

Passivity-Based Trajectory Tracking Control with Adaptive Sampling Over a Wireless Network

Emeka Eyisi and Xenofon Koutsoukos and Nicholas Kottenstette

Abstract—Uncertainty in wireless networks, such as time-varying delays and packet loss, often leads to instability and degraded performance in Networked Control Systems (NCS). Further, limited network resources impose constraints on communication between plants and controllers. In this paper, we use passivity combined with adaptive sampling to design a control architecture for trajectory tracking. The proposed architecture can tolerate time-varying delays and packet loss while efficiently utilizing network bandwidth. We provide analytical results to show passivity of the proposed networked control architecture and trajectory tracking. We demonstrate our approach using a case study on the trajectory tracking control of a robotic manipulator over a wireless network. The simulation results show the efficient utilization of network resources as well as robustness to network uncertainties.

I. INTRODUCTION

This paper addresses the challenges in designing Networked Control Systems (NCS), posed by the limitations and unreliability of wireless networks, by integrating passivity-based control with adaptive sampling. Passivity has been exploited as a major tool for the design of NCS because it provides significant advantages in dealing with network delays, data dropouts, and quantization [1]–[4]. Using passivity in NCS has resulted in many novel approaches in control over communication networks (see [2]–[5] and references therein). However, a precise formalism demonstrating how to handle adaptive sampling in order to utilize network resources, especially for control objectives such as tracking, is lacking.

Adaptive sampling has become very important due to a strong need for the efficient use of computational and network resources [6], [7], [8]. The main idea of adaptive sampling is to compute the sample interval, T_i , at which a control law needs to be executed based on certain criteria such as the current state of the system, a function of a desired system characteristics such as error or even based on the quality of service (QoS) of the communication network. Over the past years, various adaptive sampling techniques have been developed [9] [10] [11] [12] and references therein. An example of such an adaptive sampling scheme is self-triggered control. Self-triggered control, first introduced by [7], uses a model of the system dynamics to perform computations in order to predict when the violation of a defined event condition will occur and based on this prediction,

appropriate sampling intervals are determined. Several self-triggered implementations have been proposed [13] [14] [15] [16] and references therein.

The main contribution of this work is that we achieve trajectory tracking in a hierarchical NCS [17] using a passivity-based approach integrated with adaptive sampling. This approach guarantees stability of the NCS in the presence of network uncertainties. The integration of adaptive sampling not only ensures the efficient utilization of network resources but also provides the flexibility of incorporating network scheduling adaptation. Additionally, we demonstrate our approach with a case study on the trajectory tracking control of a robotic manipulator over a wireless network. We provide simulation results using Matlab/Simulink/TrueTime. We show that the robotic manipulator tracks a desired trajectory while reducing the utilization of network resources compared to the case of a fixed sampling period. We also provide simulation results to demonstrate the robustness of our approach under various network uncertainties such as time-varying delays and packet loss.

The rest of the paper is organized as follows: The problem statement is provided in Section II. The proposed architecture is introduced in Section III. Analysis of the proposed approach is provided in Section IV. Section V presents an example case study. Finally, Section VI provides concluding remarks and directions for future work.

II. PROBLEM STATEMENT

This paper considers the problem of achieving trajectory tracking in a networked control system in the presence of network uncertainties and limited network bandwidth. Consider the system model H_{mp} shown in Fig. 1, H_p (which

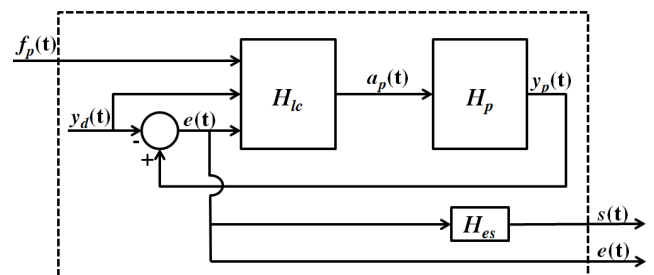


Fig. 1. System Model, H_{mp} .

could be linear or nonlinear) is a plant with input and output defined as a_p and y_p respectively. e is the error defined as the difference between the plant output, y_p and a reference trajectory, y_d . H_{lc} is a local controller which takes f_p , y_d

Emeka Eyisi, Xenofon Koutsoukos are with the EECS Department, Vanderbilt University, Nashville, TN, USA, [emeka.eyisi, xenofon.koutsoukos]@vanderbilt.edu

Nicholas Kottenstette is with WW Technology Group, USA, nicholas.e.kottenstette@ieee.org

and e as inputs and provides an output a_p which is fed into the plant. H_{es} is defined as a function that relates the output s , as shown in the Fig. 1, to the error, e .

