AAMRL-SR-90-517

OPTIMIZED FORWARD KINEMATICS FOR THE MBA EXOSKELETON AND PARTITIONED KINEMATICS FOR THE MERLIN ROBOT

Michael S. Branicky

ARTIFICIAL INTELLIGENCE LABORATORY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02193

Steven J. Remis

BIODYNAMICS AND BIOENGINEERING DIVISION
HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-6573

NOVEMBER 1990

FINAL REPORT FOR THE PERIOD JANUARY 1990 THROUGH DECEMBER 1990

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
HUMAN SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-6573

NOTICES

TECHNICAL REVIEW AND APPROVAL

AAMRL-SR-90-517

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

PETER A. LURKER, Lt Col, USAF BSC
Acting Director
Biodynamics and Bioengineering Division
Harry G. Armatrong Aerospace Medical Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
November 1990

3. REPORT TYPE AND DATES COVERED
Special Report, June 1990 - August 1990

4. TITLE AND SUBTITLE
Optimized Forward Kinematics for the MBA Exoskeleton and Partitioned Kinematics for the Merlin Robot

5. FUNDING NUMBERS
PE - 6220F
PR - 7231
TA - 38
WU - 03

6. AUTHOR(S)
Michael S. Branicky
Steven J. Remis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
Cambridge, MA 02193

AAMRL/BBA
Wright-Patterson Air Force Base, OH 45433-6573

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Aerospace Medical Research Laboratory
Human Systems Division
Air Force Systems Command
Wright-Patterson AFB, OH 45433-6573

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AAMRL-SR-90-517

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words) This paper describes a systems approach to improving the performance of a telerobotic testbed. The testbed is the Force-Reflecting Interfaces to Telemanipulators Testing System (FITTS), at the Harry G. Armstrong Aerospace Medical Research Laboratory. First, the testbed hardware is described, along with an overview of the system's communications paths. Next, the previously-determined forward kinematics for the MBA exoskeleton are outlined. Then the optimization of these kinematic equations is given. The paper also details efficient forward and inverse kinematic solutions for the Merlin industrial robots, using the method of wrist partitioning. The utility of these solutions <u>in toto</u> is that the interfacing of the two systems, given sufficient communications bandwidth between the MBA exoskeleton and the control computer, can now achieve the 4 ms compute time attainable by the Merlin hardware. This optimization puts the FITTS hardware at a milestone stage of completion, as the system is now capable of operating at its peak as a unilateral telerobotic testbed. Finally, future steps to extend FITTS for force reflection research are specified.

14. SUBJECT TERMS
Teleoperation; Robotics; Remote Control; Human Factors; Force Reflection

15. NUMBER OF PAGES
17

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Contents

1 Introduction

At the Harry G. Armstrong Aerospace Medical Research Laboratory, the Human Sensory Feedback team of the Biological Acoustics Branch is investigating technologies for intuitive human control of, and sensory feedback from, teleoperated robotic devices. To this end the Force-Reflecting Interfaces to Telemanipulators Testing System (FITTS) has been established [5]. This system is currently undergoing refinement before its use as a dedicated testbed for evaluating man-machine interfaces to teleoperators.

The goal of the FITTS system is to investigate human interfaces to telerobotic systems. The first interface to be evaluated by FITTS is the MBA Exoskeleton, shown in Figure 1. This seven degree-of-freedom (DOF), unilateral device is being used to establish performance baselines for worst-case telerobotic performance. The optimal baselines, of course, will be human hands-on task completion, although it is recognized that synergism between the operator and the robots may eventually surpass this now-optimal baseline.

Figure 1: Seven DOF MBA Exoskeleton.

The slave robots used in FITTS are the 6 DOF American Robot Corporation Merlin robots shown in Figure 2. The testbed contains a left-right pair of Merlins, although only the left-arm Merlin is currently operating at high speed. Each Merlin has a 50 *lb* payload, and can move the end effector at 5 $\frac{ft}{sec}$ throughout the workspace. The left-arm Merlin has been equipped with the optional High Speed Host Interface (HSHI), which allows the control computer to update desired position or velocity commands at 250 *Hz* via shared-memory window.

