Final Technical Report - Phase II

Firm: Barrett Technology, Inc.
Grant Number: DE-FG02-02ER83371
Project Title: Universal Telerobotic-Master Device with Advanced Feedback
Project Period: 06/27/03 through 12/26/05
Document Title: DOE-ER-83371-PhII-Final

Single-Page Project Summary

Purpose of the Research
The purpose of this research was to develop a telerobotic master device consisting of a 7-axis backdrivable robotic arm, and a pressure-sensitive grip-controller integrated with a Compact Remote Console (CRC), thus creating a highly functional teleoperation station targeted to control a 6-axis industrial robotic arm and dexterous robotic hand to be used for demolition work in a nuclear setting.

Description of Work Performed
We successfully completed the development of one of the world’s smallest brushless motor controllers due partially to funding through this grant. These controllers are used to drive the motors in the master robotic arm. We also completed the development of an improved model of a highly advanced 4 degree-of-freedom arm – this same arm is the core component in the teleoperation system. The WAM arm and a 3-axis gimbals were integrated with a commercially available CRC at our consultant’s lab at University of Tennessee. Additional support hardware and software were combined to tie the master control system to an existing industrial robot in the lab. A master controller for a dexterous hand was developed and became an integral part of the gimbals handle. Control algorithms were developed and the software was written and implemented. The entire system was then debugged and tested.

Research Results
Results of the prototype system are promising. The WAM Arm, gimbals, hand controller and CRC were successful integrated. Testing of the system to control the 6-axis industrial arm and prototype dexterous hand showed great potential. Relatively simple tasks were successfully performed at slow speeds. Some of the testing was hampered by problems with the slave dexterous hand. This is a prototype hand being developed by Barrett under a different Phase II program. Potential improvements and advancements to the system include improving the control code, and integration of a 2nd master controller arm in order to drive a 2nd slave arm and hand. In summary, the device is a complex system with advanced features and could be used as a universal platform for efficient controlling of robotic arms performing remote tasks in unstructured and uncertain environments such as those prevalent in environmental clean up.

Potential Applications
In addition to the DOE-targeted application of teleoperation in a nuclear setting, other teleoperation tasks could greatly benefit from this new system including remote surgery, and unexploded-ordnance disposal. Additional benefits from this project include the completion of an advanced miniature motor controller for brushless DC motors, and the completion of an improved four degree-of-freedom stand-alone robot arm. The former may become a standard Barrett product, while the latter has already sold ten units since its re-introduction.

© 2006 Barrett Technology, Inc.
Brief Summary of Project Objectives

Objective 1: Complete development of the miniature motor controller

Barrett completed development of its 43 gram miniature controller with integrated optical encoder. In addition to being key to the operation of Barrett’s WAM Arm and thus the master telerobotic system, the “Puck” (Powerful Universal Controller) may stand as a viable product itself.

Objective 2: Electromechanical design of the redundant regional kinematic structure and integration with the Compact Remote Console

Barrett completed development of its improved four degree-of-freedom WAM Arm; and, along with our consultant Dr. William Hamel and his team at the University of Tennessee, successfully integrated it with the Compact Remote Console (CRC). Furthermore, Barrett’s WAM Arm stands alone as a product - since its re-introduction in 2004, we have sold ten units with additional sales pending.

Objective 3: Motorize the backdrivable gimbals of Phase I

As highlighted in the interim reports, the passive gimbals developed under Phase I and improved upon in Phase II proved to be adequate and more desirable than the proposed motorized gimbals. Additional actuators would have proved too costly, unnecessarily complex and would have added substantially to the robot arm end-tip mass.

Objective 4: Develop Neural-Mapped Pistol-Grip Grasp Control of Dexterous Slave Hands

A simple yet effective master controller was developed that consists of five force sensors integrated into the 3-axis gimbals handle and supporting hardware and software. This controller allows the teleoperator to drive both a slave arm and slave hand using a single one-handed controller.

Objective 5: Prepare haptic software in the form of an API

A focused effort on the firmware of the Puck motor controller has lead to a reliable, commercially competitive module. Barrett also put a significant effort into the development of a comprehensive Application Programmers Interface (API) for the entire WAM Arm including a haptics package.

Objective 6: Experimental Demonstration and Evaluation

The majority of the effort in this objective was handled by Dr. Hamel and his team at University of Tennessee. They successfully tested the full master-slave system with promising results.

Objective 7: Master Controller and Haptic Integration

Integration and operation of the master-slave system fell upon Dr. Hamel and his team. They successfully integrated the Barrett 4-DOF WAM arm, Barrett 3-axis Gimbals, dexterous hand pistol-grip controller, Compact Remote Console, Barrett Wraptor dexterous slave hand, and Shilling Titan II arm.

DOE Application(s)

The prototype telerobotic-master system developed under this grant is shown below in Figure 1. The system includes a Compact Remote Console (CRC) from Agile Engineering, Barrett Technology’s 4 degree-of-freedom WAM Arm, Barrett’s modified 3-axis gimbals with integral master controller for a robotic hand, and additional support hardware and computers.
The master system was used to drive a slave system consisting of a Titan II robotic arm from Schilling, a Wraptor dexterous hand prototype from Barrett, and support equipment. The slave system is shown below in Figure 2. The combined master-slave system proved to be an excellent demonstrator on advanced teleoperation, especially in a nuclear setting. This opens the door to offering a turn-key solution for a universal telerobotic master device.

**Commercial Applications**

Figure 3 below shows some of Barrett Technology’s commercially-available robotic systems. Full development of the new 2004 model of the 4-DOF WAM Arm was partially funded through this SBIR. The WAM Arm is a highly dexterous backdrivable manipulator. It is built to outperform today's conventional robots by offering extraordinary dexterity, fast dynamics, high bandwidth, zero backlash, and near-zero friction. It is the only practical arm ever built to be inherently joint-torque controllable.

The impact of the arm as a stand-alone product is likely to be the greatest overall value to the commercial sector. To date we have sold ten 4-DOF units of this model to both universities and companies with several more sales pending. Applications include machine vision and manipulation...
research, manipulation from a mobile robot platform, custom spray-painting of automobiles, and stroke-rehabilitation therapy, to name a few. Furthermore, the teleoperation-related research and findings funded through this grant will help further sales and implementation of this product as a master-unit for any teleoperation system whether or not it is related to the nuclear industry. The left image in Figure 4 shows Barrett’s 3-axis gimbals that attaches to the end of the 4-DOF arm and enables a user to comfortably control the end-tip of the arm. The gimbals axes include potentiometers allowing the system to record trajectories. This gimbals is now a commercially-available option for the 4-DOF Arm system. The right image in Figure 4 shows the modified gimbals that includes force sensors used as inputs in controlling a slave hand.

