

MULTIPLE MODEL-BASED CONTROL OF ROBOTIC MANIPULATORS: THEORY and SIMULATION

M. B. Leahy Jr. and L. D. Tellman

Air Force Institute of Technology
Department of Electrical and Computer Engineering
WPAFB OH 45433

ABSTRACT

The Multiple Model-Based Control (MMBC) technique utilizes knowledge of nominal plant dynamics and principles of Bayesian estimation to provide parameter independent trajectory tracking accuracy. The MMBC algorithm is formed by augmenting a model-based controller with a closed-loop form of Multiple Model Adaptive Estimation (Δ MMAE). The Δ MMAE uses perturbation models of the combined plant and feedback control system, along with measurements of tracking error, to provide an estimate of the plant parameters. When MMBC is applied to the robotic manipulator control problem the Δ MMAE provides a payload estimate. The model-based controller combines the a priori knowledge of robot structure with the payload estimate to produce the multiple models of the manipulator dynamics required to maintain controller accuracy. Extensive simulation studies on the first three links of a PUMA-560 have verified the algorithm's potential. MMBC provides a unique solution to the problem of maintaining trajectory tracking accuracy in uncertain payload environments.

1 INTRODUCTION

The intelligent manipulators required for advanced robotic telepresence applications must be able to fully emulate human arm motion. The manipulator's ability to duplicate the payload capacity, range of motion, speed, and tracking accuracy of the human arm system is essential if the man in the loop is to be able to operate the manipulator system in an intuitive manner. A long term goal of robotic system research at the Air Force Institute of Technology (AFIT) is the development of the enabling technologies for a manipulator system capable of human arm emulation. One of the essential components of such a system will be a control strategy that permits accurate tracking over random high speed trajectories without a priori knowledge of payload. Current industrial robot control approaches can not provide the required level of performance. In this paper we propose a new form of adaptive model-based robot control which may provide a technique for accurately tracking high speed trajectories in the presence of payload uncertainty.

Adaptive control of robotic manipulators is an area of active research. One of the most basic forms of adaptive control is the model-based approach. Experimental evaluations of model-based techniques have demonstrated their potential for improving tracking accuracy over high speed trajectories [11,1,7,15,12,2,28]. Those experiments have been conducted on both research robots [11,1,28] and existing industrial arms whose control structures have been highly modified [15,12,2]. Unfortunately, the model-based approaches patterned after the computed-torque technique

can only adapt to changes in manipulator joint configuration [9]. The tracking performance of those algorithms degrades noticeably in the presence of uncertain payloads [7], even for robots with high torque amplification drive systems [12].

Since the model-based control algorithm provides excellent tracking performance when accurate payload information is available, one approach to adaptive control has been to augment that controller with a payload adaptation mechanism [24]. A common theme in adaptive model-based control design has been the use of Lyapunov theory to develop the adaptation algorithms [24]. Experimental evaluations have attempted to estimate all equation of motion parameters that are a function of payload [7,23]. The Lyapunov based methods can be reduced to payload estimation [23] but no evaluations of adaptation based on payload estimation alone have been reported. While the experimental evaluations clearly demonstrate the potential of this form of adaptive robot control, Lyapunov based methods may not be appropriate for all possible robotic applications.

Lyapunov theory guarantees that the controller will be stable and that the steady state errors will asymptotically approach zero. Lyapunov theory does not predict how quickly the estimator will converge. Asymptotic stability is not a primary concern when the entire robotic motion may be completed in seconds and that motion is not repetitive. The global convergence proofs of Lyapunov techniques require a rigid robot assumption, and the estimation scheme can be susceptible to persistent excitation problems [24].

An alternative to the Lyapunov based approach is the use of stochastic estimation/adaptation techniques. In addition to providing a fast means of parameter adaptation the stochastic approach explicitly accounts for the numerous sources of noise and uncertainty in a real physical system. Multiple Model Adaptive Estimation (MMAE) is a Bayesian estimation approach that employs multiple Kalman filters to quickly and accurately estimate parameters in the presence of noise and uncertainty. MMAE has been successfully applied to several difficult aerospace tracking problems [21,20,3,5]. By combining the principles of MMAE and model-based control a powerful new form of adaptive model-based control was developed [27].