In addition to the MBA Exoskeleton and the Merlin robots, FITTS contains a peg-into-hole taskboard developed jointly by the Naval Oceans Systems Center - Hawaii, and AAMRL [8]. This taskboard is particularly well-suited to measuring task performance using a Fitts' Law paradigm [4]. The peg-into-hole tasks available have indices of difficulty (ID) ranging from 6 to 12, with the tasks of higher ID resulting from closer tolerances between the pegs and the holes or larger amplitudes of movement. Two preliminary studies have been completed at AAMRL using this subsystem of FITTS [7] [6] to measure the task performance degradation caused by the MBA Exoskeleton, exclusive of any slave robotic systems.

Figure 2: Six DOF Merlin Industrial Robots.

The interactions between the components of the FITTS testbed are shown in Figure 3. Joint angle data flows serially from the MBA Exoskeleton to the Compaq computer, which then calculates the endpoint position of the exoskeleton. The Compaq next computes the inverse kinematics of the Merlin robot, and feeds the joint position information to the robot's controller via shared RAM. The Merlin controller internally generates a trajectory to accomplish this motion.

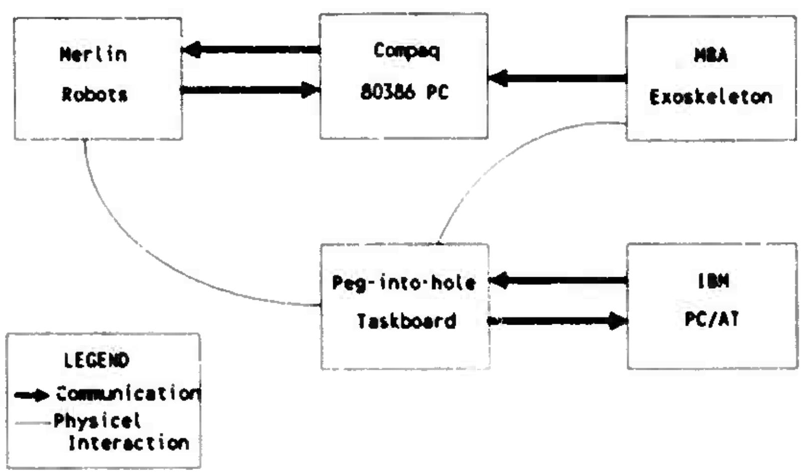

Figure 3: FITTS subsystems and their interactions.

This paper describes recent work to improve the performance of the FITTS system, preparing it first f-- unilateral, then for bilateral teleoperation. Specifically, the steps taken to optimize the kinematics code for the MBA Exoskeleton are presented. Next, the Merlin kinematics solutions, using the method of wrist partitioning, are given. It is shown that these new algorithms reduce the computational time by 75 per cent from the previous methods. An analysis of the FITTS system's communication paths is then completed to determine the improvements in communication rates needed to optimize performance for unilateral teleoperation. Finally, the process is given which will allow the FITTS testbed to evaluate bilateral man-machine interfaces, such as force-reflecting exoskeletons.

2 MBA Exoskeleton Forward Kinematics

link	θ_i	α_i	a_i	d_i
1	$-90°$	0	$L_1 - d_1 \sin 9°$	$L_2 \cos 9° - L_4$
2	$90°$	$9°$	L_3	0
3	θ_3	$-90°$	0	0
4	θ_4	$90°$	0	$L_6 - L_5$
5	θ_5	$-90°$	0	L_{12}
6	0	0	0	$L_7 + L_8 + L_9 - L_{10} + L_{11}$
7	θ_7	$90°$	0	0
8	θ_8	$-90°$	0	L_{13}
9	θ_9	0	$L_{19} - L_{20}$	$-L_{14} + L_{15} + L_{16}$
10	$90°$	$-90°$	0	0
11	$\gamma_{11} - \tan^{-1}(L_{17}/L_{18})$	$90°$	0	0
12	$-90°$	0	0	$\sqrt{L_{17}^2 + L_{18}^2}$

Table 1: D-H paramaters for the MBA exoskeleton, left arm.