We provide the following definitions of passivity [18]

. **Definition 1:** A dynamical system, H ,

$$H : \begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases} \quad (1)$$

is said to be **passive** if there exists a non-negative function $V : \mathbb{R}^n \mapsto \mathbb{R}$, called storage function, such that for all $u \in U$, for all $x(0) = x_0$, for all $y \in Y$ and $t \geq 0$, it satisfies

$$\int_0^t y^\top(\tau)u(\tau)d\tau \geq V(x(t)) - V(x_0) \geq -\beta \quad (2)$$

where $U \subset \mathbb{R}^n$ is the set of input values, $X \subset \mathbb{R}^m$ is the state space and $Y \subset \mathbb{R}^n$ is a set of output values and $\beta = V(x_0)$. If $V(x)$ is C^1 , which implies differentiable, then (2) can also be expressed as $\dot{V} \leq y^\top(t)u(t)$

1) H is said to be **strictly input passive** if

$$\int_0^t y^\top(\tau)u(\tau) - u^\top(\tau)\phi u(\tau)d\tau \geq -\beta \quad (3)$$

where $u^\top(\tau)\phi u(\tau) > 0$ for some function ϕ and $\forall u \neq 0$

2) H is **strictly output passive** if

$$\int_0^t y^\top(\tau)u(\tau) - y^\top(\tau)\eta y(\tau)d\tau \geq -\beta \quad (4)$$

where $y^\top(\tau)\eta y(\tau) > 0$ for some function η and $\forall y \neq 0$

The discrete-time equivalent of the above definitions is provided in [19].

We assume the following about the system model, H_{mp}

A1: The input-output pair, with input, f_p and output, s as shown in Fig. 1 can be defined such that a map, $H_{mp} : f_p \mapsto s$ is passive.

A2: H_{mp} is zero-state detectable [18].

A3: There exists a function, H_{es} , relating e to s with Λ defined as a positive diagonal matrix, in the form,

$$s = \dot{e} + \Lambda e \quad (5)$$

such that if $s \rightarrow 0$ then

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (6)$$

. **A4:** The desired reference trajectory, y_d , is bounded and twice differentiable.

In regards to the communication over the network, we assume the following on the packet handling of network messages:

A5: The networked packet handling mechanism is designed to ensure that no duplicate packet is processed.

A6: In the case of packet loss, if the input buffer is empty, null packets are processed.

Assumptions **A5** and **A6** ensure that no additional energy is introduced by the communication channel.

Based on these assumptions, we address two main problems in this paper:

1) We consider the problem of designing a robust hierarchical networked control system for achieving the following tracking objective in an unreliable wireless network:

a) Ensure that, y_p , the output of the plant, H_p , tracks a local reference trajectory, y_d such that

$$\lim_{t \rightarrow \infty} y_p(t) - y_d(t) = 0 \quad (7)$$

b) In the presence of a non-local reference input, r_c , which can be used to model behaviors such as the presence of an obstacle in the plant's environment, the plant output tracks a modified reference signal, y_{da} such that

$$\lim_{t \rightarrow \infty} y_p(t) - y_{da}(t) = \lim_{t \rightarrow \infty} y_p(t) - y_d(t) - r_c(t) = 0 \quad (8)$$

Hence, ensuring that the plant avoids the obstacle.

2) We consider the problem of reducing the amount of network resources utilized while ensuring that the tracking objective is achieved.

III. PASSIVITY-BASED ADAPTIVE SAMPLING CONTROL ARCHITECTURE

Fig. 2, depicts the proposed networked control architecture which we refer to as passivity-based adaptive sampling control architecture (PBASC). This architecture is designed to achieve trajectory tracking as well as the efficient use of network resources. The remote system and the networked controller are shown on the left and right sides of the network in Fig. 2 respectively. On the remote system side of the network, the block H_{mp} represents the system model introduced in Fig. 1. In what follows we describe the components of Fig. 2.

A. Adaptive Sampling Scheme

The sampling policy box, in Fig. 2 depicts an adaptive sampling scheme which outputs the sampling intervals. In order to interconnect the system, H_{mp} to the digitally implemented networked controller, a sample-and-hold mechanism is needed. Traditionally, the sample-and-hold mechanism is based on fixed sampling periods but in order to efficiently utilize network bandwidth we use an adaptive sampling scheme. The adaptive sampling mechanism can be designed based on any of the existing adaptive sampling techniques in order to generate sampling intervals. For the purpose of tracking, we need a mechanism that determines the sampling intervals based on a function of tracking error. We chose the adaptive sampling mechanism based on the self-triggered control concept described in [15]. This approach enables us to specify adaptive sampling scheme in terms of a storage function, a lyapunov-like function of the tracking error. The chosen triggering condition is based on a passive mapping.

