An Adaptive Control Scheme for Robots with Unknown Dynamics

Q.-H. Max Meng
Department of Electrical Engineering
Lakehead University
Thunder Bay, Ontario
Canada P7B 5E1

W.-S. Lu
Dept. of Electrical & Computer Engineering
University of Victoria
Victoria, B.C.
Canada V8W 3P6

Abstract – In this paper, a stable adaptive control scheme for robot manipulators with unknown dynamics is proposed. It consists of an off-line least-mean-square (LMS) type identifier to identify structured system dynamics and an on-line dynamic compensator to compensating for dynamic uncertainties. Taking advantage of the unique structure of the robot regressor dynamics, the former uses a LMS type algorithm to identify, using a set of trial data, the structured dynamic parameters of the robot while the latter uses an on-line stable parameter updating mechanism determined using Lyapunov theory to compensate for both unknown and uncertain dynamics. The off-line identified parameters are used as initial values for the on-line dynamic parameter estimation. Since both identifier and compensator are implemented using the regressor dynamics, the recursive formula for the computation of robot regressor dynamics previously proposed can be used to achieve high computational efficiency in real-time implementations. An illustrative simulation example is included to show the proposed adaptive control algorithm.

INTRODUCTION

Vast majority of industrial robots are controlled by using classic PID type regulators. PID controllers are robust to dynamic uncertainties since they do not use dynamic models in their algorithms. The computed torque control algorithm, on the other hand, generally provides better performance than a PID control algorithm, but it requires the exact dynamic model of the controlled robot to achieve this performance. The fact is that dynamic models of commercial robots are normally not available and the dynamic models of industrial robots in applications are changing with the changing robot loads and various disturbances. Therefore, neither one will provide a satisfactory control algorithm for robot applications where precision control algorithms with high performance are essential to the successful accomplishment of a task, such as space robot applications in precision assembly type of tasks. To overcome these disadvantages associated with both PID and computed torque control algorithms, model based adaptive control algorithms have been extensively studied and several stable adaptive control schemes have been proved to be effective in compensating unknown/uncertain dynamics of the controlled robot [1-5]. The previously proposed adaptive control algorithms are generally theoretically attractive but computationally not so efficient.

In this paper, a stable adaptive control scheme for robot manipulators with unknown dynamics is proposed. It consists of an off-line least-mean-square (LMS) type identifier to identify structured system dynamics and an on-line dynamic compensator to compensating for both structured and unstructured dynamic uncertainties. The former takes advantage of the unique structure of the robot regressor dynamics and uses a LMS type algorithm to precisely identify, using a set of trial data, the structured dynamic parameters of the robot, which are contained in the parameter vector of the regressor dynamics. The latter uses an on-line stable parameter updating mechanism determined using Lyapunov theory to compensate for both unknown and uncertain dynamics, with the off-line identified parameters as initial values of the dynamic parameters. The off-line identification process provides a precise robot dynamic model as a starting point while the on-line compensator updates the dynamic parameter to reflect the changing dynamics introduced by, for example, varying loads or disturbances. Since both identifier and compensator are implemented using the regressor dynamics, the recursive formula for the computation of robot regressor dynamics presented in [6] can be used to achieve high computational efficiency in real-time implementation of the proposed control algorithm. An illustrative simulation example is included to show the proposed adaptive control algorithm.

Robot Dynamics

The dynamics of an n degree-of-freedom (d.o.f.) robot manipulator can be described by

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{\tau}$ is the $n \times 1$ vector of the joint torque; $\mathbf{q}$ is the $n \times 1$ joint displacement vector; $\mathbf{H}(\mathbf{q})$ is the $n \times n$ mass matrix; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ represent torques due to centrifugal and gravity forces respectively. Two of the important properties of the dynamics (1) are as follows.

Property 1 All robot dynamics parameters such as link masses, centers of masses, etc. appear as coefficients of known functions of $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$, i.e., eqn. (1) can be written as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta} = \boldsymbol{\tau} \tag{2}$$

where $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is an $n \times r$ matrix of known functions which is referred to as the *manipulator regressor*, and $\boldsymbol{\theta}$ is an $r \times 1$ vector containing both known and unknown dynamics parameters.

Property 2 Matrix $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is skew symmetric, i.e., it satisfies

$$\mathbf{x}^T\mathbf{S}\mathbf{x} = \mathbf{x}^T(\dot{\mathbf{H}} - 2\mathbf{C})\mathbf{x} = 0.$$

The above two properties are crucial to the development and stability proof of the proposed control scheme.

