Available online at www.sciencedirect.com
ScienceDirect
Automatica 43 (2007) 1226–1233
automatica
www.elsevier.com/locate/automatica

Brief paper

Modified transpose Jacobian control of robotic systems☆

S. Ali A. Moosavian[a,*], Evangelos Papadopoulos[b]

[a]Department of Mechanical Engineering, K. N. Toossi University of Technology, P.O. Box 16765-3381, Tehran, Iran
[b]Department of Mechanical Engineering, National Technical University of Athens, 15780 Athens, Greece

Received 7 August 2005; received in revised form 17 August 2006; accepted 12 December 2006
Available online 16 May 2007

Abstract

The simplicity of Transpose Jacobian (TJ) control is a significant characteristic of this algorithm for controlling robotic manipulators. Nevertheless, a poor performance may result in tracking of fast trajectories, since it is not dynamics-based. Use of high gains can deteriorate performance seriously in the presence of feedback measurement noise. Another drawback is that there is no prescribed method of selecting its control gains. In this paper, based on feedback linearization approach a Modified TJ (MTJ) algorithm is presented which employs stored data of the control command in the previous time step, as a learning tool to yield improved performance. The gains of this new algorithm can be selected systematically, and do not need to be large, hence the noise rejection characteristics of the algorithm are improved. Based on Lyapunov's theorems, it is shown that both the standard and the MTJ algorithms are asymptotically stable. Analysis of the required computational effort reveals the efficiency of the proposed MTJ law compared to the Model-based algorithms. Simulation results are presented which compare tracking performance of the MTJ algorithm to that of the TJ and Model-Based algorithms in various tasks. Results of these simulations show that performance of the new MTJ algorithm is comparable to that of Computed Torque algorithms, without requiring a priori knowledge of plant dynamics, and with reduced computational burden. Therefore, the proposed algorithm is well suited to most industrial applications where simple efficient algorithms are more appropriate than complicated theoretical ones with massive computational burden.

Keywords: Robotics; Control; Path tracking; Transpose Jacobian algorithm; Stability

1. Introduction

Many approaches have been employed for the complex problem of controlling mechanical manipulators and robotic systems. A prime difficulty for all approaches is due to the strong non-linearities and time dependencies in the dynamics of such systems. Hence, a wide range of algorithms has been suggested to challenge this task, including Adaptive Control algorithms as proposed by Slotine and Li (1987), and Taira, Sagara, and Katoh (2000), Time-Delay Control as suggested by Youcef-Toumi and Ito (1987), Motion-rate Control as presented by Umetani and Yoshida (1989), Kelly and Moreno (2005), Artificial Neural Networks and Fuzzy Control as proposed by Meghdari, Naderi, and Alam (2005), Dominguez-Lopez, Damper, Crowder, and Harris (2004), Steil, Röthling, Haschke, and Ritter (2004), Mbede et al. (2005), Hybrid Motion/Force Control as suggested by Raibert and Craig (1981), Chiu, Lian, and Wu (2004), and Impedance Control as proposed by Hogan (1985), and Moosavian, Rastegari, and Papadopoulos (2005).

Transposed Jacobian (TJ) control is one of the simplest algorithms used to control motion of robotic manipulators. According to Craig (1989), the TJ algorithm has been arrived at intuitively, and is similar to classic PD-action controllers. In the case of using an approximate Jacobian, Miyazaki, Masutani, and Arimoto (1988) have shown that the damping matrix and the position gain matrix of this controller play an important role in system stability. Apparently, the algorithm can be applied to redundant manipulators as shown by Asari, Sato, Yoshimi, and Tatsuno (1993), and as discussed by Chiaverini, Sciavicco,

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Masaki Yamakita under the direction of Editor Mituhiko Araki.

* Corresponding author. Fax: +98 21 7733 4338.
E-mail addresses: moosavian@kntu.ac.ir (S.A.A. Moosavian), egpapado@central.ntua.gr (E. Papadopoulos).

doi:10.1016/j.automatica.2006.12.029

and Siciliano (1991) it does not fail when a singularity occurs. Hootsmans and Dubowsky (1991) have developed an extended Jacobian transpose control algorithm to improve the performance of mobile manipulator systems. Subsequently, to fulfill simplicity requirements, Bevly, Dubowsky, and Mavroidis (2000) have developed Simplified Cartesian Computed Torque (SCCT) Control algorithm for highly geared climbing robots.

The performance of TJ-based algorithms has been experimentally compared to those of different algorithms using unit quaternions on a direct-drive spherical wrist by Garcia and Kelly (2002). Papadopoulos and Moosavian (1995) have compared the performance of this simple algorithm to those of various model-based algorithms. Both experimental and simulation results show the merits of the TJ algorithm in controlling of highly non-linear and complex systems with multiple degrees-of freedom (DOF), motivating further work on this algorithm. However, since the TJ is not dynamics-based, poor performance may result in fast trajectory tracking. Use of high gains can deteriorate performance seriously in the presence of feedback measurement noise. Another drawback is that there is no formal method of selecting its control gains, and a heuristic selection of gains makes it difficult to apply.

In this paper, a new Modified Transpose Jacobian (MTJ) algorithm is developed which employs stored data of the control command in the previous time step, as a learning tool to yield an improved performance. The gains of this new algorithm can be selected more systematically, and do not need to be large, hence the noise rejection characteristics of the algorithm are improved. Stability analysis, based on Lyapunov's theorems, shows that both the standard TJ and the MTJ algorithms are asymptotically stable. Simulation results show that tracking performance of this new algorithm is comparable to that of Model-Based (MB) algorithms, without requiring a priori knowledge of plant dynamics, and with reduced computational burden.

