Lyapunov Based Sampling for Adaptive Tracking Control in Robot Manipulators: An Experimental Comparison

Pavankumar Tallapragada[1] and Nikhil Chopra[2*]

[1] Department of Mechanical Engineering, University of Maryland, College Park, 20742 MD, USA, pavant@umd.edu

[2] Department of Mechanical Engineering and The Institute for Systems Research, University of Maryland, College Park, 20742 MD, USA, nchopra@umd.edu

Abstract. In digital computer based controllers, efficient sampling mechanisms for sensors as well as controllers is of great importance. In this paper, we are interested in designing controllers that result in low average frequency of control updates while simultaneously ensuring stability of the robotic system. We experimentally investigate a non-periodic state-triggered control sampling scheme (designed through Lyapunov like analysis) for adaptive tracking controllers in robot manipulators. We implement this scheme on two well known continuous-time adaptive controllers for tracking in robot manipulators and compare their performance heuristically based on the results of experiments performed on a two link planar manipulator.

1 Introduction

Motivation The controllers in robotic systems are predominantly digital computer based. These systems inherently involve sampling, which creates a trade-off between sampling/communication cost and achieving the required task at hand, such as stabilization. These issues assume additional significance when the control signal is communicated over a network or when the controller is implemented on an embedded processor with low computational capabilities. In certain cases, the task of sensing (for example, visual feedback) may itself impose constraints on the update frequency of the control. Thus, there is a clear need to design efficient sampling mechanisms for controllers. In this paper we investigate a state based sampling technique for reducing the average frequency of control updates while ensuring stability of trajectory tracking in robot manipulators.

Related Work Traditional computer based control systems in robotics rely on periodic sampling of the sensors and computation/execution of the control [1–5]. The reason for the popularity of this paradigm is a well developed theory and the ease of analysis of such systems. However, such

*This work was partially supported by the National Science Foundation under grant 0931661.

control algorithms may need to employ high sampling rates. Recently, there has been a growing interest among the control community in what are known as event based control systems [6–9]. In these control systems, timing of control execution is not necessarily periodic and can be state dependent. By encoding the nature of the task (for example stabilization) into an even-triggering condition, it is possible to systematically design sampled-data controllers that make better use of computational and communication resources.

Predominantly, *event-triggered* controllers in the literature are essentially sampled data versions of continuous time controllers, with the sampling instants determined by state based triggering conditions. Additionally, many of these controllers may be called Lyapunov based controllers. This is due to the fact that the task of the controller, whether it is set point or trajectory tracking, can be cast as a stabilization problem and the event-triggering condition can be determined through a Lyapunov like analysis.

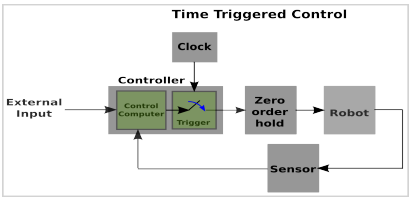
While Lyapunov based event-triggered controllers implicitly guarantee stability, they have a major drawback. These controllers rely critically on the knowledge of an accurate model of the system. For example, the results in [6,9] are general enough to hold for robotic manipulators when perfect knowledge of the system is available. However, building a model of high accuracy is a time consuming process and in many cases, it may not even be possible. Therefore, it is important to extend the design of implicitly verified event based controllers to cases where only a poor model of the system is available. This is specially important in the field of robotics, where adaptive and robust controllers are often used.

Problem Statement Motivated by this practical necessity, we have developed [10] an event based adaptive control technique for trajectory tracking that can be used even when the robot system dynamics are unknown. In this paper, this state based Lyapunov sampling/triggering scheme is utilized to realize digital computer implementations of two well known adaptive tracking controllers for robot manipulators - [11] and [12], respectively, and their performance is compared through some experiments on a two-link planar manipulator.

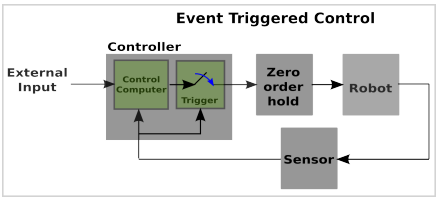
1.1 Technical Approach - Lyapunov Based Sampling

The basic idea of time-triggered control and state-triggered control (event-triggered control) is illustrated in Figure 1. The difference between the two is in the manner of determining the sampling instants. While in the time-triggered case the sampling instants are determined by an external clock, in the event-triggered case they are determined implicitly by a state dependent event-triggering condition.

Figure 2a illustrates a simple time-triggered (periodic sampling) and state-triggered (threshold-triggered) sampling. As can be seen, sampling based on threshold crossings is dependent on the signal, unlike the time-triggered sampling. Thus, the event-triggered approach may be said to sample the signal only when 'necessary'. This idea of sampling only when necessary can be made more systematic by encoding the control objective in an event-trigger.



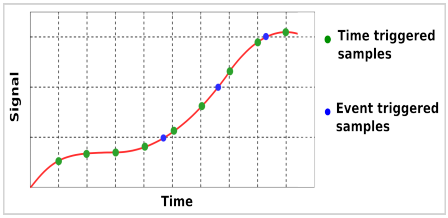
(a) Time-Triggered-Control: The sensor data is required by the controller at predetermined time instances



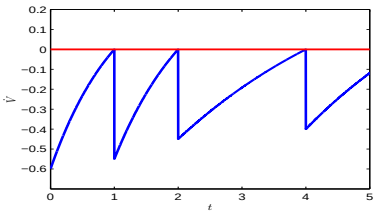
(b) Event-Triggered Control: The sensor data is required by the controller at time instants determined by a triggering function

Fig. 1: Time-triggered and event-triggered approaches to sampled data control.