Off-Line Identification Algorithm

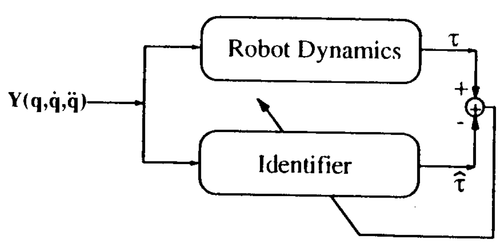

Fig. 1 Off-Line Parameter Identification Structure

The regressor dynamics defined in (2) is

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta} \tag{3}$$

or

$$\tau_i = \sum_{j=1}^{r} y_{ij}\theta_j \qquad \text{for } i = 1, \cdots, n \tag{4}$$

where τ_i is the i-th element of $\boldsymbol{\tau}$, θ_j is the j-th element of $\boldsymbol{\theta}$, and y_{ij} is the (i,j)-th element of $\mathbf{Y}$.

An off-line identification process, using a set of trial data of the robot, as shown in Fig. 1 is used to identify vector $\boldsymbol{\theta}$. The known regressor $\mathbf{Y}$ can be obtained by using the formula presented in [6].

The least-mean-square (LMS) algorithm is used in the off-line identification process. It makes the identified parameters converge to a set of parameters that minimizes the mean-square error

$$\mathcal{J} = \mathrm{E}(\|\boldsymbol{\tau} - \hat{\boldsymbol{\tau}}\|^2)$$

where $\boldsymbol{\tau}$ is the torque vector from the set of trial data of the robot dynamics and $\hat{\boldsymbol{\tau}}$ is the output torque vector of the identifier. By applying gradient descent method, the algorithm to update $\boldsymbol{\theta}$ to minimize $\mathcal{J}$ turns out to be

$$\theta_{j,new} = \theta_{j,old} + 2\mu\delta y_{ij} \qquad \text{and} \qquad \delta = (\tau_i - \hat{\tau}_i)\psi_i$$

where ψ_i is the gradient, or an approximation of it, with respect to θ_j of $\hat{\tau}_i$. The constant μ is chosen to adjust the speed of convergence and stability of the parameters during identification.

On-Line Control Algorithms

Consider the equation of motion (1). Define $\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \boldsymbol{\Lambda}\tilde{\mathbf{q}}$, where $\dot{\mathbf{q}}_d$ is the desired velocity, $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$, and $\boldsymbol{\Lambda} > \mathbf{0}$, and the control torque $\boldsymbol{\tau}$ is assigned as

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}} - \mathbf{K}\mathbf{s} \tag{5}$$

with $\mathbf{s} = \dot{\mathbf{q}} - \mathbf{q}_r$ and $\hat{\boldsymbol{\theta}}$ is an estimate of $\boldsymbol{\theta}$.

The first term of control (5) can be written as

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\bar{\boldsymbol{\theta}} + \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\Delta\boldsymbol{\theta}. \tag{6}$$

The first term on the right-hand side of eqn. (6) can be replaced by the regressor $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)$ and the parameter vector $\bar{\boldsymbol{\theta}}$, which is obtained through the off-line identification process using a set of trial data of the robot as presented in previous section.

The second term on the right-hand side of eqn. (6) will be implemented using the following on-line adaptive control algorithm.

If we choose $\boldsymbol{\Lambda} = \lambda\mathbf{I}$, $\mathbf{K} = \lambda\hat{\mathbf{H}}(\mathbf{q})$ and $\hat{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}} + \Delta\boldsymbol{\theta}$ with $\bar{\boldsymbol{\theta}}$ as an initial estimate and $\Delta\boldsymbol{\theta}$ as a small variation of $\hat{\boldsymbol{\theta}}$, the error dynamics of the system becomes

$$\ddot{\tilde{\mathbf{q}}} + 2\lambda\dot{\tilde{\mathbf{q}}} + \lambda^2\tilde{\mathbf{q}} = \hat{\mathbf{H}}^{-1}\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}} \tag{7}$$

where $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ and $\dot{\tilde{\boldsymbol{\theta}}} = \dot{\hat{\boldsymbol{\theta}}} = \Delta\dot{\boldsymbol{\theta}}$. If the estimate is updated according to

$$\Delta\dot{\boldsymbol{\theta}} = -\boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\mathbf{s}, \qquad \boldsymbol{\Gamma} > 0, \tag{8}$$

using property 2 of (1), the Lyapunov function candidate

$$\mathcal{V} = \frac{1}{2}(\mathbf{s}^T \mathbf{H}\mathbf{s} + \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\theta}})$$

and its time derivative along trajectories of (1) as

$$\dot{\mathcal{V}} = -\mathbf{s}^T(\mathbf{K} - \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}))\mathbf{s}$$

with symmetric matrix $\bar{\mathbf{C}} = (\mathbf{C} + \mathbf{C}^T)/2$ guarantee the position and velocity tracking errors converge to zero. This can also be concluded from eqn. (7) since the estimate update law (8) will drive $\Delta\boldsymbol{\theta}$ to $\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}$ ($\dot{\boldsymbol{\theta}} = 0$) and therefore $\dot{\tilde{\mathbf{q}}}$ and $\tilde{\mathbf{q}}$ approach to zero.