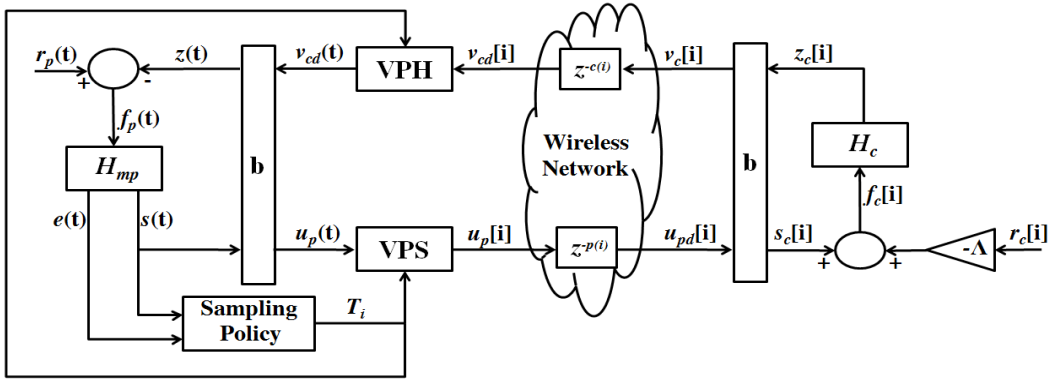


Fig. 2. PBASC Networked Control Architecture.

Our triggering mechanism is based on the system model, H_{es} , relating the tracking error, e , with the output of H_{mp} , s . Rearranging and restating (5), we obtain the system model defined as

$$\dot{e}(t) = -\Lambda e(t) + s(t); \quad (9)$$

$$s(t) = s(t_i), \quad t \in [t_i, t_{i+1}) \quad (10)$$

where $[t_i]_{i \in \mathbb{N}}$ is an increasing sequence of sampling times with $t_0 = 0$. This model represents a strictly output passive mapping with the tracking error, e , as the output and s , as the input. The system model in (9) and (10) could be represented in a minimal state-space realization, where the tracking error, e , is the state as well as the output of the system and s , is the input. This results in the state-space coefficients where $A = -\Lambda$, $B = C = I$, the identity matrix, and $D = \mathbf{0}$.

A map, Γ_d , can then be used to define the self-triggered implementation of the system model given by (9) and (10). This map determines the t_{i+1} , as a function of the tracking error output, e at the time t_i , i.e., $t_{i+1} = t_i + \Gamma_d(e(t_i))$. If we denote by T_i , the sampling interval $T_i = t_{i+1} - t_i$ we have $T_i = \Gamma_d(e(t_i))$. As described in [15], the design of a self-triggered policy involves a sequence of steps.

First, an output function to describe the evolution of the system's tracking error needs to be determined. The output function for our sampling policy is the storage function of the system defined by (9) and (10).

$$V(e) = e^T P e \quad (11)$$

where P is a positive definite matrix satisfying the passivity constraints for the system model defined by (9) and (10). The passivity constraints for the strictly output passive system can be defined by a set of Linear Matrix Inequalities (LMI) [20]. The output function defined in (11) has an estimated decay rate, ρ_0 which can be computed based on the approach in [21].

Next, we define a performance specification in terms of the tracking error output. Our performance function is an exponentially decaying function of the output function with an initial value as the current sampled error. The performance

specification is defined as:

$$D(t) = V(e(t_i)) \exp^{-\rho(t-t_i)} \quad (12)$$

In order to guarantee that the performance function bounds the output function, the decay rate of the performance specification is chosen as $\rho < \rho_0$.

With the output function and performance specification, we determine a continuous time triggering function. From (11) and (12), the triggering condition is given as

$$h_c(t_i, e(t), e(t_i)) := V(e(t)) - V(e(t_i)) \exp^{-\rho(t-t_i)} \leq 0; \quad (13)$$

for some $0 < \rho < \rho_0$.

Finally, a self-triggered policy can then be determined from the continuous time triggering function. In order to check when the triggering condition defined in (13) is violated, we consider a discrete-time implementation based on a discrete step size, $\Delta \in \mathbb{R}^+$ since no digital implementation can be used to perform this check. With, T_i defined as the sampling interval, let T_{\min} and T_{\max} be defined as the minimum and maximum sampling intervals respectively. Also $N_{\min} := \lfloor T_{\min}/\Delta \rfloor$, $N_{\max} := \lfloor T_{\max}/\Delta \rfloor$. The discrete-time implementation can be defined as follows:

$$h_d(e(t_i), n) := h_c(t_i, e(t), e(t_i)) \quad \forall n \in [0, N_{\max}] \quad \forall i \in \mathbb{N} \quad (14)$$

From this discrete-time implementation, the map $\Gamma_d : \mathbb{R}^n \mapsto \mathbb{R}^+$, for computing the next sampling interval or time for the tracking error system model given in (9) is given by:

$$\Gamma_d(e(t_i)) := \max\{T_{\min}, n_i \Delta\} \quad (15)$$

where

$$n_i := \max_{n \in \mathbb{N}} \{n \leq N_{\max} | h_d(e(t_i), c) \leq 0, c = 0, \dots, n\} \quad (16)$$

The lower and upper bounds of the sampling intervals are explicitly enforced by T_{\min} and T_{\max} respectively. The upper bound enforces robustness of the implementation and limits computational complexity. The discrete time step, Δ , can be chosen based on desired accuracy and computational complexity.

B. Wave Variables

The blocks denoted as \mathbf{b} in Fig. 2 each represents the scattering transformation. The scattering transformation is used in order to convert power variables, such as input and output signals of plants and controllers, to wave variables and vice versa. By transmitting wave variables over the network, the communication channel maintains passivity even in the presence of time-varying delays and packet loss [3].

