Multiple-Loop Self-Triggered Model Predictive Control for Network Scheduling and Control

Erik Henriksson, Daniel E. Quevedo, *Senior Member, IEEE*, Edwin G. W. Peters, Henrik Sandberg, and Karl Henrik Johansson, *Fellow, IEEE*

***Abstract*— We present an algorithm for controlling and scheduling multiple linear time-invariant processes on a shared bandwidth-limited communication network using adaptive sampling intervals. The controller is centralized and not only computes at every sampling instant the new control command for a process but also decides the time interval to wait until taking the next sample. The approach relies on model predictive control ideas, where the cost function penalizes the state and control effort as well as the time interval until the next sample is taken. The latter is introduced to generate an adaptive sampling scheme for the overall system such that the sampling time increases as the norm of the system state goes to zero. This paper presents a method for synthesizing such a predictive controller and gives explicit sufficient conditions for when it is stabilizing. Further explicit conditions are given that guarantee conflict free transmissions on the network. It is shown that the optimization problem may be solved offline and that the controller can be implemented as a lookup table of state feedback gains. The simulation studies which compare the proposed algorithm to periodic sampling illustrate potential performance gains.**

***Index Terms*— Networked control systems, predictive control, process control, scheduling, self-triggered control, stability.**

I. Introduction

WIRELESS sensing and control systems have received increased attention in the process industry in recent years. Emerging technologies in low-power wake-up radio enable engineering of a new type of industrial automation systems where sensors, controllers, and actuators communicate over a wireless channel. The introduction of a wireless medium in the control loop gives rise to new challenges, which need to be handled [1]. The aim of this paper is to address the problem of how the medium access to the wireless channel could be divided between the loops, taking the process dynamics into consideration. We investigate the possibilities to design a self-triggered controller that adaptively chooses the sampling period for multiple control loops. The aim is to reduce the amount of generated network traffic, while maintaining a guaranteed level of performance in respect of driving the initial system states to zero and the control effort needed.

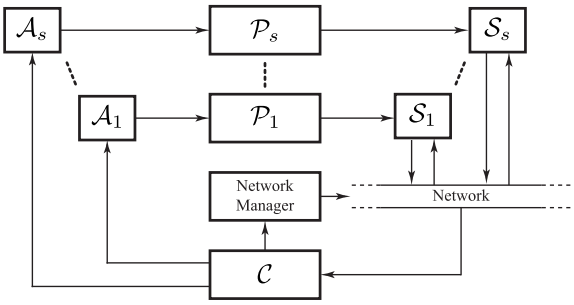

Fig. 1. Actuators $\mathcal{A}$ and processes $\mathcal{P}$ are wired to the controller $\mathcal{C}$ while the sensors $\mathcal{S}$ communicate over a wireless network, which in turn is coordinated by the *network manager*.

Consider the networked control system in Fig. 1, which shows how the sensors and the controller are connected through a wireless network. The wireless network is controlled by a *network manager* that allocates medium access to the sensors and triggers their transmissions. This setup is motivated by current industry standards based on the IEEE 802.15.4 standard [2], [3], which utilizes this structure for wireless control in process industry. Here, the triggering is in turn generated by the controller that, in addition to computing the appropriate control action, dynamically determines the time of the next sample by a self-triggering approach [4]. In doing so, the controller gives varying attention to the loops depending on their state, while trying to communicate only few samples. To achieve this, the controller must, for every loop, trade control performance against intersampling time and give a quantitative measure of the resulting performance.

The main contribution of this paper is to show that a self-triggering controller can be derived using a receding horizon control formulation, where the predicted cost is used to jointly determine what control signal to be applied as well as the time of the next sampling instant. Using this formulation, we can guarantee a minimum and a maximum time between samples.

We will initially consider a single-loop system. We will then extend the approach to the multiple-loop case, which can be analyzed with additional constraints on the

Manuscript received December 4, 2013; revised October 31, 2014; accepted January 10, 2015. Date of publication April 30, 2015; date of current version October 12, 2015. Manuscript received in final form February 9, 2015. This work was supported in part by the Australian Research Council Discovery Project Funding Scheme under Project DP0988601, in part by the Knut and Alice Wallenberg Foundation, in part by the Swedish Governmental Agency for Innovation Systems through the WiComPi Project, and in part by the Swedish Research Council under Grant 2007-6350 and Grant 2009-4565. Recommended by Associate Editor O. Stursberg.

