'94 KACC (1994. 10. 17 ~ 20)

Adaptive Control for Robot Manipulators Executing Fine Motion Tasks

Vicente Parra-Vega * and Suguru Arimoto
Faculty of Engineering, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 166, Japan.

Abstract

A passivity-based adaptive controller for robots executing fine motion tasks is proposed. The robot dynamics is modelled such that it is subject to holonomic constraints and hence it can be treated as a particular case of constrained motion tasks. Energy-motivated stability analysis is used to conclude the asymptotic stability. Remarks regarding the structure of the controller are given. A computer simulations study is presented and a robust constraint stabilization algorithm is also proposed.

Keywords: Fine Robot Motion, Global Stability, Holonomic Systems, Adaptive Control, Passivity, Constraint Stabilization.

1 Introduction

Steadily the industrial needs are moving toward tasks that demand finer trajectory tracking capabilities. There are high precision tasks that demand almost perfect trajectory tracking such as optical inspection, surgical operation, space applications, etc. These kinds of tasks are termed fine motion tasks. Several approaches have been proposed which rely on the exact knowledge of system dynamics whose control objective is pure trajectory tracking of the end-effector.

In this paper a rather different approach is taken: the robot dynamics is treated as a dynamical system subject to an algebraic constraint ([1],[3]). This constrained manifold can be interpretated as a virtual boundary which the manipulator cannot transpass. This boundary can be seen as a growth surface of the enviroment and the control objective is trajectory tracking on this growth surface *and* stabilization of the associate Lagrange multiplier (the augmented system is balanced now by a Lagrange multiplier). Thus this problem can be treated as a particular case of constraint motion.

Recently a novel approach for hybrid motion for robot manipulators was proposed ([3],[4]). That scheme has potential applications since it controls the Lagrange multiplier (which in hybrid motion stands for the contact force). That scheme is a passivity-based model-based adaptive controller for robot manipulators under holonomic constraints. The underlying structure of that controller relies on a joint space decomposition in such a way that convergence of position tracking errors and boundedness of the Lagragian are guaranteed.

As byproducts of that formulation, this paper presentes a model-based adaptive controller for robot manipulators whose task is defined as tracking a desired trajectory above a surface of an object without colliding with it, *i.e.* fine motion task. For this kind of tasks, usually the end-effector is equipped with a tool (camera, painting gun, etc.) which makes neccesary the design of an adaptive controller to compensate inertial and gravity loads of the tool and those of the robot itself.

The dynamics of the robot manipulator is now subject to kinematic constraints such that an augmented model can describe precisely the robot dynamics constrained to lie onto a smooth invariant manifold defined by the growth constraint surface. In this situation a Lagragian multiplier arises in the right hand side of the robot dynamics. Contrary to the constrained motion case where the Lagragian multiplier stands for the contact force at the contact point, in fine motion tasks the Lagrange multiplier associated with the constraint does not have a physical meaning. It is the Lagrange multiplier in the calculus of variations. Using the gradient of the growth constraint surface, two transformations are derived in order to formulate a passivity-based adaptive controller[1]. The global convergence of tracking errors and the boundedness of the Lagrange multiplier are shown via energy-motivated stability arguments. Computer simulations show the performance of the controller and it is compared against the popular computer torque method and model-based adaptive control (Slotine and Li's).

*email:vega@arimotolab.t.u-tokyo.ac.jp

For simulations purposes, typical constraint stabilization is used to keep the system onto the constraint manifold via a simple PD controller that plays the role of a "numerical impedance". Here a sliding mode control term is proposed to carry out the robust constraint stabilization.

This paper is organized as follows: Section 2 describes the robot dynamics subject to holonomic constraints and states a key assumption. In Section 3 a global joint space decomposition is derived. Section 4 proposes an asymptotically stable adaptive controller which guarantees globally the convergence of tracking errors. Section 5 a robust sliding mode constraint stabilization algorithm is proposed. Section 6 shows computer simulations, and finally Section 7 presents some discussions and concluding remarks.