On the left hand side of Fig. 2, the control signal, $z(t)$ and the system's output signal, $s(t)$ are transformed into the wave domain through the scattering transformation. The scattering transformation produces the continuous wave variables $u_p(t)$ and $v_{cd}(t)$, which are related to the signals, $z(t) \in \mathbb{R}^m$ and $s(t) \in \mathbb{R}^m$ as follows:

$$\frac{1}{2}(u_p^\top(t)u_p(t) - v_{cd}^\top(t)v_{cd}(t)) = s^\top(t)z(t). \quad (17)$$

The wave variables $u_p(t)$ and $v_{cd}(t)$ can be described by the following expressions:

$$u_p(t) = \frac{1}{\sqrt{2b}}(bs(t) + z(t)); \quad v_{cd}(t) = \frac{1}{\sqrt{2b}}(bs(t) - z(t)); \quad (18)$$

where $b \in \mathbb{R}_0^+$.

On the right hand side of Fig. 2, the scattering transformation produces discrete wave variables $u_{pd}[i]$ and $v_c[i]$, which are related to the corresponding discrete control signal $z_c[i] \in \mathbb{R}^m$ and the system's discrete-time output $s_c[i] \in \mathbb{R}^m$ as follows:

$$\frac{1}{2}(u_{pd}^\top[i]u_{pd}[i] - v_c^\top[i]v_c[i]) = z_c^\top[i]s_c[i] \quad (19)$$

The wave variables $u_{pd}[i]$ and $v_c[i]$ can be described by the following expressions:

$$u_{pd}[i] = \frac{1}{\sqrt{2b}}(bs_c[i] + z_c[i]); \quad v_c = \frac{1}{\sqrt{2b}}(bs_c[i] - z_c[i]); \quad (20)$$

The wave variables $u_{pd}[i]$ and $v_{cd}[i]$ are the delayed versions of the wave variables $u_p[i]$ and $v_c[i]$ respectively such that

$$u_{pd}[i] = u_p[i - p(i)]; \quad v_{cd}[i] = v_c[i - c(i)] \quad (21)$$

in which $p(i), c(i) \in \mathbb{N}_0^+$ are the respective delays at time i as shown in Fig. 2 as $Z^{-p(i)}$ and $Z^{-c(i)}$.

C. Variable Passive Sampler and Variable Passive Hold

The blocks VPS and VPH in Fig. 2 denote the variable passive sampler and variable passive hold respectively. These blocks represent a pair of sample-and-hold components that can handle non-uniform and adaptive sampling intervals while at same time preserving passivity. These components perform their tasks based on the sampling intervals provided by the adaptive sampling policy described in Section III-A. VPS performs the task of converting continuous time wave variables to discrete time wave variables while VPH performs the task of converting discrete time wave variables to continuous time wave variables.

The VPS and VPH are designed to satisfy the inequality:

$$\begin{aligned} & \int_0^{t_N} (u_p^\top(t)u_p(t) - v_{cd}^\top(t)v_{cd}(t))dt \\ & \geq \sum_{i=0}^{N-1} T_i(u_p^\top[i]u_p[i] - v_{cd}^\top[i]v_{cd}[i]) \end{aligned} \quad (22)$$

To satisfy (22), VPS can be designed to satisfy the following inequality

$$\int_0^{t_N} u_p^\top(t)u_p(t)dt \geq \sum_{i=0}^{N-1} T_i u_p^\top[i]u_p[i], \quad (23)$$

while VPH can be designed to satisfy

$$\sum_{i=0}^{N-1} T_i v_{cd}^\top[i]v_{cd}[i] \geq \int_0^{t_N} v_{cd}^\top(t)v_{cd}(t)dt \quad (24)$$

These inequalities ensure that no energy is generated by the sample and hold devices and thus preserve passivity. Let the j th element of the column vectors $u_p(t)$ and $u_p[i]$ be defined as $u_{pj}(t)$ and $u_{pj}[i]$, respectively, where $j = 1, \dots, m$ and assume that $u_{pj}(t) = 0$; if $t < 0$. A design of the VPS that ensures condition (23) is given by

$$u_{pj}[i] = \frac{1}{\sqrt{T_{i-1}T_i}} \int_{t_{i-1}}^{t_i} u_{pj}(t)dt, \quad \forall j \in \{1, \dots, m\}. \quad (25)$$

where $u_{pj}[0] = 0$ and $u_{pj}(t)$ is the continuous-time wave variable.

In a similar manner, a design of the VPH that satisfies the condition in (24) is given by

$$v_{cd_j}(t) = v_{cd_j}[i], \quad t \in [t_i, t_{i+1}). \quad (26)$$

The proofs for both (25) and (26) are provided in [22].

D. Networked Controller

The networked controller provides control law updates at the request of the system, H_{mp} in order for the plant, H_p to track a desired trajectory. Additionally, for the case of r_c as shown in Fig. 2, the networked controller introduces an additional bias to the desired local trajectory. As a result of the introduced bias, y_p the output of the plant, H_p , tracks the desired local trajectory, y_d plus an additional offset, r_c due to the bias.

