

Scheduling Measurements and Controls over Networks - Part I: Rollout Strategies for Protocol Design

D. J. Antunes, W. P. M. H. Heemels, J. P. Hespanha, and C. J. Silvestre

Abstract— We consider networked control systems where nodes (sensors, actuators, and controller) are connected via a communication network that allows only one user to transmit at a given time. We tackle the scheduling problem of deciding which node should access the network at each transmission time so as to optimize a quadratic performance objective. Using the framework of dynamic programming, we propose a rollout strategy by which the node elected to transmit at each step is the one that leads to optimal performance over a lookahead horizon assuming that from then on nodes transmit in a periodic order. The proposed strategy leads to a protocol in which a conic state partition determines which node transmits at each step and which can outperform any given periodic protocol. Moreover, we show that some of the protocols previously proposed in the literature, such as the Maximum Error First and the dynamic protocols, can be viewed as rollout strategies for a certain dynamic programming problem. The advantages of using rollout strategies are illustrated by a numerical example.

I. INTRODUCTION

Several works have addressed the scheduling problem for networked control systems in which nodes (sensors, actuators, and controller) communicate via a shared network that allows only one node to transmit at a given time (e.g., Ethernet, CAN-BUS, Wireless 802.11). In this paper, we follow a line of research [1], [2], [3], [4], [5], [6] that considers an emulation set-up in the sense that the control algorithm for the networked control system is obtained from a previously designed stabilizing continuous-time controller and the network protocol should strive to emulate a continuous connection between controller, sensors and actuators. Hence, in this paper, only the protocol is to be designed. In a companion paper [7], we tackle the problem of simultaneously designing the protocol and the controller.

In [1], [2], Maximum Error First (MEF) protocols are proposed, in which the node that transmits at each step

Duarte Antunes and Maurice Heemels are with the Hybrid and Networked Systems Group, Department of Mechanical Engineering, Eindhoven University of Technology, the Netherlands. {D.Antunes, M.Heemels}@tue.nl

João P. Hespanha is with the Dept. of Electrical and Computer Eng., University of California, Santa Barbara, CA 93106-9560, USA. hespanha@ece.ucsb.edu

Carlos Silvestre is with the Institute for Systems and Robotics, Instituto Superior Técnico, 1046-001 Lisboa, Portugal, and the Faculty of Science and Technology, University of Macau, Taipa, Macau. cjs@isr.ist.utl.pt

This work is supported by the Dutch Science Foundation (STW) and the Dutch Organization for Scientific Research (NWO) under the VICI grant Wireless controls systems: A new frontier in automation, by the European 7th Framework Network of Excellence by the project Highly-complex and networked control systems (HYCON2-257462) and also supported by project FCT [PEst-OE/EEI/LA0009/2011], and by project FCT OBSERVFLY (PTDC/EEA-ACR/72853/2006)

is the one yielding the largest error between its current measurement/control value and the last value it sent over the network. Simulation results indicate significant advantages of using MEF protocols instead of periodic protocols, in which nodes transmit in a periodic order. See also [3] where a weighted version of the MEF protocol is considered. In a similar setup, [4], [5] propose a more general class of quadratic protocols, where the node elected to transmit is the one yielding the least value of quadratic state functions associated to each node. Conditions are given in [4], [5] to assert the stability of a given quadratic protocol considering delays and time-varying sampling intervals. When applied to a benchmark example, the stability conditions provided in [4], [5] for quadratic protocols provide less conservative bounds on delays and time-varying sampling intervals for which stability of the networked control system can be guaranteed than the stability conditions for round-robin protocols, a special case of periodic protocols in which each node transmits only once in a period. Dynamic protocols, proposed in [6], generalize quadratic protocols by allowing more than one quadratic function to be associated with each node. It is established in [6] that if the networked control system is stable for a given periodic protocol, then one can find a stabilizing dynamic protocol, for which the networked control system is also stable.

In the present paper, we tackle the scheduling problem of deciding which node should access the network at each transmission time so as to optimize a quadratic performance objective. We consider both finite and infinite horizon problems and propose the use of rollout strategies, which for the infinite horizon problem lead to stationary policies. As explained in [8], rollout strategies consist of suboptimal strategies for dynamic programming problems in which the search for optimal decisions occurs only along a lookahead horizon, assuming that from then on a base policy is used for which the cost to go is typically simple to determine. In our approach, we propose to use a periodic protocol for the base policy. We show that this rollout strategy leads to a protocol in which a conic state partition determines which node should transmit based on the current state. By construction of rollout algorithms our method outperforms any periodic protocol as long as this protocol is used as the base policy. Moreover, we establish the following connections between rollout strategies and previously considered protocols in the literature: (i) a weighted version of the MEF protocol is obtained from a rollout algorithm with a round-robin base policy for a special optimal problem and for the special case where only two nodes transmit over the network; (ii) dynamic

protocols obtained from the design procedure given in [6] can be viewed as rollout algorithms in which the base policy consists of a stochastic protocol assigning transmissions based on the state of a Markov chain. We also briefly address the relation between stochastic and periodic protocols.