We begin with the Denavit-Hartenburg (D-H) parameters for the MBA exoskeleton, as found by Gary Merrill of Systems Research Laboratories, Inc. He assigned twelve coordinate frames for each arm, shown in Figure 4 and summarized in Table 1. See [7] for more details.

In Table 1, θ_i is the mapping from x_{i-1} to x_i about the z_{i-1} axis, α_i is the mapping from z_{i-1} to z_i about the x_i axis, a_i is the distance between the z_{i-1} and z_i axes along x_i, d_i is the distance between the x_{i-1} and x_i axes along z_{i-1}, L_i is the length of link i, γ_{11} is the measured joint angle in frame 11, and the addend $\tan^{-1}(\cdot)$ term is a fixed offset for this frame.

The forward kinematics (of a single arm) of the MBA exoskeleton had been calculated on-line by iteratively loading and then multiplying the twelve transformation matrices corresponding to each of these frames. The C code that performed this operation required 768 multiplies, 576 additions, 48 trigonometric function evaluations, over 400 variable assignments, 36 nested *for* loops, and 12 function calls (with variable declarations/initializations). Each iteration of this code took 11.2 ms of compute time on a 33 MHz Compaq 80386 personal computer with an 80387 math coprocessor.

In order to improve controller bandwidth — and performance — it was deemed necessary to