2. General motion control laws

Availability of a system dynamics is always helpful in the design of a control system. Using the expressions for the kinetic and potential energy, and applying Lagrange's equations for a robotic system, the dynamics model can be obtained and has the form

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

where all gravity and non-linear velocity terms are contained in vector $\mathbf{C}$, and $\mathbf{H}$ is a positive definite matrix, function of the generalized coordinates $\mathbf{q}$. Gravity terms that cause static positioning errors in control, can be compensated separately. Therefore, it is usually assumed that the vector $\mathbf{C}$ contains only non-linear velocity terms.

The output speeds, $\dot{\hat{\mathbf{q}}}$, associated with the output variables to be controlled, $\hat{\mathbf{q}}$, are obtained from the generalized speeds $\dot{\mathbf{q}}$ using a Jacobian matrix, $\mathbf{J}_C$, as

$$\dot{\hat{\mathbf{q}}} = \mathbf{J}_C(\mathbf{q})\,\dot{\mathbf{q}}. \tag{2}$$

Assuming that this Jacobian matrix is square and non-singular, Eq. (1) can be written in terms of the output variables as follows:

$$\hat{\mathbf{H}}(\mathbf{q})\ddot{\hat{\mathbf{q}}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{Q}}(\mathbf{q}), \tag{3a}$$

where $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{Q}}$ can be obtained as

$$\begin{aligned} \hat{\mathbf{H}} &= \mathbf{J}_C^{-\mathrm{T}}\mathbf{H}\mathbf{J}_C^{-1}, \\ \hat{\mathbf{C}} &= \mathbf{J}_C^{-\mathrm{T}}\mathbf{C} - \hat{\mathbf{H}}\dot{\mathbf{J}}_C\dot{\mathbf{q}}, \\ \hat{\mathbf{Q}} &= \mathbf{J}_C^{-\mathrm{T}}\mathbf{Q}. \end{aligned} \tag{3b}$$

To control such a system, classic PID joint controllers that ignore the dynamic coupling are widely employed in industrial geared robots. These feedback controllers can effectively control the system due to the high gear ratios at the joints, Arimoto and Miyazaki (1983). However, in direct drive manipulators and space systems the need for inclusion of the system dynamics cannot be eliminated. The *Computed Torque Method* employs such a model to compensate for the non-linearities, and result in a linearized error behavior.[1] The application of Model-*Referenced Adaptive Control* to robotic manipulators is based on an adaptation algorithm which changes the controller gains so that the real output follows the referenced model within an accuracy bound, Slotine and Li (1987).

Based on the system dynamics introduced by Eq. (3), an MB (Computed Torque) control law can be developed as

$$\mathbf{Q} = \mathbf{J}_C^{\mathrm{T}}\{\hat{\mathbf{H}}[\mathbf{K}_p\mathbf{e} + \mathbf{K}_d\dot{\mathbf{e}} + \ddot{\hat{\mathbf{q}}}_{\mathrm{des}}] + \hat{\mathbf{C}}\}. \tag{4}$$

This law linearizes and decouples the system equations to a set of second order differential equations, resulting in the following error dynamics:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \mathbf{0}, \tag{5}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite gain matrices, and $\mathbf{e}$ is the tracking error defined as

$$\mathbf{e} = \hat{\mathbf{q}}_{\mathrm{des}} - \hat{\mathbf{q}}. \tag{6}$$

Under the assumption of known system dynamics structure, and known geometric and mass properties, the control law given by Eq. (4) guarantees asymptotic convergence of the tracking error to zero. However, if these assumptions are violated, the error may never converge; driving terms appear in the right-hand side of Eq. (5). In addition, this control law requires a significant computational effort[2] (see Table 1) which may not be available in most cases.

---

[1] According to Craig (1989), the idea was first proposed in Paul (1972), and named as the Computed Torque Method in Markiewicz (1973).

[2] To apply a Model-Based (Computed Torque) control law, $\hat{\mathbf{H}}$ and $\hat{\mathbf{C}}$ have to be computed. Considering Eqs. (3b), it can be seen that computation of $\hat{\mathbf{H}}$ and $\hat{\mathbf{C}}$ requires inversion of the Jacobian matrix and calculation of its time derivative which depending on the system degrees-of-freedom may be quite cumbersome. The number of matrix multiplications in obtaining these expressions, is also considerable. The required computational operations can be seen in Table 1, though the assumptions made in preparation of this table exclude the operations for inverting the Jacobian matrix and calculating its time derivative.

Table 1
Comparison of the required computational operations

Algorithm	Multiplication	Additions
TJ	$3N^2$	$3N^2-2N$
MTJ	$3N^2+2$	$3N^2-N+1$
MB	$2N^3+7N^2$	$2N^3+5N^2-4N$

As discussed in Craig (1989), if high enough gains are used, the control law of Eq. (4) can be approximated by the simpler TJ controller as

$$\mathbf{Q}=\mathbf{J}_C^{\mathrm{T}}\{\mathbf{K}_p\mathbf{e}+\mathbf{K}_d\dot{\mathbf{e}}\} \tag{7}$$

which does not require a priori knowledge of the system dynamics. Besides simplicity, an advantage of this algorithm is that if a physical singularity is encountered, the controller given by Eq. (7) may result in errors but will not fail computationally. The action of this controller can be understood by imagining generalized springs and dampers, along the variables under control, connected between the corresponding body and the desired trajectories; the stiffer the gains are, the better the tracking should be. However, due to the presence of noise and unmodeled dynamics, the use of high gains is limited in practice. Note that computation of $\hat{\mathbf{Q}}$ based on Eqs. (7) and (3b), does not result in the error dynamics given by Eq. (6), anymore.