The remainder of the paper is organized as follows. Section II sets up the general networked control scheduling problem. Section III addresses rollout strategies, and the connection with other protocols in the literature is given in Section IV. An illustrative example is given in Section V. Section VI contains concluding remarks.

Notation We denote by I_n and O_n the $n \times n$ identity and zero matrices, respectively, and by $\text{diag}(A_1, \dots, A_n)$ a block diagonal matrix with blocks A_i . For a matrix A , A^\top denotes its transpose. The notation $x(t_k^-)$ indicates the limit from the left of x at the point t_k . For dimensionally compatible matrices A and B , we define $(A, B) := [A^\top B^\top]^\top$.

II. PROBLEM SETUP

We consider a networked control system for which sensors, actuators, and a controller, are connected through a communication network, possibly shared with other users. The plant and controller are described by the following state-space model:

$$\text{Plant: } \dot{x}_P = A_P x_P + B_P \hat{u}, \quad y = C_P x_P \quad (1)$$

$$\text{Controller: } \dot{x}_C = A_C x_C + B_C \hat{y}, \quad u = C_C x_C + D_C \hat{y}. \quad (2)$$

where $x_P(t) \in \mathbb{R}^n$, $\hat{u}(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the plant's state, input, and output, respectively, and $x_C(t) \in \mathbb{R}^{n_K}$, $\hat{y}(t) \in \mathbb{R}^p$, and $u(t) \in \mathbb{R}^m$, are the controller's state, input, and output, respectively. We assume that the controller would stabilize the closed loop if the plant and the controller were directly connected, i.e., if we would have $\hat{u}(t) = u(t)$, $\hat{y}(t) = y(t)$ for all $t \geq 0$. The signals \hat{u} , u , y , and \hat{y} are partitioned as $\hat{u} = (\hat{u}^1, \dots, \hat{u}^{n_u})$, $u = (u^1, \dots, u^{n_u})$, $y = (y^1, \dots, y^{n_y})$, and $\hat{y} = (\hat{y}^1, \dots, \hat{y}^{n_y})$, where $\hat{u}^i \in \mathbb{R}^{s_i}$ and $u^i \in \mathbb{R}^{s_i}$ pertains to an actuator node $1 \leq i \leq n_u$; $y^i \in \mathbb{R}^{s_i}$ and $\hat{y}^i \in \mathbb{R}^{s_i}$ pertains to a sensor node $n_u + 1 \leq i \leq n_u + n_y$. Note that, for convenience, we use the same index $i \in \mathcal{M} := \{1, \dots, n_u + n_y\}$ to label both actuator and sensor nodes, and in our terminology a single node can be associated with several entries of the process output y or with several entries of the process input \hat{u} . With some abuse of terminology, we say that an actuator or a sensor node transmits when a transmission occurs either from the controller to an actuator, or from a sensor to the controller, respectively.

The transmission times are denoted by t_k , $k \geq 0$, and we assume that the time intervals between transmissions is constant, i.e., $t_{k+1} - t_k = \tau_s$, where τ_s is the sampling period. Between transmission times, the inputs to the plant and to the controller are held constant, i.e.,

$$\hat{u}(t) = \hat{u}(t_k), \quad \hat{y}(t) = \hat{y}(t_k), \quad t \in [t_k, t_{k+1}). \quad (3)$$

The network is assumed to impose that only one node can transmit at a given time. For a set of time instants of interest

$\mathcal{K} := \{0, \dots, k_F - 1\}$, with $k_F \in \mathbb{N} \cup \{\infty\}$, we define the following scheduling sequence

$$\sigma_k \in \mathcal{M}, \quad \text{for } k \in \mathcal{K}. \quad (4)$$

indicating that at the time t_k , σ_k is the node that transmits. Although (4) is simply a vector of scheduling decisions when $k_F < \infty$, we will use the nomenclature scheduling sequence also for this case. The components of $\hat{y}(t_k)$ or $\hat{u}(t_k)$ associated with the node σ_k that transmits at time t_k are updated by the corresponding components of $y(t_k)$ or $u(t_k)$. If we define a vector $e \in \mathbb{R}^{n_e}$, $n_e := \sum_{j=1}^{n_u+n_y} s_j$ as

$$e := (\hat{u} - u, \hat{y} - y), \quad (5)$$

this is captured by the following equation

$$e(t_k) = \Lambda_{\sigma_k} e(t_k^-), \quad (6)$$

where

$$\Lambda_j := \text{diag}(I_{\sum_{i=1}^{j-1} s_i} 0_{s_j} I_{\sum_{i=j+1}^m s_i}), \quad j \in \mathcal{M}.$$