The Multiple Model-Based Control (MMBC) technique utilizes knowledge of nominal plant dynamics and principles of Bayesian estimation to provide parameter independent trajectory tracking accuracy. The MMBC algorithm is formed by augmenting a model-based controller with a closed-loop form of Multiple Model Adaptive Estimation (Δ MMAE). The Δ MMAE uses perturbation models of the combined plant and feedback control system, along with measurements of tracking error, to provide an estimate of the variable plant parameters. The model-based con-

$$z(t) = h(q, \dot{q}, \Upsilon, a, t) \quad (3)$$

troller combines the a priori knowledge of plant structure with the parameter estimate to produce the multiple models of the plant dynamics required to maintain tracking accuracy. MMBC was developed to reduce tracking error and therefore can be considered a direct form of adaptive control [23].

The objective of this research was to develop a MMBC algorithm for robotic manipulators and determine if that algorithm has the potential to provide payload invariant trajectory tracking performance. The test platform for that investigation was the first three links of a PUMA-560. The evaluated version of MMBC incorporated a Δ MMAE to provide payload information to the feedforward dynamic compensator of a previously evaluated model-based control law with constant PD feedback gain [12,15]. The results of our investigation suggest that robotic manipulator trajectory tracking performance can be made payload invariant by application of MMBC. This paper presents those results as follows. Section two reviews the general theoretical development of MMBC for rigid robotic manipulators. In section three MMBC trajectory tracking performance is analyzed for the PUMA case study. The development of the PUMA specific version of the MMBC is presented along with analysis of extensive simulation studies and a discussion of future research directions. Conclusions are presented in section four.

2 MMBC of Rigid Robotic Manipulators

The rigid robotic manipulator implementation of MMBC is a specific application of the more general principles of Multiple Model-Based Control. The estimation principles employed in the MMBC are not restricted to rigid plant dynamics and have been applied to control of flexible space structures [8,19]. A more general theoretical development of MMBC can be found in [27]. Additional information about the principles of Multiple Model Adaptive Estimation can be found in [18].

The nonlinear equations of motion for a rigid robot can be written as a function of payload:

$$N\Upsilon(t) = [D(q, a) + N^2M]\ddot{q} + h(\dot{q}, q, a) + N^2B_m\dot{q} + \tau_s + g(q, a) \quad (1)$$

where

- n = number of links in the robot
- q, \dot{q}, \ddot{q} = n vectors of joint angles, velocities, and accelerations.
- a = a 10 element vector representing the mass, mass centroid, and radii of gyration of the unknown payload
- N = n vector of gear ratios for each joint ($\frac{\text{motor velocity}}{\text{link velocity}}$).
- $D(q, a)$ = $n \times n$ matrix of manipulator inertias which depend on the load, the position of the manipulator, and the gear ratio.
- M = diagonal $n \times n$ matrix of actuator inertia terms.
- $h(\dot{q}, q, a)$ = n vector of centripetal and Coriolis torques.
- τ_s = n vector of static friction torques.
- B_m = n vector of damping coefficients
- $g(q, a)$ = n vector of gravity loading terms.
- $\Upsilon(t)$ = n vector of joint motor torques.

The first step in applying the Multiple Model-Based Control technique to a robotic manipulator was to rewrite the non-linear equations of motion (1) in a more general form:

$$\ddot{q}(t) = f(q, \dot{q}, \Upsilon, a, z, t) \quad (2)$$

where $f(\bullet)$ and $h(\bullet)$ are in general, nonlinear functions and $z(t)$ is a p vector of measurements.

A robot system is inherently noisy. The noise sources arise from imperfect calibration, incorrectly modeled components, and imperfect measurements of the joint position. If the noises are assumed to be added linearly to Equations (2-3), the result is a stochastic non-linear differential equation of the following form: [17].

$$\begin{aligned} \ddot{q}(t) &= f(q, \dot{q}, \Upsilon, a, z, t) + G'(t)W(t) \\ z(t) &= h(q, \dot{q}, \Upsilon, a, t) + V(t) \end{aligned} \quad (4)$$

where:

- $G'(t)$ = scaling matrix for the additive system noise.
- $W(t)$ = vector of zero mean, white Gaussian dynamics driving noise.
- $V(t)$ = vector of zero mean, white Gaussian measurement noise.