Figure 3: Barrett Technology’s commercially-available products
Finally, the Puck controller shown below in Figure 5 may become a stand-alone product. We are negotiating the sale of 20 units with our first customer.
1 Objective 1: Complete development of the miniature motor controller

The core innovation of this objective was to finish the development of a miniature DC motor controller and optical encoder to operate the WAM Arm motors. Shown below in Figure 6 is Barrett Technology’s “Puck” mounted directly to the back of one of the WAM Arm motors.

![Figure 6: Puck controller mounted on motor #4 of WAM Arm](image)

On September 30, 2004, Barrett filed internationally for patent rights on the motor controller. The application has been assigned serial number: 60/615,490. We believe that it is worth reiterating that we believe that this is truly a break-through technology. The technology is expected to be especially important to the National Labs at DoE, since brushless servo motors are the only option for explosive environments, and every brushless motor requires its own motor controller. Since nearly all brushless motor controllers are bigger than the motor they drive, Barrett’s 44-gram ultra-miniature motor controller will enable the use of motors that would not have been possible in the past.

Features and specifications for the new controller include:

- size: 35mm diameter x 17mm thick
- mass: 44 grams
- fully encapsulated, potted design
- encoder is included: the controller mounts directly to the motor and contains all encoder electronics
- closed-loop motor current control
- general purpose digital I/O lines and A/D input for user peripheral devices
- CAN bus and RS232 serial compatible

Project-specific benefits of the new controller include:

- considerable reduction of internal wiring for the master controllers
- reduction of system mass
- ability to drive additional actuators for each axis
- precision control of motor currents for low torque-ripple operation
- rugged, potted construction for high reliability
• highly software configurable and reprogrammable for field upgradability

In developing this technology, the mechanical and electrical engineering team worked hand-in-hand to minimize the volume of the Puck. Figure 7 attests to this interaction, as each individual electrical component was modeled in 3D CAD by the mechanical team and then located based on thermal, electrical, and geometric constraints. This placement of components is almost always left to the electrical engineer using a standard 2D PCB layout package, a process that would have yielded far less-optimal results.

![Figure 7: CAD model of Puck](image)

As stated in the original Phase II proposal Objective, we successfully reduced the number of current transducers to a single transducer, miniaturized the entire Puck package, integrated the encoder optics, replaced the DSP with a TI model TMS320F2812, and potted the entire package in thermal compound.

Integration of the Pucks into the WAM Arm is demonstrated in Figure 8 and Figure 9. Figure 8 compares traditional “home-run” wiring architectures and our networked architecture. Figure 8a shows a schematic of a “home-run” architecture in which the motor controllers and amplifiers are bulky and by necessity are located off-board. A large controller cabinet is associated with this architecture and large numbers of wires running through the system are required. In a networked architecture, as demonstrated in Figure 8b, the controllers and encoders are located on the motors themselves, and the motor/controller assemblies rely on a single power bus and communications bus, thus greatly reducing the numbers of wires and the need for a controller cabinet. Figure 9 shows a graphical representation of our implementation of a networked architecture on the WAM 4-DOF Arm and 3-axis gimbals. Two cables now run through the WAM arm, a power cable consisting of 2 conductors and a shield, and a communications cable also consisting of two conductors and a shield.
Motors, sensors, amplifiers, and hi-level controllers

Power supplies
Controller cabinet

Amplifiers (1 per motor)

Robot structure (actively moving)

Each motor wire bundle generally consists of approximately 16 individual wires for phase power, hall effect sensors, and position feedback.

a) Conventional “home run” control schematic  
b) Networked control schematic

Figure 8: Comparison of power and control distribution architectures

Hi-Level Arm
Commands from PC

DC Power 18-90 V

WAM Arm - Safety Board

Puck #5 & 3-axis gimbals

Motor & Puck #4

Motor & Puck #3

Motor & Puck #2

Motor & Puck #1

Power & Communications Buses

Figure 9: Schematic of 4-DOF and gimbals power and control architecture
2 Objective 2: Electromechanical design of the redundant regional kinematic structure and integration with the Compact Remote Console

The telerobotic master developed is based on Barrett’s 4-DOF WAM. The WAM is a 4-degree-of-freedom (DOF) cable-driven backdrivable arm. At the end of the arm, we incorporated a 3-axis gimbals with joint potentiometers. One of these 7-DOF systems was attached to a Compact Remote Console (CRC), thus making up part of the overall telerobotic system. Much of the effort of the mechanical engineering team was focused on a redesign of the WAM Arm to improve its functionality, manufacturability, serviceability, and safety, and to incorporate it with the CRC.

Figure 10 shows the first 2004-model prototype of the WAM arm and the previous model. Numerous improvements were implemented between the two models including:

- System mass reduced from 35.5 kg to 25.4 kg (10.1 kg reduction)
- Base footprint reduced from 1300 cm² to 900 cm² (400 cm² reduction)
- Base height reduced from 20 cm to 16 cm (4 cm reduction)
- Autotensioning capability of cable transmissions added
- Integration of new miniature Puck controller
- Implementation of power and communications bus wiring (large reduction in number of electrical wires running through system)
- Implementation of new safety system

A suitable location and orientation of the WAM on the CRC was determined and the integration of the two was completed. The CRC is an integrated vision assist system from ORNL which is set up in the Robotics and Electromechanical System Laboratory (REMSL) at University of Tennessee. The CRC
has four LCD video monitors, two LCD computer monitors, and a control unit to achieve a broad range of remote operations. The WAM is mounted on a special adjustable custom built base attached to the CRC platform on the left side of the seat. The WAM base provides a convenient height for the operator and can be adjusted for any seat position. Moreover, in order to place the end effector at a convenient position to operate, some constraints have to be implemented, and the constraints are either avoiding joint limit or avoiding an obstacle.