It should be mentioned that the TJ algorithm can be applied to redundant manipulators as shown by Asari et al. (1993). The same property applies to MTJ algorithm as will be detailed below. In fact, the assumption of a square Jacobian matrix, after Eq. (2), was for developing the MB algorithm of Eq. (4). Hence, the advantages of using the TJ controller motivate further work on this algorithm, aiming at improving its performance and limiting its drawbacks.

3. MTJ control law

The TJ control law defined by Eq. (7) is now modified to achieve both precision and simplicity:

$$\mathbf{Q}=\mathbf{J}_C^{\mathrm{T}}\{\mathbf{K}_d\dot{\mathbf{e}}+\mathbf{K}_p\mathbf{e}+\mathbf{h}(t)\}, \tag{8}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite gain matrices, $\mathbf{e}$ is the tracking error defined in Eq. (6), and $\mathbf{h}(t)$ is to be determined for feedback linearization. Substituting Eq. (8) into Eq. (3), yields

$$\mathbf{K}_d\dot{\mathbf{e}}+\mathbf{K}_p\mathbf{e}=\hat{\mathbf{H}}\ddot{\mathbf{q}}+\hat{\mathbf{C}}-\mathbf{h}(t). \tag{9}$$

Considering Eq. (3), this is equivalent to

$$\mathbf{K}_d\dot{\mathbf{e}}+\mathbf{K}_p\mathbf{e}=\hat{\mathbf{Q}}-\mathbf{h}(t). \tag{10}$$

It is motivating to note that if the right-hand side (RHS) of Eq. (10) becomes equal to zero, then the tracking error converges to zero, and the algorithm works like a MB algorithm, albeit with a simpler implementation. Note that inclusion of the second derivative of the error, $\ddot{\mathbf{e}}$, in Eq. (8) results in

$$\mathbf{Q}=\mathbf{J}_C^{\mathrm{T}}\{\ddot{\mathbf{e}}+\mathbf{K}_d\dot{\mathbf{e}}+\mathbf{K}_p\mathbf{e}+\mathbf{h}(t)\} \tag{11}$$

and then

$$\ddot{\mathbf{e}}+\mathbf{K}_d\dot{\mathbf{e}}+\mathbf{K}_p\mathbf{e}=\hat{\mathbf{Q}}-\mathbf{h}(t), \tag{12}$$

which results in n error dynamics similar to that of the MB algorithms, if the RHS of Eq. (12) becomes equal to zero. However, inclusion of this signal requires acceleration measurements or an estimator, and may be difficult to obtain in practice.

To make the RHS of Eq. (10) or (12) be close to zero, a good approximation can be obtained by taking $\mathbf{h}(t)$ equal to $\hat{\mathbf{Q}}$ at a previous small time step, $\hat{\mathbf{Q}}|_{t-\Delta t}$. Inclusion of this term may result in high joint torque requirements, when relatively high $\mathbf{e}$ or $\dot{\mathbf{e}}$ exist due to disturbances. To avoid this potential problem, the standard TJ algorithm can be applied momentarily. Therefore, a *regulating factor* can be employed as

$$\mathbf{h}(t)=k\hat{\mathbf{Q}}|_{t-\Delta t}, \tag{13}$$

where k is defined as

$$k=\begin{cases}0 & \text{when } |\mathbf{e}|\geqslant\varepsilon \text{ or } |\dot{\mathbf{e}}|\geqslant\dot{\varepsilon},\\ 1 & \text{when } |\mathbf{e}|<\varepsilon \;\&\; |\dot{\mathbf{e}}|<\dot{\varepsilon},\end{cases} \tag{14}$$

where ε and $\dot{\varepsilon}$ are positive real numbers that correspond to sensitivity thresholds. To simplify the on–off switch for factor k, the following continuous expression can be used[3] :

$$k=\exp\left(-\left(\frac{|\mathbf{e}|}{e_{\max}}+\frac{|\dot{\mathbf{e}}|}{\dot{e}_{\max}}\right)\right), \tag{15a}$$

where $e_{\max}$ and $\dot{e}_{\max}$ represent another representation of the sensitivity thresholds. Note that relatively low values for sensitivity thresholds, would make the algorithm work like the standard TJ control law. Therefore, based on the expected precision, one could start by selecting relatively equivalent values for the sensitivity thresholds. Then, if the resulting errors are high, those values should be increased. It should be mentioned that choosing low values for sensitivity thresholds makes k equal to zero, which results in a TJ control law, as it is deliberately taken at initial time in the first time step. In practice, $\mathbf{K}_p$ and $\mathbf{K}_d$ can be chosen as diagonal matrices, and so factor k in Eq. (13) can be replaced by a diagonal matrix $\mathbf{K}$. Then its elements can be defined as

$$k_{ii}=\exp\left(-\left(\frac{|e_i|}{e_{\max_i}}+\frac{|\dot{e}_i|}{\dot{e}_{\max_i}}\right)\right). \tag{15b}$$

Including the error derivative in Eq. (15), introduces a sense of anticipation, without compromising the smoothness of response. Similarly, one can include another term based on the second rate of error, if available. However, this makes the algorithm more sensitive, and therefore sharp variations of actuator forces/torques may result.