2 Robot Dynamics under Holonomic Constraints

Fine motion taks can be seen as constrained motion tasks if they are defined in such a way that the robot end-effector moves on a surface defined above the environment. It is assumed that a model of the surface of the object is available and this model is twice differentiable. Let

$$\Theta(x) = 0$$

denote the m−smooth surfaces $\Theta(x) : \mathcal{R}^3 \rightarrow \mathcal{R}^m$ where $X : [x_1, x_2, x_3]$ represents the cartesian coordinates. And hence, the end effector is to move on a growth or augmented surface $\Theta'(x) = \Theta(x) + \delta = 0$. In the following $\delta > 0$ is omitted since the augmented surface can be treated simply as a new surface. There exists a smooth function $f : \mathcal{R}^n \rightarrow \mathcal{R}^3$ such that $X = f(q)$ establishes the mapping between the cartesian space and the joint space $q = [q_1, \ldots, q_n]$. The smooth surface can now be expressed in terms of q as follows

$$\varphi(q) = 0 \tag{1}$$

where $\varphi(q) : \mathcal{R}^n \rightarrow \mathcal{R}^m$. Differentiating (1) yields

$$J_\varphi(q)\dot{q} = 0 \tag{2}$$

where

$$J_\varphi(q) = \frac{\partial \varphi}{\partial q} = \frac{\partial \varphi}{\partial x} J_r \tag{3}$$

where J_r denotes the standard Jacobian matrix of the robot manipulator. In joint space the velocity vector arises onto the space tangent at the contact point to that surface, and hence the columns of (3) span the space normal at the contact point. The model of a rigid serial n-link with all revolute joints described in joint coordinates is given as follows:

$$H\ddot{q} + (B_0 + \frac{1}{2}\dot{H} + \bar{S})\dot{q} + g = U \tag{4}$$

where $H = H(q)$ denotes the $n \times n$ symmetric positive definite inertial matrix, B_0 an $n \times n$ positive definite matrix of damping coeficients, $g = g(q)$ the gravity forces, $\bar{S} = \bar{S}(q, \dot{q})$ an $n \times n$ skew symmetric matrix, and U the torque input. For fine motion tasks the robot dynamics (4) is subject to an algebraic constraint [3] $\varphi(q) = 0$. Simply calculus of variations finally leads us to a constrained robot dynamics given by

$$H\ddot{q} + (B_0 + \frac{1}{2}\dot{H} + \bar{S})\dot{q} + g = U + J_\varphi^T \lambda \tag{5}$$

$$\varphi(q) = 0 \tag{6}$$

where λ plays a role of the Lagragian multiplier in the calculus of variations.

According to the task, it can be viewed that the robot dynamics are constrained to evolve on a 2n-dimensional invariant manifold

$$\mathcal{M}_0 = \{(q, \dot{q}) \in \mathcal{R}^n \times \mathcal{R}^n : \varphi(q) = 0, J_\varphi(q)\dot{q} = 0\} \tag{7}$$

This invariant manifold implies that (5) is singular in $2n$-space. It suggests that there exist a transformation that can reduce the degrees of freedom, actually the holonomic system (5)-(6) has n degrees of freedom and m holonomic constraints to be satisfied for all time. It in turn implies that m-dof can not change its initial configuration, and therefore there are $n - m$ position-controlled degrees of freedom and m "Lagrange multiplier"-controlled degrees of freedom.

In the next section, following the guidelines of the Orthogonalization Principle, we derive two transformations that effectively decompose the state space of tracking errors and finally enable us to state global results by using only linearly independent coordinates.

3 Joint Space Decomposition

We now outline the scheme proposed in [3]. The constraints (1) are considered to be independent and hence $J_\varphi(q)$ has full row rank. Now define a partition of the joint space coordinates q as follows:

$$q = [q_1^T \quad q_2^T] \tag{8}$$

where $q_1 \in \mathcal{R}^{n-m}$ and $q_2 \in \mathcal{R}^m$. According to the implicit function theorem there exists an open set