The basic structure of a model-based controller allows the control system to be separated into a precompensator and plant block that produce a nominal output, and a feedback block that produces a perturbation output. That control structure can be represented in state space form as a perturbation regulator [26]. The precompensator element produces a nominal control input given the desired position, velocity and acceleration trajectory. Applying the nominal input to the plant generates the nominal position and velocity states. The difference between the desired and nominal states is assumed to result from the disturbances in the system, $W(t)$. The feedback gains, $K(a, t)$ attempt to drive the difference to zero. The perturbation plant, $F'(a, t)$, is the first-order result of the truncated Taylor series of $f(q, \dot{q}, \Upsilon, a, t)$.

The model of the closed-loop perturbation plant matrix can be represented by:

$$\begin{aligned} \dot{x}(t) &= F(a, t)x(t) + G(a, t)W(t) \\ z(t) &= H(t)x(t) + V(t) \end{aligned} \quad (6)$$

where:

- x = $2n$ -vector of position and velocity perturbation states
- $F(a, t) = F'(a, t) - G(a, t)K(a, t)$
- $F'(a, t)$ = a nonlinear square matrix function of a and a linear function of the states that describes the homogeneous perturbation state dynamics characteristics.
- $K(a, t)$ = a square matrix of position and velocity feedback gains
- $G(a, t)$ = a square matrix that transforms the noise into the state space.
- $z(t)$ = a p -vector of noise corrupted measurements of the error states.
- $H(t)$ = the measurement matrix that transforms the states into the measurement space.

Bayesian estimation in a multiple model configuration can be used to determine the unknown parameter a in Equation (6) [18]. The basic premise of the Δ MMAE technique is that the *variations* in the continuous parameter vector a can be discretized into a finite set of possible vector values, (a_1, a_2, \dots, a_K) . The discretization of a must be large enough that there is a discernible difference between the models but not so large as to induce unacceptable errors in the estimate. The Δ MMAE is composed of

K Kalman filters running in parallel, each of whose plant models is based upon an assumed parameter variation a_j as shown in Figure 1. For a sampled data system the individual Kalman filter equations are:

$$\hat{x}(t_i^-) = \Phi(t_{i+1}, t_i)\hat{x}(t_i^+) \quad (8)$$

$$P(t_i^-) = \Phi(t_{i+1}, t_i)P(t_i^+)\Phi^T(t_{i+1}, t_i) + Q_d(t_i) \quad (9)$$

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i)[z(t_i) - H(t_i)\hat{x}(t_i^-)] \quad (10)$$

$$P(t_i^+) = P(t_i^-) - K(t_i)H(t_i)P(t_i^-) \quad (11)$$

$$K(t_i) = P(t_i^-)H^T(t_i)[H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]^{-1} \quad (12)$$

where:

- $\hat{x}(t_i^-)$ = the estimate of the state vector at time t_i , just prior to the measurement being processed at t_i .
- $\hat{x}(t_i^+)$ = the state vector at time t_i after the measurement has been processed at t_i .
- $P(t_i^-)$ = the covariance matrix of the state at time t_i^- .
- $P(t_i^+)$ = the covariance matrix of the state at time t_i^+ .
- $z(t_i)$ = the vector of noise corrupted measurements of the error states at time t_i .
- $H(t_i)$ = the measurement matrix that transforms the states into the measurement space.
- $K(t_i)$ = the Kalman filter gain matrix at time t_i .
- $\Phi(t_{i+1}, t_i)$ = the state transition matrix associated with $F(a_j, t)$ of Equation (6), defined as the $2n \times 2n$ matrix that satisfies $\dot{\Phi}(t, t_i) = F(a_j, t)\Phi(t, t_i)$ with $\Phi(t_i, t_i) = I$.
- $Q_d(t_i) = \int_{t_i^-}^{t_i^+} \Phi(t_{i+1}, \tau)G(\tau)Q(\tau)G^T(\tau)\Phi^T(t_{i+1}, \tau)d\tau$ and $Q(t)$ is the strength of the Gaussian noise,
- $W(t): E[W(t)W^T(t + \tau)] = Q(t)\delta(\tau)$.
- $R(t_i)$ = the matrix of the Gaussian noise strength, $V(t_i): E[V(t_i)V^T(t_i)] = R(t_i)$.

