SOME EXPERIMENTAL RESULTS ON MODEL-BASED CONTROL SCHEMES[1]

Pradeep K. Khosla[2]

1. Introduction

The objective of this paper is to present an overview of our research on the analysis, synthesis, real-time implementation and performance evaluation of model-based manipulator control schemes. The model-based control schemes synthesize strategies that include a dynamical model of the manipulator in the feedback loop. Further, depending on the way that the dynamical model is incorporated, it is possible to create different types of control laws. For example, the computed-torque method [24] utilizes the model in the feedback loop in order to both decouple and linearize the system. Independent joint controllers are then designed to achieve accurate trajectory tracking and to reject unknown external disturbances. The feedforward control scheme is another model-based control method and utilizes the dynamics model in the feedforward path. The idea is that the feedforward torques/forces provide gross signals and independent joint controllers provide the correcting control signals to reject disturbances that are unknown.

The manipulator trajectory tracking problem has been studied extensively and many control formulations have been presented [28, 22, 21, 3, 4, 7]. However, the real-time implementation of model-based control schemes, with high control sampling rates, had never been demonstrated on actual manipulators. The main reasons for this were: firstly, even though the recent Newton-Euler recursive formulation [23] of the dynamics reduced the amount of computation, its real-time use was not been practical for the commercially available microprocessors; secondly, it was difficult to obtain the dynamic parameters of the manipulator because research in this area had been lacking; and finally, it is extremely difficult to model the static and dynamic friction (especially in nondirect-drive arms) that tend to be large and thus hamper the effective evaluation of control schemes.

The goal of the CMU Direct-Drive Arm project, at Carnegie Mellon University, is to provide a testbed for research in sensor-based control of manipulators. In this context, we have been evaluating high speed trajectory tracking control strategies that incorporate joint position and velocity sensors in the feedback loop. Specifically, our goal is to evaluate the effect of including the dynamics model (in the control law) on the real-time trajectory tracking performance of manipulators. To achieve this goal, we have overcome the above mentioned hindrances and have experimentally implemented and evaluated the performance of the model-based schemes on the CMU DD Arm II at a sampling rate of 2 ms [16, 17, 11]. Other researchers have also independently implemented manipulator control schemes [2, 19, 20].

This paper is organized as follows: In Section 2, we briefly outline the customization procedure and present the controller architecture that has been used for experimental implementation. Our research in identification for obtaining an accurate dynamics model is presented in Section 3. The manipulator control schemes, that have been implemented, are described in Section 4 and the experimental results presented in Section 5. Finally, in Section 6, we summarize our paper.

[1]This work was supported in part by National Science Foundation under Grant ECS-8320364

[2]Assistant Professor, Department of Electrical and Computer Engineering, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

2. Customization and Controller Architecture

The recursive Newton-Euler (N-E) inverse dynamics formulation of complexity $O(N)$ has become a standard algorithm for real-time simulation. However, practical implementation of the inverse dynamics formulation to achieve real-time sampling rates (of 2 ms.) demands an efficient algorithm that utilizes the capabilities of modern digital hardware. Our approach towards achieving fast sampling rates has been two pronged: firstly, we have proposed the customization method that reduces the computational requirements; and secondly, we have developed a sophisticated computational architecture to implement and evaluate the real-time performance of manipulator control schemes.

The concept of *customizing* the Newton-Euler (N-E) dynamic equations, of a manipulator, to reduce the computational requirements is proposed in [9, 18, 10]. In *customization* we write out the scalar recursions, without incorporating iterative loops, thereby taking advantage of the sparsity of all matrices and vectors in the dynamics formulation. Customizing thus groups algebraically all quantities that can be combined at the outset, and eliminates multiplication by one (and minus one) and zero, and addition of zero. The computational efficiency in customization is thus obtained at the cost of increased program memory requirements. Our customization technique is very general and can be applied to an arbitrary N degrees-of-freedom manipulator. Customizing the inverse dynamics equations of CMU DD Arm II has resulted in a reduction computational requirements of about 56%.

For the real-time implementation of the inverse dynamics, we have developed a high speed controller architecture that is depicted in Figure 1. We have implemented all the control law computations in floating point and have obtained a computational cycle of 1 ms. The all-digital controller consists of a Motorola M68000 based single board computer, six TMS320 based individual joint processors, and a Marinco processor. The M68000 microcomputer is the host system and controls the associated Marinco processor and the TMS320 based joint controllers. Both the joint controllers and the Marinco processor are memory mapped devices. Thus the M68000 views the Marinco and the joint controllers as its own memory and communicates by reading or writing in the shared memory. This type of architecture speeds up the data transfer operations because *handshaking* commands for external communications are avoided.