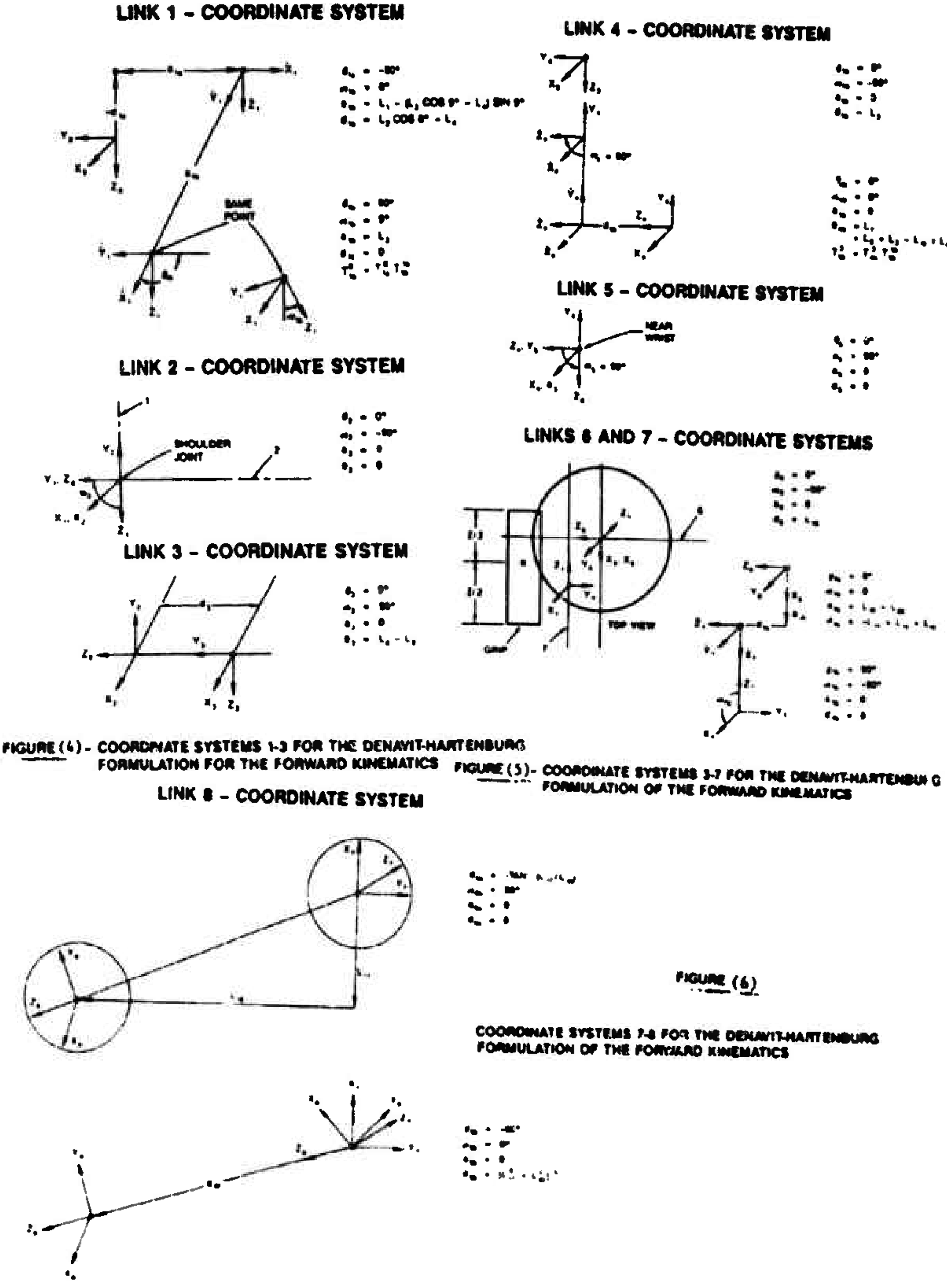

Figure 4: From [7]. Denavit-Hartenburg parameters for the MBA exoskeleton.

explicitly formulate the forward kinematics of the MBA exoskeleton by hand, and then calculate only these simplified kinematics on-line. The twelve transformation matrices were obtained from the D-H parameters and finally reduced to three. The calculations are shown below.

The general form of the transformation matrix is as given in [1]:

$$T_i = \begin{pmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where cx and sx are shorthand for $\cos x$ and $\sin x$, respectively.

Substituting each of the D-H parameters into the appropriate T_i, the twelve transformation matrices are easily found to be:

$$T_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -a_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

$$T_2 = \begin{pmatrix} 0 & -c\alpha_2 & s\alpha_2 & 0 \\ 1 & 0 & 0 & a_2 \\ 0 & s\alpha_2 & c\alpha_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

$$T_3 = \begin{pmatrix} c\theta_3 & 0 & -s\theta_3 & 0 \\ s\theta_3 & 0 & c\theta_3 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$T_4 = \begin{pmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$T_5 = \begin{pmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$T_6 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

$$T_7 = \begin{pmatrix} c\theta_7 & 0 & s\theta_7 & 0 \\ s\theta_7 & 0 & -c\theta_7 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

$$T_8 = \begin{pmatrix} c\theta_8 & 0 & -s\theta_8 & 0 \\ s\theta_8 & 0 & c\theta_8 & 0 \\ 0 & -1 & 0 & d_8 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

$$T_9 = \begin{pmatrix} c\theta_9 & -s\vartheta_9 & 0 & c\theta_9 a_9 \\ s\theta_9 & c\theta_9 & 0 & s\theta_9 a_9 \\ 0 & 0 & 1 & d_9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

$$T_{10} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

$$T_{11} = \begin{pmatrix} c\theta_{11} & 0 & s\theta_{11} & 0 \\ s\theta_{11} & 0 & -c\theta_{11} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

$$T_{12} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_{12} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

Evidently these matrices are both sparse and contain many unitary elements, hence, it is computationally advantageous to multiply them out by hand to exploit these features. In the following, we reduce from twelve to six transformation matrices, using the notation

$$T_{i-j} = T_i T_j \tag{14}$$

$$T_{1-2} = \begin{pmatrix} 1 & 0 & 0 & a_2 \\ 0 & c\alpha_2 & -s\alpha_2 & -a_1 \\ 0 & s\alpha_2 & c\alpha_2 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{15}$$

$$T_{3-4} = \begin{pmatrix} c\theta_3 c\theta_4 & -s\theta_3 & c\theta_3 s\theta_4 & -s\theta_3 d_4 \\ s\theta_3 c\theta_4 & c\theta_3 & s\theta_3 s\theta_4 & c\theta_3 d_4 \\ -s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{16}$$

$$T_{5-6} = \begin{pmatrix} c\theta_5 & 0 & -s\theta_5 & -s\theta_5 d_6 \\ s\theta_5 & 0 & c\theta_5 & c\theta_5 d_6 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

