SAND--97-3071C
 SAND97-3077C
 CONF-970469--

RECEIVED

DEC 22 1997

OSTI

A Modular Approach to Redundant Robot Control

Robert J. Anderson
 Sandia National Labs
 M.S. 1176, P.O. Box 5800
 Albuquerque, NM 87185

Abstract

This paper describes a modular approach for computing redundant robot kinematics. First some conventional redundant control methods are presented and shown to be "passive control laws", i.e. they can be represented by a network consisting of passive elements. These networks are then put into modular form by applying scattering operator techniques. Additional sub-network modules can then be added to further shape the motion. Modules for obstacle detection, joint limit avoidance, proximity sensing, and for imposing nonlinear velocity constraints are presented. The resulting redundant robot control system is modular, flexible and robust.

1.0 Introduction

Sandia National Laboratories and the Department of Energy has been involved in a number of projects involving redundant robot arms, ranging from 7-11 degrees of freedom (DOF). The 9-DOF Light Duty Utility Arm (LDUA) (Fig 1), for instance was designed for snaking around vertical obstacles in a waste storage tank while conducting inspection tasks. The Hanford test arm uses a macro-micro approach, with a 6-DOF fine-positioning robot attached to the end of a 60 ft reach positioning platform. Other redundant robot projects have involved material deposition and inspections, where redundancy is used to insure smooth steady motion for all configurations.

FIGURE 1: Light Duty Utility Arm.

One of the technologies developed as part of these and other efforts is SMART (Sequential Modular Architecture for Robotics and Teleoperation) [1, 4]. SMART allows the system developer to build up advanced telerobotic behavior by combining SMART modules in different sequences. With SMART it is assumed that all system components—such as input devices, sensors, control laws, robots, constraints, and robot environmental interaction—can be modelled by multi-dimensional networks consisting of lumped parameter positive-definite matrix-valued one-port elements connected by possibly nonlinear two-port Jacobians, and driven by independent effort and flow sources. The system is modularized by breaking up the complete closed network into discrete chunks, which can then be separately discretized and distributed among processors in a multiple-processor environment.

In this paper we will demonstrate how a redundant robot inverse kinematics routine can be implemented in a modular form in accordance with the SMART approach, allowing various additional modules to be inserted or removed, to customize the robot motion for a particular task.

2.0 Background

It is a well known result that the least norm solution to the equation,

$$\dot{x}_w = J_{w/j} \dot{x}_j \quad (1)$$

where x_w is a $m \times 1$ "world space" vector and x_j is a $n \times 1$ "joint space" vector, and $J_{w/j}$ is an $m \times n$ Jacobian matrix with $m > n$, is given by

$$\dot{x}_j = J_{w/j}^+ \dot{x}_w \quad (2)$$

where

$$J_{w/j}^+ = J_{w/j}^T (J_{w/j} J_{w/j}^T)^{-1} \quad (3)$$

is called the Moore-Penrose inverse of the Jacobian.

If the goal is to minimize a weighted joint velocity norm,

$$\|\dot{x}_j\|_W = \sqrt{\dot{x}_j^T W_j \dot{x}_j} \quad (4)$$

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

then the weighted least-norm solution [5] is given by

$$\dot{x}_j = W_j^{-1} J_{w/j}^T (J_{w/j} W_j^{-1} J_{w/j}^T)^{-1} \dot{x}_w \quad (5)$$

Finally, if the Jacobian moves through singularities such that its rank drops below m and Eq. 3 and 5 can't be computed, then a common recourse is to apply the more robust damped least squares solution [10],

$$\dot{x}_j = (J_{w/j}^T W_w J_{w/j} + \lambda)^{-1} J_{w/j}^T W_w \dot{x}_w, \quad (6)$$

with a world based weighting matrix W_w .