Application of the MTJ algorithm

$$\mathbf{Q}=\mathbf{J}_C^{\mathrm{T}}\{\mathbf{K}_d\dot{\mathbf{e}}+\mathbf{K}_p\mathbf{e}+k\hat{\mathbf{Q}}|_{t-\Delta t}\}, \tag{16}$$

[3] Similar expression has been successfully employed to regulate sliding mode controllers in Moosavian and Homaeinejad (2004) for chattering elimination.

with proper selection of the sensitivity thresholds (so that the modifying term is reasonably activated) and small time steps, results in the following error equation:

$$k_{d_i}\dot{e}_i + k_{p_i} e_i \cong 0, \tag{17}$$

where diagonal gain matrices, $\mathbf{K}_p$ and $\mathbf{K}_d$, have been used. Therefore, using Eq. (17), the control gains can be selected in a more systematic manner, as their ratio determines error time constant, and their magnitude determines the magnitude of the control command which should be adjusted based on actuator capabilities.

Considering Eq. (16), it can be deduced that for an N DOF system, calculation of the MTJ law requires $3N^2+N+2$ multiplications, and $3N^2-N+1$ additions. Comparing to the depicted results in Table 1, these are almost the same as those for the TJ law, and still significantly less compared to those needed for implementing the MB laws. Note that it is assumed that the inverse of the Jacobian matrix and its time derivative, which are required for implementing MB algorithms, are available symbolically, and hence these computations are not counted in Table 1. However, this comparison in terms of required computational effort reveals the efficiency of the TJ and the MTJ algorithms.

The above analysis reveals the simplicity (concerning a priori knowledge requirement of system dynamics) and efficiency (in terms of the required computational effort) of both the standard TJ and the new MTJ law compared to the MB algorithms. In addition, the MTJ yields approximately linearized error dynamics, and therefore an improved performance over the standard TJ algorithm. Stability analysis of the developed MTJ algorithm, based on Lyapunov's theorems, shows that both the standard and the MTJ algorithms are globally asymptotically stable (see Moosavian & Papadopoulos, 1997).

It should be noted that the MTJ law is a position control algorithm which yields an improved performance over the standard algorithm. However, to manipulate an object, application of force/impedance control laws will be required that are usually MB algorithms. For instance, the Multiple Impedance Control (MIC) is an MB algorithm that requires knowledge of the system dynamics, Moosavian et al. (2005). On the other hand, even if the system dynamics is perfectly known, its computation may require considerable process time at each step for implementing the control law. Based on the MTJ control approach proposed above, the MIC law has been recently modified to be implemented without using system dynamics as Non-Model-Based Multiple Impedance Control (NMIC), Moosavian and Ashtiani (2006). Therefore, this NMIC law is a more realistic algorithm for on-line computations in cooperating robotic systems.

Next, the performance of the new MTJ is evaluated by simulation, and compared to the standard TJ, and MB algorithms.

4. Obtained results

To focus on algorithmic aspects, a simple two link planar manipulator is considered under various conditions. Performing low-speed vs. high-speed tracking task, selection of higher gain for the TJ, and noise rejection characteristics of the proposed MTJ algorithm is investigated in these simulations, and compared to those of alternative algorithms.

The system is a 2-link planar manipulator on a horizontal plane, see Fig. 1(a). The task is tracking a trajectory defined by

$$x_{\text{des}} = \sqrt{l_1^2 + l_2^2}\cos(\omega t + \pi/4) + 0.1\sin(5\omega t),$$
$$y_{\text{des}} = \sqrt{l_1^2 + l_2^2}\sin(\omega t + \pi/4) + 0.1\sin(5\omega t). \tag{18}$$

This trajectory corresponds to a perturbed circular path, see Fig. 1(b). The motion speed along the path can be selected by setting the cyclical frequency ω.

The mass properties of the system are $m_1 = 4.0\,\text{kg}$, $I_1 = 0.333\,\text{kg m}^2$, $m_2 = 3.0\,\text{kg}$, and $I_2 = 0.30\,\text{kg m}^2$, and the link lengths are $l_1 = 1\,\text{m}$ and $l_2 = 1\,\text{m}$. The initial conditions for joint angles and derivatives are

$$(q_1(0), q_2(0), \dot{q}_1(0), \dot{q}_2(0)) = (0.03, \pi/2, 1.5, -1.0) \quad (\text{rad}, \text{rad/s})$$

which correspond to some initial position and velocity errors.

The sensitivity thresholds for the MTJ algorithm, $e_{\max}$ and $\dot{e}_{\max}$ in Eq. (15a) are taken equal to 1 m and 10 m/s, respectively. These large values for $e_{\max}$ and $\dot{e}_{\max}$, yield $k \approx 1.0$ throughout the whole duration of the simulation after the first time step (which is zero, according to the definition). The time step Δt_i is held constant, and equal to 10.0 ms. It should be mentioned that based on the explanations below Eq. (14), selecting the time step should ensure the stability. On the other hand, it must be feasible based on the system measuring and calculation time constants. Therefore, choosing 10 ms for the considered system will satisfy both restrictions. To establish a fair comparison, the gains for the algorithms under comparison are selected such that the peaks of the required joint torques are approximately equal. The Gear method for solving differential equations is used in all simulations.

4.1. Low-speed vs. high-speed tracking task

The performance of the TJ and MTJ algorithms, in terms of the end-point error in a low-speed tracking task ($\omega = 0.05\,\text{rad/s}$), is compared in Fig. 2. For the MTJ algorithm $\mathbf{K}_p = \text{diag}(30, 30)$, $\mathbf{K}_d = \text{diag}(60, 60)$, while for the TJ algorithm the gains are twice these values. It can be seen that both algorithms result in a fairly similar response. However, errors for the TJ algorithm may increase initially to higher values, before they converge to zero, see for example, $e(y)$ in Fig. 2(a).

