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 (MMAE). The 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 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 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 (MMAE). The 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-
controller 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 AMMAE 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:

\[ NT(t) = [D(q,a) + N^2M]q + h(q, q, a) + N^2B_m \dot{q} + \tau_e + g(q, a) \] (1)

where
- \( n \) is number of links in the robot
- \( q, \dot{q} \) are vectors of joint angles, velocities, and accelerations.
- \( a = 10 \) vector representing the mass, mass centroid, and radii of gyration of the unloaded robot
- \( D(q,a) = n \times n \) matrix of manipulator inertias which depend on the load, position of the manipulator, and the gear ratio.
- \( M = \) diagonal \( n \times n \) matrix of actuator inertias terms.
- \( h(q, q, a) = n \) vector of centrifugal and Coriolis torques.
- \( \tau_e = n \) vector of static friction torques.
- \( B_m = n \) vector of damping coefficients.
- \( g(q, a) = n \) vector of gravity loading terms.
- \( T(t) = n \) 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:

\[ \ddot{q}(t) = f(q, \dot{q}, T, a, z, t) \] (2)

where \( f(\cdot) \) and \( h(\cdot) \) 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:

\[ \ddot{q}(t) = f(q, \dot{q}, T, a, z, t) + G(t)W(t) \] (4)
\[ z(t) = h(q, \dot{q}, T, a, t) + V(t) \] (5)

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 [20]. 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 \( f(q, \dot{q}, T, a, t) \).

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

\[ \dot{z}(t) = F(a,t)z(t) + G(a,t)W(t) \] (6)
\[ z(t) = H(t)z(t) + V(t) \] (7)

where:
- \( z = 2n \) 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.
- \( z(t) = p \) 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 AMMAE technique is that the variations in the continuous parameter vector \( a \) can be discretized into a finite set of possible vector values, \((a_1, a_2, \ldots, a_K)\). 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 AMMAE is composed of
Kalman filters running in parallel, each of whose plant models is based upon an assumed parameter variation \(a_i\) as shown in Figure 1. For a sampled data system the individual Kalman filter equations are:

\[
\begin{align*}
\dot{x}(t_i^+) &= \dot{x}(t_i^-) + \dot{w}(t_i^-) \\
P(t_i^+) &= P(t_i^-) + \Sigma P(t_i^-) \Sigma^T + Q(t_i^-) \\
\dot{x}(t_i^-) &= \dot{x}(t_i^-) + K(t_i) (z(t_i) - H(t_i) \dot{x}(t_i^-)) \\
P(t_i^-) &= P(t_i^-) - K(t_i) H(t_i) P(t_i^-) \\
K(t_i) &= P(t_i^-) H(t_i) P(t_i^-) + R(t_i)^{-1}
\end{align*}
\]

where:

- \(\dot{x}(t_i^-)\) is the estimate of the state vector at time \(t_i^-\), just prior to the measurement being processed at \(t_i\).
- \(\dot{x}(t_i^+)\) is the state vector at time \(t_i^+\) after the measurement has been processed at \(t_i\).
- \(P(t_i^-)\) is the covariance matrix of the state at time \(t_i^-\).
- \(P(t_i^+)\) is the covariance matrix at time \(t_i^+\).
- \(z(t_i)\) is the vector of noise corrupted measurements of the error states at time \(t_i\).
- \(H(t_i)\) is the measurement matrix that transforms the states into the measurement space.
- \(K(t_i)\) is the Kalman filter gain matrix at time \(t_i\).
- \(\Phi(t_i^-, t_i^+) = \text{state transition matrix associated with} F(a_i(t_i)) \text{of Equation (6), defined as the} \ 2n \times 2n \text{matrix that}
\]

satisfies \(\dot{\Phi}(t_i^-, t_i^+) = F(a_i(t_i)) \Phi(t_i^-) \Phi(t_i^-) \Phi(t_i^-) = I\).
- \(Q(t_i) = \int_{a_i^-}^{a_i^+} \Phi(t_i, \alpha) G(\tau(t_i) \alpha) G(\tau(t_i) \alpha)^T \dot{\alpha} d\alpha\)
- \(R(t_i)\) is the strength of the Gaussian noise, \(W(t_i)E[W(t_i)W^T(t_i + \tau)] = Q(t_i)\).
- \(V(t_i) = \text{the matrix of the Gaussian noise strength, } V(t_i): \ E[V(t_i)V^T(t_i)] = R(t_i)\).