The sampling mechanism utilized in this paper may be called a Lyapunov based sampler as the state based *event-triggering condition* that determines the sampling instants is designed by a Lyapunov like analysis. This approach implicitly guarantees stability of the tracking error. The basic idea behind the Lyapunov based sampling approach is shown in Figure 2b. Ideally, if the control signal is held constant and updated only as the derivative of a Lyapunov function approaches zero from below, the average frequency of sampling may be reduced. Moreover, because this approach ensures that the derivative of a Lyapunov function, $\dot{V}$, is strictly negative at all time, it implicitly guarantees stability.



(a)



(b)

Fig. 2: (a) Time and Threshold based sampling. (b) Lyapunov based sampling - the control signal is sampled/updated only when the derivative of a Lyapunov function reaches zero.

There are two main caveats in this description of the design. The first is that continuously checking the zero-crossing of $\dot{V}$ is computationally inefficient. In addition, when only inexact information about the robot parameters is available, $\dot{V}$ cannot be computed precisely. However, it is possible to design an easy to check function (event-triggering condition) that always upper bounds $\dot{V}$ (even if its precise value is unknown at any given time).

In the next section, we present a Lyapunov based event-triggered implementation of two well known adaptive controllers for trajectory tracking.

2 Event Based Adaptive Control

In this section we present a Lyapunov based event-triggered implementation of two well known adaptive controllers for trajectory tracking in robotic manipulators. In this paper, only the essential steps in the design process are described. Detailed procedure of the design may be found in our previous works. The work in [9] is useful for general nonlinear systems under the perfect knowledge of system. On the other hand, in [10] the design procedure for a specific adaptive tracking controller for robotic manipulators is described.
Now consider a standard n-degree of freedom rigid robot model of the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u, \quad q \in \mathbb{R}^n,\ u \in \mathbb{R}^n$$

where $M : \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$, $C : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$ and $G : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. The matrix $C(q,\dot{q})$ is defined using the Christoffel symbols. Let $x_d \triangleq [q_d; \dot{q}_d] \in \mathbb{R}^n \times \mathbb{R}^n$ be the state of the desired trajectory that the robot has to track. Here the notation $[a_1; a_2]$ denotes the column vector formed by concatenating the vectors a_1 and a_2. This notation is used in this paper to refer to various concatenated vectors. Let $\tilde{q} \triangleq q - q_d$, then the tracking error is defined as $\tilde{x} \triangleq [\tilde{q}; \dot{\tilde{q}}]$. Let $u = \gamma(\xi) \in \mathbb{R}^m$ be a known continuous-time control law for trajectory tracking, where ξ is the data that the controller depends on.
Let us introduce the following notation to denote the sampled data versions of different signals in the system. The sampled data version of any signal X (which can be a scalar, a vector or a matrix) is denoted by X_s. In particular, the data sampled by the controller is denoted by ξ_s, and is defined as

$$\xi_s(t) = \xi(t_i), \quad \text{for all } t \in [t_i, t_{i+1}), \quad \text{for each } i$$

where t_i are the sampling instants. All the other sampled data signals are similarly defined. In time-triggered or periodic control systems, $t_{i+1} - t_i = T_s$ for all $i \in \{0, 1, 2, \ldots\}$, where $T_s > 0$ is a constant sampling time. On the other hand, in an event-triggered controller the time instants t_i are defined implicitly by a triggering condition. The sampled nature of the signals can be alternatively viewed as time-continuous signals, albeit with an error in their measurement. Thus, we define the measurement error as

$$e \triangleq \xi_s - \xi = \xi(t_i) - \xi, \quad \text{for } t \in [t_i, t_{i+1}),\ i \in \{0, 1, 2, ...\}$$

Note that e is discontinuous at $t = t_i$, for each i, because $e(t_i) = \xi(t_i) - \xi(t_i) = 0$ while $e(t_i^-) \triangleq \lim_{t\uparrow t_i} e(t) = \lim_{t\uparrow t_i}(\xi(t_{i-1}) - \xi(t))$. The sampled data controller is then given as

$$u_s = \gamma(\xi_s) = \gamma(\xi) + \big(\gamma(\xi_s) - \gamma(\xi)\big)$$

The second relation is useful in expressing the sampled data control as a perturbation of the continuous-time control law.
It is a well-known fact that the Lagrangian robot dynamics are linearly parametrizable [13]. Specifically, the following is true.

$$M(q)a + C(q,w)v + G(q) = Y(q,w,v,a)\theta$$

where θ is the vector of parameters, w, v and a are arbitrary real vectors of appropriate size. The specific forms of w, v and a depend on the context.
Now we give the essential elements of the event-triggered implementations of two well known adaptive controllers for tracking in robot manipulators, proposed in [11] and [12], respectively. Detailed design of the event-triggered implementation for the first one can be found in our previous work [10]. The design procedure for the latter case is similar.
Important Note: The arguments of matrix Y differ in the two cases we discuss. This has to be kept in mind because after the initial definitions, the arguments are generally avoided to keep the notation compact.