Simulation and Comparison

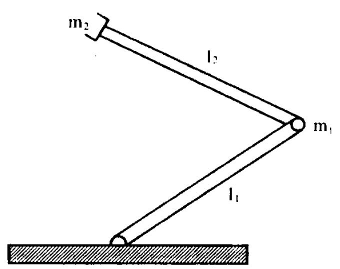

Fig. 2 A Two Degree-of-Freedom Planar Robot

A two d.o.f. planar robot as shown in Fig. 2 is used as an example system to illustrate the proposed scheme. In this simulation, it is assumed that we have no knowledge of the robot dynamic parameters at all and that the mass of each link is a point mass located at the distal end of the link. By using the formula for regressor dynamics presented in [6], the regressor $\mathbf{Y}$ of the robot is found as

$$\mathbf{Y} = \begin{bmatrix} \begin{array}{c} c_2(\ddot{q}_1 + \ddot{q}_2) \\ -s_2\dot{q}_2(\dot{q}_2 + 2\dot{q}_1) \end{array} & \ddot{q}_1 + \ddot{q}_2 & gc_{12} & gc_1 & \ddot{q}_1 \\ & & & & \\ c_2\ddot{q}_1 + s_2\dot{q}_1^2 & \ddot{q}_1 + \ddot{q}_2 & gc_{12} & 0 & 0 \end{bmatrix}$$

where $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $c_{12} = \cos(q_1 + q_2)$, and g is the gravitational acceleration constant. Parameter vector $\boldsymbol{\theta}$ is then

$$\boldsymbol{\theta}^T = [m_2 l_1 l_2 \;\; m_2 l_2^2 \;\; m_2 l_2 \;\; (m_1 + m_2) l_1 \;\; (m_1 + m_2) l_1^2]$$

The true values of the link masses and lengths are m_1=2 kg, m_2=1 kg, l_1=1 m, and l_2=1 m. Therefore the corresponding true parameter vector is

$$\boldsymbol{\theta}^T = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 \end{bmatrix}$$

In the simulation study, first the off-line identification process of Fig.1 is performed where a small set of random number is used to start the process. After using a set of trial data of the robot dynamics with 200 sampling points for 5 times (200 × 5 iterations), the identified parameter vector $\bar{\boldsymbol{\theta}}$ converges to the following values

$$\boldsymbol{\theta}^T = \begin{bmatrix} 1.0673 & 0.9262 & 1.0035 & 3.0127 & 2.9411 \end{bmatrix}$$

and its convergence profile is depicted in Fig.3. This set of values is then used as an initial estimate of the parameter vector and the control (5) and (6) is then applied to the robot to track the circle in as shown in Fig.4, which depicts the tracking performance of the proposed control scheme. This tracking performance also shows the ability of the proposed control algorithm in dealing with unstructured dynamics uncertainties since an unmodeled dynamics term, the Coulomb friction term, $\mathbf{f} = [1.75 \cdot \mathrm{sgn}(\dot{q}_1) \;\; 1.4 \cdot \mathrm{sgn}(\dot{q}_2)]^T$. is added to the robot dynamics but is not modeled in the control algorithm. Fig.5 shows the tracking errors in joint 1 and 2 by using the proposed controller. Fig.6 shows the profile of the parameter vector variation $\Delta\boldsymbol{\theta}$. Fig.7 shows the tracking performance of the robot using the modified Slotine-Li adaptive algorithm. Fig.8 shows the tracking errors in joint 1 and 2 by using the MSL adaptive controller. Fig.9 shows the parameter identification of the MSL adaptive controller, which is not as good as the identification of the off-line identifier probably because too many parameters (five) are identified and the input signals are not rich enough. Fig.10 shows the Cartesian space tracking error comparison of the two control methods with the error defined as

$$e(t) = \|\mathbf{x}_d(t) - \mathbf{x}(t)\|$$

where $\mathbf{x}_d$ is the desired position and $\mathbf{x}$ is the actual position of the robot tip in Cartesian space.

References

[1] M. W. Spong, F. L. Lewis, and C. T. Abdallah, Editors, *Robot Control: Dynamics, Motion Planning, and Analysis*, IEEE Press, 1993.

[2] T. C. Hsia, "Adaptive control of robot manipulators - a review," in *Proc. of IEEE Int. Conf. Robotics and Automation*, pp. 183-189, 1986.