$\mathcal{O} \in \mathcal{R}^{n-m}$ and a function $\Omega : \mathcal{R}^{n-m} \rightarrow \mathcal{R}^{m}$ such that

$$\varphi(q_1, \Omega(q_1)) = 0 \qquad \forall q_1 \in \mathcal{O} \tag{9}$$

For the holonomic constraint (9) to be satisfied, it implies the existence of a unique solution

$$q_2 = \Omega(q_1) \tag{10}$$

such that (9) is fulfilled $\forall t$. In order to derive the explicit description of (10), consider (2) with its corresponding partition according to (8). Then (2) can be written in the following way:

$$J_\varphi^T \dot{q} = [J_{\varphi 1}(q) \quad J_{\varphi 2}(q)]^T [\dot{q}_1 \quad \dot{q}_2] = 0 \tag{11}$$

where $J_{\varphi 1} \in \mathcal{R}^{m \times (n-m)}$ and $J_{\varphi 2} \in \mathcal{R}^{m \times m}$. Solving (11) for $\dot{q}_2$ yields

$$\dot{q}_2 = -[J_{\varphi 2}(q)]^{-1} J_{\varphi 1}(q) \dot{q}_1 \tag{12}$$

Differentiating (10) with respect to t yields

$$\dot{q}_2 = \Omega_{q_1} \dot{q}_1 \tag{13}$$

where $\Omega_{q_1} = \frac{\partial \Omega(q_1)}{\partial q_1} = -[J_{\varphi 2}(q)]^{-1} J_{\varphi 1}(q) : \mathcal{R}^{n-m} \rightarrow \mathcal{R}^{m}$. According to the partition (8) and (12), the velocity of the generalized coordinates can be written

$$\dot{q} = Q \dot{q}_1 \tag{14}$$

where $Q = \begin{bmatrix} I_{n-m} \\ \Omega_{q1} \end{bmatrix} \in \mathcal{R}^{n \times (n-m)}$ is well posed (see [7] for details).

Remark 1.- The transformation $J_\varphi(q)$ spans the null space of Q, i.e., the n-state space is decomposed into two orthogonal subspaces such that $\mathcal{R}^n$ can be written as the direct sum $\mathcal{R}^n = \mathrm{R}(J_\varphi^T) \oplus \mathrm{R}(Q)$ where $\mathrm{R}(*)$ stands for the range of $(*)$.

In the next section, we rewrite (5) in terms of error space by using this joint space decomposition trasformations and by exploiting some fundamental properties of robot dynamics. Then we proceed to design an adaptive control law.

4 Adaptive Control for Fine Motion Tasks

Design of model-based adaptive controllers exploits some fundamental properties of robot dynamics: a) the left hand side of (5) can be written in linear dependence of unknown or uncertain parameters which may include the coefficient of the sliding friction forces, b) the skew symmetric property of the matrix $\bar{S}(q, \dot{q})$, and c) the passivity property from input torque to output velocity. In order to write robot dynamics (5) in the error space we now define a linear parametrization of the robot dynamics as follows:

$$H\ddot{q} + \left\{ B_0 + \frac{1}{2}\dot{H} + \bar{S} \right\} \dot{q} + g = Y\Theta \tag{15}$$

Equation (15) can be written in terms of a nominal reference $\dot{q}_r$ as follows:

$$H(q)\ddot{q}_r + \left\{ B_0 + \frac{1}{2}\dot{H}(q) + \bar{S}(q, \dot{q}) \right\} \dot{q}_r + g = Y_r\Theta \tag{16}$$

where $\dot{q}_r = Q\{\dot{q}_{1d} - \alpha\Delta q_1\}$ and $\alpha > 0$ with $\Delta q_1 = (q_1 - q_{1d})$. Adding (16) to both sides of (5) and using $S = \dot{q} - \dot{q}_r$, robot dynamics in the S space is given as follows:

$$H\dot{S} + \left\{ B_0 + \frac{1}{2}\dot{H} + \bar{S} \right\} S = U + J_\varphi^T \lambda - Y_r\Theta. \tag{17}$$

Since $\dot{q} = Q\dot{q}_1$ sliding surface S can be written $S = Q\{\Delta\dot{q}_1 + \alpha\Delta q_1\}$. The last equation is fundamental in the problem: it projects a weighted sum of tracking errors onto the space tangent to the growth surface. Thus, tracking errors lie onto a parallel surface to the real surface defined by the environment. In this way the endpoint avoids colliding against the enviroment. At this stage the problem becomes: Design a controller U to assure the asymptotic stability of (17). To this end, we consider the following control law:

$$U = Y_r\hat{\Theta} - K_d S - J_\varphi^T \lambda_d \tag{18}$$

$$\dot{\hat{\Theta}} = -\Gamma Y_r^T S \tag{19}$$

where $K_d = K_d^T > 0$ is a diagonal $(n \times n)$ matrix, $\Gamma = \Gamma^T > 0$ is a diagonal $(p \times p)$ matrix and $\hat{\Theta}$ stands for the estimate of Θ at time t. We are now in a position to state the main result in a theorem.