2.1 Case I: Event-Triggered Implementation of Beghuis et al. [11]

In this case the controller is an event-triggered sampled-data implementation of the adaptive controller proposed in [11]. The complete system description is then as follows (description of some of the variables is given after the equations).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u_s, \quad q \in \mathbb{R}^n \tag{1}$$

$$\begin{aligned} u = \gamma(\xi) &= \hat{M}(q)\ddot{q}_d + \hat{C}(q,\rho)\dot{q}_d + \hat{G}(q) - K_d\dot{\tilde{q}} - K_p\tilde{q} \\ &= Y(q,\rho,\dot{q}_d,\ddot{q}_d)\hat{\theta} - K_d\dot{\tilde{q}} - K_p\tilde{q} \end{aligned}$$

$$u_s = \gamma(\xi_s) = Y_s\hat{\theta}_s - K_d\dot{\tilde{q}}_s - K_p\tilde{q}_s \tag{2}$$

$$\xi_s(t) = \xi(t_i), \quad \text{for all } t \in [t_i, t_{i+1}), \quad \text{for each } i$$

$$t_0 = 0$$

$$t_{i+1} = \min\{t \geq t_i : \beta(\tilde{x})L^T|\xi(t_i) - \xi(t)| \geq \sigma\alpha(\tilde{x})\}, \quad 0 < \sigma < 1 \tag{3}$$

$$\dot{\hat{\theta}} = -\Gamma^{-1}Y_s^T\psi \tag{4}$$

where the variables with the subscript s are sampled versions of the corresponding signals, the variables with a $\hat{}$ on top of them are current estimates of the corresponding variables. The desired trajectory is $[q_d; \dot{q}_d]$, the tracking error is $\tilde{x} \triangleq [\tilde{q}; \dot{\tilde{q}}] \triangleq [q - q_d; \dot{q} - \dot{q}_d]$, $\rho \triangleq \dot{q} - \lambda\tilde{q}$ and $\psi \triangleq \dot{\tilde{q}} + \lambda\tilde{q}$, where

$$\lambda = \frac{\lambda_0}{1 + \|\tilde{q}\|}, \quad \lambda_0 > 0$$

where $\|.\|$ denotes the Euclidean norm. The other gains are $K_d = K_d^T > 0$, $K_p = K_p^T > 0$, $\Gamma = \Gamma^T > 0$ and σ is a design parameter. The functions α and β are given by

$$\alpha(\tilde{x}) = k_1\left\|\dot{\tilde{q}} + \frac{\lambda}{2}\tilde{q}\right\|^2 + k_2\left\|\frac{\lambda}{2}\tilde{q}\right\|^2, \quad \beta(\tilde{x}) = \|\psi\| = \|\dot{\tilde{q}} + \lambda\tilde{q}\|$$

where k_1 and k_2 are given by

$$k_1 = K_{d,m} - 3\lambda_0 M_M - 2\lambda_0 C_M$$
$$k_2 = 4\lambda_0^{-1} K_{p,m} - K_{d,M} - 2\lambda_0 M_M - 2\lambda_0 C_M \tag{5}$$

The constants in (5), which depend on the robot dynamics and the gains of the controller are assumed to satisfy the following assumption.

(A1) Assume that the controller gains are chosen such that

$$\lambda_0 < \min\left\{ \frac{K_{d,m}}{3M_M + 2C_M}, \frac{4K_{p,m}}{K_{d,M} + K_{d,m}} \right\}$$

where $K_{d,m} \equiv \sigma_m(K_d)$, $K_{d,M} \equiv \sigma_M(K_d)$, $K_{p,m} \equiv \sigma_m(K_p)$, with $\sigma_m(.)$, $\sigma_M(.)$ the minimum and maximum eigenvalues respectively. The constants M_m, M_M and C_M satisfy

$$0 < M_m \leq \|M(q)\| \leq M_M, \quad \|C(q,w)\| \leq C_M\|w\|, \quad \text{for all } w \tag{6}$$

where w denotes an arbitrary vector.

The other assumptions we make are

(A2) The norm of the desired trajectory $[q_d; \dot{q}_d]$, and that of its first two derivatives are uniformly bounded by known constants. That is, q_d, $\dot{q}_d$, $\ddot{q}_d$ and $\dddot{q}_d$ exist for all time, and their norms are uniformly bounded by known constants d_0, d_1, d_2 and d_3, respectively.

(A3) The matrices $M(.)$, $C(.,.)$ and $G(.)$ are globally Lipschitz.

Finally, the vector L is a quantity that depends on the sampled data ξ_s. The precise form depends on the specific robot. The purpose of this vector is to bound certain quantities resulting from the perturbation due to the sampled data nature of the controller. The vector L satisfies the essential property described in Lemma 1, which is taken from [10] (and modified to use a more compact notation). In the sequel, the notation $|.|$ denotes the component-wise absolute value of a vector or matrix and a *Lipschitz vector* is similar to a Lipschitz constant. More specifically, it is a vector of non-negative elements other than the zero vector.