The CRC, as shown in Figure 11, is manufactured by Agile Engineering, Inc. and is obtained by an equipment loan agreement from ORNL. The CRC provides an ergonomic teleoperator workstation for viewing and controlling manipulators. The fitted adjustable chair with the swing-away arm allows the operator to be comfortable for hours, and it has enough free space to place the WAM server and other devices by integrating video equipment under the platform. Video is provided to the operator through four 15 in. LCD monitors and mounted by an adjustable arm. The three lower monitors are primarily used for remote viewing through different cameras. The upper monitor is ready for an extra camera when it is needed by the operator. Furthermore, the two 15” LCD computer panels are connected to the WAM server and the Wraptor control computer for system information display while operating the telerobotic system. The WAM is physically mounted on the left side of the CRC. Due to the symmetric geometry, placing the WAM on the right side is the same as mounting it on the left side.

The WAM is used as a master device in this project for the teleoperation system. This redundant robot has a cable-drive system. This feature provides zero backlash and low friction. It has built-in sensors to measure angles, forces, and torques of every joint. The WAM also has other features such as backdrivability, gravity compensation, and force-feedback capability. The backdrivability allows the WAM joints to be actuated by external forces and makes it suitable for a master. The force feedback uses the backdrivability to measure the force applied at the arm. In addition, the gravity compensation gives the operator the ability to move the arm smoothly.
As shown in Figure 12, the WAM is mounted to the CRC. The operator controls the WAM from a seated position in the CRC. Therefore, the region of operation is constrained to the region that is convenient for the operator from that seated position. Based on these the human factors, the WAM base was designed and fabricated.

The WAM base is made of aluminum alloy plates for reduced weight and durability. The upper plate of the WAM base can move back and forth via a groove. The base is rigidly mounted to support the weight of the WAM and operation force. The suitable height for the WAM base is found as 11 inches from the ground. The WAM is attached on the base and located 3 inches away from the CRC platform. This configuration is suitable for the seated position.

The WAM is a redundant manipulator which has more degrees of freedom than the number required to define a position of the end effector. In general, six degrees of freedom are required to define a point and orientation in 3D space. The extra degrees of freedom of the WAM manipulator are effectively used to improve the ability of the manipulator to avoid obstacles and singular points. On the other hand, the redundant manipulator is more complex to control than a non-redundant manipulator.
Mapping between the operational space and joint space has always been difficult for redundant manipulators due to the presence of a non-square Jacobian matrix. Since a direct pseudo inverse does not work at all times, an additional optimization algorithm is required along with the Jacobian calculation. Namely, some constraints need to be implemented, which would make the Jacobian square and non-singular. Since the WAM has seven degrees of freedom, the redundant manipulator does not have a unique solution. Therefore, functional constraints like joint limit avoidance and obstacle avoidance constraints are introduced to solve the redundancy, and a joint volume limitation constraint provides a comfortable operational space in union with the master manipulator.

**Joint Limit Avoidance**

Redundancy has been effectively used to prevent joint angles from reaching their limits while the end effector is chasing its trajectory. In the joint limit avoidance approach, any joint of the WAM can be identified as the redundant joint and constrained. However, it is wise to select a joint which would keep the elbow joint out of the way of the operator. In other words, the elbow joint is outstretched for all the trajectories. Therefore, the shoulder pitch is identified as the joint to be constrained. This occurrence of joint limit constraint can be systematically implemented such that it expands the task space properly.

**Obstacle Avoidance Constraint**

One of the main advantages of a redundant manipulator is that it can control effectively in an environment by using its redundant degree of freedom, and it can avoid many obstacles in its trajectory. When the end effector is tracking a collision free trajectory, one or more of the links might be very close to an obstacle. In such situations, the inverse kinematics gives the joint angles, which are very close to each other. As the manipulator moves along the trajectory, one or more constraints need to be implemented in order to avoid obstacles. For this purpose, obstacles are modeled as convex volumes in 3D space. The convex volumes are assumed to be made up of simple structures like cubes, spheres and cylinders, so that the exact distance between the obstacle and the manipulator link can be determined. A link is said to have avoided an obstacle if the distance from that link to the obstacle is greater than a predefined threshold distance.

**Joint Volume Limitation Constraint**

The Joint Volume Limitation Constraint (JVLC) is based on the obstacle avoidance criteria. It is another method to input a constraint into the inverse kinematics solution of the WAM, like the joint limit avoidance and obstacle avoidance. The main concern with manipulating the WAM is to keep the end effector in a convenient region of operation for the operator. One observation made from manipulating the WAM is that it is very difficult for the operator to hold the shoulder from falling. In order to provide easy operation for the operator, it is very important to prevent the shoulder from falling. One disadvantage with implementing the obstacle avoidance method in the master manipulation control is that the prevention of the free falling shoulder is only in the direction towards the obstacle. In all other directions, beyond a certain shoulder pitch angle, gravity pulls the arm. One way to solve this is to restrict the region of movement of joint four in all directions. That is exactly what the name JVLC suggests. JVLC is very much similar to the obstacle avoidance criteria except for the error definition and criteria for activating the constraint.

### 3 Objective 3: Motorize the backdrivable gimbals of Phase I

The passive gimbals developed under Phase I and improved upon in Phase II proved to be adequate and more desirable than the proposed motorized gimbals. Additional actuators would have proved too costly, unnecessarily complex, and would have added substantially to the robot arm end-tip mass.
Instead of developing a motorized gimbals, we refined the prototype of Phase I into a commercially-available option for our 4-DOF Arm. Improvements included the integration of a single Puck readily connectable to the WAM Arm power and bus structure to process the outputs from the 3 joint potentiometers. Digital filtering improved the joint resolution.

### 4 Objective 4: Develop Neural-Mapped Pistol-Grip Grasp Control of Dexterous Slave Hands

The “slave” robotic hand used for this Objective is Barrett’s Wraptor prototype shown below in Figure 14. Dr. Hamel’s team at University of Tennessee developed a simple and intuitive “master” controller (hardware and software) that is an integral part of the teloperation system. The first proof-of-concept prototype is shown in Figure 15.

![Figure 14: Barrett Technology's Wraptor prototype used as “slave” robotic hand](image)

A new Linux-based control system for the Wraptor robotic hand has been created. The system utilizes force sensors that are mounted on the 3-axis gimbals of the WAM. Thus, the teleoperator can control
both the Titan arm and the Wraptor hand from a single mechanically integrated master controller. At this point, the Titan can be controlled by the WAM with full teleoperation. The current Wraptor control software utilizes the signals from the force sensors mounted on the WAM, but it still requires some software modifications in order to allow complete control using only those force sensors.