Each of the Kalman filters is presented with the same measurement vector, $z(t_i)$ and produces a state estimate based upon its internally assumed model. The state estimate is used to generate the filter residuals, $r(t_i) = [z(t_i) - H(t_i)\hat{x}(t_i^-)]$. The residuals are passed to an executive program that computes a conditional probability, $p_j(t_i)$ (see Equations (13-14)) and the direction of the parameter variation (see Equation (16)).

$$p_j(t_i) \triangleq \text{prob}\{a = a_j \mid Z(t_i) = Z_i\} \quad (13)$$

$$p_j(t_i) = \frac{f_{s(t_i)|a, Z(t_{i-1})}(z_i \mid a_j, Z_{i-1})p_j(t_{i-1})}{\sum_{k=1}^K f_{s(t_i)|a, Z(t_{i-1})}(z_i \mid a_k, Z_{i-1})p_k(t_{i-1})} \quad (14)$$

where:

- $Z(t_{i-1})$ = the measurement history up to time t_{i-1}
- $f_{s(t_i)|a, Z(t_{i-1})}(z_i \mid a_j, Z_{i-1})$ = the conditional probability that the j^{TH} filter was correct. For the assumed Gaussian distribution it has the form $\frac{1}{(2\pi)^n |A|^{1/2}} e^{(-1/2r^t A^{-1} r)}$ where $A(t_i) = [H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]$.
- the denominator scales the conditional probability such that $\sum_{j=1}^K p_j(t_i) = 1$

The conditional mean of the parameter variation Δa at t_i is given by:

$$\Delta a(t_i) \triangleq \sum_{j=1}^K a_j p_j(t_i) \quad (15)$$

Therefore $\Delta a(t_i)$ is the smoothed optimal Bayesian estimate of the parameter variation.

The sign on the residuals from the Kalman filters indicates whether Δa is to be added to the current value of \hat{a} in the feed-forward element or subtracted from it as shown below:

$$\hat{a}(t_i) = \hat{a}(t_{i-1}) + \Delta a(t_i) \text{SIGN}[f(r(t_i))] \quad (16)$$

where \hat{a} is the actual parameter estimate output from the Δ MMAE and $f(\bullet)$ is a degree of design freedom used to maximize performance for a specific application.

For the robot control case \hat{a} represents an estimate of the payload vector and the MMBC law is defined as:

$$NT(t) = [D(q, \hat{a}) + N^2 M][\ddot{q}_d + 2\zeta\omega_n \dot{e} + \omega_n^2 e] + h(\dot{q}, q, \hat{a}) + N^2 B_m \dot{q} + \tau_s + g(q, \hat{a}) \quad (17)$$

where

- $e = q_d - q$, the joint position error vector
- ζ = diagonal matrix of desired damping ratios
- ω_n = diagonal matrix of desired natural frequency

The achilles heal of the MMBC approach is the potentially large number of Kalman filters required to estimate all ten elements of the payload vector. However, a little knowledge of manipulator dynamics allows a great reduction in algorithm complexity. Experimental evaluations have shown that most of the degradation in model-based algorithm performance can be recovered with just knowledge of payload mass and centroid [2,12,13]. If the length of the links is large, compared to the distance of the payload centroid from the end-effector axis, only mass information is required [2,12,13]. Therefore, \hat{a} is a scalar and the Δ MMAE is reduced to a single bank of filters. When physical insight can not reduce the number of Kalman filters other techniques can be applied to reduce the computational burden [8,19].

3 PUMA Case Study

The objective of this case study was to determine if the MMBC algorithm has the potential to provide payload invariant trajectory tracking performance. The test case for that effort was the first three links of a PUMA-560. The PUMA-560 was an appropriate test case because [13,15]:

- its trajectory tracking performance has been extensively studied,
- tracking performance is a function of payload,
- reducing the payload vector to just the mass parameter has minimal impact on large link tracking performance, and
- existing facilities could be modified to experimentally evaluate algorithm performance if the simulation tests were successful.

The ability to accurately compensate for payload with only mass information reduces a to the scalar case. The reduction to a scalar estimation task is appropriate for an initial evaluation. If the MMBC could not perform adequately for single parameter adaption there would be no point investigating more complex payload estimation requirements.

Without loss of generality, the payload was assumed to be a point mass rigidly attached to the end of the third link. The general form of the MMBC shown in Equation (17) can be reduced to

$$NT(t) = [D(q, \hat{a}) + N^2 M]\ddot{q}_d + h(\dot{q}, q, \hat{a}) + N^2 B_m \dot{q}_d + \tau_s + g(q, \hat{a}) + [K_v \dot{e} + K_p e] \quad (18)$$

without degradation of PUMA trajectory tracking performance [15]. The K_p and K_v PD gain values are identical to those employed in previous experimental evaluations [12,15]. The constant PD gains were designed to produce critically damped response when the link was in its minimal inertial configuration.