Fig. 3 shows the end-point tracking error in a high-speed tracking ($\omega = 2.0\,\text{rad/s}$). As shown in this figure, the MTJ algorithm results in smaller tracking errors, and therefore is preferred. This poor performance of the TJ algorithm is due to the fact that it is not dynamics-based. However, one would expect that by selecting very high gains, its performance can be improved.

To investigate this possibility, the previous gain values for the MTJ are used, while for the TJ fairly high gains are selected, see Table 2. Besides, the task speed is reduced to $\omega = 1.0\,\text{rad/s}$.

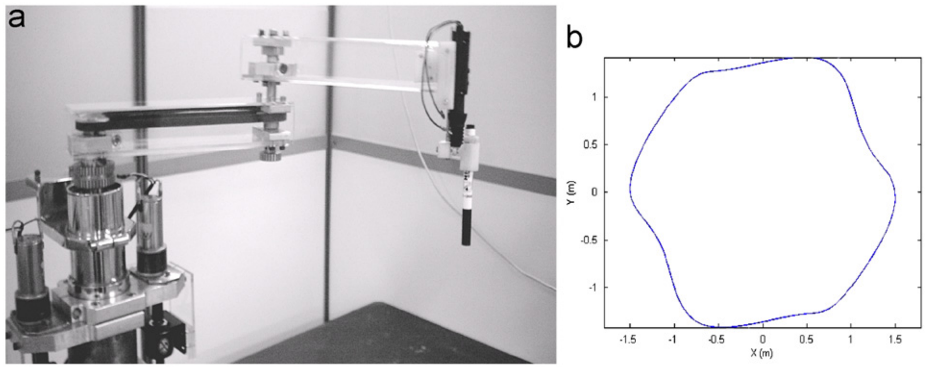

Fig. 1. (a) A two-link planar manipulator and (b) desired tracking path.

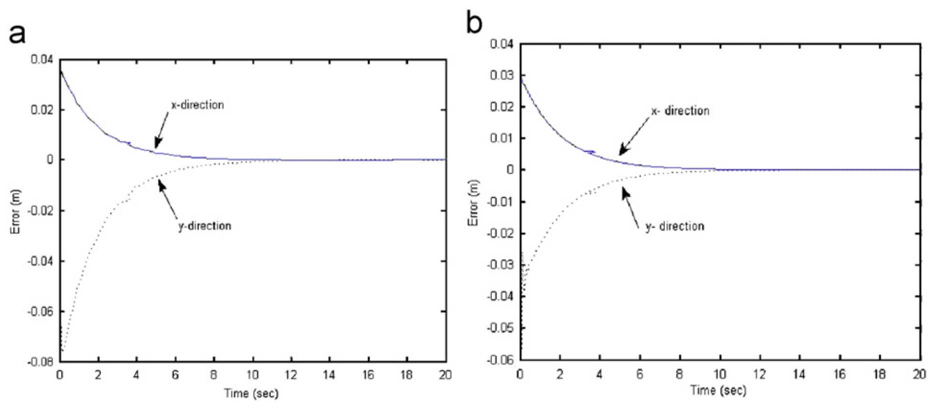

Fig. 2. Tracking errors for low-speed task: (a) TJ and (b) MTJ algorithm.

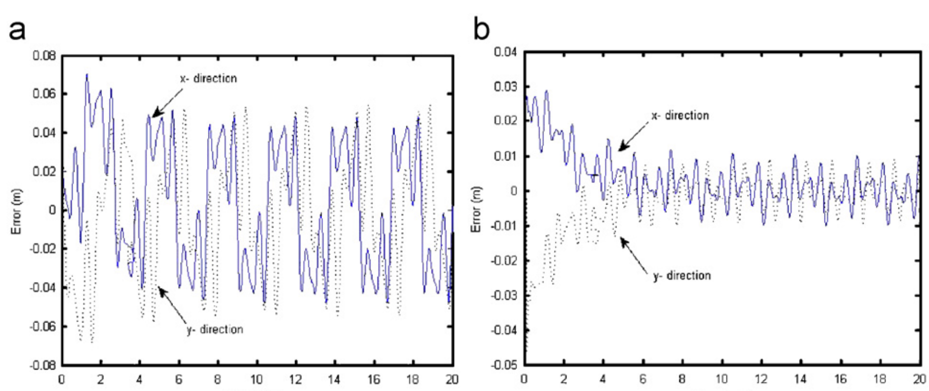

Fig. 3. Tracking errors for high-speed task: (a) TJ and (b) MTJ algorithm.

Table 2
Selected gains for alternative algorithms, Example 1

Algorithm	$\mathbf{K}_p$	$\mathbf{K}_d$
TJ	diag(150, 150)	diag(300, 300)
MTJ	diag(30, 30)	diag(60, 60)
MB, case 1	diag(8, 8)	diag(4, 4)
MB, case 2	diag(30, 30)	diag(60, 60)

Here, in addition to the TJ and MTJ algorithms, two cases of model-based (MB) algorithms are also considered. In the first case, it is assumed that the mass properties are completely known, while in the second one, the mass properties of the dynamics model in the controller are perturbed by 10% with respect to the *true* values. For the perfect MB, the chosen gains are fairly low which correspond to a settling time of 2.0 s, and a damping ratio of 0.7. For the second MB case, these low gains result in relatively large tracking errors, therefore they