E. Henriksson is with Öhlins Racing AB, Upplands Väsby, Sweden (e-mail: erike02@ee.kth.se).

H. Sandberg and K. H. Johansson are with the ACCESS Linnaeus Centre, School of Electrical Engineering, Royal Institute of Technology, Stockholm 100 44, Sweden (e-mail: hsan@ee.kth.se; kallej@ee.kth.se).

D. E. Quevedo is with the Department of Electrical Engineering (EIM-E), University of Paderborn, 33098 Paderborn, Germany (e-mail: dquevedo@ieee.org).

E. G. W. Peters is with The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: edwin.g.w.peters@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2015.2404308

communication pattern. The results presented herein are extensions to [5]. These results have been extended to handle multiple control loops on the same network while maintaining control performance and simultaneously guaranteeing conflict-free transmissions on the network.

The development of control strategies for wireless automation has become a large area of research in which, up until recently, most efforts have been made under the assumption of periodic communication [6]. However, the idea of adaptive sampling is receiving increased attention. The efforts within this area may coarsely be divided into the two paradigms of event- and self-triggered control. In event-triggered control [7]–[15], the sensor continuously monitors the process state and generates a sample when the state violates some predefined condition. Self-triggered control [4], [16]–[18] utilizes a model of the system to predict when a new sample needs to be taken to fulfill some predefined condition. A possible advantage of event-over self-triggered control is that the continuous monitoring of the state guarantees that a sample will be drawn as soon as the design condition is violated, thus resulting in an appropriate control action. The self-triggered controller will instead operate in open loop between samples. This could potentially be a problem as disturbances to the process between samples cannot be attenuated. This problem may, however, be avoided by good choices of the intersampling times. The possible advantage of self-over event-triggered control is that the transmission time of sensor packets is known *a priori* and hence we may schedule them, enabling sensors and transmitters to be put to sleep in between samples and thereby saving energy.

The research area of joint design of control and communication is currently very active, especially in the context of event-triggered control. In [19], a joint optimization of control and communication is solved using dynamic programming by placing a communication scheduler in the sensor. In [20]–[22], the control law and event-condition are codesigned to match performance of periodic control using a lower communication rate. In [23], this idea is extended to decentralized systems.

The use of predictive control is also gaining popularity within the networked control community [24]–[27]. In [28], predictive methods and vector quantization are used to reduce the controller-to-actuator communication in multiple input systems. In [29], model predictive control (MPC) is used to design multiple actuator link scheduling and control signals. There have also been developments in using MPC under event-based sampling. In [30], a method for trading control performance and transmission rate in systems with multiple sensors is given. In [31], an event-based MPC is proposed where the decision to recalculate the control law is based on the difference between predicted and measured states.

The problem addressed in this paper, namely, the joint design of a self-triggering rule and the appropriate control signal using MPC has been less studied than its event-triggered counterpart. In [32], an approach relying on an exhaustive search that utilizes sub-optimal solutions giving the control policy and a corresponding self-triggering policy is presented. In [33], it is suggested that a portion of the open-loop trajectory produced by the MPC should be applied to the process. The time between reoptimizations is then decided via a self-triggering approach.

The approach taken in this paper differs from the above two in that the open loop cost we propose the MPC to solve is designed to be used in an adaptive sampling context. Further, our extension to handle multiple loops using a self-triggered MPC is new and so is the guarantee of conflict free transmissions.

The outline of this paper is as follows. In Section II, the self-triggered network scheduling and control problem is defined and formulated as a receding horizon control problem. Section III presents the open-loop optimal control problem for a single loop, to be solved by the receding horizon controller. The optimal solution is presented in Section IV. Section V presents the receding horizon control algorithm for a single loop in further detail and gives conditions for when it is stabilizing. The results are then extended to the multiple-loop case in Section VII, where conditions for stability and conflict free transmissions are given. The proposed method is explained and evaluated on the simulated examples in Section VIII. The concluding discussion is made in Section IX.