Lemma 1. *Suppose that assumptions* (A2),(A3) *and conditions* (6) *hold. Also assume that* $K_p > 0$ *and* $K_d > 0$*. Then, there exists a Lipschitz vector* L *that depends only on the sampled data, and the uniform bound on* $\dot{q}_d$ *such that*

$$\|(Y_s - Y)\theta\| + \|Y_s(\hat{\theta}_s - \hat{\theta})\| + \|K_d(\dot{\tilde{q}}_s - \dot{\tilde{q}}) + K_p(\tilde{q}_s - \tilde{q})\| \leq L^T|e|$$

Equations (2)-(3) provide a complete description of the event-triggered controller. The condition that implicitly defines the sampling instants, (3), is the triggering condition. Finally, (4) is the adaptation law for estimating the dynamic uncertainty in the system. The following result, taken from [10], says that with the event-triggered controller, the tracking error in the closed loop system globally converges to zero.

Theorem 1. *Under assumptions* (A1)-(A3) *and dynamics* (1)-(4)*, the tracking error,* $\tilde{x} = [\tilde{q}; \dot{\tilde{q}}]$ *globally asymptotically converges to zero.*

The proof of this result relies on a Lyapunov like analysis with the candidate Lyapunov function

$$V(\tilde{q},\dot{\tilde{q}}) = \frac{1}{2}\psi^T M(q)\psi + \frac{1}{2}\tilde{q}^T K_p \tilde{q} + \frac{1}{2}\tilde{\theta}^T \Gamma \tilde{\theta}$$

where $\tilde{\theta} \triangleq \hat{\theta} - \theta$. Note that, due to uncertain dynamics of the robot, $M(q)$, $\tilde{\theta}$ and thus $V(\tilde{q},\dot{\tilde{q}})$ are not known precisely. However, it can be shown that the derivative of the Lyapunov function along the flow of the closed loop system (1)-(4) satisfies

$$\begin{aligned}\dot{V} &\leq -\alpha(\tilde{x}) + \psi^T\left[(Y_s - Y)\theta + Y_s(\hat{\theta}_s - \hat{\theta}) - K_d(\dot{\tilde{q}}_s - \dot{\tilde{q}}) - K_p(\tilde{q}_s - \tilde{q})\right] \\ &= -\alpha(\tilde{x}) + \beta(\tilde{x})L^T|e| \leq -(1-\sigma)\alpha(\tilde{x})\end{aligned}$$

which involves only the known quantities. The triggering condition (3) is derived from the second last relation.

2.2 Case II: Event-Triggered Implementation of Slotine-Li [12]

In this case, we present an event-triggered sampled-data implementation of the adaptive controller from [12]. This implementation has a structure similar to CASE I, with only minor differences.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u_s, \quad q \in \mathbb{R}^n \tag{7}$$

$$u = \gamma(\xi) = \hat{M}(q)a + \hat{C}(q,\dot{q})v + \hat{G}(q) - Kr = Y(q,\dot{q},v,a)\hat{\theta} - Kr \tag{8}$$

$$u_s = \gamma(\xi_s) = Y_s\hat{\theta}_s - Kr_s \tag{9}$$

$$\xi_s(t) = \xi(t_i), \quad \text{for all } t \in [t_i, t_{i+1}), \quad \text{for each } i$$

$$t_0 = 0$$

$$t_{i+1} = \min\{t \geq t_i : \beta(\tilde{x})L^T|\xi(t_i) - \xi(t)| \geq \sigma\alpha(\tilde{x})\}, \quad 0 < \sigma < 1 \tag{10}$$

$$\dot{\hat{\theta}} = -\Gamma^{-1}Y_s^T r \tag{11}$$

where $r \triangleq \dot{\tilde{q}} + \lambda\tilde{q}$, $v \triangleq \dot{q}_d - \lambda\tilde{q}$, $a \triangleq \ddot{q}_d - \lambda\dot{\tilde{q}}$, $\lambda > 0$, $K > 0$, $\Gamma = \Gamma^T > 0$ are constant gains and σ is a design parameter. Note that, unlike in CASE I, α and β are positive constants, which are given by

$$\alpha(\tilde{x}) = \min(K, K\lambda^2)\|\tilde{x}\|^2, \quad \beta(\tilde{x}) = \sqrt{(\lambda^2+1)}\|\tilde{x}\|$$

The Lipschitz vector satisfies a similar property as in CASE I, which is summarized in the following Lemma.

Lemma 2. *Suppose that assumptions* (A2),(A3) *and conditions* (6) *hold. Also assume that $K_p > 0$ and $K_d > 0$. Then, there exists a Lipschitz vector L that depends only on the sampled data, and the uniform bound on $\dot{q}_d$ such that*

$$\|(Y_s - Y)\theta\| + \|Y_s(\hat{\theta}_s - \hat{\theta})\| + \|K(r_s - r)\| \leq L^T|e|$$

Again, it can be shown that with the event-triggered controller the tracking error in the closed loop system globally converges to zero, which is captured in the following result.