$$T_{7-8} = \begin{pmatrix} c\theta_7 c\theta_8 & -s\theta_7 & -c\theta_7 s\theta_8 & s\theta_7 d_8 \\ s\theta_7 c\theta_8 & c\theta_7 & -s\theta_7 s\theta_8 & -c\theta_7 d_8 \\ s\theta_8 & 0 & c\theta_8 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{18}$$

$$T_{9-10} = \begin{pmatrix} -s\theta_9 & 0 & -c\theta_9 & c\theta_9 a_9 \\ c\theta_9 & 0 & -s\theta_9 & s\theta_9 a_9 \\ 0 & -1 & 0 & d_9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{19}$$

$$T_{11-12} = \begin{pmatrix} 0 & c\theta_{11} & s\theta_{11} & s\theta_{11} d_{12} \\ 0 & s\theta_{11} & -c\theta_{11} & -c\theta_{11} d_{12} \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{20}$$

Finally, these six are reduced to only three transformation matrices. These three matrices are shown below, using the notation

$$T_{i-l} = T_{i-j} T_{k-l} \tag{21}$$

and we adopt the additional convention of using c_i and s_i to represent $\cos\theta_i$ and $\sin\theta_i$, respectively:

$$T_{1-4} = \begin{pmatrix} c_3 c_4 & -s_3 & c_3 s_4 & a_2 - d_4 s_3 \\ c\alpha_2 s_3 c_4 + s\alpha_2 s_4 & c\alpha_2 c_3 & c\alpha_2 s_3 s_4 - s\alpha_2 c_4 & d_4 c\alpha_2 c_3 - a_1 \\ s\alpha_2 s_3 c_4 - c\alpha_2 s_4 & s\alpha_2 c_3 & s\alpha_2 s_3 s_4 + c\alpha_2 c_4 & d_4 s\alpha_2 c_3 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{22}$$

$$T_{5-8} = \begin{pmatrix} c_5 c_7 c_8 - s_5 s_8 & -c_5 s_7 & -c_5 c_7 s_8 - s_5 c_8 & d_8 c_5 s_7 - d_6 s_5 \\ s_5 c_7 c_8 + c_5 s_8 & -s_5 s_7 & -s_5 c_7 s_8 + c_5 c_8 & d_8 s_5 s_7 + d_6 c_5 \\ -s_7 c_8 & -c_7 & s_7 s_8 & d_8 c_7 + d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{23}$$

$$T_{9-12} = \begin{pmatrix} c_9 & -s_9 c_{11} & -s_9 s_{11} & -d_{12} s_9 s_{11} + a_9 c_9 \\ s_9 & c_9 c_{11} & c_9 s_{11} & d_{12} c_9 s_{11} + a_9 s_9 \\ 0 & -s_{11} & c_{11} & d_{12} c_{11} + d_9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{24}$$

These three matrices were coded in the C programming language on the Compaq computer. The remaining two matrix multiplications,

$$T_{1-8} = T_{1-4} T_{5-8} \tag{25}$$

and

$$T_{1-12} = T_{1-8} T_{9-12} \tag{26}$$

were explicitly written in terms of the matrix elements, with multiplications by 0, 1, and -1 removed by hand. This technique, though sub-optimal, is significantly more efficient than the previous method of loading and multiplying each matrix using nested *for* loops. The resulting code has only 112 multiplications, 75 additions, 14 trigonometric function evaluations, about 75 variable assignments, no *for* loops, and no function calls; its cycle time is 0.7 ms on the Compaq (versus 11.2 ms for the previous method).

Further optimisation can be achieved in two ways, if desired. The first procedure is to explicitly combine the three matrices T_{1-4}, T_{5-8}, and T_{9-12} off-line. This results in obtaining the final transformation matrix, T_{1-12}, without requiring any interim matrix computations. The computational advantage of this method comes from reducing the number of interim variable assignments and register manipulations. Unfortunately, the complexity of these matrices suggests the use of a symbolic math program to perform these calculations.

The second optimisation procedure is to search the final matrices (or matrix if the previous optimisation method is also used) for all common factors. Each common factor should be computed once, and the resultant of this calculation used at each occurrence of the factor. This process will reduce the number of multiplications, trigonometric function evaluations, and register manipulations. Although this process can often save a significant amount of time, in our case the number of common factors appears to be small; thus the savings would not be very great.