II. Self-Triggered Networked Control Architecture

We consider the problem of controlling $s \geq 1$ processes $\mathcal{P}_1$ through $\mathcal{P}_s$ over a shared communication network as in Fig. 1. The processes are controlled by the controller $\mathcal{C}$ that computes the appropriate control action and schedule for each process. Each process $\mathcal{P}_\ell$ is given by a linear time-invariant (LTI) system

$$x_\ell(k+1) = A_\ell x_\ell(k) + B_\ell u_\ell(k) \qquad x_\ell(k) \in \mathbb{R}^{n_\ell}, \quad u_\ell(k) \in \mathbb{R}^{m_\ell}. \tag{1}$$

The controller works in the following way: at time $k = k_\ell$, sensor $\mathcal{S}_\ell$ transmits a sample $x_\ell(k_\ell)$ to the controller that then computes the control signal $u_\ell(k_\ell)$ and sends it to the actuator $\mathcal{A}_\ell$. Here, $\ell \in \{1, 2, \ldots, s\}$ is the process index. The actuator in turn will apply this control signal to the process until a new value is received from the controller. Jointly with deciding $u_\ell(k_\ell)$ the controller also decides how many discrete time steps, say $I_\ell(k_\ell) \in \mathbb{N}^+ \triangleq \{1, 2, \ldots\}$, it will wait before it needs to change the control signal the next time. This value $I_\ell(k_\ell)$ is sent to the *network manager*, which will schedule the sensor $\mathcal{S}_\ell$ to send a new sample at time $k = k_\ell + I_\ell(k_\ell)$. To guarantee conflict-free transmissions on the network, only one sensor is allowed to transmit at every time instance. Hence, when deciding the time to wait $I_\ell(k_\ell)$, the controller must make sure that no other sensor $\mathcal{S}_q$, $q \neq \ell$, already is scheduled for transmission at time $k = k_\ell + I_\ell(k_\ell)$.

We propose that the controller $\mathcal{C}$ should be implemented as a receding horizon controller that for an individual loop ℓ at every sampling instant $k = k_\ell$ solves an open-loop optimal control problem. It does so by minimizing the infinite-horizon quadratic cost function

$$\sum_{l=0}^{\infty} \left(\|x_\ell(k_\ell + l)\|^2_{Q_\ell} + \|u_\ell(k_\ell + l)\|^2_{R_\ell} \right)$$

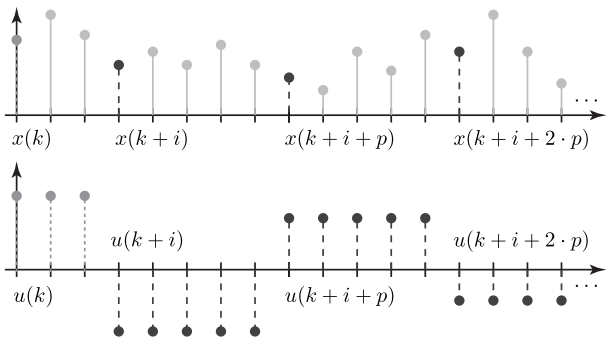

Fig. 2. Prediction horizon. Here, typical signal predictions are shown with $I(k) = 3$ and $p = 5$.

subject to the user defined weights Q_ℓ and R_ℓ, while considering system dynamics. In Section III, we will embellish this cost function such that control performance, inter sampling time, and overall network schedulability also are taken into consideration.

Remark 1: The focus is on a networked system where the network manager and controller are integrated in the same unit. This means that the controller can send the schedules that contain the transmission times of the sensors directly to the network manager. This information needs to be transmitted to the sensor node such that it knows when to sample and transmit. For example, using the IEEE 802.15.4 superframe, described in [3], this can be done without creating additional overhead. Here, the network manager broadcasts a beacon in the beginning of each superframe, which occurs at a fixed time interval. This beacon is received by all devices on the network and includes a schedule of the sensors that are allowed to transmit at given times.

III. Adaptive Sampling Strategy

For pedagogical ease, we will in this section study the case when we control a single process on the network allowing us to drop the loop index ℓ. The process we control has dynamics

$$x(k+1) = Ax(k) + Bu(k),\ x(k) \in \mathbb{R}^n, \quad u(k) \in \mathbb{R}^m \tag{2}$$

and the open-loop cost function we propose the controller to minimize at every sampling instant is

$$J(x(k), i, \mathcal{U}) = \frac{\alpha}{i} + \sum_{l=0}^{\infty} \left(\|x(k+l)\|_Q^2 + \|u(k+l)\|_R^2 \right) \tag{3}$$