Theorem 2. *Under assumptions* (A2)-(A3) *and dynamics* (7)-(11)*, the tracking error,* $\tilde{x} = [\tilde{q}; \dot{\tilde{q}}]$ *globally asymptotically converges to zero.*

The proof of this result is again based on a Lyapunov like analysis with the candidate Lyapunov function

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2} r^T M(q) r + \lambda K \tilde{q}^T \tilde{q} + \frac{1}{2} \tilde{\theta}^T \Gamma \tilde{\theta} \tag{12}$$

where $\tilde{\theta} \triangleq \hat{\theta} - \theta$. Note that, due to uncertain dynamics of the robot, $M(q)$, $\tilde{\theta}$ and thus $V(\tilde{q}, \dot{\tilde{q}})$ are not known precisely. However, it can be shown that the derivative of the Lyapunov function along the flow of the closed loop system (7)-(11) satisfies

$$\begin{aligned} \dot{V} &\leq -\alpha(\tilde{x}) + r^T \left[(Y_s - Y)\theta + Y_s(\hat{\theta}_s - \hat{\theta}) - K(r_s - r) \right] \\ &= -\alpha(\tilde{x}) + \beta(\tilde{x}) L^T |e| \leq -(1 - \sigma)\alpha \|\tilde{x}\|^2 \end{aligned}$$

In the next section, we present the dynamic model of a two link planar manipulator and the corresponding L vectors.

3 Two Link Planar Manipulator

In this section we describe the dynamic model of a planar two-link revolute joint arm, with both the joints driven by motors mounted at the base. We choose this model because of a similar driving mechanism in PHANToM Omni. A schematic of the arm is shown along with the generalized coordinates in Figure 3. The $M(q)$, $C(q, \rho)$ and $G(q)$ matrices

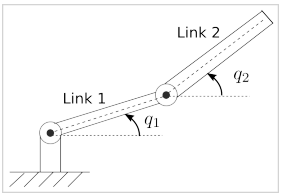


Fig. 3: A schematic of a two link planar revolute manipulator with the second link remotely driven from base of Link 1.

can be easily derived from the Euler-Lagrange equations or can be found in books on robot modeling (for example, see pages 262-264 [13]). The regressor matrix, $Y(q, w, v, a)$, is given as

$$Y = \begin{bmatrix} a_1 & a_2 \cos(q_2 - q_1) - v_2 \sin(q_2 - q_1) w_2 & 0 & \cos(q_1) & 0 \\ 0 & a_1 \cos(q_2 - q_1) + v_1 \sin(q_2 - q_1) w_1 & a_2 & 0 & \cos(q_2) \end{bmatrix}$$

and the column vector of parameters is given as

$$\begin{aligned} \theta &= \left[\theta_1;\ m_2 l_1 l_{c_2};\ m_2 l_{c_2}^2 + I_2;\ (m_1 l_{c_1} + m_2 l_1) g;\ m_2 l_{c_2} g \right] \\ \theta_1 &= m_1 l_{c_1}^2 + m_2 l_1^2 + I_1 \end{aligned} \tag{13}$$

Now, the vector L is presented for the two cases that we considered in Section 2.

CASE I:

$$L = \begin{bmatrix} \bar{\theta}_2(\phi_{s,1} + d_1\phi_{s,2}) + \bar{\theta}_4 \\ \bar{\theta}_2(\phi_{s,1} + d_1\phi_{s,2}) + \bar{\theta}_5 \\ \bar{\theta}_2 d_1 \\ \bar{\theta}_2 d_1 \\ \bar{\theta}_2\phi_{s,1} + \bar{\theta}_4 \\ \bar{\theta}_2\phi_{s,1} + \bar{\theta}_5 \\ \bar{\theta}_2(|\rho_{s,1}| + d_1) \\ \bar{\theta}_2(|\rho_{s,2}| + d_1) \\ \bar{\theta}_1 + \bar{\theta}_2 \\ \bar{\theta}_3 + \bar{\theta}_2 \\ \mathbf{0} \end{bmatrix} + N + D$$

$$\phi_{s,1} = |\rho_{s,1}\dot{q}_{d,s,1}| + |\rho_{s,2}\dot{q}_{d,s,2}| + |\ddot{q}_{d,s,1}| + |\ddot{q}_{d,s,2}|$$

$$\phi_{s,2} = \lambda_s(1 + 2|\tilde{q}_{s,1}| + 2|\tilde{q}_{s,2}|)$$

where $\bar{\theta}_i$ is a known upper bound on the parameter θ_i, $\rho_{s_k} = \dot{q}_{s,k} - \lambda_s\tilde{q}_{s,k}$, for $k = 1, 2$, d_1 is the uniform bound on $|\dot{q}_{d,1}|$ and $|\dot{q}_{d,2}|$. The notation $\mathbf{0}$ denotes a vector of zeros of appropriate dimension. The vectors N and D are given as $N = [\mathbf{0}^T,\ \text{Column-wise sum of } |Y_s|\,]^T$ and $D = [K_p; K_p; K_d; K_d; \mathbf{0}]$, respectively. Notice that most of the elements in these vectors are constants or easily computable functions of the sampled data.

CASE II:

$$L = \begin{bmatrix} \bar{\theta}_2(\phi_s + \lambda(\mu + d_1)) + \bar{\theta}_4 \\ \bar{\theta}_2(\phi_s + \lambda(\mu + d_1)) + \bar{\theta}_5 \\ \bar{\theta}_2(\lambda + |v_{s,1}|) + \bar{\theta}_1\lambda \\ \bar{\theta}_2(\lambda + |v_{s,2}|) + \bar{\theta}_3\lambda \\ \bar{\theta}_2\phi_s + \bar{\theta}_4 \\ \bar{\theta}_2\phi_s + \bar{\theta}_5 \\ \bar{\theta}_2(\mu + d_1 + |v_{s,1}|) \\ \bar{\theta}_2(\mu + d_1 + |v_{s,2}|) \\ \bar{\theta}_1 + \bar{\theta}_2 \\ \bar{\theta}_3 + \bar{\theta}_2 \\ \mathbf{0} \end{bmatrix} + N + D$$