3 Merlin Robot Kinematics

Since the Merlin robot has a spherical wrist, its kinematic formulation may be partitioned into position (first three DOF) and orientation (last three DOF) [9]. The forward and inverse equations for position of the Merlin robot were found using the geometric approach [3]; those for orientation are more easily found because the wrist is spherical [10]. The complete solution is very similar to that of a PUMA 560 robot.

The following sections relate the forward and inverse solutions for the Merlin without describing the process in detail. We again adopt cx and sx as shorthand for $\cos x$ and $\sin x$, respectively, and the additional convention of using c_i and s_i to represent $\cos\theta_i$ and $\sin\theta_i$, respectively. In the sections which follow, we use the general form of a transformation matrix

$$T = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{27}$$

with $T[i][j]$ representing the ijth entry of this matrix, and $\theta_{[1..6]}$ representing the array of joint angles for the Merlin, beginning with the base and working towards the final hand roll.

3.1 Position

Given $\theta_{[1..6]}$, the forward equations to the wrist are as follows:

$$r_{\parallel} = l_2 c\theta_2 + l_3 c\theta_3 \tag{28}$$

$$p_x = -d_2 s\theta_1 + r_{\parallel} c\theta_1 \tag{29}$$

$$p_y = d_2 c\theta_1 + r_{\parallel} s\theta_1 \tag{30}$$

$$p_z = l_2 s\theta_2 + l_3 s\theta_3 \tag{31}$$

where $d_2 = 12.0$, $l_2 = 17.38$, $l_3 = 17.24$ (units are inches), θ_3 is measured absolutely, and $r_{||}$ is the vector from the base coordinate frame to the first wrist coordinate frame.

The inverse equations for wrist position, given p_x, p_y, and p_z, are the following:

$$r_{||} = \sqrt{p_x^2 + p_y^2 - d_2^2} \quad (32)$$

$$c = \frac{l_3^2 - l_2^2 + r_{||}^2 + p_z^2}{2l_3} \quad (33)$$

$$\theta_1 = \text{atan2}(r_{||}, d_2) - \text{atan2}(p_x, p_y) \quad (34)$$

$$\theta_3 = \text{atan2}(p_z, r_{||}) + \text{atan2}(r_{||}^2 + p_z^2 - c^2, c) \quad (35)$$

$$\theta_2 = \text{atan2}(p_z - l_3 s\theta_3, r_{||} - l_3 c\theta_3) \quad (36)$$

where θ_3 is again given as an absolute angle.

In implementing the above solutions, it should be noted that the Merlin controller gives the signs of θ_2 and θ_3 opposite to convention, and hence this must be accounted for by negating these two angles before (forward) and after (inverse) the above calculations are performed.

3.2 Orientation

Before listing the equations for the orientation of the Merlin wrist, we first define the following common factors in these equations:

$$c_{23} = \cos(\theta_3 - \pi/2) \quad (37)$$

$$s_{23} = \sin(\theta_3 - \pi/2) \quad (38)$$

$$k_1 = c_4 c_5 c_6 - s_4 s_6 \quad (39)$$

$$k_2 = s_4 c_5 c_6 + c_4 s_6 \quad (40)$$

$$k_3 = s_4 c_5 s_6 - c_4 c_6 \quad (41)$$

$$k_4 = c_{23} k_1 - s_{23} s_5 c_6 \quad (42)$$

$$k_5 = c_4 c_5 s_6 + s_4 c_6 \quad (43)$$

$$k_6 = c_{23} c_4 s_5 + s_{23} c_5 \quad (44)$$

With these common factors defined, the forward kinematics equations for wrist orientation simplify to the following:

$$T[0][0] = c_1 k_4 + s_1 k_2 \quad (45)$$

$$T[1][0] = s_1 k_4 - c_1 k_2 \quad (46)$$

$$T[2][0] = -s_{23} k_1 - c_{23} s_5 c_6 \quad (47)$$

$$T[0][1] = c_1(-c_{23} k_5 + s_{23} s_5 s_6) - s_1 k_3 \quad (48)$$

$$T[1][1] = s_1(-c_{23} k_5 + s_{23} s_5 s_6) + c_1 k_3 \quad (49)$$

$$T[2][1] = s_{23} k_5 + c_{23} s_5 s_6 \quad (50)$$

$$T[0][2] = -c_1 k_6 - s_1 s_4 s_5 \quad (51)$$

$$T[1][2] = -s_1 k_6 + c_1 s_4 s_5 \quad (52)$$

$$T[2][2] = s_{23} c_4 s_5 - c_{23} c_5 \quad (53)$$

The inverse equations for the wrist are given below. We will show each angle's solution in turn after defining some temporary constants for each. First θ_1 and θ_3 must be found with the inverse solution for wrist position. To solve for wrist orientation we also need c_{23} and s_{23} (where the subscript $_{23}$ represents $\theta_2 + \theta_3$). Note that

$$c_{23} = \cos(\theta_3 - \pi/2) \quad (54)$$

$$s_{23} = \sin(\theta_3 - \pi/2) \quad (55)$$

because of the absolute measurement of θ_3, and because θ_2 has an initial alignment of $\pi/2$ from the base frame.

For θ_4, we have

$$t_1 = T[2][2]s_{23} - T[0][2]c_1c_{23} - T[1][2]s_1c_{23} \quad (56)$$

$$t_2 = T[1][2]c_1 - T[0][2]s_1 \quad (57)$$

$$\theta_4 = \text{atan2}(t_2, t_1) \quad (58)$$

For θ_5, we have

$$t_1 = -T[0][2]c_1s_{23} - T[1][2]s_1s_{23} - T[2][2]c_{23} \quad (59)$$

$$t_2 = T[2][2]s_{23}c_4 - T[0][2](c_1c_{23}c_4 + s_1s_4) - T[1][2](s_1c_{23}c_4 - c_1s_4) \quad (60)$$

$$\theta_5 = \text{atan2}(t_2, t_1) \quad (61)$$

For θ_6, we have

$$t_1 = T[0][0](c_5(c_1c_{23}c_4 + s_1s_4) - c_1s_{23}s_5) \quad (62)$$

$$+T[1][0](c_5(s_1c_{23}c_4 - c_1s_4) - s_1s_{23}s_5) - T[2][0](s_{23}c_4c_5 + c_{23}s_5) \quad (63)$$

$$t_2 = T[2][0]s_{23}s_4 - T[0][0](c_1c_{23}s_4 - s_1c_4) - T[1][0](s_1c_{23}s_4 + c_1c_4) \quad (64)$$

$$\theta_6 = \text{atan2}(t_2, t_1) \quad (65)$$

To move the robot wrist to $\theta_{[1\ldots6]}$, realize that the Merlin distinguishes between 1°, 361°, and -359°, but the atan2 function only returns values in the range of ±180°. To account for this discrepancy, a section of C code was added to the inverse solution immediately following the computation of θ_4. This code keeps track of θ_4 as the wrist rolls about its axis, and adds ±180° as often as necessary to prevent a "flip" of 180° when moving to the new position. The code also tracks θ_6, and uses the same technique to prevent this joint from "flipping."

This completes the forward and inverse kinematics solutions for the Merlin robot. These equations were hand-optimized and coded in the C computer programming language. The full (six DOF) inverse solution can be calculated in less than 1 ms on the Compaq computer.

4 Conclusions

With the changes described in this paper, calculating the forward kinematics of the MBA exoskeleton and the inverse kinematics of the Merlin, plus transferring this data from the Compaq to the Merlin takes less than 3 ms (the minimum update time of the Merlin robots is 4 ms). However, the

FITTS hardware cannot yet operate above this 250 Hz goal due to the 32.8 *kbaud* serial (RS-422) link between the MBA exoskeleton and the Compaq computer.

It is evident we will achieve maximum throughput for this system by increasing the bandwidth of MBA-Compaq communications. Using the current communications protocol, transferring joint angle data from the exoskeleton to the Compaq takes, at best, 8 ms [2] for one arm of the exoskeleton. Some of this delay is due to the Compaq computer polling the port too often during data transfer, essentially preventing data flow from the exoskeleton to the computer during these polls. This problem has been corrected by making the data transfer interrupt-driven.