The networked controller, H_c , is an event-based proportional controller defined as

$$z_c[i] = H_c(f_c[i]) = K_c(s_c[i] - \Lambda r_c[i]) \quad (27)$$

where the proportional gain, $K_c = \text{diag}[K_{c1}, K_{c2}, \dots, K_{cm}]$ is a positive diagonal matrix. $r_c \in \mathbb{R}^m$ is the bias reference input and Λ is a positive diagonal matrix used in (5).

The networked controller is strictly input passive [23] with an input-output mapping : $f_c \mapsto z_c$. This implies that

$$\sum_{i=0}^{N-1} T_i z_c^\top[i]f_c[i] \geq \delta \sum_{i=0}^{N-1} T_i f_c^\top[i]f_c[i] - \beta_2 \quad (28)$$

where $\delta, \beta_2 \in \mathbb{R}_0^+$

IV. ANALYSIS

This section presents the main analytical results that prove stability and trajectory tracking in the proposed PBASC architecture. Due to space limitations, the proofs are omitted here and are provided in [22].

Theorem 1: Consider the proposed PBASC architecture shown in Fig. 2, if the components of the architecture satisfy their individual passivity constraints such that H_{mp} is passive, VPH and VPS are both passive, the networked controller is strictly input passive and the assumptions **A5** and **A6** hold, then the closed loop system described in Fig. 2 is passive. Additionally, if $K_c = bI$ and $r_p(t) = \mathbf{0}$, the input-output mapping $r_c \mapsto s$ shown in Fig. 2 is strictly output passive [23].

Theorem 2: Consider the proposed PBASC architecture shown in Fig. 2, with the system, H_{mp} and the event-based networked controller described by (27). Assuming the disturbance input, $r_p(t) = \mathbf{0}$, then

$$\lim_{t \rightarrow \infty} e_a(t) = \lim_{t \rightarrow \infty} y_p(t) - y_{da}(t) = \mathbf{0}. \quad (29)$$

where y_{da} and e_a are the modified desired trajectory and modified tracking error respectively in the presence of a bias r_c .

Corollary 1: In the absence of a networked controller bias, that is with the input vector $r_c = \mathbf{0}$, the plant system, H_p tracks the desired trajectory.

V. CASE STUDY

This case study involves the trajectory tracking control of a robotic manipulator over a wireless network using the PBASC architecture. Due to space limitations, a brief description of the robot is provided, the detailed model with figures of the described robot is provided in [22].

A. Derivation of H_{mp} and H_{es} for the robotic manipulator

The Euler-Lagrange equations of motion for an n-degree-of-freedom robotic manipulator, neglecting friction, can be generally described by [24]:

$$M(y_p)\ddot{y}_p + C(y_p, \dot{y}_p)(\dot{y}_p) + g(y_p) = \tau; \quad (30)$$

where $y_p(t) \in \mathbb{R}^n$ is the vector of joint angles, $\tau(t) \in \mathbb{R}^n$ is the input torque vector, $M(y_p) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(y_p, \dot{y}_p) \in \mathbb{R}^{n \times n}$ is the matrix of centrifugal and coriolis effects, and $g(y_p) \in \mathbb{R}^n$ is the gravity vector.

In order to obtain the passive mapping, H_{mp} and the tracking error function, H_{es} for the robotic manipulator in the form described in (2) and (5) respectively, we sought the sliding mode technique employed in [25] where s is introduced as a sliding variable.

With the tracking error, e defined as $y_p(t) - y_d(t)$, we choose the local controller, H_{lc} as:

$$\tau = M(y_p)\ddot{y}_r + C(y_p, \dot{y}_p)\dot{y}_r + g(y_p) + f_p; \quad (31)$$

where f_p is the input to the system, H_{mp} . Let y_r and \dot{y}_r be defined as

$$y_r = y_d - \lambda \int_0^t e; \quad \dot{y}_r = \dot{y}_d - \Lambda e; \quad (32)$$

where Λ , as defined in Section II, is a positive diagonal matrix. By combining (30), (31) and (32), we obtain the following

$$M(y_p)\dot{s} + C(y_p, \dot{y}_p)s = f_p; \quad (33)$$

where s is defined as

$$s = \dot{y}_p - \dot{y}_r = \dot{e} + \Lambda e. \quad (34)$$

The expression in (33) results in the system, H_{mp} , a passive input-output mapping from $f_p \mapsto s$. Additionally, (34) results in the desired function, H_{es} relating the system output, s to the tracking error, e . Hence, the obtained H_{mp} and H_{es} satisfy the assumptions stated in Section II.