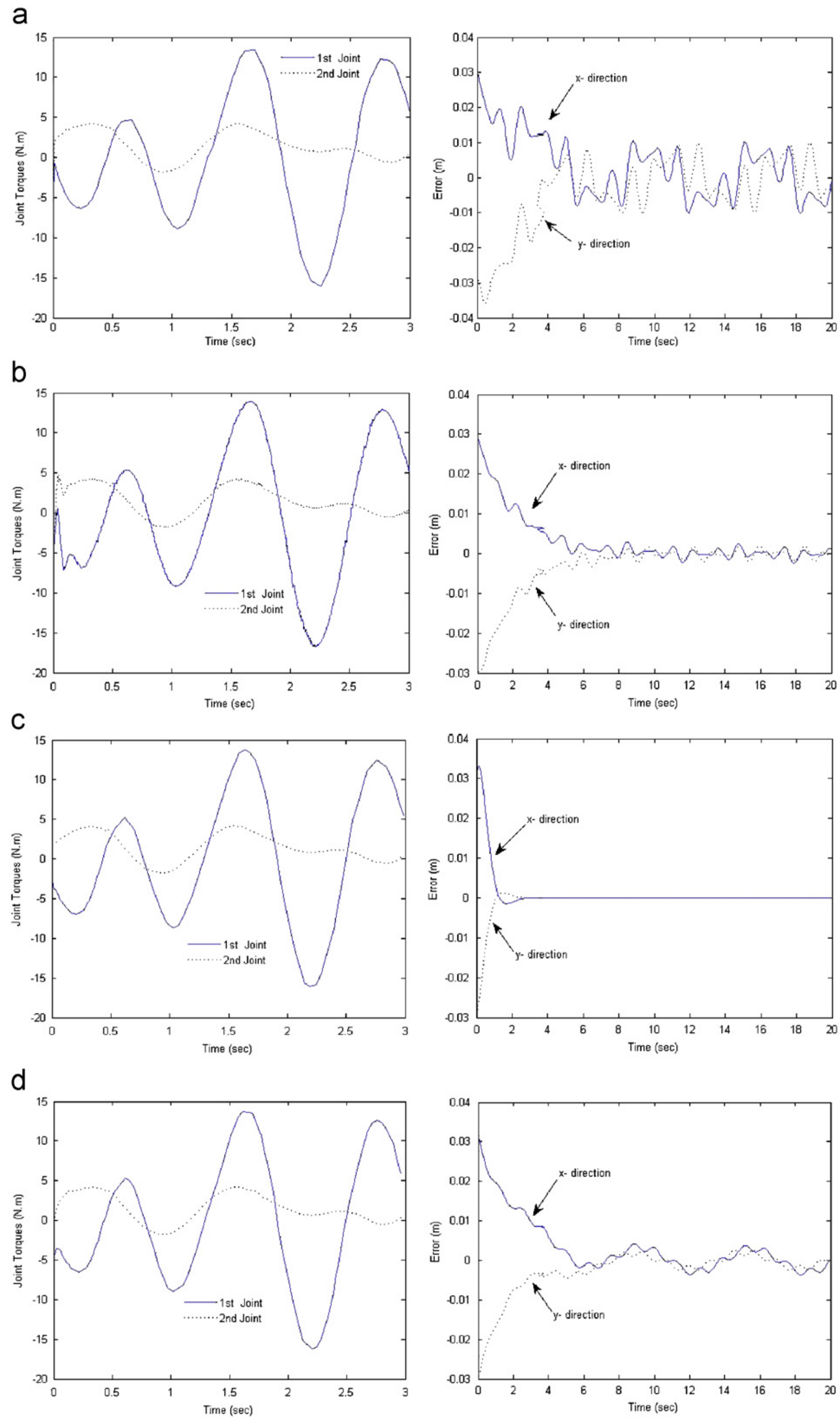

Fig. 4. Joint torques and tracking errors: (a) TJ with high gains, (b) MTJ, (c) MB, case 1, and (d) MB, case 2.

are selected equal to the ones for the MTJ. As Fig. 4 shows, due to properly adjusted gains, the peaks of joint torques for all four algorithms are about the same, which as mentioned before establishes a fair comparison. Nevertheless, it can be seen that, even with relatively very high gains for the TJ, the resulting tracking errors of the MTJ are still about five times smaller than the ones of the standard TJ, and even better than the ones of the perturbed MB (case 2) algorithm. In other words, the