3.1 Δ MMAE Implementation

The process of transforming the stochastic non-linear differential equation (4) to a perturbation regulator (6) required the partial derivative of Equation (4) with respect to q and \dot{q} evaluated at the nominal q, \dot{q}, τ, a_j . A program using MACSYMA [10] commands was developed to provide a symbolically reduced set of equations for $F'(a_j, t)$ [27]. The first three links of the PUMA were modeled using the equations and inertial parameters developed by Tarn in [25]. The friction and motor damping information was from experimental evaluations by Leahy and Saridis [15].

The plant matrix used by the Kalman filters was:

$$F(a_j, t_i) = (F'(a_j, t_i) - G(a_j, t_i)K) \quad (19)$$

where K represents a block diagonal matrix of the constant position and velocity gains, K_p and K_v . $W(t)$ was assumed to be added to the nominal torque. Therefore the $G(a_j, t_i)$ matrix which transforms the torque noise into the state space had the following form:

$$G(a_j, t_i) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ (D(q(t_i), a_j) + N^2 M)^{-1} \end{bmatrix} \text{ desired} \quad (20)$$

The only measurements available on the PUMA-560 are the actual joint positions. Therefore the only measurements input to the Kalman filters are the error in the position states. Since $z(t_i)$ is a linear function of the position states:

$$H(t_i) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (21)$$

The $F(a_j, t_i)$ and $G(a_j, t_i)$ matrices are payload and trajectory dependent, but can be assumed constant over the sample period [27,13]. Therefore, $\Phi(t_i, t_{i-1})$ and $Q_d(t_i)$ were accurately approximated by:

$$\Phi(t_i, t_{i-1}) \approx I + F(a_j, t_i)\Delta t + 1/2 F^2(a_j, t_i)\Delta t^2 \quad (22)$$

$$Q_d(t_i) \approx G(t_i)QG^T(t_i)\Delta t \quad (23)$$

where Δt is the sample period.

The value of measurement noise, V was determined from the resolution of the encoders. The probability density function of the noise is uniform with zero mean and covariance equal to $\pm 1/2$, the encoder resolution, and was approximated by a Gaussian distribution with identical mean and covariance. Additional noise information can be easily added via shaping filters [17]. The dynamics driving noise, $Q(t)$, was tuned to provide the best performance of the Δ MMAE. Once $Q(t)$ was selected, that value was held constant for all test trajectories. The initial conditions of $\hat{x}(t_0)$ and $P(t_0)$ were assumed to be zero. Further details about the filters and their tuning may be found in [26].

The procedure used to discretize the parameter space so that the different Kalman filters are based on sufficiently different

models is outlined in [26]. An optimal technique for the discretization of a was beyond the scope of this research. Previous PUMA research suggested that a reasonable discretization could be achieved with only three levels. Therefore, $K = 3$ and the a_j values for the filters were set at 0.0, 2.5 and 5.0 Kg. The residual function, $f(r(t_i))$, used in Equation (16) was simply the joint 2 residual from the 2.5 Kg filter.

3.2 Simulation Studies

The tracking performance of the MMBC was initially evaluated by digital simulation. Every effort was made to produce simulation results that would produce a valid indicator of real PUMA performance. The feedforward compensator, PD servo loop, and the Δ MMAE were all updated with new measurement information at a 142 Hz sample rate. 142 Hz corresponds to the fastest sample rate supported by our experimental evaluation environment. The test trajectories have been utilized in previous PUMA evaluations [15,12]. Robot motion was simulated by a 4th order Runge-Kutta integration of Equation (1) which also included Gaussian dynamic driving noise and uniform measurement noise [27]. Due to space limitations only the error profiles of joint 2 are included. A more complete set is in [16].

The tracking performance of the MMBC algorithm was compared to two other forms of model-based control. Both forms of Single Model-Based Control (SMBC) were just realizations of Equation (18) without payload adaption, i. e. the value of \hat{a} was held constant. The difference between the two SMBC algorithms was in the constant \hat{a} value. Worst case model-based control performance was simulated by a version of the SMBC with $\hat{a} = 0$. Peak model-based performance was simulated by artificially informing the second version of SMBC of the payload value used by the arm simulator. Ideally the performance of the MMBC and the artificially informed SMBC would be identical.