$$\phi_s = |v_{s,1}\dot{q}_{s,1}| + |v_{s,2}\dot{q}_{s,2}| + |a_{s,1}| + |a_{s,2}|, \qquad \mu = \sqrt{\frac{V_{est}}{\alpha_1}}$$

where $\bar{\theta}_i$ is a known upper bound on the parameter θ_i, $\rho_{s_k} = \dot{q}_{s,k} - \lambda_s\tilde{q}_{s,k}$, for $k = 1, 2$, d_1 is the uniform bound on $|\dot{q}_{d,1}|$ and $|\dot{q}_{d,2}|$. The notation $\mathbf{0}$ denotes a vector of zeros of appropriate dimension. The vectors N and D are given as $N = [\mathbf{0}^T,\ \text{Column-wise sum of } |Y_s|\,]^T$ and $D = [\lambda K; \lambda K; K; K; \mathbf{0}]$, respectively. The quantity V_{est} is an upper bound on the value of the Lyapunov function (12) at the sampling instant. Even though the precise value of the Lyapunov function cannot be known, it can still be bounded by known quantities.

$$\alpha_1\|\tilde{x}_s\|^2 \leq V(\tilde{q}_s, \dot{\tilde{q}}_s) \leq \alpha_2\|\tilde{x}_s\|^2 + \frac{1}{2}H^T\Gamma H = V_{est}$$

where α_1 and α_2 are positive constants. These constants can be found from either the lower and upper bounds on $\|M(q)\|$, M_m and M_M, respectively or from the individual bounds on the parameters, $\bar{\theta}_i$. H is a vector such that $H_i \geq |\tilde{\theta}_{s,i}|$ for each i. This vector can be estimated as

$$H_i = \begin{cases} \bar{\theta}_i - \hat{\theta}_{s,i}, & \text{if } \hat{\theta}_{s,i} < \bar{\theta}_i \\ \hat{\theta}_{s,i}, & \text{if } \hat{\theta}_{s,i} \geq \bar{\theta}_i \end{cases}$$

Notice that computation of the L vector is more involved than in CASE I. This is mainly because in CASE I, the arguments ρ, $\dot{q}_d$ and $\ddot{q}_d$ in the Y matrix are all uniformly bounded by known constants and do not have to be estimated at each sampling instant.

4 Experiments

In the experimental results presented here, the position variables of the desired trajectory were chosen as

$$q_{d,1} = -0.4(\cos(0.8t) - 1.1), \quad q_{d,2} = -0.4(\cos(0.3\pi t) - 1) - (\pi/2)$$

The signals $\dot{q}_d$, $\ddot{q}_d$ and $\dddot{q}_d$ were defined simply as the corresponding derivatives of q_d. The control gains and the parameters were heuristically chosen as

$$\bar{\theta} = [0.0035, 0.0035, 0.002, 0.2, 0.1]^T$$
$$d_1 = 0.5, \quad h_l = 10^{-8}, \quad \sigma \in \{0.95, 0.6, 0.2\}$$
CASE I: $\lambda_0 = 0.7,\ K_d = 0.03,\ K_p = 0.7$
$$\Gamma = \text{diag}([30, 40, 50, 10, 10]^T)$$
CASE II: $\lambda = 1.5, \quad K = 0.03, \quad \Gamma = \text{diag}([250, 350, 400, 0.8, 0.9]^T)$

where d_1 is the uniform upper bound on $|\dot{q}_{d,1}|$ and $|\dot{q}_{d,2}|$ and h_l is a lower bound on $(\theta_1\theta_3 - \theta_2^2)$, which can be easily shown to be positive for a two link manipulator. Using these quantities, M_M, M_m and C_M can be estimated as

$$M_M = \frac{\bar{\theta}_1 + \bar{\theta}_3 + \sqrt{(\bar{\theta}_1 + \bar{\theta}_3)^2 - 4h_l}}{2}$$
$$M_m = \frac{\bar{\theta}_1 + \bar{\theta}_3 - \sqrt{(\bar{\theta}_1 + \bar{\theta}_3)^2 - 4h_l}}{2}, \quad C_M = \bar{\theta}_2$$

which are useful to compute α_1 and α_2 in CASE II. Finally, the initial conditions of the system were chosen as

$$[q_1, q_2, \dot{q}_1, \dot{q}_2]^T(0) = [0, -\pi/2, 0, 0]^T$$
$$\hat{\theta}(0) = [0.0001, 0.0001, 0.0001, 0.01, 0.001]^T$$

The choice of such low initial values for $\hat{\theta}$ is motivated by the fact that initial torques will be lower in the absence of knowledge of the system

parameters. Notice from (13) that the first three parameters of the robot involve quadratic terms in the length, while the last two involve only linear terms in the length and also involve acceleration due to gravity, g. Thus, for the PHANToM Omni, which is a desktop robot with small link lengths and small link masses, the last two parameters are larger than the others by some orders of magnitude. Thus intuitively, the adaptation gain, Γ^{-1}, has to be chosen such that $\hat{\theta}_4$ and $\hat{\theta}_5$ evolve at a faster rate. The other control gains were also chosen by heuristic means. For example, K_d in CASE I and K in CASE II multiply velocity terms. Since the joint velocity measurements obtained from the device were very noisy, we have chosen relatively smaller values for these gains to limit buzzing, or high frequency and high magnitude torques.