The TMS320 based joint processors are used to provide the controller interface to the joint resolvers and the servo amplifiers. At each sampling instant the TMS320 reads the digital value of the position and velocity of the corresponding joint. These values are converted to a floating point representation and stored in the memory. Upon completion, the M68000 is signalled which then transfers these values to the Marinco processor for the computation of the control law. The M68000 also serves the bookkeeping purpose by recording the values of the joint positions, velocities and torques. After having transferred the measured data to the Marinco, the M68000 commands it to start the computation. Upon completion, the Marinco interrupts the host and the control torques are transferred to the TMS320 processors by the M68000 host. The above sequence of events continues every sampling instant until the system is commanded to stop.

Besides developing the hardware architecture, we have also developed a software monitor that provides the user interface. The monitor allows us to download trajectories from the Vax 750, execute the trajectories and also store the recorded trajectories on the Vax. The transfers between the Vax computer and the M68000 host take place over the Ethernet. Further implementational details may be found in [12]. In the ensuing sections, we describe our theoretical contributions in the area of identification and also present the results of our experimental implementation and evaluation of model-based control methods.

3. Identification of the Dynamics Model

One of the fundamental assumptions in the computed-torque scheme is that the model of the manipulator is accurately known. In order to satisfy this assumption, and evaluate the role of real-time dynamics model compensation, we have proposed techniques to estimate the dynamics parameters of an N degrees-of-freedom manipulator. The proposed identification algorithms are amenable to both off-line and on-line applications and particularly suited to real-time estimation of the pay-load dynamics parameters. The off-line estimation algorithm [15] is based on the Lagrange-Euler formulation and the on-line algorithm [13] is based on the Newton-Euler formulation. The results of the experimental implementation of the identification algorithms for the 6 degrees-of-freedom CMU DD Arm II are presented in [12]. Other researchers have also addressed the problem of estimating the dynamics parameters [1, 25]. In the sequel, we briefly present the underlying ideas of our identification algorithm that is based on the Newton-Euler formulation [23].

In order to derive the estimation algorithm, we first investigated some properties of the inverse dynamics formulation. We observed that in the Newton-Euler formulation, the classical link inertia tensors $\mathbf{I}_i$ and the link masses m_i appear linearly in the dynamics model, but the link masses are multiplied by linear and/or quadratic functions of the center-of-mass vectors $\mathbf{s}_i$ and nonlinear functions of the joint position variables θ_i. In contrast, the Lagrange formulation, which utilizes the pseudo-inertia matrices, has been shown to be linear in the dynamic parameters [15]. The pseudo-inertia matrices are formed by first expressing the classical inertia tensors about the link coordinate frames and then combining their elements linearly. We thus infer that if the Newton-Euler model is reformulated such that the link inertia tensors are expressed about the link coordinate frames instead of the link center-of-mass coordinate frame, the modified Newton-Euler formulation will be also linear in the center of the mass vectors $\mathbf{s}_i$. This nonlinear transformation is:

$$\mathbf{I}'_i=\mathbf{I}_i+m_i(\mathbf{s}_i^T\mathbf{s}_i\mathbf{E}-\mathbf{s}_i\mathbf{s}_i^T) \tag{1}$$

and is also known as Steiner's law. In the above transformation $\mathbf{I}_i$ is the classical link inertia tensor about the center-of-mass of link i, $\mathbf{I}'_i$ is the corresponding inertia tensor about the link i coordinate frame and $\mathbf{E}$ is the 3×3 identity matrix.

If we assume that we have nominal values of the dynamics parameters, say from engineering drawings, we can obtain the torque/force error model as [15]:

$$\varepsilon_i=\tau_i-\tau_i^0=\phi_i^T\Delta\psi_i \quad \mathit{for} \quad i=1,2,\cdots,N \tag{2}$$

where τ_i is the applied torque/force to link i, τ_i^0 is the value of the torque/force, as computed by an inverse dynamics model using the nominal values [23], ε_i is the input torque/force error of link i, $\Delta\psi_i$ is the correction vector of unknown parameters that affect the torque/force of link i, and ϕ_i is the nonlinear vector function of the kinematic parameters and the output measurements (joint positions, velocities and accelerations). The torque/force error model (2) relates the error torque/force of link i to corresponding modeling inaccuracies in the dynamic parameters.