These equations have served as a basis for building more elaborate algorithms. A typical variation is to use the null space of the Psuedo-inverse to optimize some other parameter. For instance, joint limits and obstacles can be avoided by using a gradient projection algorithm [7],

$$\dot{x}_j = J_{w/j}^+ \dot{x}_j + (I - J_{w/j}^+ J_{w/j}) \nabla H \quad (7)$$

where $H(x_j)$ is a function to be maximized and ∇H is its gradient. Other researchers have augmented the Jacobian or have enhanced the damped least squares weighting to help shape the behavior [8][10].

We will be taking a different approach to redundant robot control in this paper. First, rather than focusing solely on the joint velocity mapping, we will focus on both the velocity and force mappings. Forces will be used to measure the ability of the system to follow a trajectory while satisfying multiple constraints. In many cases the redundancy is not adequate for satisfying excessive constraints and a large force may arise. Representing and conveying this force is important both for a bilateral tele-operator who can then adjust his motion accordingly, and for a trajectory generator which must then replan its path.

Second, rather than attempting to perturb joint velocities only in the null space, we will make perturbations directly in joint space. This will insure that the appropriate corrective action will always occur, although it may generate motion perturbations and corrective forces in world space.

Finally, the gradient projection based solutions have difficulty dealing with velocity based constraints, since this creates an algebraic loop in Equation 7. The approach presented allows velocity constraints to be readily implemented.

3.0 Damped least squares in network form

In this section, inverse kinematics algorithms will be recast in network form to demonstrate that many of the optimal approaches of the previous section can be derived directly from Kirchoff's laws.

Consider the circuit shown below, with a desired world space velocity source, $v_d_w = \dot{x}_d_w$, a non-square Jacobian (represented by a transformer), $J_{w/j}$, positive definite joint space damping (a resistor), Bl_j , and positive definite general impedance, Z_w .

FIGURE 2: Continuous Damped Least Squares Network

Applying Kirchoff's law to this network results in the equations below:

$$\begin{aligned} v_j &= Bl_j^{-1} J_{w/j}^T f_w \\ &= Bl_j^{-1} J_{w/j}^T Z_w (v_d_w - v_w) \\ &= Bl_j^{-1} J_{w/j}^T Z_w (v_d_w - J_{w/j} v_j) \\ &= (I_j + Bl_j^{-1} J_{w/j}^T Z_w J_{w/j})^{-1} Bl_j^{-1} J_{w/j}^T Z_w v_d_w \\ &= (Bl_j + J_{w/j}^T Z_w J_{w/j})^{-1} J_{w/j}^T Z_w v_d_w \end{aligned} \quad (8)$$

Because Bl_j and Z_w are both positive definite, this is also equivalent to,

$$v_j = Bl_j^{-1} J_{w/j}^T (Z_w^{-1} + J_{w/j} Bl_j^{-1} J_{w/j}^T)^{-1} v_d_w. \quad (9)$$

For an infinite impedance, Z_w , Equation 9, reduces to the weighted least norm (Eq. 5), with weighting $W = Bl_j$. For a finite impedance, Z_w , Equation 8 reduces to the damped least squares solution, (Eq. 6), with $W = Z_w$ and $\lambda = Bl_j$.

Thus as researchers have used the the least-squares equations as a basis for building redundant robot controllers, we will use the network shown in Figure 2 as the basis for building a modular network-based redundant robot controller.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

3.1 Including position feedback

The first step in making a bilateral solution that drives the world position error to zero if possible and generates a force error if not, is the inclusion of a spring, K_w , in the impedance, Z_w . Assigning,

$$Z_w = \frac{K_w}{s} + B2_w \quad (10)$$

results in the figure shown below (Fig 3):