The state of the networked control system is defined by the vector $x := (x_P, x_C, e)$, $x(t) \in \mathbb{R}^{n_x}$, $n_x := n + n_K + n_e$, and can be described by

$$\begin{aligned} \dot{x}(t) &= A_{CL} x(t), \quad t \geq 0, \quad t \neq t_k, \quad k \geq 0 \\ x(t_k) &= J_{\sigma_k} x(t_k^-), \end{aligned} \quad (7)$$

where $J_j := \text{diag}(I_{n+n_K}, \Lambda_j)$, $j \in \mathcal{M}$, and

$$\begin{aligned} A_{CL} &= \begin{bmatrix} I \\ C_e \end{bmatrix} \begin{bmatrix} A_{xx} & A_{xe} \end{bmatrix}, & A_{xe} &= \begin{bmatrix} B_P & B_P D_C \\ 0 & B_C \end{bmatrix} \\ A_{xx} &= \begin{bmatrix} A_P + B_P D_C C_P & B_P C_C \\ B_C C_P & A_C \end{bmatrix}, & C_e &= \begin{bmatrix} 0 & -C_C \\ -C_P & 0 \end{bmatrix}. \end{aligned} \quad (8)$$

Note that A_{xx} is Hurwitz due to the assumption that the controller would stabilize the closed loop if plant (1) and controller (2) were directly connected.

Defining $x_k := x(t_k^-)$, $x_k \in \mathbb{R}^{n_x}$, we can write

$$x_{k+1} = A_{\sigma_k} x_k, \quad k \geq 0, \quad (9)$$

where $A_{\sigma_k} := \bar{A} J_{\sigma_k}$,

$$\bar{A} := e^{A_{CL} \tau_s}. \quad (10)$$

We consider the following cost

$$J(x_0, \sigma) := \begin{cases} \sum_{k=0}^{k_F-1} x_k^\top Q_{\sigma_k} x_k + x_{k_F}^\top \bar{Q} x_{k_F}, & \text{if } k_F < \infty \\ \sum_{k=0}^{\infty} x_k^\top Q_{\sigma_k} x_k, & \text{if } k_F = \infty \end{cases} \quad (11)$$

where the matrices Q_j , $j \in \mathcal{K}$, and \bar{Q} are assumed to be positive semi definite, and we shall be interested in the problem of finding a scheduling sequence $\sigma_k, k \in \mathcal{M}$ to minimize $J(x_0, \sigma)$. Since this problem is combinatorial in general we shall propose suboptimal strategies which lead to policies taking the form $\sigma_k = \mu_k(x_k)$, $k \in \mathcal{K}$, for some functions $\mu_k, k \in \mathcal{K}$.

The cost matrices Q_j , $j \in \mathcal{K}$, and \bar{Q} in (11) are in general problem specific, but can also be seen as tuning knobs. In fact, by properly choosing these matrices we shall be able to relate the results presented here with previous results in the literature (cf. Section IV). These matrices can also be obtained from an optimal quadratic cost for (1), (2), taking the form $J_C = \int_0^{t_F} x_P(t)^\top Q_C x_P(t) + \hat{u}(t)^\top R_C \hat{u}(t) dt$, for $t_F > 0$, in which case we have

$$Q_{\sigma_k} = \int_0^{\tau_s} J_{\sigma_k}^\top e^{A_{CL}^\top s} (\text{diag}(Q_C, 0_{n_{\mathcal{K}+p+m}}) + U_C^\top R_C U_C) e^{A_{CL} s} J_{\sigma_k} ds, \quad (12)$$

$U_C = [D_C C_P \quad C_C \quad I \quad D_C]$, and \bar{Q} is defined as Q_{σ_k} by replacing τ_s by $\min\{t_F - k\tau_s, k \geq 0, t_F - k\tau_s > 0\}$.

III. ROLLOUT STRATEGIES

We start by showing how to compute the cost (11) for a periodic base policy in Section III-A, and then we present the scheduling protocol that a rollout strategy with this periodic base policy leads to in Section III-B. In Section III-C we discuss this rollout strategy.

A. Base Policy

To define a periodic protocol, we consider a set of h consecutive scheduling decisions σ_k denoted by

$$(v_0, \dots, v_{h-1}), \quad (13)$$

where $v_\ell \in \mathcal{M}$, $\ell \in \mathcal{H}$, $\mathcal{H} := \{0, 1, \dots, h-1\}$, which are periodically repeated as explained next. If we let $[k]_h$ denote the remainder after division of k by h , we have

$$\sigma_k = \theta_k^\kappa, \quad k \in \mathcal{K}, \quad (14)$$

where

$$\theta_k^\kappa := v_{[k+\kappa]_h}, \quad k \in \mathcal{K}, \quad (15)$$

for some $\kappa \in \mathcal{H}$ that characterizes the initial condition of the periodic scheduling θ_k^κ . We make the following assumption.