Using this idea, we have generated the identification model of the 3 DOF positioning system of the CMU DD Arm II that is described by the following equations:

$$\begin{aligned}\varepsilon_3=\;&I_{3xz}[\theta_1^2-2C_3\theta_1^2+2\theta_1\theta_2-4C_3^2\theta_1\theta_2+\theta_2^2-2C_3^2\theta_2^2]\\&+I_{3yz}[-C_3\theta_2-C_3\theta_1]+I_{3yy}[\theta_3]\\&+(I_{3zz}-I_{3xx})[2C_3S_3\theta_1\theta_2+C_3S_3\theta_1^2+C_3S_3\theta_2^2]\\&+I_{3xy}[-S_3\theta_2-S_3\theta_1]\\&+m_3s_{3z}[-d_3C_3\theta_2-d_3C_3\theta_1-gS_3+a_1C_2C_3\theta_1^2\\&-a_1C_3S_2\theta_1+a_2C_3\theta_2^2+2a_2C_3\theta_1\theta_2+a_2C_3\theta_1^2]\\&+m_3s_{3x}[-d_3S_3\theta_2-d_3S_3\theta_1+gC_3+a_1C_2S_3\theta_1^2\\&-a_1S_3S_2\theta_1+a_2S_3\theta_2^2+2a_2S_3\theta_1\theta_2+a_2S_3\theta_1^2]\end{aligned} \tag{3}$$

$$\begin{aligned}\varepsilon_2=\;&I_{2yy}[\theta_1+\theta_2]+I_{3zz}[C_3^2\theta_2+C_3^2\theta_1]\\&+I_{3xx}[S_3^2\theta_2+S_3^2\theta_1]\\&+m_2s_{2x}[2a_2\theta_1+a_1C_2\theta_1+a_1S_2\theta_1^2+2a_2\theta_2]\\&+m_2s_{2z}[-a_1C_2\theta_1^2+a_1S_2\theta_1]\\&+m_3s_{3y}[a_1C_2\theta_1^2-a_1S_2\theta_1-2d_3\theta_2-2d_3\theta_1]\end{aligned} \tag{4}$$

$$\varepsilon_1=I_{1zz}\theta_1+m_1s_{1x}2a_1\theta_1 \tag{5}$$

where $C_i=cos\theta_i$ and $S_i=sin\theta_i$ and all the inertial elements are expressed about the link coordinate frames.

From these equations, we observe that:

- The torque of the third link is affected by all the elements of the inertia matrix and also by the two elements, m_3s_{3x} and m_3s_{3z}, of the center-of-mass vector. However, we can only identify seven of the eight parameters in (3) because the inertia elements I_{3xx} and I_{3zz} occur as a linear combination.
- The torque of link 2 is affected by two inertia elements of the third link, one inertia element of the second link, two elements of the center-of-mass vector of the second link and one element of the center-of-mass vector of the third link. All these parameters occur independently and can be uniquely identified. Note that we could identify only the linear combination $I_{3zz}-I_{3xx}$ from the torque/force error model of the third link whereas we can identify I_{3zz} and I_{3xx} independently from the torque/force error model of the second link.
- The torque of the first link is affected by two independent dynamic parameters, I_{1zz} and m_1s_{1x}, which can be identified uniquely.

Every link of a manipulator has ten dynamics parameters that must be estimated; these include the six elements of the symmetric inertia tensor of a link, the three elements of the center-of-mass vector and the mass of the link. Thus a manipulator with N links has $10N$ dynamics parameters which must be estimated to obtain a complete model. However, as has been noted, not all of the $10N$ parameters can be estimated. Thus before starting the estimation procedure, it is important to categorize the dynamics parameters as: uniquely identifiable, identifiable in linear combinations only, and unidentifiable. Our research has shown that it is indeed possible to do so based only on the kinematics description of the manipulator. An algorithm to categorize the dynamics parameters is presented in [12].

Another important aspect of the estimation procedure is determining trajectories that will estimate all the identifiable parameters. Such trajectories are called *persistently exciting* trajectories [6]. Our categorization algorithm does not only categorize the parameters but it also provides with a method of determining whether a given trajectory is persistently exciting. The problem of choosing persistently exciting trajectories, however, remains an open research issue [12].

The proposed identification algorithm together with the categorization procedure has served to satisfy an important assumption in model-based control of manipulators. We have implemented the identification algorithm on the six degrees-of-freedom CMU DD Arm II and obtained the estimates of dynamics parameters [13]. These estimated parameters have also been used to implement and evaluate the model-based control schemes on the CMU DD Arm II.

4. Robot Control Methods

The manipulator control problem revolves around computing the joint torques/forces that must be applied in order to track the desired joint trajectories. This problem is extremely challenging because the manipulator dynamics are described by a set of highly coupled and nonlinear differential equations. Consequently, many researchers have proposed control algorithms for the control of these complex systems. However, as pointed out earlier in the paper, there has been a lack of experimental evaluation of the proposed control methods. In this section, we describe the model-based control methods that were experimentally implemented on the CMU DD Arm II.

Independent Joint Control Scheme (IJC)

In this scheme, linear PD control laws are designed for each joint of the robot based on the assumption that the joints are decoupled and linear. The control signal applied to the joints at each sampling instant is computed as:

$$\tau = \mathbf{K}_p(\theta_d - \theta) + \mathbf{K}_v(\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d \tag{6}$$

where τ is the 6 x 1 vector of applied control torques and, $\mathbf{K}_p$ and $\mathbf{K}_v$ are 6 x 6 diagonal position and velocity gain matrices, respectively, and $\mathbf{J}$ is the constant and diagonal $N{\times}N$ matrix of link inertias at a typical position. The variables θ_d and θ are the desired and measured joint positions, respectively.