FIGURE 3: Damped least square network with stiffness

with defining equations,

$$\begin{aligned} v_j &= B1_j^{-1} J_{w/j}^T Z_w (vd_w - J_{w/j} v_j) \\ &= B1_j^{-1} J_{w/j}^T (B2_w (vd_w - J_{w/j} v_j) + K_w (xd_w - \Lambda(x_j))) \\ &= (I_j + B1_j^{-1} J_{w/j}^T B2_w J_{w/j})^{-1} \\ &\quad * B1_j^{-1} (J_{w/j}^T Z_w vd_w + K_w (xd_w - \Lambda(x_j))) \end{aligned} \quad (11)$$

$$x_j(t) = \int_0^t v_j dt + x_j(0) \quad (12)$$

This will generate a force proportional to the tracking error across the spring which will either drive the system in the correct direction, or serve as a measure of the inability to track the commanded motion. Because this continuous equation is derived from a passive network, i.e. it consists of passive elements and independent sources, it is passive and always stable. Unfortunately, it is not directly implementable in its current form.

3.2 The discretized network.

The problem with these equations is that integrating Equation 12 on a digital computer induces a sample delay, and this leads to non-passivity and potential instability.

The resulting circuit with a nonpassive dependent source fc_K , replacing the spring is shown below.

FIGURE 4: DiscreteDamped Least Squares Network

The combined discrete equations for this system are given by:

$$fc_K(k) = K_w (xd_w(k) - \Lambda_{w/j}(x_j(k-1))) \quad (13)$$

$$\begin{aligned} v_j(k) &= (I_j + B1_j^{-1} J_{w/j}^T B2_w J_{w/j})^{-1} \\ &\quad * B1_j^{-1} (J_{w/j}^T Z_w vd_w + fc(k)) \end{aligned} \quad (14)$$

$$x_j(k) = x_j(k-1) + T v_j(k), \quad (15)$$

where T is the sampling time and k is the sample instant. In this system, the dependent effort source, fc_K , attempts to emulate a passive system, but fails at high frequencies due to the phase lag introduced by sampling. If the gains in the system are too high; i.e. if the stiffness, K_w , is too high, the damping, $B1_w$ is too low, or the Jacobian singular values get too high, the system will go unstable. In any case, the position loop can always be made stable by further reducing the sample period, T . This may be possible off-line, but is not a viable option for real-time operations.

4.0 Modularizing the system

It has recently been shown [4] that any system that can be represented as a multi-dimensional network, as shown in Fig. 2, can be arbitrarily modularized using scattering theory methods. The system is divided into any number of sub-networks, typically consisting of one-ports and two-ports. Input-output equations for each subnetwork are computed separately with outputs serving as inputs to its neighbors during the next compute cycle. By using scattering theory methods, passivity is maintained for arbi-

trary sample delays. For instance, a system that is functionally equivalent to Figure 3 can be broken into the modules shown below:

FIGURE 5: Modularizing the DLS system

The names, KB1, TRAJ, GEN_INV_KIN, and B1, refer respectively to a spring-damper terminator, trajectory generator, general inverse kinematics module, and damping terminator as used by the SMART system. The input-output equations for the B1, KB1 and TRAJ modules have been derived in other papers [1, 4]. Here we will derive the discrete equations for the GEN_INV_KIN module.

4.1 Finding the Scattering Operator Map

Scattering theory has been applied to numerous discretized tele robotic systems [3, 6, 9]. By sampling input and output waves rather than effort and flow variables, many of the problems that arise from sampling passive systems can be eliminated. The scattering operator mapping implements a change of variables from effort, f , and flow, v , variables to input and output waves (a and b respectively).

$$a = f + Z_0 v \quad (16)$$

$$b = f - Z_0 v \quad (17)$$

The term Z_0 , is the "characteristic impedance," a

arbitrary constant that the developer selects to help match the magnitudes of the effort and flow variables. The scattering operator, S , maps input waves to output waves for a given subnetwork:

$$b = S a \quad (18)$$

The steps for computing the scattering operator for a network were given in [4] and are summarized below:

- Separate memoryless non-linear and linear-time-invariant (LTI) components into different subnetworks.
- Drive the ports of each subnetwork with Thevenin equivalent effort sources in series with constant impedance (Z_0) (as shown below in Fig 6 for a one-port).
- Compute the velocities entering each port, v_i , as a function of the input waves, $v_i = (a_1, \dots, a_n, f_c)$.