where $\alpha \in \mathbb{R}^+$ is a design variable that is used to trade off the cost of sampling against the cost of control and $i = I(k) \in \{1, 2, \ldots, p\}$ is the number of discrete time units to wait before taking the next sample, where $p \in \mathbb{N}^+$ is the maximum number of time units to wait until taking the next sample. Further, the design variables $0 < Q$ and $0 < R$ are symmetric matrices of appropriate dimensions. We optimize this cost over the constraint that the control sequence $\mathcal{U} \triangleq \{u(k), u(k+1), \ldots\}$ should follow the specific shape shown in Fig. 2, for some fixed period p. The period p is the maximum amount of time steps the sensor is allowed to wait before taking the next sample. If $p = 1$, a sample is taken at every time instance and a regular receding horizon cost is obtained. If one, on the other hand, would select a large value of p, very long sampling intervals can be obtained. This can though affect the control performance significantly in case there are disturbances in the system that make the size of the state increase. Thus, the number of discrete time units $i = I(k) \in \{1, 2, \ldots, p\}$ to wait before taking the next sample $x(k+i)$ as well as the levels in the control sequence $\mathcal{U}$ are free variables over which we optimize. Note that neither the state nor the control values have a constrained magnitude.

The constraint on the shape of the control trajectory $\mathcal{U}$ is motivated by the idea that this sequence is applied to the actuator from time k until time instant $k+i$. At time $x(k+i)$, we take a new sample and redo the optimization. By this method, we get a joint optimization of the control signal to be applied as well as the number of time steps to the next sampling instant. The reason for letting the system be controlled by a control signal with period p after this is that we hope for the receding horizon algorithm to converge to this sampling rate. We will provide methods for choosing p so that this happens in Section V. The reason for wanting convergence to a down sampled control is that we want the system to be sampled at a slow rate when it has reached steady state, while we want it to be sampled faster during the transients.

If one includes the above-mentioned constraints, then (3) can be rewritten as

$$\begin{aligned} J(x(k), i, \mathcal{U}(i)) = \frac{\alpha}{i} &+ \sum_{l=0}^{i-1} \left(\|x(k+l)\|_Q^2 + \|u(k)\|_R^2 \right) \\ &+ \sum_{r=0}^{\infty} \left(\sum_{l=0}^{p-1} \left(\|x(k+i+r\cdot p+l)\|_Q^2 \right.\right. \\ &\left.\left. + \|u(k+i+r\cdot p)\|_R^2 \right) \right) \end{aligned} \tag{4}$$

where i and $\mathcal{U}(i) \triangleq \{u(k), u(k+i), u(k+i+\eta\cdot p), \ldots\}$, $\eta \in \mathbb{N}^+$, are the decision variables over which we optimize. The term α/i reflects the cost of sampling. We use this cost to weight the cost of sampling against the classical quadratic control performance cost. For a given $x(k)$, choosing a large α will force i to be larger and hence give longer intersampling times. By the construction of the cost, we may tune Q, R, and α via simulations to get the desired sampling behavior while maintaining the desired control performance. One could imagine a more general cost of sampling. Here, however, we found that α/i is sufficient to be able to trade control performance for communication cost (see simulations in Section VIII-B).

IV. Cost Function Minimization

Having defined the open-loop cost (4), we proceed by computing its optimal value. We start by noticing that even though we have a joint optimization problem, we may state it as

$$\underset{i}{\text{minimize}} \Big(\underset{\mathcal{U}(i)}{\text{minimize}}\ J(x(k), i, \mathcal{U}(i)) \Big). \tag{5}$$

We will use this separation and start by solving the inner problem that of minimizing $J(x(k), i, \mathcal{U}(i))$ for a given value of i. To derive the solution, and for future reference, we need to define some variables.

Definition 1: We define notation for the lifted model as

$$A^{(i)} = A^i, \quad B^{(i)} = \sum_{q=0}^{i-1} A^q B.$$

and notation for the generalized weighting matrices associated with (4) as

$$\begin{aligned} Q^{(i)} &= Q^{(i-1)} + A^{(i-1)^T} Q A^{(i-1)} \\ R^{(i)} &= R^{(i-1)} + B^{(i-1)^T} Q B^{(i-1)} + R \\ N^{(i)} &= N^{(i-1)} + A^{(i-1)^T} Q B^{(i-1)} \end{aligned}$$

where $i \in \{1, 2, \ldots, p\}$, $Q^{(1)} = Q$, $R^{(1)} = R$, and $N^{(1)} = 0$.

Using Definition 1, it is straightforward to show the following lemma.