Computed-Torque Control Scheme (CT)

This scheme utilizes nonlinear feedback to decouple the manipulator by using the arm dynamics model and compensating, in real-time, for the dynamic and gravitational forces that vary as the arm configuration changes. The control torque τ is computed using the following equation:

$$\mathbf{u} = \mathbf{K}_p(\theta_d - \theta) + \mathbf{K}_v(\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d$$
$$\tau = \tilde{\mathbf{D}}(\theta)\mathbf{u} + \tilde{\mathbf{h}}(\theta,\dot{\theta}) + \tilde{\mathbf{g}}(\theta) = \mathit{InverseArm}(\theta,\dot{\theta},\mathbf{u}) \tag{7}$$

where $\mathbf{D}(\theta)$ is the position dependent inertial matrix, $\tilde{\mathbf{h}}(\theta,\dot{\theta})$ is the vector of centrifugal and Coriolis forces and $\tilde{\mathbf{g}}(\theta)$ is the vector of gravitational forces and " ~ " indicates that the estimated arm dynamics model is used in the computation.

Reduced Computed Torque Scheme (RCT)

The effect of the velocity dependent Coriolis and centrifugal term $\tilde{\mathbf{h}}(\theta,\dot{\theta})$ on the trajectory tracking performance of manipulators has been a subject of controversy [8, 26]. To study this effect, we compute the control torque excluding the velocity dependent terms in the inverse dynamics in (7). The control torque is therefore:

$$\tau = \tilde{\mathbf{D}}(\theta)\mathbf{u} + \tilde{\mathbf{g}}(\theta) \tag{8}$$

Feedforward Dynamics Compensation Scheme (FED)

The feedforward dynamics compensation technique is based on the premise that the gross torque for trajectory tracking is provided by applying the joint torques computed from the inverse dynamics model $\mathit{InverseArm}(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$, where the subscript d suggests that the desired trajectory is used in the computation. This feedforward signal is then augmented with the feedback signal derived from linear independent joint controllers which are assumed to correct for the small deviations in trajectory tracking. The control torque τ is therefore:

$$\tau = \tilde{\mathbf{D}}(\theta_d)\ddot{\theta}_d + \tilde{\mathbf{h}}(\theta_d,\dot{\theta}_d) + \tilde{\mathbf{g}}(\theta_d) + \mathbf{J}[\mathbf{K}_p(\theta_d - \theta) + \mathbf{K}_v(\dot{\theta}_d - \dot{\theta})] \tag{9}$$

where the first three terms are the feedforward compensation torque and the last term is the torque due to the feedback controller and $\mathbf{J}$ is the $N{\times}N$ diagonal matrix of link inertias at a typical position.

Reduced Feedforward Compensation Scheme (RFED)

The reduced feedforward compensation scheme computes the control torque by substituting the constant diagonal inertia matrix instead of $\mathbf{D}(\theta_d)$ in the first term in (9) as:

$$\tau = \mathbf{J}[\mathbf{K}_p(\theta_d - \theta) + \mathbf{K}_v(\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d] + \tilde{\mathbf{h}}(\theta_d,\dot{\theta}_d) + \tilde{\mathbf{g}}(\theta_d) \tag{10}$$

where τ is the N vector of applied control torques and $\mathbf{J}$ is the $N{\times}N$ diagonal matrix of link inertias at a typical position. This scheme has been implemented to study the effect of approximating the position dependent inertia matrix by a constant diagonal matrix.

An important consideration in implementation of the above mentioned schemes are the choice of the position and velocity gain matrices. Further, the above control schemes assume that the actuators can be modeled as a set of double integrators. In order to confirm this, we performed experiments to determine the characteristics of each actuator. It suffices to mention that the assumptions were indeed satisfied. Further details regarding choosing controller gains and experimental determination of actuator characteristics are presented in [12, 16, 17].

5. Experimental Results

We have implemented the five control schemes, presented in the previous section, and evaluated their real-time performance both experimentally and analytically. The objective of our experiments were: first, to compare the performance of the computed-torque, feedforward, and independent joint control schemes; second, to evaluate the effect of Coriolis and Centrifugal torques on the trajectory tracking performance; third, to underscore the need for including the off-diagonal elements of the inertia matrix in the control torque computation; fourth, to evaluate the effect of sampling rates on the performance of model-based control methods; and finally, to investigate the stability of the model-based control schemes in the presence of modeling errors.

The control schemes were implemented on the CMU DD Arm II with a control sampling period of 2 ms. Further, the schemes were tested with many different trajectories that were designed to study the effects of dynamics compensation. For the sake of brevity, we present only the highlights of our conclusions in this paper.