Assumption 1: We assume that the base policy (14), (15) is such that the system (9) is asymptotically stable when σ_k is given by (14), (15), with $k_F = \infty$, i.e., we assume that the matrix $A_{v_{h-1}} \dots A_{v_1} A_{v_0}$ has all its eigenvalues inside the unit disk.

As usual, by asymptotic stability we mean that the state of (9) converges to zero $x_k \rightarrow 0$ when $k \rightarrow \infty$, for any initial condition x_0 .

The following proposition summarizes how to compute the cost of the periodic policy.

Proposition 2: Suppose that Assumption 1 holds and suppose that $k_F > h$. Then, the cost (11) is given by

$$J_{\text{base}, \kappa}(x_0) := x_0^\top (P_\kappa + \bar{P}_\kappa(k_F, 0)) x_0,$$

where $\{P_\kappa, \kappa \in \mathcal{H}\}$ is the unique solution to

$$P_\kappa = A_{v_\kappa}^\top P_{[\kappa+1]_h} A_{v_\kappa} + Q_{v_\kappa}, \quad \kappa \in \mathcal{H}, \quad (16)$$

and

$$\bar{P}_\kappa(k_F, 0) := \Phi(k_F, 0)^\top (\bar{Q} - P_{\theta_{k_F}^\kappa}) \Phi(k_F, 0),$$

$$\Phi(k_F, 0) := A_{\theta_{k_F-1}^\kappa} \dots A_{\theta_1^\kappa} A_{\theta_0^\kappa}.$$

□

B. Rollout Policies

We propose to choose at each time the node to transmit as the one that leads to optimal performance over a fixed lookahead horizon, assuming that from then on a periodic base policy is used. In other words, for each time iteration ℓ , $0 \leq \ell \leq k_F - 1$, the schedules

$$\sigma_\ell, \sigma_{\ell+1}, \dots, \sigma_{\ell+H-1}$$

are assumed to be free variables, where H denotes the length of the lookahead horizon, while

$$\sigma_{\ell+H}, \sigma_{\ell+H+1}, \dots$$

are fixed and follow a periodic policy as in (14) and (15). The free scheduling variables are denoted by $\nu = (\nu_0, \dots, \nu_{H-1})$, i.e.,

$$\sigma_k = \nu_{k-\ell}, \quad \text{for } k \in \{\ell, \dots, \ell+H-1\} \quad (17)$$

and the fixed scheduling variables can be written as

$$\sigma_k = \theta_{k-(\ell+H)}^\kappa, \quad \text{for } k \in \{\ell+H, \dots, k_F-1\}. \quad (18)$$

Note that at time $\ell+H$ the base policy is assumed to start at an initial schedule v_κ , determined by κ . We consider that $\kappa \in \mathcal{H}$ is also a decision variable, and the decision set is denoted by $\mathcal{I} := \mathcal{M}^H \times \mathcal{H}$, i.e., $(\nu, \kappa) \in \mathcal{I}$. We also allow for $H = 0$, in which case κ is the only decision variable. The length of the lookahead horizon is a fixed constant, but naturally needs to be adapted when the time iteration is close to the terminal time iteration k_F , i.e.,

$$H(\ell) := \min(H_c, k_F - 1 - \ell), \quad (19)$$

where $0 \leq H_c \leq k_F - 1$ is a constant. The dependency of H on ℓ is omitted hereafter. The process is restarted at each step, in a similar fashion as in Model Predictive Control (MPC) [9].

We describe next a protocol which, as established in the sequel, corresponds to the rollout algorithm just described. We separate the cases $H = 0$ and $H > 0$. As in Proposition 2, we assume that the periodic base policy stabilizes the networked control system.

Protocol 1: At each time iteration ℓ take the scheduling decision σ_ℓ as

$$\sigma_\ell = w_0^\ell, \quad (20)$$

where:

- (i) if $H > 0$, w_0^ℓ is the first entry of the vector $w^\ell = (w_0^\ell, \dots, w_{H-1}^\ell)$ obtained from

$$(w^\ell, \kappa^\ell) = \underset{(\nu, \kappa) \in \mathcal{I}}{\text{argmin}} x_\ell^\top R_{\nu, \kappa, \ell} x_\ell, \quad (21)$$

where

$$R_{\nu, \kappa, \ell} := \sum_{m=0}^{H-1} \Psi(m, 0)^\top Q_{v_m} \Psi(m, 0) + \Psi(H, 0)^\top (P_\kappa + \bar{P}_\kappa(k_F - (\ell+H), 0)) \Psi(H, 0), \quad (22)$$

$\Psi(m, 0) := A_{v_{m-1}} \times \dots \times A_{v_1} \times A_{v_0}$, $\Psi(0, 0) = I$, and $P_\kappa, \bar{P}_\kappa(k_F - (\ell+H), 0)$ are described in Proposition 2.