Lemma 1: It holds that

$$\begin{aligned} &\sum_{l=0}^{i-1} \left(\|x(k+l)\|_Q^2 + \|u(k)\|_R^2 \right) \\ &\qquad = x(k)^T Q^{(i)} x(k) + u(k)^T R^{(i)} u(k) + 2x(k)^T N^{(i)} u(k) \end{aligned}$$

and $x(k+i) = A^{(i)} x(k) + B^{(i)} u(k)$.

Lemma 2: Assume that $0 < Q$ and $0 < R$, and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Then

$$\begin{aligned} \min_{\mathcal{U}(i)} \sum_{r=0}^{\infty} \Bigg(&\sum_{l=0}^{p-1} \left(\|x(k+i+r\cdot p+l)\|_Q^2 \right. \\ &\left. + \|u(k+i+r\cdot p)\|_R^2 \right) \Bigg) = \|x(k+i)\|_{P^{(p)}}^2 \end{aligned}$$

and the minimizing control signal characterizing $\mathcal{U}(i)$ is given by

$$u(k+i+r\cdot p) = -L^{(p)} x(k+i+r\cdot p).$$

In the above

$$\begin{aligned} P^{(p)} &= Q^{(p)} + A^{(p)^T} P^{(p)} A^{(p)} \\ &\quad - \left(A^{(p)^T} P^{(p)} B^{(p)} + N^{(p)} \right) L^{(p)} \\ L^{(p)} &= \left(R^{(p)} + B^{(p)^T} P^{(p)} B^{(p)} \right)^{-1} \\ &\quad \times \left(A^{(p)^T} P^{(p)} B^{(p)} + N^{(p)} \right)^T. \end{aligned} \tag{6}$$

Proof: The proof is given in Appendix A. □

Using the above results, we may formulate the main result of this section as follows.

Theorem 1 (Closed Form Solution): Assume that $0 < Q$ and $0 < R$, and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Then

$$\min_{\mathcal{U}(i)} J(x(k), i, \mathcal{U}(i)) = \frac{\alpha}{i} + \|x(k)\|_{P^{(i)}}^2 \tag{7}$$

where

$$\begin{aligned} P^{(i)} &= Q^{(i)} + A^{(i)^T} P^{(p)} A^{(i)} \\ &\quad - \left(A^{(i)^T} P^{(p)} B^{(i)} + N^{(i)} \right) L^{(i)} \\ L^{(i)} &= \left(R^{(i)} + B^{(i)^T} P^{(p)} B^{(i)} \right)^{-1} \\ &\quad \times \left(A^{(i)^T} P^{(p)} B^{(i)} + N^{(i)} \right)^T \end{aligned} \tag{8}$$

and $P^{(p)}$ is given by Lemma 2. Denoting the vector of all ones in $\mathbb{R}^n$ as $\mathbf{1}_n$, the minimizing control signal sequence is given by

$$\mathcal{U}^* = \left\{ -L^{(i)} x(k) \mathbf{1}_i^T, -L^{(p)} x(k+i+r\cdot p) \mathbf{1}_p^T, \ldots \right\}, \quad r \in \mathbb{N}$$

where also $L^{(p)}$ is given by Lemma 2.

Proof: The proof is given in Appendix B. ■

Now, getting back to the original problem (5), provided that the assumptions of Theorem 1 hold, we may apply it giving that

$$\min_{i, \mathcal{U}(i)} J(x(k), i, \mathcal{U}(i)) = \min_i \left\{ \frac{\alpha}{i} + \|x(k)\|_{P^{(i)}}^2 \right\}.$$

Unfortunately, the authors are not aware of any method to solve this problem in general. However, if i is restricted to a known *finite* set $\mathcal{I}^0 \subset \mathbb{N}^+$, we may find the optimal value within this set for a given value of $x(k)$ by simply evaluating $\frac{\alpha}{i} + \|x(k)\|_{P^{(i)}}^2 \ \forall i \in \mathcal{I}^0$ and by this obtaining the i, which gives the lowest value of the cost. This procedure gives the optimum of (5). Note that the computational complexity of finding the optimum is not necessarily high, as we may compute $P^{(i)} \ \forall i \in \mathcal{I}^0$ offline prior to execution.

V. Single-Loop Self-Triggered MPC

Having presented the receding horizon cost, we now continue with formulating its implementation in further detail. We will assume that $0 < Q$ and $0 < R$, and that the down-sampled pair $(A^{(p)}, B^{(p)})$ is controllable. Let us also assume that the *finite* and *nonempty* set $\mathcal{I}^0 \subset \mathbb{N}^+$ is given and let

$$\gamma = \max \mathcal{I}^0. \tag{9}$$

From this, we may use Theorem 1 to compute the pairs $(P^{(i)}, L^{(i)}) \ \forall i \in \mathcal{I}^0$ and formulate our proposed receding horizon control algorithm for a single loop.