Theorem 1 *Consider robot dynamics (5) in closed loop with the adaptive controller (18)-(19). The global asymptotic stability of tracking errors is assured with bounded λ and bounded parameter estimates in the $\mathcal{L}_\infty$-sense, i.e.,*

$$q(t) \rightarrow q_d(t) \quad \textit{and} \quad (\lambda(t), \hat{\Theta}) \in \mathcal{L}_\infty$$

as $t \rightarrow \infty$.

Proof.- First consider the Lyapunov function

$$V = \frac{1}{2}\left\{ S^T H S + \Delta\Theta^T \Gamma^{-1} \Delta\Theta + \Delta q_1^T P \Delta q_1 \right\} \tag{20}$$

Computing the total derivative of (20) along with (18) yields

$$\dot{V} = -S^T K S + S^T J_\varphi^T \Delta\lambda + \Delta q_1^T P \Delta \dot{q}_1 \qquad (21)$$

where $K = (K_d + B_0) > 0 \in \mathcal{R}^{n \times n}$ is a diagonal matrix and $\Delta\lambda = \lambda - \lambda_d$. According to the orthogonality between Q and J_φ^T and if we chose $P = 2K\alpha$, we obtain.

$$\dot{V} = -\Delta q_1^T \alpha^T K_1 \alpha \Delta q_1 - \Delta \dot{q}_1^T K_1 \Delta \dot{q}_1 - S_p^T R^T K_2 R S_p \qquad (22)$$

where $K_1 \in \mathcal{R}^{(n-m)\times(n-m)}$ and $K_2 \in \mathcal{R}^{m \times m}$ stands for the partition of K, $R = -J_{\varphi 2}^{-1} J_{\varphi 1}$, and $S_p = \Delta \dot{q}_1 + \alpha \Delta q_1$. Applying Barbalat's Lemma and assuming that $q_d \in C^3$ and $\lambda_d \in C^1$, it turns out the global convergence of position and velocity tracking errors can be concluded [5]:

$$q(t) \to q_d(t) \quad \text{and} \quad \dot{q}(t) \to \dot{q}_d(t) \quad \text{as} \quad t \to \infty \qquad (23)$$

Boundedness of Θ is immediately concluded from (21) and (22). The boundedness of λ is more involved. Similarly to [6], if the desired trajectories are bounded and $\ddot{q}_d$ is uniformly continuous all over t, then it can be shown from (17) and (18) that in the limit

$$Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\Delta\theta = J_\varphi^T \Delta\lambda \qquad (24)$$

Equations (20) and (22) in the other hand imply the convergence of the state trajectories $(q, \dot{q})$ and the boundedness of $\Delta\Theta$. Thus since $q \in C^3$, the left hand side of eq.(24) is bounded. Since the constraint surface $\varphi(q) = 0$ is assumed to be C^2, and $J_\varphi^T(q)$ full column rank, then the boundedness of the Lagragian λ can be concluded, namely

$$\Delta\lambda(t) \in \mathcal{L}_\infty. \qquad (25)$$

Remark 2.- Since the true parameters Θ are unknown, computation of λ becomes an issue to discuss. λ can be obtained if the second derivative of the constrained surface $\varphi(q, \dot{q}, u, \Theta, \lambda, t) = 0$ is solved for λ by using the best guess of Θ. In this case what we obtain by solving $\varphi(q, \dot{q}, u, \Theta, \lambda, t) = 0$ is $\tilde{\lambda}$ instead of λ where $\tilde{\lambda} = \lambda - \delta_\lambda$ for $|\delta_\lambda| > 0$. Note that λ_d can be written as $\lambda_d = \lambda_{d1} + \delta_\lambda$, thus $\Delta\lambda = \lambda - \lambda_{d1}$.