The mechanical integration of the master controller for the Wraptor and the WAM required a modification of the original gimbals. The sensors needed to be mounted on the handle such that they are easily accessible to each of the five fingers. The development of the new WAM handle with five force sensors has been completed; it is shown in Figure 16. The new handle is rotated 90° from the original; this enables more accessible and comfortable placement of the sensors.

![Figure 16: modified gimbals with integrated force sensors](image)

Analysis has been performed in order to determine positions for the sensors on the handle such that each sensor is easily accessible to a human finger, for a variety of hand sizes. Additionally, since the sensors are stationary and very thin, the handle was designed such that the operator can easily feel the precise locations of the active areas on the sensors.

Tekscan resistive force sensors were selected because they allow a variable input to allow the operator to control the grasping torque. They act as a variable resistor in the circuit; as the force applied to the sensing area increases, the resistance decreases. Each sensor is connected to a separate force-to-voltage converter circuit, which is connected to the data acquisition board. Data acquisition is performed by using a National Instruments data-acquisition board; the current software is able to acquire and utilize these signals from the force sensors.

The current control strategy uses a set of automated trajectories that can be initiated by the teleoperator. The operator has the capability to command the Wraptor to an initial open approach position, grasp several objects and release the objects by commanding the Wraptor fingers back to the open position. For development purposes, the current system utilizes the sensors only to select whether to grasp the
reciprocating saw, the bandsaw or a cylindrical pipe. An intuitive graphical user interface has been created that completes the control of the Wraptor.

Analysis has been performed to determine the most stable grasp trajectories and configurations for the three main objects of interest: the reciprocating saw, the bandsaw and a cylindrical pipe. In all three cases, the grasp trajectories are commanded in the following manner. First, the Wraptor is commanded to an initial open position to approach the object. Next, velocity commands are sent to wrap the fingers around the object. Finally, the joint torques reach preset maximum values and the motors stall. The test results were promising; they showed that the Wraptor is able to grasp these three objects in a stable manner, plus a sphere and a disc, with no mechanical modifications, although the current bandsaw presents some challenges. Figure 17 shows the Wraptor holding the reciprocating saw.

![Figure 17: Wraptor holding reciprocating saw](image)

Furthermore, tests were performed in which both the cylindrical pipe and reciprocating saw were successfully grasped from a test fixture, moved to another position and replaced on the test fixture. Additionally, the Wraptor was successfully used to hold the reciprocating saw while cutting a 2”x4” piece of wood. These tests were performed using both the WAM and the Linux/C based Wraptor control system developed at REMSL. These tests will be discussed in detail in a later section. Figure 18 shows the Wraptor holding the reciprocating saw at an initial cutting position.
A Linux based graphical user interface that allows position and velocity control of individual Wraptor joints as well as automated grasp commands for the three main objects has been created. Based on the Barrett software, the Wraptor joint positions, velocities and torques can easily be commanded. Some parts of this user interface were created to facilitate the development of the final control system, rather than for use in the final control system itself. The graphical user interface has three main windows. The basic operations window allows the user to connect, initialize and zero the Wraptor and choose between a teleoperation mode and a joint control mode. The joint control mode or “manual control” window has the capability to send position and velocity commands to individual joints. The teleoperation control window displays the force sensor data and allows the user to execute the automated grasp commands.

5 Objective 5: Prepare haptic software in the form of an API

With regard to the objectives laid out in the proposal Barrett performed well. A focused effort on the firmware of the motor controller has lead to a reliable, commercially competitive module. Work on the data packet protocol for motor controller network communications led to a more efficient and flexible system that will allow additional modules such as sensors and user interaction devices to be placed on the same network. Upon a detailed review of the costs/benefits of moving feedforward and other control algorithms to the motor controller, Barrett decided that it was more practical to move this functionality to the API. The resulting system had no degradation in performance.

Efforts on the API were also successful. Bugs in the prototype communications driver from phase I were located and fixed. Initialization and shutdown routines were created that use logical default parameters and ease the workload of the programmer when writing short applications. Safety and controller health monitoring functionality in the API was deferred and moved to a separate piece of hardware instead. This separation of operation and safety hardware has lead to a much more robust system for use with a person in the workspace when the WAM is used as a teleoperation master.

Cartesian position and force control where implemented successfully. The redundant degree of freedom in the WAM while operated in Cartesian control affords additional operator safety by allowing the
robot arm to seek a low energy potential when the operator is in the robot's path of motion. This redundancy means that when the robot moves and a person is in the way, the robot will naturally wrap around the person while it seeks its destination rather than trying to force its way through the person.

Haptics primitives were also implemented successfully. Spheres and planes were used initially as a proof of concept before more complicated shapes were integrated.

Barrett put a significant effort into the development of a comprehensive Application Programmers Interface (API). To provide a useful set of haptics routines, a great deal of infrastructure was developed. The value of this infrastructure is so great that Barrett decided to provide the full source code to its customers with the hope that they can learn from it and extend its use both in specific applications using the WAM and other robotics projects.

Core features of this library include:

- Written in C in a style meant to be easily portable to other systems.
- The source code is provided.
- The code is well documented. This documentation can be extracted into html or a word document using the well-known free documentation tool Doxygen.
- A N-link serial chain robot kinematics & dynamics solving module.
- An analytic geometry haptics library suitable for basic haptic simulations and virtual fixturing. This library is organized to be easily extended for more advanced and application specific solutions.
- Full-speed data logging facilities that allow the programmer to quickly specify arbitrary information to be tracked. Data is recorded without interrupting the critical real-time control loop.
- Convenient text based configuration file facilities allow rapid iteration of robot control parameters without recompiling the code.
- Comprehensive debugging layer to help programmers find errors in their code and ease the difficulty in working with a real-time system where there are many programming constraints.
- Seven detailed example programs that teach a programmer new to the system how to use most of the features of this API.
- A very efficient math library suitable for real-time systems was created. Functionality includes vector/matrix math, quaternion math, and digital filtering.

During the course of this work several innovations were created.

- A comprehensive notation for vector and matrix math was created to make algorithms more clear in code and suitable for automated equation generation in Computer Algebra Systems (CAS) software like Mathematica.
- In the trajectory-following code that was implemented, a distinction was made between the path (position component) and the trajectory (velocity component) to provide a greater flexibility in working with analytically generated trajectories and space curves for haptic guides.
- A separation of geometry from the haptic surface normal and tangential impedance response was made. This separation not only increases flexibility of the library but also creates a clear focal point for creating new interaction experiences. The increased flexibility allowed an intern with no previous haptics experience and no hand-holding to create a new surface interaction quickly.