Remark 2: Note that the control signal value is sent to the actuator at the same time as the controller requests the scheduling of the next sample by the sensor.

Remark 3: The proposed algorithm guarantees a minimum and a maximum intersampling time. The minimum time is 1 time step in the time scale of the underlying process (2) and the maximum inter sampling time is γ time steps. This implies that there is some minimum attention to every loop independent of the predicted evolution of the process.

Remark 4: Even though we are working in uniformly sampled discrete time, the state is not sampled at every time instant k. Instead, the set of samples of the state actually taken is given by the set $\mathcal{D}$, assuming that the first sample is taken at $k = 0$, is given by

$$\mathcal{D} = \{x(0), x(I(0)), x(I(I(0))), \ldots\}. \tag{10}$$

Algorithm 1 Single Loop Self-Triggered MPC

1) *At time $k = k'$ the sample $x(k')$ is transmitted by the sensor to the controller.*
2) *Using $x(k')$ the controller computes*

$$I(k') = \arg\min_{i\in\mathcal{I}^0} \frac{\alpha}{i} + \|x(k')\|^2_{P^{(i)}}$$
$$u(k') = -L^{(I(k'))}x(k')$$

3) a) *The controller sends $u(k')$ to the actuator which applies $u(k) = u(k')$ to (2) until $k = k' + I(k')$.*
   b) *The network manager is requested to schedule the sensor to transmit a new sample at time $k = k' + I(k')$.*

VI. Analysis

Having established and detailed our receding horizon control law for a single loop, we continue with giving conditions for when it is stabilizing. Letting $\lambda(A)$ to denote the set of eigenvalues of A, we first recall the following controllability conditions.

Lemma 3 [34]: The system $(A^{(i)}, B^{(i)})$ is controllable if and only if the pair (A, B) is controllable and A has no eigenvalue $\lambda \in \lambda(A)$ such that $\lambda \neq 1$ and $\lambda^i = 1$.

Using the above, and the results of Section IV, we may now give conditions for when the proposed receding horizon control algorithm is stabilizing.

Theorem 2 (Practical Stability [35]): Assume $0 < Q$ and $0 < R$ and that (A, B) is controllable. If we choose $i \in \mathcal{I}^0 \subset \mathbb{N}^+$ and $p = p^*$ given by

$$p^* = \max\{i | i \in \mathcal{I}^0, \forall\lambda \in \lambda(A)\ \lambda^i \neq 1 \text{ if } \lambda \neq 1\} \quad (11)$$

and apply Algorithm 1, then the system state is ultimately bounded as per the following:

$$\frac{\alpha}{\gamma} \leq \lim_{k\to\infty} \min_{i\in\mathcal{I}^0} \left(\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}}\right) \leq \frac{\alpha}{\epsilon}\left(\frac{1}{p^*} - (1-\epsilon)\frac{1}{\gamma}\right).$$

In the above, γ is as in (9) and ϵ is the largest value in the interval $(0, 1]$ which $\forall i \in \mathcal{I}^0$ fulfills

$$(A^{(i)} - B^{(i)}L^{(i)})^T P^{(p^*)}(A^{(i)} - B^{(i)}L^{(i)}) \leq (1-\epsilon)P^{(i)}$$

and is guaranteed to exist.

Proof: The proof is given in Appendix C. ■

Remark 5: The bound given in Theorem 2 scales linearly with the choice of α.

Assumption 1: Assume that $\nexists\lambda \in \lambda(A)$ except possibly $\lambda = 1$ such that $|\lambda| = 1$ and the complex argument $\angle\lambda = (2\pi/\gamma)\cdot n$ for some $n \in \mathbb{N}^+$.

Lemma 4: Let Assumption 1 hold, then $p^* = \gamma$.

Proof: The proof is given in Appendix D. ■

Corollary 1 (Asymptotic Stability of the Origin): Assume $0 < Q$ and $0 < R$, and that (A, B) is controllable. Further assume that either Assumption 1 holds or $\alpha = 0$. If we choose $i \in \mathcal{I}^0 \subset \mathbb{N}^+$ and $p = p^*$ given by (11) and apply Algorithm 1, then

$$\lim_{k\to\infty} x(k) = 0.$$

Proof: The proof is given in Appendix E. ■

From the above results, we may note the following.