5.1. Comparative Performance Evaluation

One of the important constituents of experimental evaluation of the control schemes is the selection of trajectories. These must be selected such that the experimental results while being simple to analyze also depict the effects that are being evaluated. In order to accomplish this, we initially selected two simple but illustrative trajectories.

The first trajectory (also called Trajectory T1), depicted in Figure 2, was used to compare the performance of the IJC and CT control schemes. In this trajectory, only joint 2 is commanded to move while all the other joints are commanded to hold their positions dynamically. The second trajectory (also called Trajectory T2), depicted in Figures 3 and 4, was used to compare the performance of CT and FED control schemes. In this trajectory, joints 1 and 2 were commanded to move in opposite directions while all the other joints were commanded to maintain their positions dynamically. Our experiments with these simple trajectories have shed light on the role of dynamics in model-based manipulator control. Our experimental observations and conclusions are also supported by theoretical analysis and are are described in detail in [16, 17, 14]. These are also summarized in the succeeding paragraphs.

By comparing the performance of the computed-torque and the independent joint control schemes, we have demonstrated that CT scheme clearly outperforms the IJC scheme [16]. The trajectory performance of the CT scheme while executing trajectory T1 is shown is Figure 5. Figure 6 depicts the error for joint 2 (for trajectory T1) for the IJC, CT and RCT schemes. The maximum trajectory tracking errors of the first three joints are shown in Table 1. It had been intuitively argued before that the effect of the Coriolis and centrifugal forces becomes important only at high speeds [26]. Our experiments clearly demonstrate that not compensating for the Coriolis and centrifugal terms (as in RCT scheme) introduces significant trajectory tracking errors even at low velocities of 1 rad/sec.

In order to evaluate the effect of the off-diagonal terms in the manipulator inertia matrix, we compared the performance of the CT, FED and RFED schemes. Figures 7 and 8 depict the trajectory tracking performance of joints 1 and 2 for all the three schemes. It is clear that not incorporating the off-diagonal terms, in the RFED scheme, results in loss of accuracy in tracking when compared to the FED scheme. We have demonstrated through experiments and further by theoretical analysis that it is possible for the off-diagonal terms to completely dominate the computation of the control torques [17]. In comparing the performance of the CT and FED schemes, we observed that if an *accurate* model is available then both the schemes perform equally well. In such a circumstance, using the FED scheme offers some distinct computational advantages compared to the CT scheme.

5.2. Effect of Sampling Rates and Modeling Errors

In designing a real-time control system one of the important parameters to be selected is the sampling rate. For a computation intensive model-based control law, choosing a small sampling rate will result in complicated hardware. While on the other hand, choosing a large sampling rate may result in loss of trajectory tracking accuracy and also the stability of the system. The selection of sampling rates is well understood for linear time invariant systems. However, it is not easy to choose sampling rates for a complex and nonlinear system such as a manipulator. To provide some insight into the effect of sampling rates in robot control we conducted two types os experiments. First, we compared the performance of each scheme as the sampling rate was changed and second, we compared the relative performance of the CT and IJC schemes at different sampling rates. The details of our experimental results are presented in [14].

Figure 9 depicts the performance of the CT scheme as the sampling rate is changed from 2 to 5 ms and Figure 10 depicts the performance of the IJC schemes under similar conditions. While the performance of the CT scheme deteriorates only marginally, the performance of the IJC scheme becomes completely unacceptable. We conducted experiments with several different trajectories and our conclusions about the effect of control sampling rates can be summarized as follows:

1. If the gains are selected for a lower sampling rate and then if the sampling rate is increased, while keeping the gains fixed, there is no appreciable improvement in the performance of both the CT and the IJC schemes.
2. At lower sampling rates the CT scheme outperforms the IJC method. Even though the disturbance rejection ratio of both the schemes is diminished, it does not appreciably affect the CT method because of the compensation for the nonlinear and coupling terms. Whereas it affects the IJC method because the disturbance that is constituted by the nonlinear and the coupling terms is not rejected appreciably.
3. If the maximum possible gains are selected for the chosen sampling rates then the performance of CT at a higher sampling rate is better than its performance at a lower sampling rate. A similar conclusion is also drawn for the IJC scheme.

Our last conclusion is especially significant because it suggests that a higher sampling rate does not only imply improved performance but it also allows us to achieve high stiffness. It is desirable for a manipulator to have high stiffness so that the effect of unpredictable external disturbances on the trajectory tracking performance is significantly reduced.

One of the principal objections cited against the computed-torque scheme is that it is very sensitive to modeling errors [5]. In order to address this issue, we performed experiments wherein we changed the model of the manipulator drastically. In the first set of experiments, we changed the elements of the inertia matrix by 20%. We then performed the trajectory tracking experiments with this new model. The initial results of our experiments can be found in [12]. While we noticed an increase in the trajectory tracking errors as a function of the modeling errors, we did not note any instability. We have shown that the instability is dependent on the structure of the manipulator inertia matrix and not on the modeling errors. Similar observations have also been made together with theoretical justifications in [27].