6 Objective 6: Experimental Demonstration and Evaluation

The experimental strategy is a three-stage process with progressively increasing complexity that allows the performance properties of the teleoperation system to be demonstrated and evaluated in a
systematic manner. Note that additional testing was done under Objective 7, during the initial integration and debugging phase.

6.1 First Stage – proof-of-concept test

Equipment: modified WAM arm installed on CRC, Titan II manipulator, and Wraptor hand with modified Gimbals control interface.

Purpose: Determine the possibilities of grasping objects and tools.

Strategy:

1. The operator sits in the CRC.
2. The objects are placed in the tool stand within the arm reach.
3. The arm is moved toward the object to a position convenient for a grasp.
4. The Wraptor grasps the object.
5. The Wraptor releases the object.
6. The arm is shifted away from the tool stand.

After full system integration (see Objective 7), the arm was teleoperated toward the tool rack where a pipe was picked up and released. This task was successfully performed.

6.2 Second stage – testing system functionality with straightforward tasks

Equipment: modified WAM arm with master device for controlling a Barrett Wraptor hand, CRC, Wraptor hand installed on Titan II manipulator, tool stand with objects.

Purpose: Test the coordination of the UTMD with the master device and D&D manipulators (TII manipulator) in teleoperation.

Strategy:

1. The operator sits in the CRC.
2. Objects are placed in the tool stand within the arm reach
3. The arm is moved toward the objects to position convenient for a grasp.
4. The Wraptor grasps the object.
5. The object is moved away from the tool stand.
6. The object is returned to the tool stand.
7. The object is released.
8. The arm is shifted away from the tool stand.

In the second stage, the complexity of the experiment was increased and new steps were added to the existing first stage. It included showing the abilities of WAM and Titan II to move objects from position to position. It also included the testing of the grasp quality. A pipe was successfully grasped, moved away from the stand and put back in its original place. The grasp has been proven to be firm enough to hold and move objects of simple shape. UMTD has been proven to supply enough robustness to perform all the mentioned steps. A video of the entire procedure was recorded and is available.
6.3 **Third stage – performing complex tasks**

Equipment: UTMD with master device for Wraptor hand, CRC, Wraptor hand installed on TII manipulator, reciprocating saw, tool stand, and wood for cutting.

Purpose: Test the functionalities of UTMD and Barrett Wraptor hand and evaluate the overall system performance in complex tasks

Strategy:

1. The operator sits in the CRC.
2. The reciprocating saw is placed in the tool stand within the arm reach.
3. A 2”x 4”x 8 ft. piece of wood is mounted on the tool stand.
4. The operator moves the arm toward the tool to a position convenient for a grasp.
5. The tool is grasped.
6. The saw is moved toward the cutting point.
7. The saw is turned on.
8. The cutting operation is performed.
9. Steps “o” and “p” are repeated for another cut
10. The saw is turned off.
11. The tool is returned to the stand.
12. The tool is released.
13. The arm is shifted away.

The third stage was proposed to test the ability of performing long and complex tasks. However, this task could not be performed with a full success due to the problems with the grasp quality of the Wraptor during high vibrations. The reciprocating saw strongly vibrates during its normal operation, causing the tool to slide or “walk” out of the fingers and the fingers to open slightly. However, these problems were mitigated and a cutting operation was successfully performed. [Note that improvement of the Wraptor dexterous hand is outside the scope of this project, but is in fact funded under a separate Phase II SBIR.] All steps were performed successfully and videotaped. Described experiments are shown in the picture sequence in Figure 19.
The experimental results prove the ability of the integrated system to perform teleoperation tasks. It has been shown that the basic operations, including grasping and moving objects and performing cutting operations, are possible using the WAM with cooperation of Titan II manipulator and the Wraptor hand. However, the dexterity and the execution time for those tasks imply the need for certain improvements. As mentioned, the Wraptor improvements are ongoing under a separate Phase-II grant.
7 Objective 7: Master Controller and Haptic Integration

Teleoperation is widely used in human adverse environments, such as nuclear waste remediation and space exploration. Teleoperation usually involves master and slave devices. One commonly used teleoperation system features a master device that is suitable for joint space control, for example, a miniature of the slave.

In this research, a new teleoperation system was set up; a teleoperation-control strategy was established for the system, and the algorithmic method for Cartesian space mapping was implemented. In the proposed teleoperation system, a Whole-Arm Manipulator (WAM™) arm, which is a seven degree-of-freedom (7-D) manipulator made by Barrett Technology®, Inc., is used as a master manipulator, and the slave is a Titan II 6-D manipulator by the Schilling Robotics® LLC. Due to the kinematic dissimilarity between the two manipulators, Cartesian space mapping is used in the research.

The proposed mapping method can prevent numerical problems such as sudden jumps of the slave manipulator, as shown by the simulation and demonstration experimental results. A distributed PC based control system involving three computers is used for the WAM-Titan II system. The communication among the computers is via a dedicated local area network (LAN). A simulation framework is also set up for system development and testing. The proposed teleoperation system is part of a dual arm teleoperation platform of the University of Tennessee, Knoxville, and the Oak Ridge National Laboratory (ORNL). The complete system diagram is shown in Figure 20.

![Figure 20: Teleoperation system architecture](image-url)

There are two teleoperation systems in the architecture: the Minimaster-Titan II (R) teleoperation, which is still under development, shown as a dotted line, and the WAM-Titan II (L) teleoperation, which is the focus of this grant, shown as blue dashed line. The second teleoperation system also has two teleoperation subsystems: the WAM-Titan II teleoperation, which is discussed in detail in this section, and a master and slave hand, which are discussed in a separate section in this report.
In the remainder of this section, the technical issues associated with the WAM-Titan II teleoperation system are discussed in the following aspects:

1. System Hardware and Software Integration
2. Mapping Algorithm and Control Strategy
3. System Debugging and Analysis

### 7.1 System Hardware and Software Integration

#### Functions and requirements

The following functions and requirements are achieved in the teleoperation system:

1. The Titan II moves in a controlled manner with smooth trajectory.
2. The Titan II has free movement in its whole working space, with the exception of the singularity regions.
3. The software implementation has no noticeable effect on the performance of the WAM control software and Titan II control software.
4. A high bandwidth of teleoperation loop frequency can be achieved with the Ethernet-based communication network.
5. The mapping algorithm is designed for position mapping and orientation mapping simultaneously.
6. Different operation modes can be chosen from the user interface for operation convenience.
7. Experimental results showed that operation accuracy can be controlled well if sufficient video feedback is available.