Remark 6: If the assumptions of Theorem 2 hold, then Corollary 1 can be used to ensure asymptotic stability of the origin, except in the extremely rare case that the underlying system (A, B) becomes uncontrollable under down sampling by a factor γ (see Lemma 3). Otherwise, one may use Lemma 4 to redesign $\mathcal{I}^0$ giving a new value of γ that recovers the case $p^* = \gamma$.

Remark 7: It is worth noticing that the developed framework is not limited to just varying the time to the next sample i as in Fig. 2. One could expand this using a multistep approach wherein the intersampling time is optimized over several frames before reverting to constant sampling with period p.

VII. Extension to Multiple Loops

Having detailed the controller for the case with a single loop on the network and given conditions for when it is stabilizing, we now continue with extending to the case when we control multiple loops on the network, as described in Fig. 1. The idea is that the controller $\mathcal{C}$ now will run s such single-loop controllers described in Algorithm 1 in parallel, one for each process $\mathcal{P}_\ell$, $\ell \in \mathbb{L} = \{1, 2, \ldots, s\}$, controlled over the network. To guarantee conflict-free communication on the network, the controller $\mathcal{C}$ will, at the same time, centrally coordinate the transmissions of the different loops.

A. Cost Function

We start by extending the results in Section III to the case when we have multiple loops. The cost function we propose the controller to minimize at every sampling instant for loop ℓ is then

$$J_\ell(x_\ell(k), i, \mathcal{U}_\ell(i)) = \frac{\alpha_\ell}{i} + \sum_{l=0}^{i-1}\left(\|x_\ell(k+l)\|^2_{Q_\ell} + \|u_\ell(k)\|^2_{R_\ell}\right) + \sum_{r=0}^{\infty}\left(\sum_{l=0}^{p_\ell-1}\left(\|x_\ell(k+i+r\cdot p_\ell+l)\|^2_{Q_\ell} + \|u_\ell(k+i+r\cdot p_\ell)\|^2_{R_\ell}\right)\right)$$

derived in the same way as (4) now with $\alpha_\ell \in \mathbb{R}^+$, $0 < Q_\ell$ $0 < R_\ell$ and period $p_\ell \in \mathbb{N}^+$ specific for the control of process $\mathcal{P}_\ell$ given by (1). From this, we can state the following.

Definition 2: We define the notation in the multiple-loop case following Definition 1. For a matrix E_ℓ, e.g., A_ℓ and Q_ℓ, we denote $(E_\ell)^{(i)}$ by $E_\ell^{(i)}$.

Theorem 3 (Closed-Form Solution): Assume that $0 < Q_\ell$ and $0 < R_\ell$, and that the pair $(A_\ell^{(p)}, B_\ell^{(p)})$ is controllable. Then

$$\min_{\mathcal{U}_\ell(i)} J_\ell(x_\ell(k), i, \mathcal{U}_\ell(i)) = \frac{\alpha_\ell}{i} + \|x_\ell(k)\|^2_{P_\ell^{(i)}}$$

where

$$P_\ell^{(i)} = Q_\ell^{(i)} + A_\ell^{(i)^T} P_\ell^{(p_\ell)} A_\ell^{(i)} - \left(A_\ell^{(i)^T} P_\ell^{(p_\ell)} B_\ell^{(i)} + N_\ell^{(i)}\right) L_\ell^{(i)}$$

$$L_\ell^{(i)} = \left(R_\ell^{(i)} + B_\ell^{(i)^T} P_\ell^{(p_\ell)} B_\ell^{(i)}\right)^{-1} \left(A_\ell^{(i)^T} P_\ell^{(p_\ell)} B_\ell^{(i)} + N_\ell^{(i)}\right)^T$$

and

$$\begin{aligned} P_\ell^{(p_\ell)} &= Q_\ell^{(p_\ell)} + A_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} A_\ell^{(p_\ell)} \\ &\quad - \left(A_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} B_\ell^{(p_\ell)} + N_\ell^{(p_\ell)}\right) L_\ell^{(p_\ell)} \\ L_\ell^{(p_\ell)} &= \left(R_\ell^{(p_\ell)} + B_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} B_\ell^{(p_\ell)}\right)^{-1} \\ &\quad \times \left(A_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} B_\ell^{(p_\ell)} + N_\ell^{(p_\ell)}\right)^T. \end{aligned}$$