Applying these techniques to the inverse-kinematic algorithm results in a two-stage operation. First the non-linear Jacobian subnetwork is given by [4],

$$\begin{bmatrix} b_w \\ b_j \end{bmatrix} = \begin{bmatrix} f_w - Z_0 v_w \\ f_j + Z_0 v_j \end{bmatrix} = \begin{bmatrix} D_w - I_w & D_w J_w/j \\ J_w^T D_w J_w/j - I_w^T & J_w^T D_w J_w/j \end{bmatrix} \begin{bmatrix} a_w \\ a_j \end{bmatrix} \quad (19)$$

where,

$$D_w = 2(I_w + J_w/j J_w^T)^{-1} \quad (20)$$

Next the linear-time-invariant scattering map is computed by driving the "previous" and "next" ports with Thevenin equivalent driving sources, as is shown below in Figure 7. Note that we've also added a small series damping term, $B1$, to the network, and distributed its damping equally on both ends of the subnetwork. This damping term provides better transient response in the system without requiring any additional computation during the main update cycle.

FIGURE 7: LTI Subnetwork with driving sources

Applying Kirchoff's laws to this network leads to the wave equations:

$$\Delta v = \frac{(a_p + a_n - 2fc)}{Z_0 + 2B2_w + B1/2} \quad (21)$$

$$\begin{aligned} \begin{bmatrix} b_p \\ b_n \end{bmatrix} &= \begin{bmatrix} a_p \\ a_n \end{bmatrix} - 2Z_0 \begin{bmatrix} v_p \\ -v_n \end{bmatrix} \\ &= \begin{bmatrix} \frac{B1}{2Z_0 + B1} & \frac{2Z_0}{2Z_0 + B1} \\ \frac{2Z_0}{2Z_0 + B1} & \frac{B1}{2Z_0 + B1} \end{bmatrix} \begin{bmatrix} a_p \\ a_n \end{bmatrix} - Z_0 \begin{bmatrix} \Delta v \\ \Delta v \end{bmatrix} \end{aligned} \quad (22)$$

where the driving position corrector term is a dependent effort source defined as before by,

$$fc_K = K_w(xd_w - \Lambda_{w/j}(x_j(k-1))) \quad (23)$$

These equations could be encapsulated in a module and directly combined with other SMART modules to shape the behavior of a redundant robot. Before doing this however, it is possible to further improve the performance of the inverse kinematics algorithm by decoupling the high-frequency component of the algorithm and compensating for its non-passivity. This is done in the next section.

5.0 Extending the passivity range

Once scattering theory has been applied, the only non-passive element of the control equations is the position feedback compensation. (Eq. 23), which is used to compensate for any drift in integrating the joint velocities. Only the high frequencies of Eq. 23 are non-passive due to the sampling phase lag. The low frequencies emulate a passive spring. Thus we would like to keep the low frequency behavior, and improve the passivity of the high frequency component of the system.

Consider the following frequency separation technique with low pass filter time constant, τ :

$$\begin{aligned} fc_K &= K_w(xd_w - \Lambda_{w/j}(x_j)) \\ &= \frac{(\tau s + 1)}{(\tau s + 1)} (K_w(xd_w - \Lambda_{w/j}(x_j))) \\ &= \frac{\tau K_w}{(\tau s + 1)} (vd_w - J_{w/j} v_j) \\ &\quad + \frac{K_w}{(\tau s + 1)} (xd_w - \Lambda_{w/j}(x_j)) \end{aligned}$$

By defining $B3_w = K_w \tau$, we get,

$$fc_K = \frac{B3_w K_w}{(B3_s + K_w)} (vd_w - J_{w/j} v_j) + fc_{Klow},$$

where fc_{Klow} is the low pass filtered position compensation. This equation is equivalent to the filter response of the circuit shown below (Fig 8):

FIGURE 8: Frequency separation of fc

Thus the high-frequency component of the dependent source can be replaced with passive network elements, and only the low-frequency component of the system is retained.