The second issue of concern is the transfer rate. To send joint angle data for both arms of the exoskeleton in less than 1 ms, the exoskeleton must send 32 8-bit bytes (14 DOF plus grippers), in addition to appropriate start/stop bits and handshaking signals. A quick calculation shows 320 *kbaud* is necessary for the RS-422 link to support this throughput requirement. Another approach is to send the data in parallel, using a 32 *kbaud* transfer rate. Because of the desire to be able to locate the exoskeleton some distance from the control computer, the differentially-driven serial port is the better solution, but the baud rate must be increased above 320 *kbaud*. Since 1 *Mbaud* is often achievable with RS-422 ports, the plan is to implement 1 *Mbaud* on this system.

Once this communications upgrade is completed, the FITTS hardware will be able to perform at its peak for unilateral teleoperation. However, several issues need to be addressed before the system is ready for bilateral teleoperation. One issue is deciding which controller should compute the forward kinematics of the new interfaces. Another is determining which computer should calculate feedback joint torques for force reflection, using the well-known relationship

$$\tau = \mathbf{J}^T \mathbf{F} \tag{66}$$

where τ is the [6 x 1] vector of joint torques, $\mathbf{J}^T$ is the transpose of the Jacobian of the bilateral interface, and $\mathbf{F}$ is the [6 x 1] vector of force/torque information $[F_x, F_y, F_z, T_x, T_y, T_z]$ from the Merlin's end effector.

The issues listed above, and others, remain to be answered before FITTS becomes the modular testbed of bilateral human interfaces, as it is designed. The basic plan is to use the Compaq computer to control the Merlins, and as the center for data exchange between masters and slaves. This implies that the bilateral devices must compute their own forward kinematics, outputting only the final transformation matrix to the Compaq. These interfaces must also be prepared to accept 6 DOF force/torque information from the Compaq, and compute the appropriate joint torques for force feedback. A determining factor in this plan is to keep the Merlin update rate above 250 Hz.

This paper showed that the FITTS testbed has been established, and is being refined at AAMRL. This testbed will soon be able to operate for unilateral teleoperation, comparing various human interface devices for their ease of operation. A preliminary plan has also been presented which will allow FITTS to become a modular testbed for evaluating the various bilateral human-interfaces currently used as input devices to teleoperators.

References

[1] Asada, Haruhiko and J. E. Slotine, *Robot Analysis and Control*, John Wiley & Sons, New York, 1986.

[2] Branicky, M. S., "Control and Modeling for Improved Human Sensory Feedback," *1990 LGFP/DoD Fellowship Summer Research Program Final Report*, Universal Energy Systems, Inc., 1990.

[3] Craig, John J., *Introduction to Robotics*, Addison-Wesley Publishing Company, Reading, MA, 1986.

[4] Fitts, P. M., "The Information Capacity of the Human Motor System in Controlling the Amplitude of a Movement," *Journal of Experimental Psychology*, 46: pp. 381-391, 1954.

[5] Remis, Steven J. and D. K. Nelson, "Force-reflecting Interfaces to Telemanipulators Testing System (FITTS) Interim User's Guide," *AAMRL Special Report*, AAMRL-SR-90-509, 1990.

[6] Remis, Steven J. and D. W. Repperger "Quantifying Performance Degradation Due to the Man-Machine Interface of Telemanipulators," *AAMRL Special Report*, AAMRL-SR-90-510, 1990.

[7] Repperger, Dan W., S. J. Remis, and G. Merrill, "Performance Measures of Teleoperation Using an Exoskeleton Device," *Proc. of the 1990 IEEE Int. Conf. on Robotics and Automation*, pp. 552–557, 1990.

[8] Spain, H., *Peg-in-Hole Taskboard Documentation*, Version 1.0. NOSC-HI, Code 531, March 1989.

[9] Stanisic, M. M., personal conversations with S.J. Remis. Summer, 1989.

[10] W. A. Wolovich. *Robotics: Basic Analysis and Design*. Holt, Rinehart and Winston, 1987.

☆U.S. Government Printing Office 1993 — 750-061/60281