#### System Overview

The overall Master system is shown in Figure 21 with the Slave system in the background.

The Wraptor interface with force sensors has been mounted on the WAM handle (Figure 22). For this purpose, a new D shaped handle has been designed and manufactured. The previous one did not
provide sufficient dexterity for maneuvering and using sensors. The video output for the Wraptor was connected to the CRC in order to provide necessary information for the operator (Figure 23).

![Figure 22, WAM with a new handle and sensors](image1)

The Wraptor has been mounted on the Titan II arm using a long disc-shaped adapter. This configuration provided unrestricted movement abilities. It is presented in a picture below.

![Figure 23, Operator's control center](image2)
Test Mockup and Support Equipment

In order to perform the experiments and meet the goals of the project, specific equipment was needed. A full list of the equipment used for the experiments is shown below:

1. Schilling Titan II arm mounted on a beam
2. Hydraulic power unit for Titan II
3. PC-104 – high level controller for Titan II
4. Barrett WAM master device
5. Barrett Wraptor robotic hand BH8-600 series
6. Wraptor interface with force sensors
7. Wraptor control PC
8. Compact Remote Console (CRC)
9. Three cameras for teleoperation
10. Server PC
11. Inverse kinematics PC
12. Wraptor control PC
13. Roller’s Fox reciprocating saw, type 55
14. Tool stand/rack
15. 5” pipes for grasping
16. 2” x 4” x 8 ft. piece of wood for cutting
17. Cables and other accessories

WAM 4-DOF details

The master manipulator in the system is the WAM manipulator, which has seven degrees-of-freedom; the geometric configuration is shown in Figure 25. The WAM system features a gravity and friction compensation function for joints J1, J2, J3, and J4. The last three joints make up the Gimbals. The D-H parameters of the WAM are given in Table 1.
Table 1, D-H parameters for the WAM

<table>
<thead>
<tr>
<th></th>
<th>Joint 1</th>
<th>Joint 2</th>
<th>Joint 3</th>
<th>Joint 4</th>
<th>Joint 5</th>
<th>Joint 6</th>
<th>Joint 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (mm)</td>
<td>0</td>
<td>0</td>
<td>0.045</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d (mm)</td>
<td>0</td>
<td>0</td>
<td>0.55</td>
<td>0</td>
<td>0.1547</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>α (deg)</td>
<td>π/2</td>
<td>π/2</td>
<td>π/2</td>
<td>π/2</td>
<td>π/2</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>θ (deg)</td>
<td>θ₁</td>
<td>θ₂</td>
<td>θ₃</td>
<td>θ₄-π/2</td>
<td>θ₅-π/2</td>
<td>θ₆</td>
<td>θ₇</td>
</tr>
</tbody>
</table>

**Titan II details**

The slave manipulator in the research is the Titan II manipulator, which is hydraulic driven and has six degrees of freedom. The geometric configuration of the Titan II is shown in Figure 26. The D-H parameters for the Titan II are given in Table 2.

The original Titan II comes with a gripper fitted at the end plate of the sixth joint; in this research, a three fingered hand, called the Wraptor, is used to replace the gripper.
Table 2, Titan II D-H parameters

<table>
<thead>
<tr>
<th></th>
<th>Joint 1</th>
<th>Joint 2</th>
<th>Joint 3</th>
<th>Joint 4</th>
<th>Joint 5</th>
<th>Joint 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (inch)</td>
<td>4.77</td>
<td>33.50</td>
<td>19.00</td>
<td>5.25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d (inch)</td>
<td>7.67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.61</td>
</tr>
<tr>
<td>α (deg)</td>
<td>π/2</td>
<td>0</td>
<td>0</td>
<td>-π/2</td>
<td>π/2</td>
<td>0</td>
</tr>
<tr>
<td>θ (deg)</td>
<td>θ₁</td>
<td>θ₂</td>
<td>θ₃</td>
<td>θ₄</td>
<td>θ₅+π/2</td>
<td>θ₆</td>
</tr>
</tbody>
</table>

Compact Remote Console (CRC) details

The CRC provides three live video views for the teleoperation. Their purpose is to provide sufficient position and orientation information of the Titan II, the tool, and the task object. Currently, the three views are front, back and left view; operation experience indicates that an additional top view will be beneficial for the operation. Figure 11 shows the views layout on the CRC.

The Control System and Software

The teleoperation system software runs on two PCs and one PC-104. The system diagram is shown in Figure 27.

![Figure 27, WAM Titan II teleoperation system diagram](image)

The second teleoperation system includes a WAM PC, a server PC, a PC-104, and a development PC. The first three PCs in this path are the actual teleoperation system. The development PC is used to develop the Titan II control software and download it to the PC-104 through Qnet™, or QNX® networking. The second path includes the WAM PC, the server PC, and the simulation PC; this path is used for the system design and testing. All five computers are connected through a dedicated LAN.

The WAM PC hosts the low-level control software. It also has an Ethernet client module, which sends WAM joint angles at 20 Hz to the server PC. Communication between the low-level control program and the Ethernet module is by a RTAI FIFO [1]. The entire program in the WAM PC runs under Linux/RTAI (Real Time Application Interface), which is a hard real time Linux kernel extension founded by the department of aerospace engineering of Politecnico di Milano (DIAPM) [1]. The low-level WAM control-loop rate is 500 Hz.

The PC-104 hosts the low-level control program for the Titan II. It also has an Ethernet client module; communication between the two is by shared memory and semaphores. The Titan II low-level control software runs under QNX, and its control loop update rate is 200 Hz.

The server PC receives joint angles from the WAM PC and the PC-104, executes the high level position mapping, and generates the Titan II joint angle command, which is then sent to the PC-104 via Ethernet. The server PC is a RedHat Linux PC.

The simulation PC has a commercial software package, Roboworks™, for robot simulation under a Windows operating system. It has a TCP/IP module to receive inputs via Ethernet.
In order to support precise velocity mapping, the difference between the clock cycles in the PC-104 and the WAM PC must be sufficiently small, which is guaranteed by using realtime operating systems (RTOS). The teleoperation loop update rate is 20 Hz, which is much lower than the bandwidth of a dedicated LAN, so time delay is not an issue in this project.