6. Summary

In this paper we have presented an overview of our research in model-based control. The three important contributions of our research are: architectures for real-time computation and implementation of inverse dynamics, algorithms for dynamics parameter estimation, and real-time implementation and evaluation of model-based control schemes. Our experimental results have established the importance of dynamics compensation. Besides, the experiments have and also shed light on the role of the various terms in the dynamics model and their effect on the performance of the control schemes.

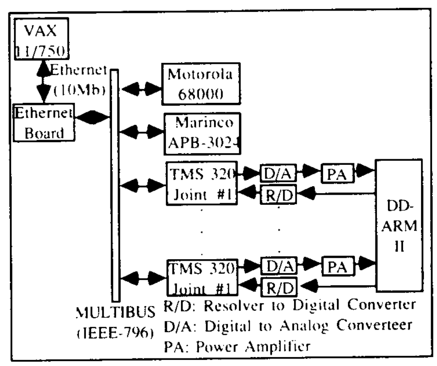

Figure 1: Controller Architecture of CMU DD Arm II

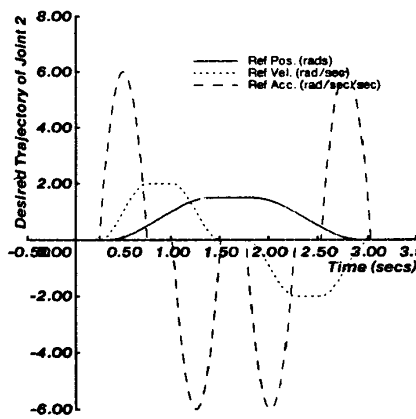

Figure 2: Desired Trajectories for Joint 2 [Trajectory T1]

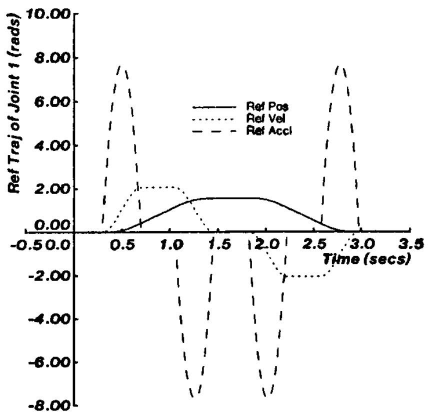

Figure 3: Desired Trajectories for Joint 1 [Trajectory T2]

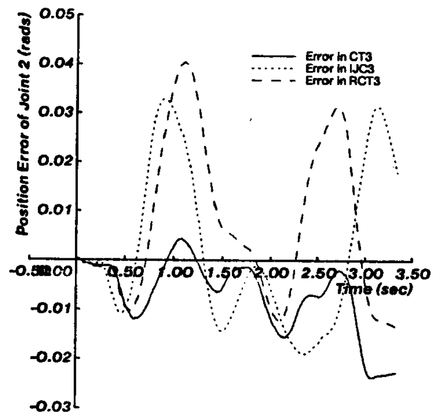

Figure 6: Position Tracking Errors of Joint 2 [Trajectory T1]

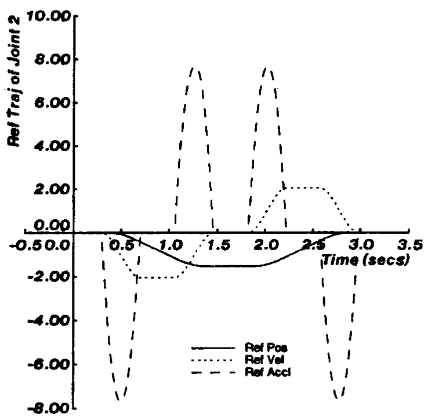

Figure 4: Desired Trajectories for Joint 2 [Trajectory T2]

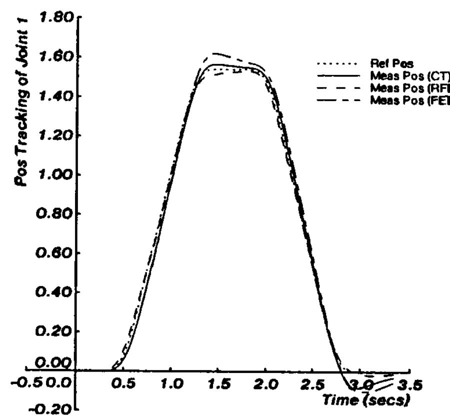

Figure 7: Position Tracking of Joint 1 [Trajectory T2]