[3] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: a tutorial," in *Proc. of the 27th Conf. on Decision and Control*, pp. 1575-1584, 1988.

[4] J. E. Slotine and W. Li, "Adaptive manipulator control: a case study," *IEEE Trans. on Automatic Control*, vol. 33, no. 11, pp. 995-1003, 1988.

[5] J. J. Craig, *Adaptive Control of Mechanical Manipulators*. Reading, MA: Addison-Wesley, 1988.

[6] W.-S. Lu and Q.-H. Meng, "Regressor formulation of robot dynamics: computation and applications," to appear in *IEEE Trans. Robotics and Automation*, 1993.

Fig.3 Parameter Convergence Profile

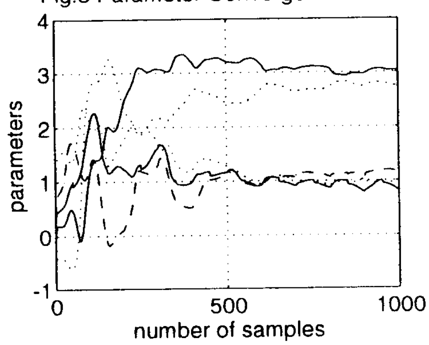

Fig.4 Tracking of Proposed Controller

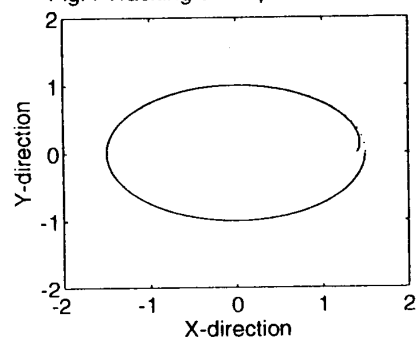

Fig.5 Tracking Errors in Joint Space

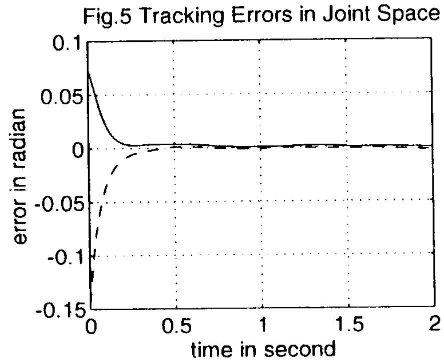

Fig.6 Delta-Theta Updating Profile

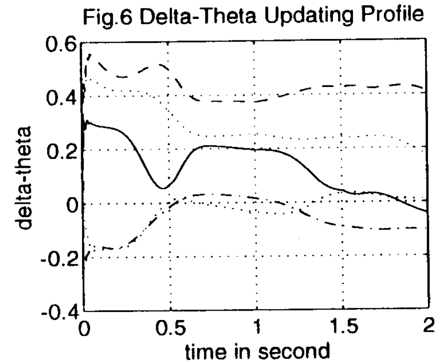

Fig.7 Tracking of MSL Controller

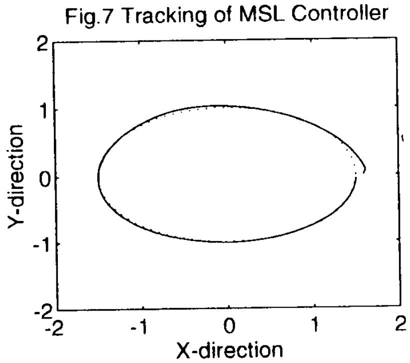

Fig.8 MSL Tracking Errors in Joint Space

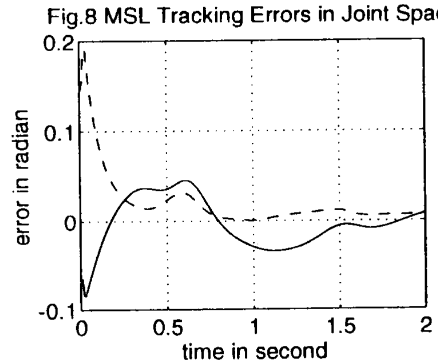

Fig.9 MSL Parameter Estimation Profile

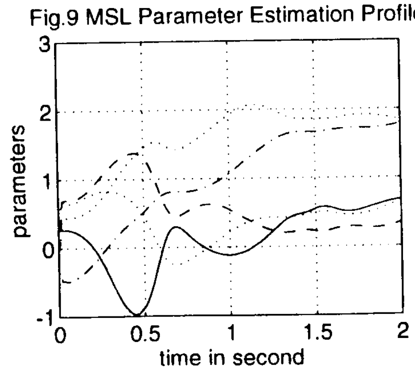

Fig. 10 Cartesian Tracking Error Comparison