(ii) if $H = 0$, $w_0^\ell = v_{\bar{\kappa}_\ell}$ where

$$\hat{\kappa}_\ell = \operatorname{argmin}_{\kappa \in \mathcal{H}} x_\ell^\top (P_\kappa + \bar{P}_\kappa(k_F - (\ell + H), 0)) x_\ell. \quad (23)$$

□

As stated next, this protocol does in fact correspond to the rollout algorithm described above, and it always outperforms the corresponding periodic base policy. Let $J_{\text{rollout}}(x_0)$ be the cost (11) for the system (9) with initial condition x_0 when σ_k is chosen according to Protocol 1.

Theorem 3: The rollout scheduling algorithm (17), (18) is determined by at each iteration ℓ choosing σ_ℓ as described in Protocol 1. Moreover,

$$J_{\text{rollout}}(x_0) \leq \min_{\kappa \in \mathcal{H}} J_{\text{base}, \kappa}(x_0) \quad (24)$$

for every initial condition $x_0 \in \mathbb{R}^{n_x}$.

□

The fact that rollout strategies outperform the corresponding base policy is a general property of these policies [8], but, as explained in [8], it is typically hard to prove that (24) holds with a strict inequality. It is also mentioned in [8] that these policies typically largely outperform the corresponding base policy in practice. We shall illustrate this performance improvement with an example in Section V.

C. Discussion on Protocol 1

We provide next further important comments on Protocol 1 and Theorem 3, concerning: (i) the case where $k_F = \infty$; (ii) the full state knowledge assumption of the Protocol 1.

1) *Conic state partition and stability* ($k_F = \infty$): It follows from the Assumption 1, that $\Phi(r, 0) \rightarrow 0$, as $r \rightarrow 0$, and this implies in turn that $\bar{P}_\kappa(r, 0) \rightarrow 0$, where $\Phi(r, 0)$ and $\bar{P}_\kappa(r, 0)$ are described in Proposition 2. Thus, if $k_F = \infty$, the scheduling law (20), (21) becomes time-invariant, i.e., independent of ℓ , depending only on x_ℓ , or in other words, the rollout policy becomes stationary. Let $S_j, j \in \{1, \dots, n_y + n_u\}^H \times h$ be an indexation of the matrices $R_{v, \kappa}$ obtained from eliminating the time-varying dependency from $R_{v, \kappa, k}$ by letting $k_F \rightarrow \infty$. Furthermore, corresponding to this indexation, consider a map d_1 assigning to each $j \in \mathcal{M}$ the corresponding first component ν_0 of the vector $\nu = (\nu_0, \dots, \nu_{H-1})$ of the matrix $R_{v, \kappa}$ corresponding to j . Then, we can write (20), (21) when $k_F = \infty$ as

$$\sigma_\ell = d_1(j), \quad (25)$$

where j is obtained from

$$j := \operatorname{argmin}_{j \in \mathcal{M}_D} x_\ell^\top S_j x_\ell. \quad (26)$$

In this case, the scheduling depends only on the state (and not on time ℓ) and is determined by a zero-symmetric conic state partition since the same schedule is applied for a state x_ℓ and for αx_ℓ , where $\alpha \in \mathbb{R} - \{0\}$. The protocol (25), (26) as the same structure of the quadratic protocols proposed in [4], [5], and of the dynamic protocols proposed in [6], and we shall discuss this relation in more detail in Section IV.

As stated in the next proposition the system (9) with Protocol 1 when $k_F = \infty$ is asymptotically stable, if the matrices $Q_k, k \in \mathcal{K}$, are positive definite.

Proposition 4: Suppose that Assumption 1 holds, that $k_F = \infty$, and that $Q_k, k \in \mathcal{K}$, are positive definite. Then the system (9) with Protocol 1 is asymptotic stable.

□

2) *Full-State knowledge assumption:* One difficulty in implementing the protocol (25), (26) is that one needs to assume that the full-state is known, which is only true in very special cases. In the companion paper [7], using a different problem formulation in which we aim at simultaneously designing the protocol and the controller based on the data transmitted over the network up to a scheduling decision time t_ℓ , we are able to find a state estimator that comes out naturally from the solution to the problem and which can be implemented in a distributed way.

IV. CONNECTION WITH EXISTING PROTOCOLS

We start by addressing the connection between rollout protocols and Maximum Error First (MEF) protocols [1], [2] in Section IV-A and then address the connection with dynamic protocols [6] in Section IV-B.