Remark 3.- Notice that there is no need to know λ_{d1} nor δ_λ. The best guess of Θ can be obtained by running several times the controller for free motion (Slotine and Li's approach). Experimental data indicates that a consistent envelope of parameters Θ can be obtained by using $a * sin(wt)$ as the desired trajetories with different amplitude a and differents frequencies w. After we have obtained such estimates it is still advisable to keep the adaptive control loop in the controller because adaptive controllers outperform its nonadaptive version ([8]).

5 Robust Constraint Stabilization via Sliding Modes

The augmented robot dynamics is in fact a set of n overdetermined diferential equations since the state must satisfy an algebraic constraint for all time. This equation behaves as a very stiff system and several pathological behaviors are identified trying to solve DAE via a numerical method for ODEs. Thus a DAS's numerical integration scheme must be used to carry out simulations

Solving DAEs restricts the solution onto the solution manifold (7) rather than in the flat Euclidian space. Therefore it requires to correct the solution produced by the integration method. In order to avoid the real solution to drift far off the solution manifold $\mathcal{M}_0$, we briefly review the succesful "Baumgarte's method" which makes the solution manifold a stable attractor by adding a PD-type controller to the kinematic constraint at the acceleration level,

$$\ddot{\varphi}(q) = 0 \Longrightarrow \ddot{\varphi}(q) = u_{cs} \qquad (26)$$

It bounds the numerical error with respect to the constraint equations. When the constraints are not satisfied, the PD-type additional control term u_{cs} acts as a spring-damper such that it pulls the solution back onto $\mathcal{M}_0$. The controller is given by

$$u_{cs} = -(a+b)\dot{\varphi}(q) - ab\varphi(q) \qquad (27)$$

where $a, b >> 0$. Substituting (27) into (26) we obtain

$$\ddot{\varphi}(q) + (a+b)\dot{\varphi}(q) + ab\varphi(q) = 0 \qquad (28)$$

This method removes the double eigenvalue of the second derivative of (26) to the left side of the complex plane such that it modifies the stiffness of the closed loop system. Inside the closed loop there exist an inner control loop for constraints stabilization purpose that affects the performance of the closed loop system. The resulting constraint force (the Lagrange multiplier) does not strictly satisfy $\varphi(q) = 0$ but $\varphi(q) = \epsilon$, $\epsilon \neq 0$ such that the closed loop system is stabilized. The direction of the acceleration changes slightly so as to satisfy the position and velocity constraints. This simple method avoids the nonlinear error propagation such that the orthogonality of Q and $J_\varphi^T(q)$ still holds.

Other types of control for contraints stabilization purposes can be employed such as PID-type or even nonlinear or adaptive; it depends on the particular numerical method chosen so as to the stability region of (28) lies on the stability region of chosen integration method. Since we do not know Θ (see remark 2) a robust constraint stabilization algorithm is required