Re-computing the scattering operator for the LTI subnetwork results in the following circuit (Fig 9):

FIGURE 9: Enhanced LTI subnetwork with driving sources attached

Because of the balance in the circuit, it can be simplified to the circuit shown below, where a_s is the average applied effort source:

The equations for this network are given below:

$$\Delta v = (Z_s(B_3s + K_w) + B_3K_w)^{-1} * (B_3s + K_w)(a_s - fc_{low}) \quad (24)$$

where

$$Z_s = Z_0/2 + B1_w/4 + B2_w, \quad (25)$$

$$a_s = \frac{a_p + a_n}{2} \quad (26)$$

and

$$fc_{low} = \frac{K_w}{(\tau s + 1)}(xd_w - \Lambda_{w/j}(x_j)). \quad (27)$$

As before we get the wave equations:

$$\begin{aligned} \begin{bmatrix} b_p \\ b_n \end{bmatrix} &= \begin{bmatrix} a_p \\ a_n \end{bmatrix} - 2Z_0 \begin{bmatrix} v_p \\ -v_n \end{bmatrix} \\ &= \begin{bmatrix} \frac{B1}{2Z_0 + B1} & \frac{2Z_0}{2Z_0 + B1} \\ \frac{2Z_0}{2Z_0 + B1} & \frac{B1}{2Z_0 + B1} \end{bmatrix} \begin{bmatrix} a_p \\ a_n \end{bmatrix} - Z_0 \begin{bmatrix} \Delta v \\ \Delta v \end{bmatrix} \end{aligned} \quad (28)$$

The joint position estimate, x_j , is computed by integrating the joint velocity,

$$v_j = (b_j - a_j)/2*Z_0. \quad (29)$$

Equations 15, 19, 20, 24-27, represent the input-output equations for a GEN_INV_KIN module¹. By collecting these equations as a module and then updating this module in a system of similarly generated modules, we can customize the behavior of the redundant robot controller to suit the task at hand.

1. Applying Tustin's method to Eq. 24 will result in the discrete equation needed for final implementation.

FIGURE 11: Redundant Robot Controller with Sensor Avoidance, Nonlinear velocity avoidance, and virtual obstacle avoidance

6.0 Shaping the motion using additional modules

Using SMART it is possible to build robot control systems by combining network based modules to create behaviors. A SMART Editor provides a graphical interface for combining modules on a template, testing for valid connections, and automatically generating code for the resulting system. The GEN_INV_KIN module is the result of the work presented in this paper. This module takes as arguments, the Denavit-Hartenburg (DH) parameters for the robot to be modeled, the stiffness and damping gains, and the passivity roll-off time constant, τ . In this section we describe how to satisfy multiple competing objectives for a redundant control system by replacing the B1 damper one-port with various other modules.

The SMART icons for implementing the modular damped least squares algorithm corresponding to Figure 5 are shown below:

FIGURE 10: SMART modules for simple damped least squares

6.1 Joint travel constraints

The first additional objective for redundant kinematics is the avoidance of joint limits. To avoid joint limits, an energy penalty boundary must be implemented whenever the robot approaches its joint limits. This energy penalty function must transition smoothly from free space motion to constrained motion.

As has been shown [1], this type of energy penalty can be implemented by using a nonlinear switching function Jacobian connected to a stiff spring. In free space the switch is off and no flow is directed into the spring-damper system. As the limits are approached, the switch becomes positive, flow is directed into the spring damper, and a large penalty force is generated.

6.2 Non-linear velocity constraints

One of the primary problems of relying on linear viscous damping, such as the B1 module above, to shape the velocity is that there is no guarantee that velocities will be kept within hard limits. Likewise, velocities that are well within hard slew-rate limits are unfairly penalized.