Initial evaluations duplicated the test conditions employed in previous evaluations of payload effects on PUMA performance [12]. The arm was commanded to move from $(0^\circ, -135^\circ, 135^\circ)$ to $(90^\circ, -90^\circ, 45^\circ)$ in 1.5 seconds. Payload was constant at 2.3 Kg and the MMBC \hat{a} value was initialized to zero. The performance of the MMBC didn't meet the ideal, but as Figure 2 illustrates that the Δ MMAE can very quickly provide an estimate of payload that will significantly reduce the tracking error. The Δ MMAE locked onto a payload estimate by 0.4 seconds, and the oscillations in the transient region have minimal effect on tracking performance. Examples of Δ MMAE payload estimates are in [27]. The peak tracking and final position errors of the MMBC were very close to the artificially informed SMBC and a significant improvement over those produced by the SMBC without payload information.

To determine if the MMBC performance was payload invariant the single versus multiple model comparison was performed over an ensemble of payloads. The payload mass was varied from 0.9 Kg in 1 Kg increments and the tracking performance simulated. The results were averaged to get a true indication of MMBC performance. Figure 3 compares the mean tracking accuracy of the MMBC and the artificially informed SMBC. Even for simulation results, the differences between the two algorithms were minimal. MMBC error was always within one standard deviation (σ) of the ideal. For this set of test conditions, the MMBC demonstrated the potential to provide payload invariant tracking performance.

To provide a more thorough test of MMBC capabilities, a task was simulated where the robot picks up an unknown payload and while in motion, inadvertently drops the payload. The payload was initially set to 2.3 Kg and was reset to zero, 0.6 seconds into the trajectory. The drop time was after the initial acquisition period and before the peak velocity. The MMBC and SMBC were initialized with \hat{a} values of 0.0 Kg and 2.3 Kg respectively. The artificially informed SMBC algorithm switches payload information to 0.0 Kg at 0.6 seconds. As the Figure 4 shows, the MMBC rapidly adapted to both payload changes maintaining excellent tracking performance. Similar results were produced for other payload values and drop times [27].

The dependence of algorithm performance on trajectory was also evaluated. While the shape of the error profiles varies slightly from trajectory to trajectory the dependence on payload was minimal [16]. A more optimal set of tuning parameters may eliminate the slight trajectory dependence.

3.3 Discussion

Our initial development and evaluation identified several areas of future research. More general techniques for linearizing manipulator dynamics have been proposed [22,4]. The computational advantage of those techniques, if any, over our symbolic formulation will be investigated. The results presented here required a bank of three filters employing two states for each joint. Additional results indicate that the payload estimation can be accomplished by only monitoring the motion of a single joint. A reduction in the size of the filter bank may also be possible. Simulation studies are underway to evaluate the feasibility of those modifications. The ability of the MMBC to provide payload invariant tracking for manipulators that require payload mass and centroid information is also under investigation.

The MMBC is designed to take advantage of the latest advances in microprocessor technology. VHSIC researchers at AFIT have developed an application specific processor (ASP) with a high speed floating point double precision adder and multiplier controlled by the microcode in a laser programmable ROM [6]. The microcode required to implement a Kalman filter on that chip is currently under development. The Kalman filter ASP will reduce the computational time of the individual Kalman filters used in MMBC to the microsecond range.

4 Conclusion

The Multiple Model-Based Control (MMBC) technique provides a unique solution to the problem of maintaining robotic manipulator trajectory tracking accuracy in uncertain payload environments. Robotic applications of MMBC are not restricted by excitation requirements or rigid body dynamics. The MMBC algorithm utilizes knowledge of nominal manipulator dynamics and principles of closed-loop multiple model Bayesian estimation to quickly and accurately adapt to payload variations. Extensive simulation studies have demonstrated the payload independent trajectory tracking accuracy of MMBC. The simulation results clearly warrant experimental evaluation and further testing of the algorithm's potential. Current MMBC research is concentrated on experimentally evaluating tracking performance. Refinements to the MMBC technique should produce a robot control algorithm with the ability to emulate human arm gross motion.

Acknowledgments

This research was partially supported by the USAF Armstrong Aerospace Medical Research Laboratory Robotic Telepresence program. The authors would like to thank Dr Peter Maybeck for his MMAE expertise.