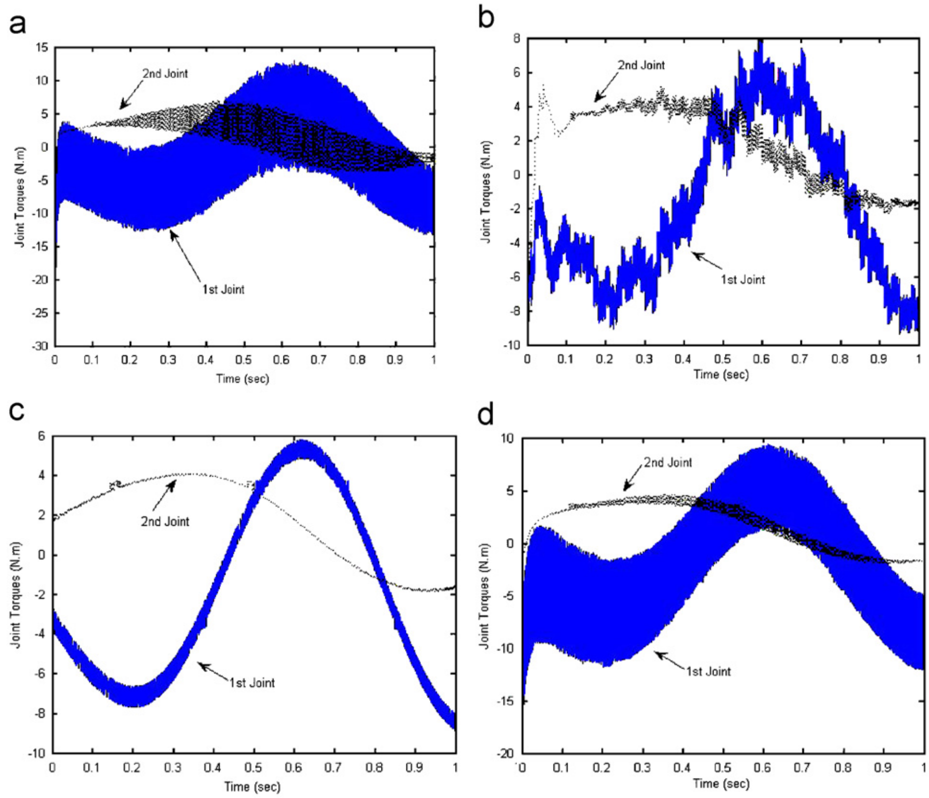

Fig. 5. Joint torques in the presence of noisy feedback: (a) TJ, high gains, (b) MTJ, (c) MB, case 1, and (d) MB, case 2.

advantage of MB laws is lost if the parameters are not known exactly.

It should be mentioned that the total energy consumption of each algorithm for performing this task, given by the time integral of $\sum_{i=1}^{2}|\tau_i \dot{q}_i|$, is about the same, i. e. in correspondence to Fig. 4, (a) 153, (b) 156, (c) 153, and (d) 154 J.

4.2. Noise rejection characteristics

In practice, noise will corrupt available feedback. Therefore, one should examine the noise rejection capabilities of would be implemented algorithms, especially of those that rely on high gains. The previous simulation is repeated now, assuming that measurements of joint angles and their rates are corrupted by white noise whose amplitude is 2% of the output's magnitude. Although the performance in terms of the average tracking errors is almost the same as before, the variation in the required torques is larger. As shown in Fig. 5, the required torques for the MTJ algorithm are almost as smooth as for a perfect MB control, while the noise rejection characteristics for the TJ algorithm are poorer, due to the high gains employed. It should be noted that for the MB algorithms, noisy feedback affects the elements of controller dynamics, which in the presence of uncertainties (the second MB case) results in a poor noise rejection characteristics, see Fig. 5(d).

It can be concluded that for better tracking, higher gains are required for the TJ algorithm, and these lead to poor noise rejection characteristics. Also, high frequency inputs can excite flexible system modes, and consequently decrease the accuracy, and the useful life of a system. Hence, it is confirmed that using high gains is not a viable option. On the other hand, the new MTJ algorithm, by being an approximation of a feedback linearization algorithm, does not require high gains, or a high computational power, while its performance is comparable to that of the MB algorithms.

5. Conclusions

This paper presented the new MTJ control which yields a better performance (in terms of tracking errors, with the same requirements of actuator forces/torques) compared to the standard Transposed Jacobian (TJ) algorithm. The MTJ controller approximates a feedback linearization solution, using stored data of the control command in the previous time step as a learning tool, with no need to a priori knowledge of the plant dynamics. Therefore, unlike an MB algorithm, it is not affected by inaccuracies in mass properties. Stability analysis, based on Lyapunov theorems, shows that both the standard and the MTJ algorithms are globally asymptotically stable. A two link manipulator was simulated to investigate different aspects of the performance of the new proposed algorithm. It was shown that the performance of the MTJ controller is comparable to that of a perfect MB algorithm, with the advantage that less computational power is needed. The substantially reduced computational requirements compared to the MB, and the good tracking and noise rejection performance characteristics in comparison with the TJ, suggest that the MTJ algorithm is a promising alternative. Therefore, the new MTJ algorithm can be considered as a good candidate in the control of industrial robots, where simple efficient algorithms are more appropriate than complicated theoretical ones.

References

Arimoto, S., & Miyazaki, F. (1983). Stability and robustness of PID feedback control for robot manipulators of sensory capability. *Proceedings of the first international symposium on robotics research* (pp. 783–799). Cambridge, MA.

Asari, Y., Sato, H., Yoshimi, T., & Tatsuno, K. (1993). Development of a model-based remote maintenance robot system (IV)—a practical stiffness control method for redundant robot arm. *Proceedings of the first international conference on intelligent robots and systems*, Yokohama, Japan.

Bevly, D., Dubowsky, S., & Mavroidis, C. (2000). A simplified Cartesian-computed torque controller for highly geared systems and its application to an experimental climbing robot. *Journal of Dynamic Systems, Measurement, and Control*, *122*, 27–32.

Chiaverini, S., Sciavicco, L., & Siciliano, B. (1991). Control of robotic systems through singularities. *Proceedings of the international workshop on nonlinear and adaptive control*: *Issues in robotics*, Grenoble, France, 1990. Lecture notes in control & information sciences, Vol. 162.

Chiu, C. S., Lian, K. Y., & Wu, T. C. (2004). Robust adaptive motion/force tracking control design for uncertain constrained robot manipulators. *Automatica*, *40*(12), 2111–2119.

Craig, J. (1989). *Introduction to robotics, mechanics and control*. Reading, MA: Addison-Wesley.

Dominguez-Lopez, J. A., Damper, R. I., Crowder, R. M., & Harris, C. J. (2004). Adaptive neurofuzzy control of a robotic gripper with on-line machine learning. *Robotics and Autonomous Systems*, *48*, 93–110.

Garcia, E., & Kelly, R. (2002). Experimental evaluation of task space regulators using unit quaternions on a direct-drive spherical wrist. *Proceedings of the IEEE international conference on robotics and automation*, Washington, DC, USA, May.