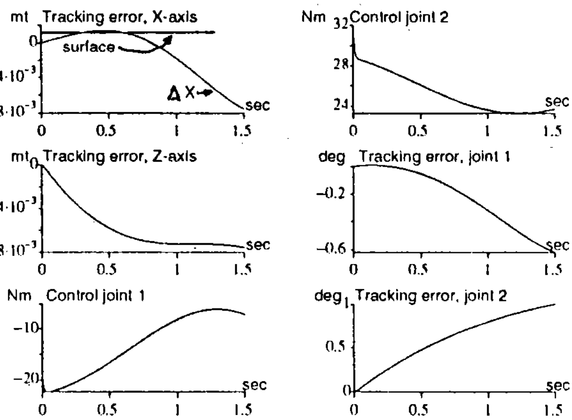

Figure 1: Computer torque control performance

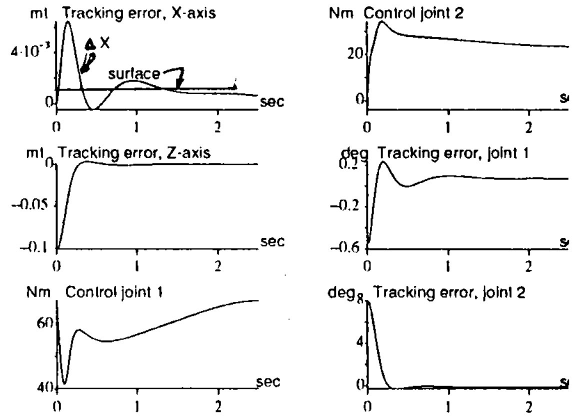

Figure 2: Slotine and Li's control performance

to cope with the parametric uncertainty in (refeq-p1). To this end, a sliding mode constraint stabilization controller is proposed in this section. Let $S_{cs} = \dot{\varphi}(q) + \nu\varphi(q)$ be a sliding surface, for $\nu > 0$. The purpose is to induce a sliding regime onto $S_{cs} = 0$ despite of the uncertainty on Θ (in (27) we are using the best guess of Θ which differs from the real Θ). To this end consider a sliding mode control law given by

$$u_{cs} = -a\dot{\varphi}(q) - bsgn(S_{cs}) \qquad (29)$$

it can easily be proved that such controller fulfills the condition for the existence of a sliding mode around $S_{cs} = 0$. Equation (29) is discontinuous and Discontinuities in DAEs are very critical from the numerical integration viewpoint. DAEs solver must accomodate such discontinuities without cutting step size every time encountered because if time step h is reduced the jacobian matrix (if Newton iteration is used) becomes ill-conditioned, the round-off error makes it imposible to solve. To overcome this problem a simple saturation function is used instead of the discontinuous fuction $sign(*)$.

We tested in a computer simulation study this kind of constraint stabilization controller and we compared against several others well stablished constraint stabilization controller (PD, PID and adaptive); this computer simulations study reports the superior performance of (28). Plots are omitted.

6 Computer Simulations

Computer simulations on a simple two DOF robot manipulator acting in the vertical plane $X - Z$. Suppose that the fine motion task is to slide the end-

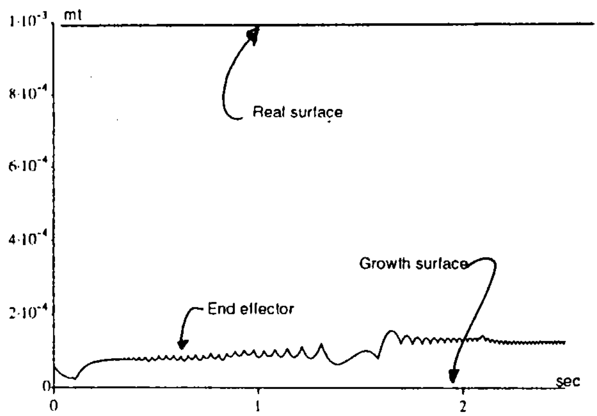

Figure 3: Tracking error in Z axis

effector tool (a camera, for instance) in front of a surface (a mirror, for instance) with a precision within 1 mm without colliding with the surface. The surface is placed parallel to the Z-axis. The end-effector travels $0.16mt/s$. The initial configuration is $\Delta Z = -0.1mt$ and $\Delta X = -0.001mt$ (the end-effector is onto the growth surface). Mass and inertial parameters of both links were considered to be uncertain with -10 % error. The control gains were chosen as follows: $\Gamma = 5I$ $K_d = (50, 10)$, and $\alpha = 10$. Link parameters are $l_1 = 1mt$, $l_2 = 0.8mt$, $l_{c1} = 0.42mt$, $l_{c2} = 0.57$, $m1 = 10kg$, $m2 = 7kg$, $I_1 = 0.2Kgm^2$, and $I_2 = 0.1Kgm^2$. There is not an intuitive way to choose λ_d, in this simulation we chose $\lambda_d = 10N$. The boundary layer in the sliding mode constraint stabilization algorithm is 0.01 and parameters of eq.(28) were $a = 10$, $b = 50$ and $\nu = 10$.