B. Evaluation

Using simulations, we evaluate the case study on a robotic manipulator. The setup involves the passive mapping, H_{mp} for the robot and an event-based networked controller, H_c , communicating over a wireless network as shown in Fig. 2. The dynamics of H_{mp} and H_c are implemented using Matlab/Simulink models while TrueTime is used to model the dynamics of the wireless network. The network protocol used is 802.11b, with a speed/bandwidth of 11Mbps. The robot has four degrees-of-freedom and is modeled using four points of mass. The point masses are: $m_1 = 0.362kg$, $m_2 = 0.401kg$, $m_3 = 0.059kg$ and $m_4 = 0.177kg$. The design parameters for the self-triggered policy are $T_{min} = 0.01s$, $T_{max} = 0.1s$, $\Delta = 0.001s$ and $P = 0.5 * I$. The other parameters are $b = 1$, $K_c = I$, and $\Lambda = 10 * I$.

The goal of these experiments is for the robot to track a specified trajectory while efficiently using the network resources. Due to limited space, we focus on joint 2 of the robotic arm with a rest position of 0.17 radians. The desired local trajectory with respect to the home position is $y_{d2} = 0.5 \sin(\frac{2\pi}{5})$. Additionally, we evaluate the introduction of a bias, by the networked controller, which modifies the desired reference trajectory. Such a bias can be viewed as the presence of an obstacle in the robotic manipulator's path which can only be perceived by the networked controller.

1) *PBASC Approach vs Traditional Approach.*: This experiment considers nominal network conditions with no additional delays and packet losses. We model the presence of an obstacle as a step reference input, $r_{cj} = 0.9$, which is introduced between the interval from 3 seconds to 10 seconds. Figs. 3a and 3c show the plots for joint 2 trajectory and the sampling intervals respectively using the PBASC approach. We compare the plots from our nominal scenario with the case of a traditional sampling alternative in the same framework but instead using a fixed sampling period (FSP) of T_{min} . Figs. 3b and 3d show the trajectory and sampling intervals plots respectively for the FSP approach. Comparatively, it can be seen that both approaches closely track the specified trajectory and in the presence of an obstacle,

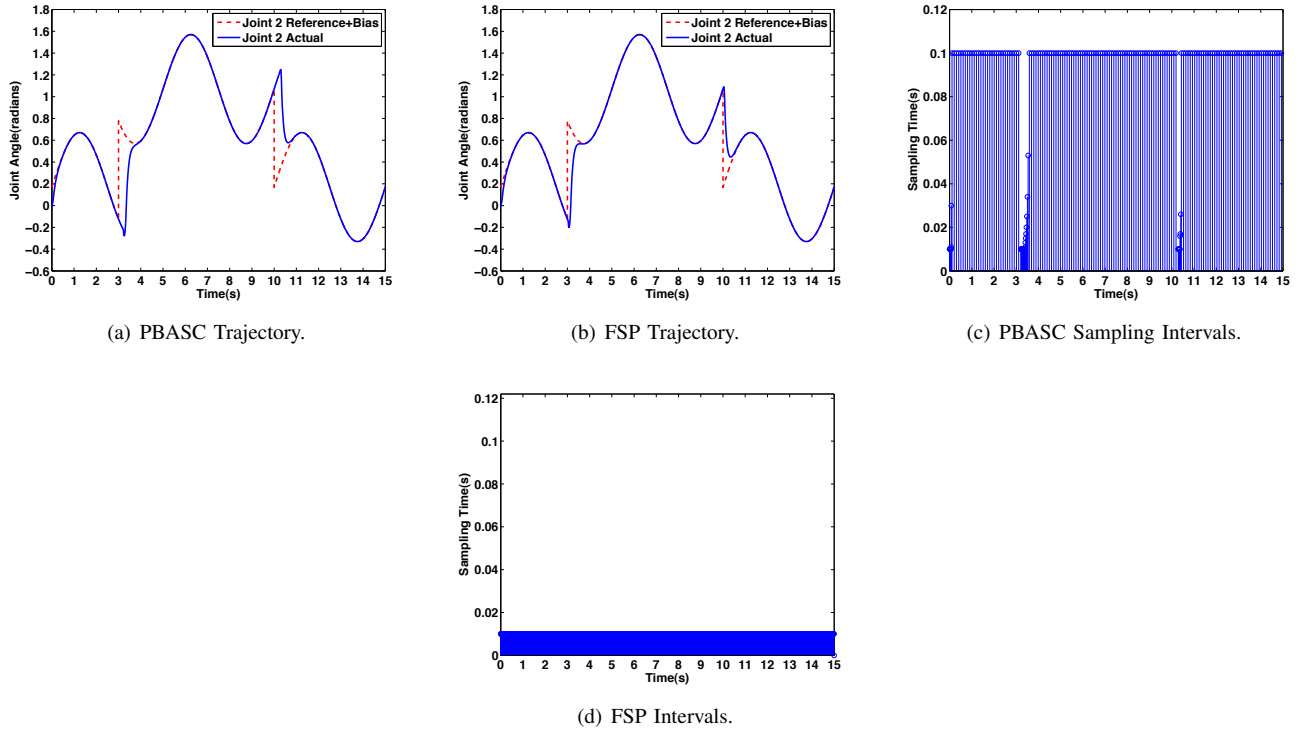


Fig. 3. PBASC Approach vs. FSP Approach in Nominal Case.

the trajectories are adjusted with a bias of 0.9, although the FSP approach seems to respond quicker initially. The average tracking error, the average absolute difference between the joint's trajectory and the desired trajectory for the PBASC and FSP approaches are 0.1064 and 0.04 respectively. On the other hand, using PBASC approach, fewer control law updates are required in order to maintain tracking thereby reducing the amount of network resources utilized.