A. Maximum Error First

A weighted version of the MEF protocols [1], [2], can be described as follows. Consider a partition of the error vector e defined in (27) into components $e^i \in \mathbb{R}^{s_i}, i \in \mathcal{M}$, pertaining to node i , i.e.,

$$e = (e^1, e^2, \dots, e^{n_u + n_y}), \quad (27)$$

where $e^i := (\hat{u}^i - u^i)$, if $i \in \{1, \dots, n_u\}$, and $e^i := (\hat{y}^{i-n_u} - y^{i-n_u})$, if $i \in \{n_u + 1, \dots, n_u + n_y\}$. Then the MEF protocol is defined by

$$\text{MEF: } \sigma_k = \operatorname{argmax}_{j \in \mathcal{M}} e_k^j \top M_j e_k^j, \quad k \in \mathcal{K}, \quad (28)$$

for some positive definite matrices $M_j, j \in \mathcal{M}$. It is also useful to define the round-robin protocol which is a periodic protocol taking the form (14), (15) with

$$(v_0, v_1, \dots, v_{h-1}) = (1, 2, \dots, n_u + n_y), \quad h = n_u + n_y. \quad (29)$$

To establish a link between the rollout strategy and MEF protocols we start by making the following assumption.

Assumption 5: Let h and v be defined by (29). Then, we assume that there exists matrices $S_\kappa, \kappa \in \mathcal{H}$, with the following structure

$$S_\kappa = \begin{bmatrix} X & 0 \\ 0 & E_\kappa \end{bmatrix} \quad (30)$$

for positive definite matrices $X \in \mathbb{R}^{(n+n_\kappa) \times (n+n_\kappa)}$, and positive definite matrices $E_\kappa \in \mathbb{R}^{n_e \times n_e}, \kappa \in \mathcal{H}$, such that

$$(J_{v_{\lfloor \kappa+1 \rfloor h}} \bar{A})^\top S_{\lfloor \kappa+1 \rfloor h} J_{v_{\lfloor \kappa+1 \rfloor h}} \bar{A} - S_\kappa < 0, \quad \kappa \in \mathcal{H}. \quad (31)$$

□

It can be shown that the existence of general positive definite matrices $S_\kappa, \kappa \in \mathcal{K}$, that satisfy (31) is a necessary and sufficient stability condition for (9) to be asymptotically stable when the nodes transmit in a round-robin fashion. The

restriction (30) makes Assumption 5 stricter than assuming asymptotic stability of (9) for a round-robin protocol. However, as we state next this assumption is satisfied if the sampling period τ_s in (10) is sufficiently small.

Proposition 6: There exists $\epsilon > 0$, such that Assumption 5 holds for $\tau_s \in [0, \epsilon)$, for $X > 0$ such that

$$A_{xx}^\top X + X A_{xx} < 0, \quad (32)$$

and for $E_\kappa, \kappa \in \mathcal{H}$, such that

$$\Lambda_{v_{\lfloor \kappa+1 \rfloor h}}^\top E_{\lfloor \kappa+1 \rfloor h} \Lambda_{v_{\lfloor \kappa+1 \rfloor h}} - E_\kappa < 0, \quad \kappa \in \mathcal{H}. \quad (33)$$

□

Note that there exists $X > 0$ such that (32) holds due to the assumption that A_{xx} is Hurwitz and, e.g., the following matrices satisfy (33)

$$E_\kappa = \text{diag}(c_1^\kappa I_{s_1}, \dots, c_h^\kappa I_{s_h}), \quad \kappa \in \mathcal{H},$$

where $c^\kappa := \text{circ}_\kappa([h \ h-1 \ \dots \ 2 \ 1])$, $\kappa \in \mathcal{H}$, and $\text{circ}_j(a)$ denotes a right circular shift of the vector a by j units, e.g., $\text{circ}_0([1 \ 2 \ 3]) = [1 \ 2 \ 3]$, $\text{circ}_1([1 \ 2 \ 3]) = [3 \ 1 \ 2]$.

The connection between rollout protocols and Maximum Error First (MEF) protocols is summarized in the next result. Under Assumption 5, we define the following positive definite matrices

$$\bar{Q}_\kappa := S_\kappa - (J_{v_{\lfloor \kappa+1 \rfloor h}} \bar{A})^\top S_{\lfloor \kappa+1 \rfloor h} J_{v_{\lfloor \kappa+1 \rfloor h}} \bar{A}, \quad \kappa \in \mathcal{H}. \quad (34)$$

Theorem 7: Suppose that Assumption 5 holds, let h and v be defined by (29), and consider that $k_F = \infty$. If the cost matrices in (11) are given by

$$Q_j = J_j^\top \bar{Q}_{j-1} J_j, \quad j \in \mathcal{M}, \quad (35)$$

then a rollout policy with $H = 0$ and round-robin periodic base policy yields the following protocol

$$\sigma_k = v_{\bar{k}}, \quad (36)$$

where

$$\bar{k} = \text{argmin}_{i \in \mathcal{H}} e_i^\top \Lambda_{v_i}^\top E_i \Lambda_{v_i} e_k. \quad (37)$$

In particular, when the number of nodes is equal to two, i.e., $h = n_y + n_u = 2$, one obtains the following MEF protocol

$$\text{MEF: } \sigma_k = \text{argmax}_{j \in \{1,2\}} e_k^j \top M_j e_k^j, \quad k \in \mathcal{K}, \quad (38)$$

for $M_1 = \Lambda_2^\top E_2 \Lambda_2$ and $M_2 = \Lambda_1^\top E_1 \Lambda_1$.