All controllers were simulated under the same conditions of -10% on the knowledge of the parameters

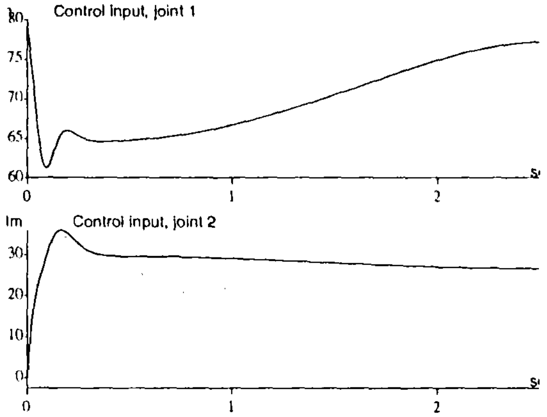

Figure 4: Control input

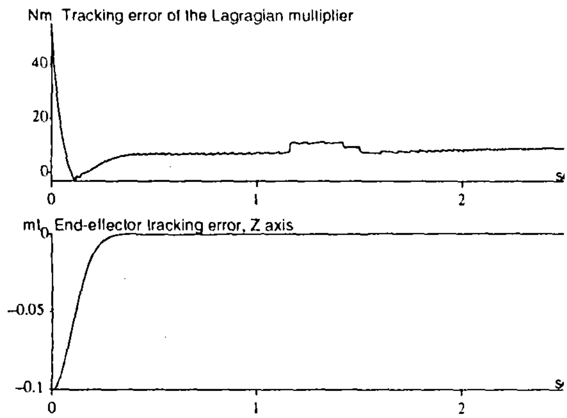

Figure 5: Tracking errors

Θ. As it can be seen in Fig.1(a) computer torque controller (poles placed at -240 and -250) makes the end-effector collide at $t = 0.3sec$. Fig.2(a) shows the performance of Slotine and Li's controller where it can be appreciated that during transient response the end-effector also collides with the enviroment at $t = 0.1sec$. In contrast Fig.3 shows that our proposed controller attains the required tracking precision onto the enviroment with smooth controllers. It is interesting to note that the proposed robust constraint stabilization algorithm proposed in Section 5 worked for any simulated conditions while Bumargarte's method failed under some conditions (using both equivalent set of parameters).

7 Discussion and conclusions

An asymptotically stable adaptive controller for fine motion tasks for robot manipulators was proposed. It was also proposed a robust constraint stabilization algorithm to cope with parametric uncertainties in the manifold defined by the augmented surface. A comparative computer simulation study reports the performance of the proposed controller against the computer torque method and Slotine and Li's.

References

[1] Dooley, J. R., (1994) "A Strategy for Fine Motion Manipulation of robots Using Kinematic Constraints", *The Proc. of the IEEE Robotics and Automation Conf.*, Sn. Diego, pp. 656-661.

[2] Slotine, J.J. and Li, W., (1987) "On the adaptive control of manipulators," *Int. Journal of Robotics Research,"* Vol.6, pp.49-59.

[3] Parra-Vega, V., Arimoto, S., Liu Y.H. and Naniwa, T., (1994) "Model-Based Adaptive Hybrid Control of Robot Manipulators under Holonomic Constraints," Proceedings of The SY.RO.CO. 1994, Italy.

[4] Arimoto, S., Liu, Y.H. and Naniwa, T., 'Model-Based Adaptive Hybrid Control for Geometrically Constrained Robots,' *Proc. of the 1993 IEEE Int. Conf. of Robotics and Automation*, 618-623 (1993).

[5] Narendra K. and Annaswamy A., Stable Adaptive Systems, Prentice Hall, 1989.

[6] Sadeg, N and Horowitz, R. , (1989) "Stability and Robustness of Adaptive of Adaptive Controllers for Robotics Manipulators," *Int. Journal of Robotics Research*, Vol.9, No.3.

[7] Blajer, W., Schielhen, W. and Schirm, W., (1994) "A Projective Criterion to the Coordinate Partitiong Method for Multibody Dynamics", *Archive of Applied Mechanics*, Vol. 64., pp.86-98.

[8] Whitcomb, L.W., Rizzi, A. and Koditschek, (1993) "Comparative Experiments with a New Adaptive Controller for Robot Arms", *Trans. on Robotics and Automation*, Vol.9, pp.59-70.