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, \(\dot{r}(t_i) = z(t_i) - H(t_i) \dot{x}(t_i^-)\). 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)).

\[
p_j(t_i) \triangleq \text{prob}\{a_j | Z(t_i) = Z_i\} \quad (13) \\
p_j(t_i) = \frac{f(a_j, Z(t_i), x(t_i^-), p_j(t_i-1))}{\sum_{a_{i,j}} f(a_j, Z(t_i), x(t_i^-), p_j(t_i-1))} \quad (14)
\]

where:

- \(Z(t_i-1)\) is the measurement history up to time \(t_i-1\).
- \(f(a_j, Z(t_i), x(t_i^-), p_j(t_i-1))\) is the conditional probability that the \(j^{th}\) filter was correct. For the assumed Gaussian distribution it has the form \(\prod_{t=1}^{t_i} \frac{1}{\sqrt{2\pi} \sigma^2} e^{-(1/2)(x(t_i)^2/\sigma^2)}\)
- \(A(t_i) = [H(t_i)P(t_i^-) H(t_i) + R(t_i)]\)
- \(\text{the denominator scales the conditional probability such that} \sum_{j=1}^{K} p_j(t_i) = 1\)

The conditional mean of the parameter variation \(\Delta a\) at \(t_i\) is given by:

\[
\Delta a(t_i) \triangleq \sum_{j=1}^{K} a_j p_j(t_i)
\]

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 \(\Delta a\) is to be added to the current value of \(\dot{a}\) in the feed-forward element or subtracted from it as shown below:

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

where \(\dot{a}\) is the actual parameter estimate output from the \(\Delta\text{MAE}\) and \(f(r)\) is a degree of design freedom used to maximize performance for a specific application.

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

\[
N^T(t) = [D(q, \dot{a}) + N^2 M][\ddot{q} + 2\Omega^\omega \dot{q} + \omega_\omega^2 \dot{q}] + \dot{h}(q, \dot{a}) \quad (17)
\]

where:

- \(\dot{e} = \ddot{q} - q\), the joint position error vector
- \(\zeta = \text{diagonal matrix of desired damping ratios}\)
- \(\omega_\omega = \text{diagonal matrix of desired natural frequency}\)

The achilles heel 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, \(\dot{a}\) is a scalar and the \(\Delta\text{MAE}\) 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:

\[
N^T(t) = [D(q, \dot{a}) + N^2 M][\ddot{q} + h(q, \dot{a}) + N^2 B_{\omega}\dot{\Omega} + \Omega^\omega \Omega + g(q, \dot{a}) + [\dot{\Omega} e + \Omega e]] \quad (18)
\]
without degradation of PUMA trajectory tracking performance [15]. The \(K_p\) and \(K_d\) 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 AMMAE 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 \(q, \dot{q}, T, \tau\). A program using MACSYMA [10] commands was developed to provide a symbolically reduced set of equations for \(F'(aj, t)\) [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:

\[
F'(aj, t) = (F'(aj, t) - G(aj, t)K)
\]  

(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 \(G(aj, t)\) matrix which transforms the torque noise into the state space had the following form:

\[
G(aj, t) = 
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & (Dg(t), a) + N^2 M)^{-1} & desired
\end{bmatrix}
\]  

(20)

The only measurements available on the PUMA-660 are the actual joint positions. Therefore the only measurements input to the Kalman filters are the error in the position states. Since \(z(t)\) is a linear function of the position states:

\[
H(t) = 
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]  

(21)

The \(F'(aj, t)\) and \(G(aj, t)\) matrices are payload and trajectory dependent, but can be assumed constant over the sample period [27,13]. Therefore, \(\Phi(t, t-1)\) and \(Q(t)\) were accurately approximated by:

\[
\Phi(t, t-1) = I + F'(aj, t)\Delta t + 1/2 F''(aj, t)\Delta t^2
\]  

(22)

\[
Q(t) = G(t)Q^G(t)\Delta t
\]  

(23)

where \(\Delta t\) 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 AMMAE. Once \(Q(t)\) was selected, that value was held constant for all test trajectories. The initial conditions of \(Z(t)\) and \(P(t)\) 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, \(f(t(t_i))\), 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 AMMAE 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 \(\dot{a}\) was held constant. The difference between the two SMBC algorithms was in the constant \(\dot{a}\) value. Worst case model-based control performance was simulated by a version of the SMBC with \(\dot{a} = 0\). 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 \(\dot{a}\) value was initialized to zero. The performance of the MMBC didn't meet the ideal, but as Figure 3 illustrates that the AMMAE can very quickly provide an estimate of payload that will significantly reduce the tracking error. The AMMAE locked onto a payload estimate by 0.4 seconds, and the oscillations in the transient region have minimal effect on tracking performance. Examples of AMMAE 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 verse 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. MMBE error was always within one standard deviation (\(\sigma\)) 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 $\alpha$ 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 APT 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 [8]. 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 robust 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

Figure 1: ΔMMAE Block Diagram

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

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

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