□

Proof: A rollout protocol with $H = 0$, with $k_F = \infty$ and a round-robin base policy can be written as (23) where the matrices $P_\kappa, \kappa \in \mathcal{H}$, are obtained from (16). If (35) holds, then the matrices $P_\kappa, \kappa \in \mathcal{H}$ that satisfy (16) and the matrices $S_\kappa, \kappa \in \mathcal{H}$ that satisfy (34) are related by

$$P_\kappa = J_{v_\kappa}^\top S_\kappa J_{v_\kappa}, \quad \kappa \in \mathcal{H}. \quad (39)$$

Replacing (39) in (23) when $k_F = \infty$ we obtain (36), (37). The last part of the theorem, follows by direct replacement. ■

B. Dynamic Protocols

To establish the connection between rollout protocols and dynamic protocols [6] we need to define a class of base policies, different from periodic. We consider the following stochastic policies

$$\begin{aligned} \text{Prob}[\omega_{k+1} = j | \omega_k = i] &= \mu_{ij}, \quad i, j \in \mathcal{M}_D \\ \sigma_k &= d(\omega_k), \quad k \geq 0, \end{aligned} \quad (40)$$

where ω_k is a Markov Chain with $m_D > m$ states, $\mathcal{M}_D := \{1, \dots, m_D\}$, and $d : \mathcal{M}_D \rightarrow \mathcal{M}$ is a map that assigns a node to each state of the Markov chain. We shall be interested in an infinite time interval, i.e., $k_F = \infty$.

The system (9) with scheduling (40) is Mean Square Stable ($\lim_{k \rightarrow \infty} \mathbb{E}[x_k^\top x_k] = 0, \forall x_0$) if and only if for every $\hat{Q}_i > 0, i \in \mathcal{M}_D$, there exists $S_i > 0, i \in \mathcal{M}_D$, such that

$$\sum_{j=1}^{m_D} \mu_{ij} A_{d(i)}^\top S_j A_{d(i)} - S_i = -\hat{Q}_i, \quad \forall i \in \mathcal{M}_D \quad (41)$$

(cf. [10]). If (41) holds then one can also conclude (cf. [10]) that for the system (9) with protocol (40), we have

$$J_{MC, \kappa}(x_0) = x_0^\top S_\kappa x_0, \quad \kappa \in \mathcal{M}_D \quad (42)$$

where

$$J_{MC, \kappa}(x_0) := \mathbb{E}\left[\sum_{k=0}^{\infty} x_k^\top \hat{Q}_{\omega_k} x_k | \omega_0 = \kappa\right], \quad \kappa \in \mathcal{M}_D. \quad (43)$$

If we choose $\hat{Q}_i, i \in \mathcal{M}_D$ as

$$\hat{Q}_i = Q_{d(i)}, \quad i \in \mathcal{M}_D, \quad (44)$$

for matrices $Q_j, j \in \mathcal{M}$, we can interpret (43) as the expected value of a cost taking the form (11).

Using similar ideas to the ones presented in Section III to obtain Protocol 1 we can obtain that a rollout policy with $H = 0$ having this stochastic policy as the base policy is described by

$$\sigma_k = d(\text{argmin}_{j \in \mathcal{M}_D} x^\top S_j x). \quad (45)$$

Moreover, we can use the same arguments as in Theorem 3 and Proposition 4 to obtain the following result. Let $J_{\text{rollout,MC}}(x_0)$ denote the cost (11) for the system (9) with initial condition x_0 when σ_k is chosen according to (45).

Theorem 8: Suppose that the system (9) with scheduling (40) is Mean Square Stable, and in particular, that there exists positive definite matrices $S_\kappa, \kappa \in \mathcal{M}_D$ such that (41) holds for $\hat{Q}_\kappa, \kappa \in \mathcal{M}_D$, taking the form (44). Then

$$J_{\text{rollout,MC}}(x_0) \leq \min_{\kappa \in \mathcal{M}_D} J_{MC, \kappa}(x_0). \quad (46)$$

for every initial condition $x_0 \in \mathbb{R}^{n_x}$. Moreover, the system (9) with protocol (45) is asymptotically stable. □

One can confer that the protocol defined by (45) has exactly the same structure of the dynamic protocols [6] which are also synthesized under the assumption that there exists matrices S_κ such that (41) holds for every $\hat{Q}_j, 1 \leq j \leq m_D$. Actually the last part of Theorem (9) is a special case

of [6, Th. 1], which is obtained using a different line of reasoning.

Stochastic protocols have been proposed before in the literature (cf. [11], [12]). Although we shall not discuss in detail the relation with periodic protocols we state the following result, which indicates that the appeal to consider Markov Chain base policies may be limited.