2) *Time-Varying Delays.*: This experiment considers the effect of time-varying delays. To simulate the case of time-varying delays, starting with the nominal case we incorporate a disturbance node in the network with a sampling period of 0.05 seconds. The disturbance node floods the network with packets based on a Bernoulli process with parameter d . The disturbance node samples a uniformly distributed random variable $X[k] \in [0, 1]$ every 0.05 seconds. If $X[k] > d$, a disturbance packet is forced on the network. Fig. 4a shows the trajectory of joint 2 in the presence of time-varying delays. Compared to the nominal case using the PBASC approach, due to the presence of time delays it takes the system a little longer to adjust its trajectory in order to track the modified trajectory. The average tracking error in the presence of time delays is 0.117. Also, the impact of the delay can also be seen in Fig. 4b, as more control updates are requested in order to achieve tracking. The overall system still maintains stability in the presence of time-varying delays.

3) *Packet Losses.*: This experiment demonstrates the effect of packet losses on the PBASC approach. We consider a lossy network whereby packets containing either sensor updates or control signals can be lost in the communication

channel. The packet loss conditions are designed using a probabilistic Bernoulli loss model. A packet is dropped if a sampled probability is less than a specified probability. We consider the case of thirty percent probability of packet loss. The average tracking error with packet loss is 0.1485. From Figs. 4c and 4d, the effect of packet losses is clearly evident from the plots by the deteriorated performance in tracking but the overall system remains stable.

VI. CONCLUSION AND FUTURE WORK

In this paper, a passivity-based adaptive sampling control architecture for trajectory tracking is introduced. The approach integrates passivity and adaptive sampling to address the challenges of guaranteeing stability in the presence of network uncertainties and efficiently utilizing limited network resources. A case study on the trajectory tracking control of a robotic manipulator over a wireless network in the presence of network uncertainties is used to demonstrate the approach.

Our future work includes extensions to a more dynamic controller and the control of multiple remote systems.

VII. ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation (CNS-1035655, CCF-0820088), U.S. Army Research Office (ARO W911NF-10-1-0005) and Lockheed Martin. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

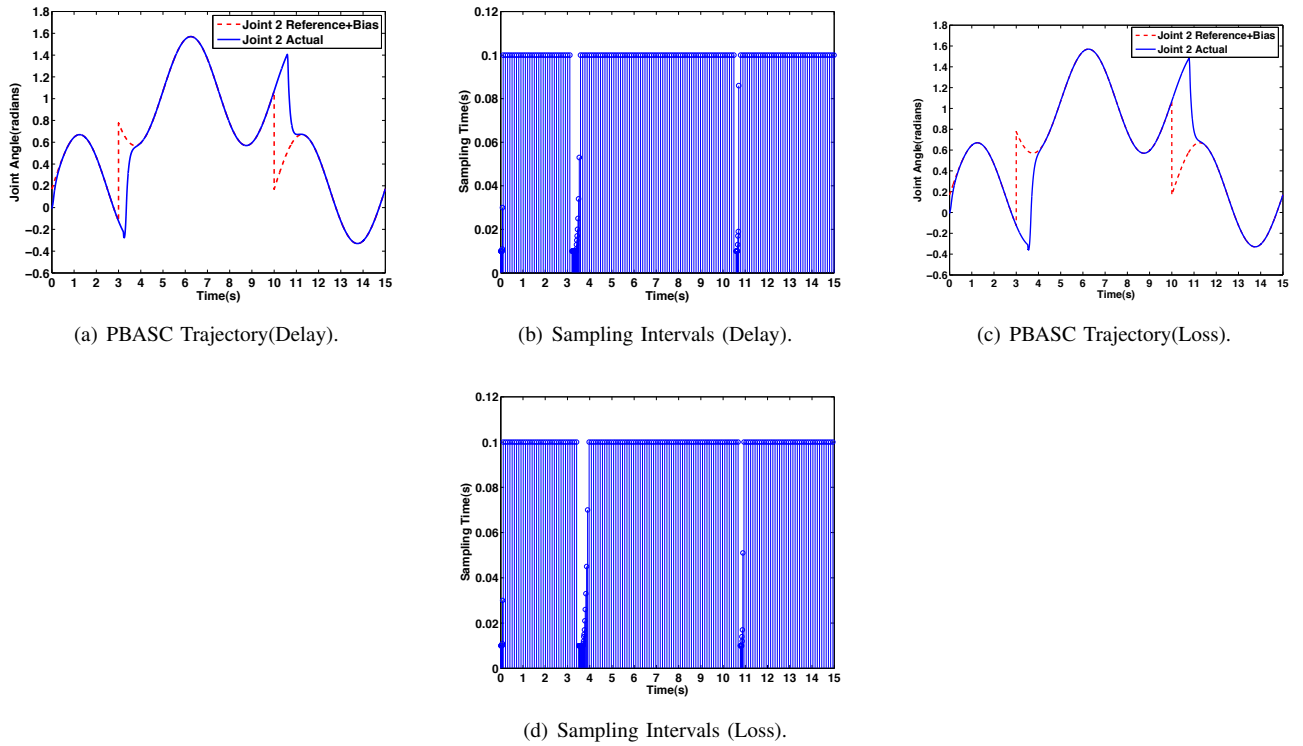


Fig. 4. PBASC Approach with Network Uncertainties.