**Titan II Low-Level Control Software**

The Titan II low-level control software is provided by ORNL with the Titan II manipulator. The software takes joint angles as commands from shared memory and performs low-level position servo control of the Titan II. The control software is developed using C language under a real time operating system QNX and is hosted in a PC-104.

**WAM Low-Level Control Software**

The WAM low-level control software is provided by Barrett Technology with the WAM manipulator. The software is developed using C language under a real time operating system RTAI/Linux. The WAM low-level control software is a self-contained module, and gravity/friction compensation is one of many kinds of functionalities provided in the software. The WAM low-level control software is hosted in the WAM PC.

**High-Level Control Software**

The high-level control software is a newly designed software library for the teleoperation system. The main function of the high-level control software is to perform Cartesian-space mapping and to implement the control strategy. The high-level control software has four modules: the WAM direct kinematics, the Titan II inverse kinematics, the WAM Ethernet Module and the Titan II Ethernet Module. The software system diagram is shown in Figure 28.

![Figure 28, WAM Titan II teleoperation system software diagram](image)

The system data flow is described in the following steps:

1. Titan II initial joint angles are sent to the high-level control software through QNX message passing scheme and Titan II Ethernet Module.
2. The WAM is operated in gravity-compensation-enabled state, and joint angles are measured along with the time stamps and then sent to a RTAI FIFO.
3. The WAM Ethernet Module receives the WAM joint data and sends them to the high-level control software through ethernet.
4. The Cartesian-space velocities (three linear velocities and three angular velocities) of the WAM end point are generated based on the WAM forward-kinematics calculation.
5. The WAM Cartesian-space velocities are used as inputs to the Titan II inverse-kinematics calculation, and the Titan II joint velocities are generated.
6. The Titan II joint positions are generated based on the joint velocities generated in step 5.
7. The Titan II joint positions are sent to the Titan II low-level control software.
8. The Titan II low-level control software receives the joint position command through the Titan II Ethernet Module and shared memory and does position servo control of the Titan II.

Once step 8 is finished, the system will start over from step 1. Steps 1 and 8 happen in the PC-104, which hosts the Titan II low-level-control software; Steps 2 and 3 happen in the WAM PC, which hosts the WAM low-level-control software; Step 4, 5, and 6 happen in the server PC, which hosts all the high-level Cartesian-space mapping algorithm and the control strategy.

There is a user interface in the high-level control software as shown in Figure 29.

![Figure 29: User interface](image)

### 7.2 The Mapping Algorithm and Control Strategy

For a 6-D Titan II manipulator, the Jacobian is a 6 by 6 square matrix, and mapping of a 7-D master to a 6-D slave manipulator can be solved analytically. But in practice, techniques must be adopted to avoid uncontrollable slave movements that can result from numerical problems when the Titan II approaches a singularity configuration. Generally speaking, a teleoperation system can have two fundamental control schemes: open loop and closed loop. Under these two schemes, various control methods have been studied in the past. The common objective of these methods is to try to attain good performance in the event of redundancy or singularity, and in these cases, the Jacobian matrices are rank deficient.

One commonly utilized method was the damped least square (DSL) inverse proposed by Y. Nakamura and H. Hanafusa [2], which augments the Jacobian to a full ranked matrix when it is rank deficient. It is reported that the DSL has poor performance near the singularity [3]. Optimization-based algorithms can be found in [3, 4]; these methods generally use some type of weighting matrix, and the selection of the weighting matrix is based on a heuristic tuning method. It is also reported that the optimization-based algorithm could be 20 times slower than the analytical solution [3].

A closed-loop scheme is formed by feeding back the actual Cartesian positions of the slave. The purpose of including feedback is to reduce the tracking errors and stabilize the system [5].

An open-loop scheme includes a forward-kinematic (FK) routine and followed by an inverse-kinematic (IK) routine. Four steps are involved in this scheme, and its general procedure is outlined as follows:
where,

\[ v_{\text{car}} \] - cartesian space velocity
\[ v_{\text{wamjt}}, \Delta \theta_{\text{wamjt}}, J_{\text{wam}} \] - WAM joint velocity, angle, jacobian
\[ v_{\text{titanjt}}, \Delta \theta_{\text{titanjt}}, J_{\text{titan}}^+ \] - Titan II joint velocity, angle, pseudo-inverse jacobian
\[ \Delta t \] - time step

Step (1) and step (4) both have a time increment, and the two time steps must be the same in order to obtain exact tracking. In this system, the timers that keep track of the time steps are in two different computers, and thus make it critical to have RTOS on the WAM PC and the PC-104. The clock difference between the two timers under RTOS can be neglected, since the time delay of RTOS is in the magnitude of several microseconds, which, together with the LAN time delays, introduce some tracking errors in the system. The effects of those errors are fairly small [6, 7].

The FK routine calculates the Cartesian-space velocities from the measurement of the master joint angles; and the Cartesian-space velocities are then used as command inputs to the slave IK routine, which produces the joint velocities of the slave; the joint angles are finally used as slave-actuator command inputs.

A new technique based on an open-loop scheme analytical solution is studied in this research, and the proposed method has good performance near the singularities.

As mentioned earlier, the Jacobian of the Titan II is a square matrix, and the analytical inverse Jacobian always exists. By theory, the IK problem has a closed-form analytical solution. In reality, problems like abrupt joint-angle jumps due to numerical problems can happen when the Titan II is near its singular point. For example, when joints 2 through 5 are aligned, the Titan II cannot move farther in the singular direction, which is generally referred to as a degree-of-freedom loss.

If a Cartesian-space velocity command as shown in Equation (2) is used, then the Titan II moves to its singular point quickly with its initial configuration shown in Figure 30.

\[
\begin{align*}
\{v_x, v_y, v_z, \omega_x, \omega_y, \omega_z\} &= \{2, 3, 4, 0, 0, 0\} \\
v_x, v_y, v_z &= \text{velocity in x, y, and z axis (inch/sec)} \\
\omega_x, \omega_y, \omega_z &= \text{angular velocity about x, y, and z axes (rad/sec)}
\end{align*}
\]
When the Titan II is approaching the singularity, the simulation result shows that the joint angles jump abruptly between adjacent time steps (0.05 seconds). This type of behavior will cause actuator saturation and thus should be avoided. Figure 31 shows the Titan II joint velocity simulation results. Position tracking is lost due to the singularity, which is shown in Figure 33.