Proposition 9: If there exists a scheduling taking the form (40) that yields (9) Mean Square Stable, then there also exists a periodic scheduling (14), (15) that yields (9) stable. □

V. ILLUSTRATIVE EXAMPLE

Consider a linearized model of an inverted pendulum

$$A_P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B_P = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_P = [1 \quad 0], \quad D_P = 0.$$

and a performance criterium

$$\int_0^\infty (y(t)^2 + \hat{u}(t)^2) dt. \quad (47)$$

The continuous-time controller is taken as $\hat{y}(t) = Kx_C(t)$ where $K = [-2.141 \quad -2.197]$ is the optimal linear quadratic regulator gains for the problem and $x_C(t)$ is a state estimate obtained by the following observer

$$\dot{x}_C(t) = (A_P + LC_P + B_PK)x_C(t) - Ly(t)$$

with $L = [-4.317 \quad -4.316]^T$. Plant and controller are connected by a network, which allows transmissions at $t_k = k\tau_s$, where the sampling period is given by $\tau_s = 0.1$. The plant sends measurements $y(k_1\tau_s)$ to the controller, and the controller sends the actuation values $u(k_2\tau_s)$ to the plant through the same exclusive network, i.e., k_1 and k_2 belong to different sets whose union is \mathbb{N} . Thus we wish to determine at each transmission step whether to sample or to control. Suppose that the initial conditions of the networked control system are given by $x_P(0) = [1 \quad 0]^T$, $x_C(0) = [0 \quad 0]^T$, $e(0) = [0 \quad 0]^T$. The cost (47) (which can be written in the form (11), (12)) of a base policy where controller and sensor transmit in a round-robin fashion, i.e., σ_k has period 2, and the cost obtained with the rollout strategy described in Theorem 3, with a lookahead horizon $H = 3$, are the following

Periodic	Rollout
4.835	3.850

We can see that the proposed rollout algorithm clearly outperforms the base policy. Figure 1 plots the continuous-time output signals from the plant and the controller, for both rollout and base policies. The scheduling sequence obtained with the rollout policy from time step $k = 0$ to $k = 34$ is given by

	s	s	$ssaaaaaaaaasasasasasasasassass$	a
k	0	1	2-33	34

where s stands for a sensor transmission and a for an actuator transmission.

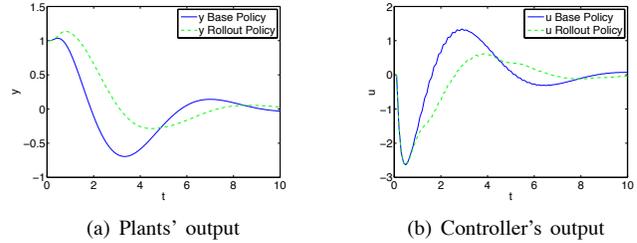


Fig. 1. Measurement and Control

VI. CONCLUSIONS

In this paper we explored the use of rollout algorithms for the scheduling problem in networked control systems. One of the main advantage of using rollout strategies is that they are never worse than the corresponding periodic strategy used as a base policy. Moreover, we interpret existing as rollout strategies for certain optimal problems. Simulation results show that rollout algorithms can indeed significantly outperform periodic protocol, while keeping computations within reasonable limits.

REFERENCES

- [1] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *Control Systems Technology, IEEE Transactions on*, vol. 10, no. 3, pp. 438–446, may. 2002.
- [2] G. Walsh and H. Ye, "Scheduling of networked control systems," *Control Systems Magazine, IEEE*, vol. 21, no. 1, pp. 57–65, feb. 2001.
- [3] D. Dacic and D. Nesić, "Observer design for wired linear networked control systems using matrix inequalities," *Automatica*, vol. 44, no. 11, pp. 2840–2848, 2008.
- [4] M. C. F. Donkers, W. P. M. H. Heemels, N. van de Wouw, and L. L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *Accepted for IEEE Transactions on Automatic Control*, 2011.
- [5] M. C. F. Donkers, W. P. M. H. Heemels, D. Bernadini, A. Bemporad, and V. Shneer, "Stability analysis of stochastic networked control systems," *Accepted for Automatica*, 2011.
- [6] D. Antunes, J. P. Hespanha, and C. Silvestre, "Stochastic networked control systems with dynamic protocols," in *American Control Conference, 2011. ACC '11.*, Jul 2011.
- [7] D. Antunes, W. Heemels, J. P. Hespanha, and C. Silvestre, "Scheduling measurements and controls over networks - part ii: Rollout strategies for simultaneous protocol and controller design," Mar. 2012, to be presented at the American Control Conference (ACC), 2012.
- [8] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.
- [9] J. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2001.
- [10] O. L. V. Costa, M. Fragoso, and R.P.Marques, *Discrete-Time Markov Jump Linear Systems*. Springer, 2005.
- [11] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109805003663>
- [12] M. Tabbara, D. Nesić, and A. Teel, "Stability of wireless and wireline networked control systems," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1615–1630, sep. 2007.