Experiments were performed on a two-link planar manipulator. PHANToM Omni was used as a test bed. For the experiments presented here, only the second and third joint were kept active, to keep the calculations simple. The first joint was never actuated and the remaining joints were either removed or constrained to a fixed position. The OpenHaptics 3.0 [14] was used to program the PHANToM Omni for performing tasks such as reading the sensors and controlling the joint torques. The OpenHaptics 3.0 API does not provide the capability to arbitrarily choose the sampling and control update instants. The sensors are sampled and control torques are updated with a roughly constant period (1ms for the results presented here). Hence, in the experiments the event-triggering condition was checked roughly every 1ms.

Some experimental results for Case I are presented in Figure 4, while the results for Case II are presented in Figure 5. These figures show the desired joint positions and the actual joint positions as the robot is tracking. Notice that for smaller values of σ the tracking performance increases - this is specially clear in joint 2 data of Figure 5. On the other hand, there is a greater overshoot for $\sigma = 0.2$ in the same figure. The improvement in tracking with decrease in the design parameter σ value is to be expected because it has a clear inverse relationship with the average sampling frequency.

From these figures it appears that the tracking error convergence rate and tracking performance are better in Case I than in Case II, particularly with respect to Joint 2. It must be pointed out that there is considerable friction in the joints, which has not been included in the system model. This friction accounts for some of the persisting tracking error. Similarly, to avoid 'buzzing' some of the gains had to be limited to smaller values as the joint velocity measurements are noisy. Another possible factor influencing the bad tracking of Joint 2 in CASE II is the quadratic dependence on joint velocity in the control law (8).

The observed minimum inter-update times and average frequency in experiments are reported in Table 1. In the Phantom Omni system, the control inter-update period is inherently fixed at 1ms, which thus limits the minimum inter-update time of the control value. Figure 6 shows the cumulative frequency distribution of the control inter-update times for different values of σ and for each of the cases. Finally, the irregular state based sampling and control updates is illustrated in Figure 7.

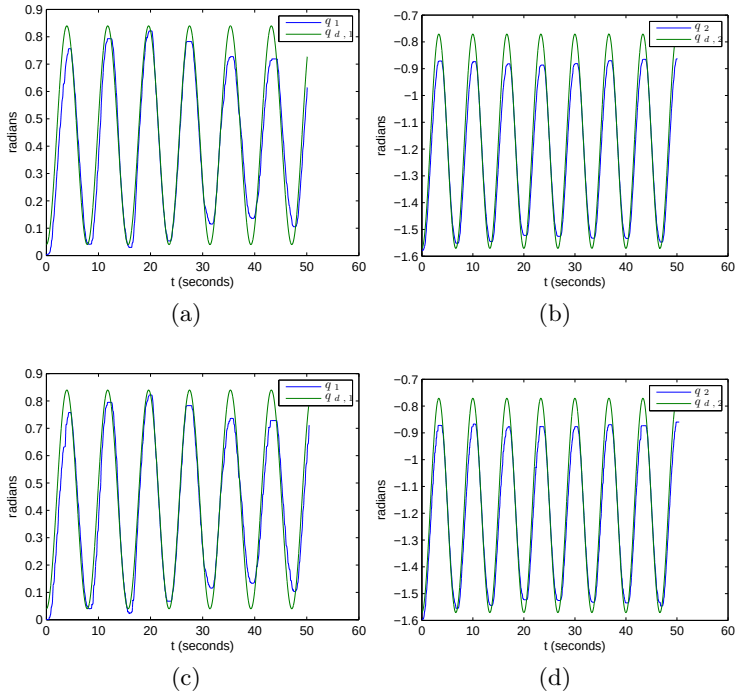


Fig. 4: The desired joint positions and the actual positions of the robot in Case I. (a), (b) $\sigma = 0.2$, (c), (d) $\sigma = 0.95$

Table 1: The observed minimum inter-update times and average frequency in experiments.

	Case I		Case II	
σ	Observed min. inter-update time (s)	Observed avg. Frequency (Hz)	Observed min. inter-update time (s)	Observed avg. Frequency (Hz)
0.2	4.7×10^{-4}	164	7×10^{-4}	298
0.6	7.4×10^{-4}	64	9.7×10^{-4}	106
0.95	9.9×10^{-4}	40	9.9×10^{-4}	63

5 Discussion and Conclusions

In this paper, a non-periodic state based control sampling technique was experimentally investigated for two well known controllers for tracking in robot manipulators. The experimental results demonstrate the promise that event based algorithms hold in robotic applications.
In the experiments, the parameters and control gains were not optimized. They were chosen by trial-and-error and some heuristic insights to give