The solution is to provide a nonlinear damping term, which remains negligible while the robot is within its velocity limits and ramps up sharply as the robot approaches hard velocity limits. This type of nonlinear damping is easily implemented within the SMART architecture, as is described in [4]. The SMART CLAMP module implements both a joint limit constraint and a nonlinear velocity constraint.

6.3 Avoiding obstacles

In many cases redundancy was designed into the system to enable obstacle avoidance while achieving positioning objectives. Obstacle avoidance modules can be developed for the SMART architecture which model obstacles and generate virtual forces whenever an object is approached. One such module is the OBSTACLE module, which was described in [3]. The OBSTACLE module represents the world as a collection of convex primitives. Then during real-time, the distance between each pair of convex objects is computed and used to generate virtual springs at points of nearest contact.

6.4 Sensor feedback

One of the main justifications for pursuing a real-time modular algorithm for redundant robot control is the ability to integrate sensors in an easy and intuitive fashion. Whole-arm proximity sensors, such as Sandia's WHAP sensor, can be integrated into the redundant controller using an algorithm nearly identical to the obstacle avoidance algorithm. These sensors will then naturally shape the motion of the arm to avoid contact with any obstacles. As with the other constraint perturbations shown here, the perturbation occurs in joint space rather than in the null-space of the world motion, and thus obstacle avoidance is assured, although the motion may be perturbed or halted due to insufficient redundancy.

7.0 Conclusions

In this paper we demonstrated how a damped least squares algorithm can be cast as a passive control law. Using this formulation we generated a scattering operator map for an inverse kinematics module. Analyzing its sources of non-passivity lead to an improvement in the basic algorithm. These scattering equations were then used as a basis for building a modular redundant robot

kinematics system, whereby additional constraint modules could be applied separately or in combination to achieve different objectives as shown in Figure 11.

8.0 Acknowledgements

This work was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract number CE-AC04-94-AL85000.

9.0 References

- [1] R. J. Anderson, "SMART: A Modular Control Architecture for Telerobotics", *IEEE Robotics and Automation Society Magazine*, pp 10-18, Sept. 95.
- [2] R. J. Anderson and M. W. Spong, "Bilateral Control of Teleoperators with Time Delay," *IEEE Transactions on Automatic Control*, vol. 34, pp.494-501, Sept. '89.
- [3] R. J. Anderson, "Teleoperation with Virtual Force Feedback", *Experimental Robotics III*, Springer-Verlag. Lecture Notes in Control and Information Sciences 200, 1993, pp. 366-375.
- [4] R. J. Anderson, "A Modular Approach to Robot Control", '96 *IEEE International Conference on Robotics and Automation*, St. Paul, Minn, pp. 699-705.
- [5] T. F. Chan and R. V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Manipulators", *Proceedings of the '93 IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA. May 1993 pp 395-402.
- [6] C. A. Lawn and B. Hannaford, "Performance Testing of Passive Communication and Control in Teleoperation with Time Delay", *Proceedings of the '93 IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA. May 1993 pp 775-781.
- [7] A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Trans. on Systems Man & Cybernetics*, SMC-7, No. 12, 1977.
- [8] A. A. Maciejewski, and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research*, Vol. 4, No. 3, pp. 109-117, 1985.
- [9] G. Niemeyer, J. J. Slotine, "Stable Adaptive Teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, pp.152-162, Jan. 1991.
- [10] Seraji, H. and Colbaugh, R., "Improved Configuration Control of Redundant Robots", *Journal of Robotic Systems*, Vol 7, No. 6, 1990.

M98001676

Report Number (14) SAND--97-3097 C.
CONF-970469--

Publ. Date (11) 199712

Sponsor Code (18) DOE/EM, XF

JC Category (19) UC-2000, DOE/ER

DOE