REFERENCES

- [1] H. Gao, T. Chen, and T. Chai, "Passivity and passification for networked control systems," *SIAM J. on Control and Optimization*, vol. 46, no. 4, pp. 1299–1322, 2008.
- [2] N. Kottenstette, J. Hall, X. Koutsoukos, P. Antsaklis, and J. Szti-napovits, "Passivity-based design of wireless networked control systems for robustness to time-varying delays," in *29th IEEE Real-Time Systems Symposium*, 2008, pp. 15–24.
- [3] N. Chopra, P. Berestesky, and M. Spong, "Bilateral teleoperation over unreliable communication networks," *IEEE Trans. on Control System Technology*, vol. 16, no. 2, pp. 304 – 313, 2008.
- [4] S. Hirche, T. Matiakis, and M. Buss, "A distributed controller approach for delay-independent stability of networked control systems," *Automatica*, vol. 45, pp. 1828–1836, August 2009.
- [5] S. Stramigioli, C. S. , A. J. van der Schaft, and C. Fantuzzi, "Sampled data systems passivity and discrete port-hamiltonian systems," *IEEE Trans. on Robotics*, vol. 21, no. 4, pp. 574 – 587, 2005.
- [6] R. Dorf, M. Farren, and C. Phillips, "Adaptive sampling frequency for sampled-data control systems," *IRE Trans. on Aut. Control*, vol. 7, no. 1, pp. 38 – 47, jan 1962.
- [7] M. Velasco, J. M. Fuertes, and P. Marti, "The self triggered task model for real-time control systems," in *24th IEEE Real-Time Systems Symposium (work in progress)*, 2003, pp. 67–70.
- [8] K. Astrom and B. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems," in *41st IEEE Conf. on Decision and Control*, 2002, pp. 2011 – 2016.
- [9] W. Heemels, A. Teel, N. van de Wouw, and D. Netic, "Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance," *IEEE Trans. on Aut. Control*, vol. 55, no. 8, pp. 1781 –1796, aug. 2010.
- [10] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. on Automatic Control*, vol. 52, no. 9, pp. 1680 – 1685, 2007.
- [11] P. Otanez, J. Moyne, and D. Tilbury, "Using deadbands to reduce communication in networked control systems," in *American Control Conf.*, vol. 4, 2002, pp. 3015 – 3020.
- [12] J. Colandairaj, G. Irwin, and W. Scanlon, "Wireless networked control systems with qos-based sampling," *IET Control Theory Applications*, vol. 1, no. 1, pp. 430 –438, january 2007.
- [13] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Trans. on Aut. Control*, vol. 55, no. 9, pp. 2030 – 2042, 2010.
- [14] X. Wang and M. Lemmon, "Self-triggered feedback control systems with finite-gain \mathcal{L}_2 -stability," *IEEE Trans. on Aut. Control*, vol. 45, no. 3, pp. 452 – 467, 2009.
- [15] M. Mazo and P. Tabuada, "Input-to-state stability of self-triggered control systems," in *48th IEEE Conf. on Decision and Control*, Dec. 2009, pp. 928 –933.
- [16] M. Mazo Jr. and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *47th IEEE Conf. on Decision and Control*, 2008, pp. 435 – 440.
- [17] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099 – 1111, 2003.
- [18] C. Byrnes, A. Isidori, and J. Willems, "Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems," *IEEE Trans. on Aut. Control, IEEE Transactions on*, vol. 36, no. 11, pp. 1228 –1240, nov 1991.
- [19] C. Byrnes and W. Lin, "Losslessness, feedback equivalence, and the global stabilization of discrete-time nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 39, no. 1, pp. 83 –98, jan 1994.
- [20] N. Kottenstette and P. Antsaklis, "Relationships between positive real, passive dissipative, and positive systems," in *American Control Conf.*, 2010, pp. 409 –416.
- [21] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [22] E. Eyiisi, X. Koutsoukos, and N. Kottenstette, "Passivity-Based Trajectory Tracking Control with Adaptive Sampling Over a Wireless Network," Technical Report ISIS-12-104.
- [23] A. van der Schaft, *L₂-Gain and Passivity in Nonlinear Control*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999.
- [24] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," in *27th IEEE Conf. on Decision and Control*, 1988, pp. 1575–1584.
- [25] J. J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. on Aut. Control*, vol. 33, no. 11, pp. 995 –1003, Nov. 1988.