References

- [1] C. H. An, C. G. Atkeson, J. D. Griffiths, and J. M. Hollerbach. Experimental evaluation of feedforward and computed torque control. *Proc. of the IEEE Int. Conf. on Rob. and Auto.*, 165-168, March/April 1987.
- [2] H. Asada and K. Hara. Load sensitivity analysis and adaptive control of a direct drive arm. *Proc. of the ACC*, 799-805, 1986.
- [3] M. Athans et al. The stochastic control of the f-8c aircraft using a multiple model adaptive control (mmac) method - part 1: equilibrium flight. *IEEE Trans. on Automatic Control*, AC-22(5):768-780, October 1977.
- [4] C. A. Balafoutis, P. Misra, and R. V. Patel. Recursive evaluation of linearized dynamic robot models. *IEEE J. of Rob. and Auto.*, RA-2(3):146-154, Sept. 1986.
- [5] R. F. Berg. Estimation and prediction for maneuvering target trajectories. *IEEE Trans. on Automatic Control*, AC-28(3):294-304, March 1983.
- [6] J. H. Comtois. *Architecture and Design For A Laser Programmable Double Precision Floating Point Application Specific Processor*. Master's thesis, Air Force Institute of Technology, Air University, December 1988. GE/ENG/88-5.
- [7] J. J. Craig, P. Hsu, and S. S. Sastry. Adaptive control of mechanical manipulators. *Int. J. of Rob. Res.*, 6(2):16-28, Summer 1987.
- [8] K. A. Drew and P. S. Maybeck. Moving bank multiple model adaptive estimation applied to flexible spacestructure control. *Proc. of the 26th IEEE CDC*, 1249-1257, December 1987.
- [9] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill Book Company, New York, 1987.
- [10] Symbolics Inc. *VAX UNIX MACSYMA Reference Manual*. November 1985.
- [11] P. K. Khosla and T. Kanade. Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators. *Int. J. of Rob. Res.*, 7(1), February 1988.
- [12] M.B. Leahy Jr. Dynamics based control of vertically articulated manipulators. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1046-1056, April 1988.
- [13] M.B. Leahy Jr. Dynamics based control of vertically articulated manipulators with variable payloads. *Accepted for publication in Int. Journal of Robotics Research*, 1989.
- [14] M.B. Leahy Jr. Experimental analysis of robot control: a performance standard for the puma-560. *IEEE Int Sym on Int Control*, 1989.
- [15] M.B. Leahy Jr. and G.N. Saridis. Compensation of industrial manipulator dynamics. *Int. J. of Rob. Res.*, 8(4), August 1989.
- [16] M.B. Leahy Jr. and L. D. Tellman. Multiple model-based control of robotic manipulators. *Technical Report ARSL-89-2*, 1989.
- [17] P. S. Maybeck. *Stochastic Models, Estimation and Control*. Volume 1, Academic Press, Inc., London, 1979.
- [18] P. S. Maybeck. *Stochastic Models, Estimation and Control*. Volume 2, Academic Press, Inc., London, 1979.

- [19] P. S. Maybeck and K. P. Hentz. Investigation of moving-bank multiple model adaptive algorithms. *Journal of Guidance, Control and Dynamics*, 10(1):90-96, 1987.
- [20] P. S. Maybeck and S. K. Rogers. Adaptive tracking of multiple hot spot target in images. *IEEE Trans. on Automatic Control*, AC 28(10):937-943, October 1983.
- [21] P. S. Maybeck and W. L. Zicker. Mmae-based control with space-time point process observations. *IEEE Trans. on Aerospace and Electronic Systems*, AES 21(3):292-300, May 1985.
- [22] C. P. Neuman and J. J. Murray. Linearisation and sensitivity functions of dynamic robot models. *IEEE Trans. on Sys., Man, and Cyber.*, SMC-14(6):805-818, Nov/Dec 1984.
- [23] G. Niemeyer and Slotine J.-J. E. Performance in adaptive manipulator control. *Proc of 27th IEEE CDC*, 1585-1591, December 1988.
- [24] R. Ortega and M. W. Spong. Adaptive motion control of rigid robots: a tutorial. *Proc of 27th IEEE CDC*, 1575-1584, December 1988.
- [25] T.J. Tarn and A.K. Bejczy. *Dynamic Equations for PUMA-560 Robot Arm Technical Report SSM-RL-85-02*. Dept. of Systems Science and Mathematics, Washington University, St. Louis, MO, July 1985.
- [26] L. D. Tellman. *Multiple Model-Based Robotic Control: Development and Initial Evaluation*. Master's thesis, Air Force Institute of Technology, Air University, October 1988.
- [27] L. D. Tellman and M.B. Leahy Jr. Multiple model-based control: development and initial evaluation. *Technical Report ARSL-89-1*, accepted at the 28th IEEE CDC, 1989.
- [28] K. Youcef-Toumi and A. T. Y. Kuo. High speed trajectory control of a direct-drive manipulator. *Proc. of the 26th IEEE CDC*, 2209-2214, December 1987.