Hogan, N. (1985). Impedance control: An approach to manipulation. *ASME Journal of Dynamic Systems, Measurement and Control*, *107*(1), 1–24.

Hootsmans, N. A. M., & Dubowsky, S. (1991). Large motion control of mobile manipulators including vehicle suspension characteristics. *Proceedings of IEEE international conference on robotics and automation*, Sacramento, CA, USA, April.

Kelly, R., & Moreno, J. (2005). Manipulator motion control in operational space using joint velocity inner loops. *Automatica*, *41*(8), 1423–1432.

Markiewicz, B. (1973). *Analysis of the computed torque drive method and comparison with conventional position servo for a computed-controlled manipulator*. Jet Propulsion Laboratory Technical Memo 33-601, March.

Mbede, J. B., Ele, P., Mveh-Abia, C. M., Youssoufi, T., Graefe, V., & Ma, S. (2005). Intelligent mobile manipulator navigation using adaptive neuro-fuzzy systems. *Information Sciences*, *171*, 447–474.

Meghdari, A., Naderi, D., & Alam, M. R. (2005). Neural-network-based observer for real-time tipover estimation. *Mechatronics*, *15*(8), 989–1004.

Miyazaki, F., Masutani, Y., & Arimoto, S. (1988). Sensor feedback using approximate Jacobian. *Proceedings of the USA–Japan symposium on flexible automation—crossing bridges*, *advances in flexible automation and robotics*, Minneapolis, MN, USA.

Moosavian, S. A. A., & Ashtiani, H. R. (2006). Cooperative object manipulation using non-model-based multiple impedance control. *Proceedings of the IEEE international conference on control applications* (*CCA*), October 4–6, Munich, Germany.

Moosavian, S. A. A., & Homaeinejad, M. R. (2004). Regulated sliding mode control of satellite rotation. *Proceedings of the IFAC workshop on generalized solutions in control problems* (*GSCP-2004*), Russia, September 22–26.

Moosavian, S. A. A., & Papadopoulos, E. (1997). Control of space free-flyers using modified transpose Jacobian algorithm. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, Grenoble, France, September 8–13.

Moosavian, S. A. A., Rastegari, R., & Papadopoulos, E. (2005). Multiple impedance control for space free-flying robots. *AIAA Journal of Guidance, Control, and Dynamics*, *28*(5),

Papadopoulos, E., & Moosavian, S. A. A. (1995). Dynamics & control of space free-flyers with multiple arms. *Journal of Advanced Robotics*, *9*(6), 603–624.

Paul, R. P. (1972). *Modeling trajectory calculation, and servoing of a computer controlled arm*. Technical Report AIM-177, Stanford University Artificial Intelligence Laboratory.

Raibert, M. H., & Craig, J. J. (1981). Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement & Control*, *126*(2), 126–133.

Slotine, J. J. E., & Li, W. (1987). Adaptive robot control: A new perspective. *Proceedings of the 26th IEEE conference on decision and control*, Los Angeles, CA, USA.

Steil, J. J., Röthling, F., Haschke, R., & Ritter, H. (2004). Situated robot learning for multi-modal instruction and imitation of grasping. *Robotics and Autonomous Systems*, *47*, 129–141.

Taira, Y., Sagara, S., & Katoh, R. (2000). Digital adaptive control of space robot manipulators using transpose of generalized Jacobian matrix. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, Takamatsu, Japan, October 31–November 5.

Umetani, Y., & Yoshida, K. (1989). Resolved motion rate control of space manipulators with generalized Jacobian matrix. *IEEE Transactions on Robotics and Automation*, *5*(3), 303–314.

Youcef-Toumi, K., & Ito, O. (1987). Controller design for systems with unknown dynamics. *Proceedings of American control conference*, Minneapolis, USA.

S. Ali A. Moosavian received his B.S. degree in 1986 from Sharif University of Technology and the M.S. degree in 1990 from Tarbiat Modaress University (both in Tehran), and his Ph.D. degree in 1996 from McGill University, all in Mechanical Engineering. He is currently an Associate Professor at the Mechanical Engineering Department at K. N. Toosi University of Technology (Tehran). He teaches courses in the areas of robotics, dynamics, automatic control, analysis and synthesis of mechanisms. His research interests are in the areas of dynamics modeling, and motion/impedance control of terrestrial and space robotic systems. He has published more than 75 articles in journals and conference proceedings. He is a Member of the IEEE, and one of the founders of the ARAS Research Center for Design, Manufacturing and Control of Robotic Systems, and Automatic Machineries.

Evangelos Papadopoulos received his Diploma from the National Technical University of Athens (NTUA) in 1981, and his M.S. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT) in 1983 and 1991, respectively, all in Mechanical Engineering. From 1985 to 1987, he was an analyst with the Hellenic Navy, Athens, Greece. In 1991 he was appointed a Lecturer at the Department of Mechanical Engineering at MIT. He then joined McGill University and the Centre for Intelligent Machines (CIM) as an Assistant Professor, and was tenured in 1997. Currently, he is an Associate Professor with the Mechanical Engineering Department at the NTUA. He teaches courses in the areas of systems, controls, mechatronics and robotics. His research interests are in the area of robotics, including space, mobile and underwater robotics, modeling and control of dynamic systems, haptic devices, mechatronics, and design. In 2003, he served as a Guest Editor for the *Advances in Robot Dynamics and Control* issue of the *ASME/IEEE Transactions on Mechatronics*. He has published more than 100 articles in journals and conference proceedings. He is a Senior Member of the AIAA and of the IEEE, and a member of the ASME, the NYAS, the Technical Chamber of Greece (TEE) and the Sigma Xi.