 76 Electrical Engineering 2013, pp.31-38.

[92] Zˇeljko Sˇ itum and Josˇko Petric, *A pneumatically actuated ball and beam system*, International Journal of Mechanical Engineering Education 36/3, pp.225-234, Downloaded from: ijj.sagepub.com, at UNIV NORTH TEXAS LIBRARY on February 19, 2015.

[93] BJ Furman. Part of a set lecture notes, ME 106 Intro to Mechatronics, Topic: *Stepper vs. DC Motors and Other Actuators*. Mechanical Engineering, 17 March 2008.

[94] National Instruments, *Motor Fundamentals*, NI White Papers, Publish Date: Oct 07, 2014. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[95] Yokogawa, *CENTUM VP, System Overview (FCS Overview*), Yokogawa Electric Corporation, Nakacho, Musashino-shi, Tokyo, Tech. Rep. TI 33K01A10-50E, Aug. 2011.

[96] DeltaV Whitepaper, *DeltaV Control Module Execution*, Emerson Process Management, January 2013. [Online]. Available: www.DelatV.com [Accessed: Sep 28, 2015].

[97] Siemens Energy & Automation, *DCS or PLC? Seven Questions to Help You Select the Best Solution*, Siemens Process Automation, 2007. [Online]. Available: www.sea.siemens.com

[Accessed: Sep 28, 2015].

[98] National Instruments, *NI Single-Board RIO General-Purpose Inverter Controller Features*, NI White Papers, Publish Date: Dec 05, 2012. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[99] Shelley Gretlein, Gerardo Garcia and Joel Summer, *DSPs, Microprocessors and FPGAs in Control*, Industry Insights, DSP for Control and Inspection, March 2006.

[100] ___, *Feed-forward in position-velocity loops*, PT Design, September, 2000.

[101] Michal Malek, Pavol Makys, Marek Stulrajter, *Feedforwrd Control of Electrical Drives-Rules and Limits*, Power Engineering and Electrical Engineering, Vo: 9 | No: 1, March, 2011, pp. 35 -42.

[102] Richard W. Armstrong Jr., *Load to Motor Inertia Mismatch: Unveiling The Truth*, Presented at Drives and Controls Conference, Telford, England, 1998.

[103] Sharp, GP2Y0A02YK0F: *Distance Measuring Sensor Unit Measuring distance: 20 to 150 cm Analog output type*, E4-A00101EN datasheet, Dec.01.2006.

[104] AMETEK PITTMAN, *Encoders: E30 Incremental Optical Encoder*, E30B datasheet. [Online], available: www.pittman-motors.com [Accessed: Sep 28, 2015].

[105] National Instruments, *Quadrature Encoder Velocity and Acceleration Estimation with CompactRIO and LabVIEW FPGA*, NI White Papers, Publish Date: Feb 03, 2012. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[106] National Instruments, *CompactRIO Integrated Systems with Real-Time Controller and Reconfigurable Chassis*, NI cRIO-907x datasheet. Last Revised: 2014-11-06 07:14:18.0 [Online], available: www.ni.com [Accessed: Sep 28, 2015].

[107] National Instruments, *Creating Custom Motion Control and Drive Electronics with an*

FPGA-Based COTS System, NI White Papers, Publish Date: Jan 15, 2014. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[108] National Instruments, *Using NI CompactRIO Scan Mode with NI LabVIEW Software*, NI White Papers, Publish Date: Sep 19, 2013. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[109] National Instruments, *±10 V, Simultaneous Analog Input, 100 kS/s, 4 Ch Module*, NI 9215 datasheet. Last Revised: 2014-11-06 07:14:15.0 [Online], available: www.ni.com [Accessed: Oct 10, 2015].

[110] *NI 9505, DC Brushed Servo Drive: Operating Instructions and Specifications*, National Instruments Corp, Austin, TX, 2006–2010, pp. 1-40.

[111] *NI 9219, 4-Channel, 24-Bit, Universal Analog Input Module: Operating Instructions and Specifications*, National Instruments Corp, Austin, TX, 2007–2009, pp. 1-44.

[112] National Instruments, *5 V/TTL, Bidirectional Digital I/O, 32 Ch Module*, NI 9403 datasheet. Last Revised: 2014-11-06 07:14:25.0 [Online], available: www.ni.com [Accessed: Oct 10, 2015].

[113] National Instruments, *24 V Sourcing Digital Output, 8 Ch Module*, NI 9474 datasheet. Last Revised: 2014-11-06 07:14:25.0 [Online], available: www.ni.com [Accessed: Oct 10, 2015].

[114] Ametek Pittman, Haydon Kerk Motion Solutions Engineering Development Center, *Brush DC Gearmotor-GM9236S027-R1-SP123*, GM9236S027-R1-SP datasheet. 10/27/2014 8:58:23 PM [Online], available: http://prototypes.haydonkerk.com [Accessed: Oct 10, 2015].

[115] Anaheim Automation, *17YPG Series - High Torque Stepper Gearmotor*, L010469

datasheet. 10/27/2014 8:58:23 PM [Online], available: www.anaheimautomation.com [Accessed: Nov 10, 2015].

[116] National Instruments, *±10 V, Analog Output, 100 kS/s, 4 Ch Module*, NI 9263 datasheet. Last Revised: Last Revised: 2014-11-06 07:14:40.0 [Online], available: www.ni.com [Accessed: Oct 10, 2015].

[117] National Instruments, *Low Cost, Low Power Instrumentation Amplifier*, AD620 datasheet, REV. E, 1999. [Online], available: www.analog.com [Accessed: Oct 10, 2015].

[118] Baumer, *Photo electric sensors: FADK 14U4470/IO*, 11014497. [Online], available: www.baumer.com [Accessed: Oct 10, 2015].

[119] _, *Control Signal Transmission*, Internet: www.pceducation.mcmaster.ca/Instrumentation [Accessed: Oct 10, 2015].

[120] Christer Rameback, *Process Automation -History and Future*, ABB Automation Technology, 2003-09-16.

[121] *MD30C 30A DC Motor Driver data manual*, Cytron Technologies, V1.2, June 2014, 81300 Skudai, Johor, Malaysia.

[122] National Instruments, *Optimizing your LabVIEW FPGA VIs: Parallel Execution and Pipelining*, NI White Papers, Publish Date: Mar 03, 2012. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[123] National Instruments, *C Series WSN Gateway*, NI 9795 datasheet, Last Revised: 2014-11-06 07:15:07.0. [Online], available: www.analog.com [Accessed: Oct 10, 2015].

[124] *NI WSN-3202, 24-Bit, NI Wireless Sensor Network 4-Channel, 16-Bit Analog Input Node: Operating Instructions and Specifications*, 372775E-01, National Instruments Corp, Austin, TX, Nov 10, 2010, pp. 1-34.

[125] National Instruments, *Connect LabVIEW to Any PLC Using OPC*, NI White Papers, Publish Date: Nov 21, 2012. [Online]. Available: www.ni.com [Accessed: Sep 28, 2015].

[126] Ossi Kaltiokallio, Lasse M. Eriksson, and Maurizio Bocca, *On the Performance of the PIDPLUS Controller in Wireless Control Systems*, 18th Mediterranean Conference on Control & Automation Congress Palace Hotel, Marrakech, Morocco, June 23-25, 2010, pp. 707-714.