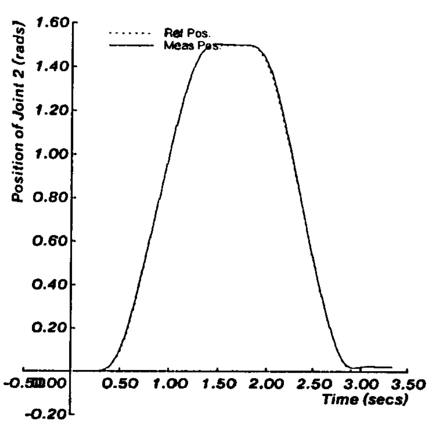

Figure 5: Position Tracking in CT3 [Trajectory T1]

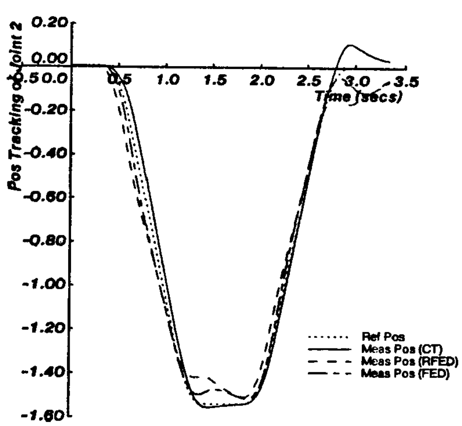

Figure 8: Position Tracking of Joint 2 [Trajectory T2]

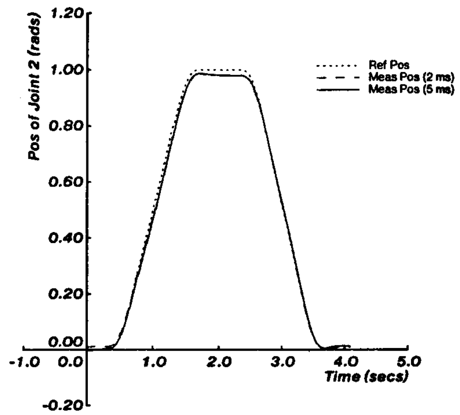

Figure 9: Performance of CT as a Function of Sampling Period

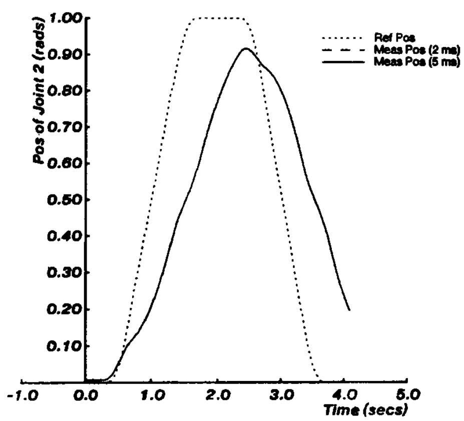

Figure 10: Performance of IJC as a Function of Sampling Period

	CT3		RCT3		IJC3	
Joint No.	Pos Error (rads)	Vel Error (rads/sec)	Pos Error (rads)	Vel Error (rads/sec)	Pos Error (rads)	Vel Error (rads/sec)
1	0.022	0.07	0.034	0.13	0.036	0.18
2	0.023	0.16	0.04	0.26	0.032	0.165
3	0.008	0.005	0.015	0.019	0.018	0.09