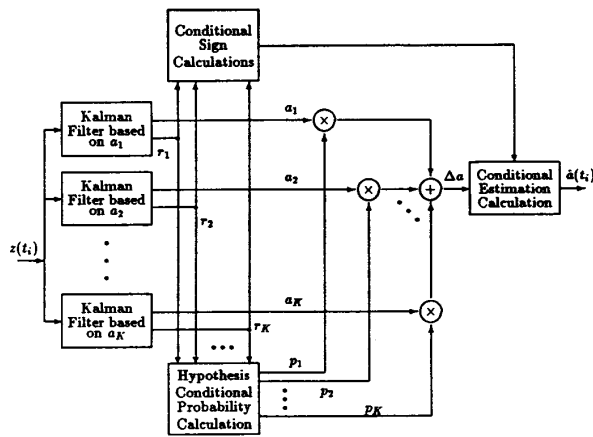


Figure 1: Δ MMAE Block Diagram

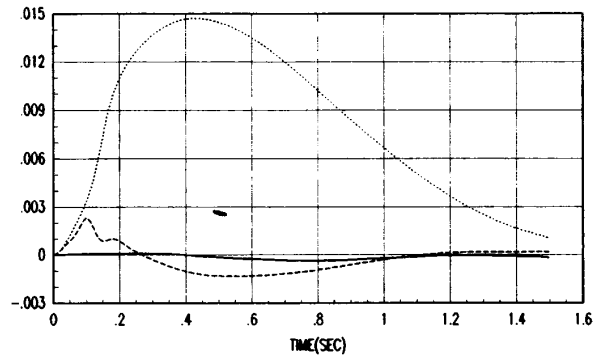


Figure 2: Trajectory 1 Tracking Error for Link 2 with 2.3kg Payload

—	SMBC with Full Payload Information
...	SMBC w/o Payload Information
- - -	MMBC

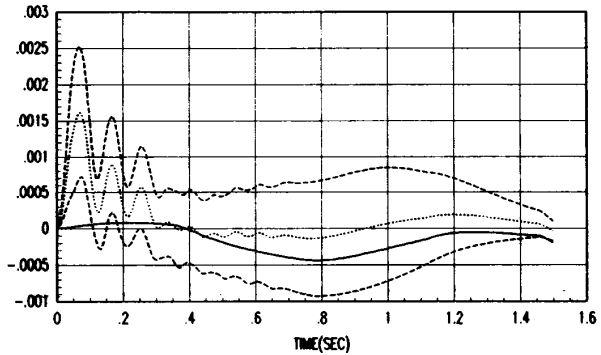


Figure 3: Trajectory 1 Tracking Error for Link 2 over Payload Ensemble

—	Mean SMBC with Full Payload Information
...	Mean MMBC
- - -	Mean MMBC +/- One Sigma

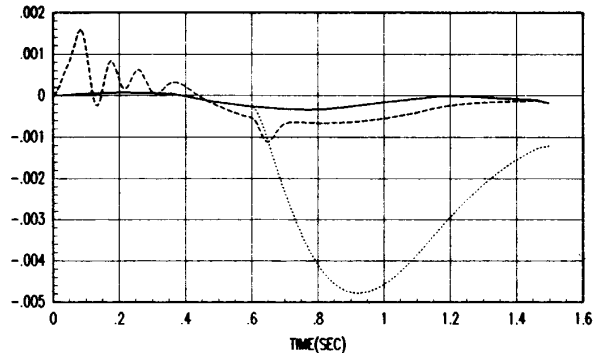


Figure 4: Trajectory 1 Tracking Error for Link 2 with Dropped Payload

—	SMBC with Full Payload Information
...	SMBC w/o Payload Information
- - -	MMBC