From Figure 31, the Titan II-joint-velocity numerical results undergo large jumps under the constant Cartesian-speed command. It is commonly known that when using the procedure step (2) shown in Equation (1) to solve an IK problem, the Jacobian matrix must have a sufficiently small condition number, which is defined as the ratio between the largest and the smallest singular values. Otherwise, any small variation in the command input will be disproportionately amplified by the inverse calculation, thus causing abrupt changes in the output joint angular velocities.

This type of situation can be avoided by setting a heuristic threshold for the condition number, and if a small singular value goes below the threshold it is set to zero before the inverse routine. However this is not exactly what occurred in the simulation because the input commands are constant. Figure 32 plots the ratios of the largest singular value to each single singular value. First observation is that the condition number has abrupt changes. Further examination of Figure 31 and Figure 32 show that the
joint velocity jumps happen at the exact locations where the condition number is undergoing large changes.

The explanation of this situation is that because of the abrupt condition number change between adjacent Jacobian matrices, especially when a large condition number is involved, some command velocities are disproportionately amplified by the inverse calculation, thus resulting in very abrupt
output-speed changes. A heuristic technique is used by setting an appropriate condition-number threshold in this case. For example, by setting the condition number threshold to 15, and keeping everything else unchanged, a new simulation result shows a much smoother joint velocity trajectory. The results are shown in Figure 34 and Figure 36.

Figure 34, Titan II joint velocities

Figure 35, Ratio between largest and every single singular value of Titan II Jacobian matrix
By comparing the position errors as shown in Figure 33 and Figure 36, it is apparent that the errors in the x direction for both cases are very large. The cause of these large errors is that the Titan II is near a singularity; however, in the y and z directions, the errors are relatively small. All errors are in similar magnitude before and after a condition-number threshold is used. When the Titan II is moving away from a singular point, the performance of the analytical solution is very good without setting a condition-number threshold. For example, when changing the command in Equation (1) to \{-2, 3, 4, 0, 0, 0\}, the Titan II starts from the same initial configuration. The Titan II will move away from the singular point. Numerical simulation results are shown in Figure 37 and Figure 38.
The joint velocities are smooth and tracking errors are small. Notice that the condition-number threshold is not set. Based on the above analysis, the conclusion is that by setting an appropriate condition-number threshold when the Titan II is moving into a region near a singularity point, the analytical IK solution can perform well near the singularity.

Even though the proposed method is theoretically correct, there is a practical issue: the orientation calculation is sensitive to the sensor noise, which makes it very difficult to control the orientation when the sensor noise is large.

In this research, the Cartesian-space mapping method can do position and orientation control simultaneously. Three other joint-to-joint controls are provided for better orientation adjustment; particularly, the elbow (J4) is used as a master to control the three wrist joints of the Titan II. Experimental results show that the proposed control strategy is able to achieve good position control and orientation control when the gimbals sensor data is subjected to significant noise. The basic idea is to decouple the position control and orientation control such that the operator can focus on either position control or orientation control of the Titan II, not both. The control strategy can be explained as follows: Cartesian-space position velocities generated in step 4 are used to control Titan II position, but the orientation velocities are set to be zeros such that the orientation of the Titan II does not change during the position mapping. So, strictly speaking, both position and orientation are controlled simultaneously, but the orientation is controlled to be constant. Once the operator wants to control the orientation, he can do a mode change through the user interface. Experimental results show that good position and orientation control are achievable with the proposed control strategy, with good video feedback.

Based on the above analyses, the proposed control strategy can be divided into three functional blocks:

1. An FK routine for the WAM
2. An IK routine for the Titan II
3. A joint level orientation control for the WAM and Titan II
7.3 System Debugging and Analysis

The experiment result in this section is mainly for the verification of the Cartesian-space mapping method, so the focus is on tracking-error analysis. For visual demonstration, in lieu of using the Titan II manipulator, a 3-D Titan model was used.

The experimental results are obtained as follows. As the WAM is moved by following a prescribed trajectory, the WAM joint-angle measurements are sent to the server PC for FK and IK routines, which calculate the Titan II joint angles, and then these joint angles are sent to the Roboworks Titan II model in the simulation PC for a visual demonstration. In addition, the experimental results are plotted by using Matlab® for error analysis.

The WAM initial position and base frame can be shown in Figure 39 (left), and the initial position and base frame of the Titan II are shown in Figure 39 (right).

Two WAM trajectories were tested. The first one was to open the elbow and then close it back to the initial position. The second one was to lift the shoulder and open the elbow simultaneously, and then move them back to the initial positions.

The purpose of the first test is to see the performance when the Titan II is far away from the singularity, while the second one is to check the performance while the Titan II is near the singularity.

In the first test, the condition-number threshold is not set. In the second, the condition-number threshold is set to 10 when the Titan II is moving towards a singular point and 50 while it is moving away from the singular point. When the Titan II is far away from the singularity, the threshold is not set.

The results of the first test are shown in Figure 40 through Figure 43. The results of second test are shown in Figure 44 through Figure 47. The selection of the condition-number threshold depends on the geometric construction of the manipulator; in this research, an operational space condition number map was used to help determine an appropriate threshold; aside from that, it is basically a trial and error method.
Figure 40, Titan II position in x, y, and z axes

Figure 41, Titan II position error in x, y, and z axes
Figure 42, Titan II joint velocities

Figure 43, Titan II velocity errors in x, y and z axes
Figure 44, Titan II position in x, y, and z axes

Figure 45, Titan II position error in x, y, and z axes
Figure 46, Titan II joint Velocities

Figure 47, Titan II velocity errors in x, y and z axes

Figure 40 and Figure 41 show that good position and velocity tracking are achieved when the Titan II is far away from the singularity points.

Figure 40 and Figure 41 show that when the Titan II moves into a region near the singular point, before 4 seconds, position and velocity tracking are lost. When it moves out of this region, after 4.5 seconds, position and velocity tracking are achieved approximately.
In all cases, the Titan II joint movements are smooth and the Titan II is in a controllable state even when it is close to a singular point.

In conclusion, the Universal Telerobotic Master Device is a user-friendly system with advanced features that create a universal platform for efficient controlling of robotic arms equipped with dexterous hands. The system is ideal for remote tasks in unstructured and uncertain environments such as those prevalent in environmental clean up and similar applications.

8 References