Table 1: Maximum Tracking Errors for Trajectory in Figure 2

References

1. An, C.H., Atkeson, C.G. and Hollerbach, J.M., Estimation of Inertial Parameters of Rigid Body Links of Manipulators. In *Proceedings of the 24th CDC.* Florida, December, 1985.
2. An, C.H., Atkeson, C.G. and Hollerbach, J.M., Experimental Determination of the Effect of Feedforward Control on Trajectory Tracking Errors. In Bejczy, A.K. (editor), *Proceedings of 1986 IEEE Conf. on Robotics & Automation*, pp. 55-60. IEEE, San Francisco, CA, April 7-10, 1986.
3. Brady, M., et al. (editors). *Robot Motion: Planning and Control.* MIT Press, Cambrdige, MA, 1982.
4. Dumas, R. and Samson, C. Robust Nonlinear Control of Robotic Manipulators: Implementation Aspects and Simulations. In *Robotics Research: The Next Five Years and Beyond.* SME, Bethlehem, PA, August, 1984.
5. Egeland, O. On the Robustness of the Computed-Torque Technique in Manipulator Control. In *Proceedings of the 1986 IEEE International Conf. on Robotics & Automation*, pp. 1203-1208. IEEE, San Francisco, CA, April 7-10, 1986.
6. Goodwin, G.C. and Sin, K.S. *Adaptive Filtering, Prediction and Control.* Prentice-Hall, Englewood Cliffs, NJ, 1984.
7. Hewit, J.R. and Burdess, J.S. Fast Dynamic Decoupled Control for Robotics Using Active Force Control. *Mechanism and Machine Theory* 16(5):535-542, 1981.
8. Hollerbach, J.M. Dynamic Scaling of Manipulator Trajectories. *Journal of Dynamic Systems, Measurement, and Control* 106(1):102-106, March, 1984.
9. Kanade, T., Khosla, P.K., and Tanaka, N. Real-Time Control of the CMU Direct Drive Arm II Using Customized Inverse Dynamics. In Polis, M.PO. (editor), *Proceedings of the 23rd IEEE Conf. on Decision and Control*, pp. 1345-1352. Las Vegas, NV, December 12-14, 1984.
10. Khalil, W., Kleinfinger, J.F. and Gautier, M. Reducing the Computational Burden of the Dynamic Model of Robots. In *Proceedings IEEE Conf. on Robotics & Automation*, pp. 525-532. IEEE, San Francisco, CA, April 1986.
11. Khosla, P.K. Determining Identifiable Parameters in Dynamic Robot Models. In *Proceedings of the 1986 IEEE Conf. on Robotics & Automation.* IEEE, San Francisco, CA, April 1986.
12. Khosla, P.K. *Real-Time Control and Identification of Direct-Drive Manipulators.* PhD thesis, Dept. of Electrical & Computer Eng., Carnegie Mellon University, August 1986.
13. Khosla, P.K. Estimation of Robot Dynamics Parameters: Theory and Application. In *Proceedings of the Second International IASTED Conference on Applied Control and Identification.* ACTA Press, Los Angeles, CA, December 10-12, 1986.
14. Khosla, P.K. Choosing Sampling Rates for Robot Control. In *Proceedings of 1987 IEEE Conf. on Robotics & Automation.* IEEE, Raleigh, NC, March, 1987.
15. Khosla, P.K. and Kanade, T. Parameter Identification of Robot Dynamics. In Franklin, G.F. (editor), *Proceedings of the 24th CDC*, pp. 1754-1760. Florida, December 11-13, 1985.
16. Khosla, P.K. and Kanade, T. Real-Time Implementation and Evaluation of Model-Based Controls on CMU DD ARM II. In Bejczy, A.K. (editor), *1986 IEEE International Conf. on Robotics & Automation.* IEEE, April 7-10, 1986.
17. Khosla, P.K. and Kanade, T. Experimental Evaluation of the Feedforward Compensation and Computed-Torque Control Schemes. In Stear, E.B. (editor), *Proceedings of the 1986 ACC.* AAAC, Seattle, WA, June 18-20, 1986.
18. Khosla, P.K. and Neuman, C.P. Computational Requirements of Customized Newton-Euler Algorithms. *Journal of Robotic Systems* 2(3):309-327, Fall, 1985.
19. Leahy, M.B., Valavanis, K.P. and Saridis, G.N. The Effects of Dynamics Models on Robot Control. In *Proceedings of the 1986 IEEE Conf. on Robotics and Automation.* IEEE, San Francisco, CA, April 1986.
20. Leborgne, M., Dumas, R., Borrelly, J., Samson, C. and Espiau, B. *Nonlinear Control of Robot Manipulators Part 2: Simulation and Implementation of a Robust Control Method.* Technical Report, IRISA/INRIA, 35042 RENNES Cedex, France, 1985.
21. Lee, C.S.G. Robot Arm Kinematics, Dynamics, and Control. *Computer* 15(12):62-80, December, 1982.
22. Luh, J.Y.S., Walker, M.W. and Paul, R.P. Resolved-Acceleration Control of Mechanical Manipulators. *IEEE Trans. on Automatic Control* 25(3):468-474, June 1980.
23. Luh, J.Y.S., Walker, M.W. and Paul, R.P. On-Line Computational Scheme for Mechanical Manipulators. *Journal of Dynamic Systems, Measurement, and Control* 102(2_:69-76, June 1980.
24. Markiewicz, B.R. *Analysis of the Computed-Torque Drive Method and Comparison with the Conventional Position Servo for a Computer-Controlled Manipulator.* Technical Memorandum 33-601, Jet Propulsion Laboratory, Pasadena, CA, March 1973.
25. Mukerjee, A. and Ballard, D.H. Self-Calibration in Robot Manipulators. In Fu, K.S. (editor), *Proceedings of the IEEE International Conf. on Robotics & Automation*, pp. 1050-1057. St Louis, Missouri, March 25-28, 1985.
26. Paul, R.P. *Robot Manipulators: Mathematics, Programming and Control.* MIT Press, Cambridge, MA, 1981.
27. Samson, C. *Une Approche Pour La Synthese et L'Analyse De La Commande Des Robots Manipulateurs Rigides.* Technical Report, INRIA, France, May 1987.
28. Slotine, J-J.E. The Robust Control of Robot Manipulators. *International Journal of Robotics Research* 4(2):81-100, Summer 1985.