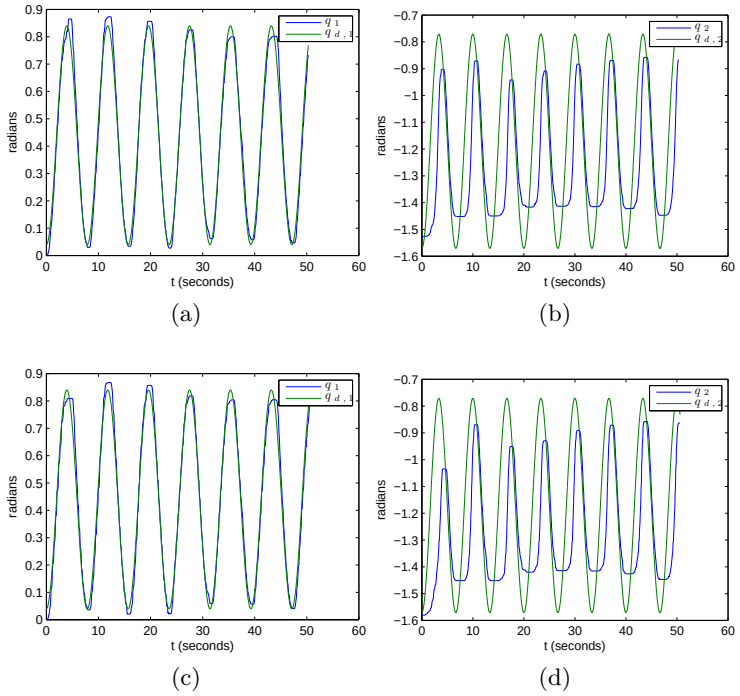


Fig. 5: The desired joint positions and the actual positions of the robot in Case II. (a), (b) $\sigma = 0.2$, (c), (d) $\sigma = 0.95$

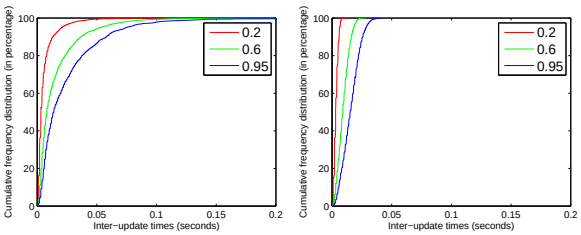
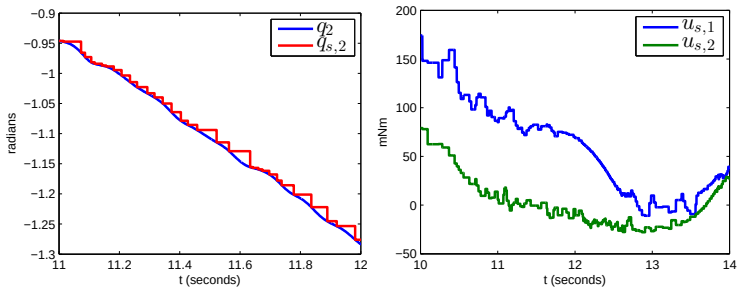


Fig. 6: The cumulative frequency distribution of the control inter-update times in the experiments for different values of σ.

'good' performance, in terms of low average sampling rates and minimizing tracking error. Similarly, the results presented here are for a single desired trajectory. Yet, these results provide valuable insights about the numerous factors affecting the performance.

The experimental results presented here suggest that the controller in CASE I is better suited for Lyapunov based sampling. It must be pointed



(a) (b)

Fig. 7: Irregular state based sampling and control updates.

out that in CASE II, joint 1 is tracked very well. However, joint 2 tracking is not good. No significant improvement was observed by changing the control gains. By changing the adaptation gains, there was some improvement in tracking performance, though at the cost of increased average sampling frequency. Another possible cause of bad tracking performance and higher average sampling rates in CASE II, is the highly noisy joint velocity data. These factors will be investigated more systematically in future studies.

References

[1] P.K. Khosla. Choosing sampling rates for robot control. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 169–174, 1987.

[2] T.J. Tarn, S. Ganguly, A.K. Ramadorai, G.T. Marth, and A.K. Bejczy. Experimental evaluation of the nonlinear feedback robot controller. In *IEEE International Conference on Robotics and Automation*, pages 1638–1644, 1991.

[3] D. Simon, E. Castillo Castaneda, and P. Freedman. Design and analysis of synchronization for real-time closed-loop control in robotics. *IEEE Transactions on Control Systems Technology*, 6(4):445–461, 1998.

[4] M.B. Leahy Jr. Industrial manipulator control with feedforward dynamic compensation. In *IEEE Conference on Decision and Control*, pages 598–603, 1988.

[5] G. Alici and R.W. Daniel. Experimental comparison of model-based robot position control strategies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 76–83, 1993.

[6] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

[7] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, 2008.

[8] X. Wang and M.D. Lemmon. Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability. *IEEE Transactions on Automatic Control*, 54:452–467, 2009.

[9] P. Tallapragada and N. Chopra. On event triggered trajectory tracking for control affine nonlinear systems. In *IEEE Conference on Decision and Control and European Control Conference*, pages 5377–5382, 2011.

[10] P. Tallapragada and N. Chopra. Lyapunov based sampling for adaptive tracking control in robot manipulators. *IEEE Transactions on Robotics*. Submitted.

[11] H. Berghuis, R. Ortega, and H. Nijmeijer. A robust adaptive controller for robot manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1876–1881, 1992.

[12] J. Slotine and L. Weiping. Adaptive manipulator control: A case study. *IEEE Transactions on Automatic Control*, 33(11):995–1003, 1988.

[13] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc., New York, 2006.

[14] SensAble Technologies *OpenHaptics Toolkit 3.0*, Jan 2009, http://www.sensable.com/products-openhaptics-toolkit.htm.