Denoting the vector of all ones in $\mathbb{R}^n$ as $\mathbf{1}_n$, the minimizing control signal sequence is given by

$$\mathcal{U}_\ell^* = \left\{ - L_\ell^{(i)} x_\ell(k) \mathbf{1}_i^T, -L_\ell^{(p_\ell)} x_\ell(k + i + r \cdot p_\ell) \mathbf{1}_{p_\ell}^T, \ldots \right\}, \quad r \in \mathbb{N}.$$

Proof: Application of Theorem 1 on the individual loops. □

B. Multiple-Loop Self-Triggered MPC

To formulate our multiple-loop receding horizon control law, we will reuse the results in Section V and apply them on a per loop basis.

For each process $\mathcal{P}_\ell$ with dynamics given in (1), let the weights $\alpha_\ell \in \mathbb{R}^+$, $0 < Q_\ell \; 0 < R_\ell$, and period $p_\ell \in \mathbb{N}^+$ specific to the process be defined. If we further define the *finite* set $\mathcal{I}_\ell^0 \subset \mathbb{N}^+$ for loop ℓ, we may apply Theorem 3 to compute the pairs $(P_\ell^{(i)}, L_\ell^{(i)})$ $\forall\, i \in \mathcal{I}_\ell^0$. Provided of course that the pair $(A_\ell^{(p_\ell)}, A_\ell^{(p_\ell)})$ is controllable.

As discussed in Section II, when choosing i, the controller must take into consideration what transmission times other loops have reserved as well as the overall network schedulability. Hence, at time $k = k_\ell$, loop ℓ is restricted to choose $i \in \mathcal{I}_\ell(k_\ell) \subseteq \mathcal{I}_\ell^0$ where $\mathcal{I}_\ell(k_\ell)$ contains the feasible values of i that gives collision-free scheduling of the network. Note that at time $k = k_\ell$, the control input u_ℓ and time for the next sample I_ℓ only have to be computed for a *single* loop ℓ. How $\mathcal{I}_\ell(k_\ell)$ should be constructed when multiple loops are present on the network is discussed further later in this section.

We may now continue with formulating our proposed algorithm for controlling multiple processes over the network. The following algorithm is executed whenever a sample is received by the controller.

Remark 8: When the controller is initialized at time $k = 0$, it is assumed that the controller has knowledge of the state $x_\ell(0)$ for all processes $\mathcal{P}_\ell$ controlled over the network. It will then execute Algorithm 2 entering at step 2, in the order of increasing loop index ℓ.

Algorithm 2 Multiple Loop Self-Triggered MPC

1) *At time $k = k_\ell$ the sample $x_\ell(k_\ell)$ of process $\mathcal{P}_\ell$ is transmitted by the sensor $\mathcal{S}_\ell$ to the controller $\mathcal{C}$.*
2) *The controller $\mathcal{C}$ constructs $\mathcal{I}_\ell(k_\ell)$.*
3) *Using $x_\ell(k_\ell)$ the controller $\mathcal{C}$ computes*

$$I_\ell(k_\ell) = \arg \min_{i \in \mathcal{I}_\ell(k_\ell)} \frac{\alpha_\ell}{i} + \|x_\ell(k_\ell)\|^2_{P_\ell^{(i)}}$$

$$u_\ell(k_\ell) = -L_\ell^{(I(k_\ell))} x_\ell(k_\ell).$$

4) a) *The controller $\mathcal{C}$ sends $u_\ell(k_\ell)$ to the actuator $\mathcal{A}_\ell$ which applies $u_\ell(k) = u_\ell(k_\ell)$ to (1) until $k = k_\ell + I_\ell(k_\ell)$.*
   b) *The* Network Manager *is requested to schedule the sensor $\mathcal{S}_\ell$ to transmit a new sample at time $k = k_\ell + I_\ell(k_\ell)$.*

Note that at time k step 3 only needs to be performed for loop l.

C. Schedulability

What remains to be detailed in the multiple-loop receding horizon control law is a mechanism for loop ℓ to choose $\mathcal{I}_\ell(k_\ell)$ to achieve collision-free scheduling. We now continue with giving conditions for when this holds.

First, we note that when using Theorem 3, we make the implicit assumption that it is possible to apply the corresponding optimal control signal sequence $\mathcal{U}_\ell^*$. For this to be possible, we must be able to measure the state $x_\ell(k)$ at the future time instances

$$\begin{aligned} S_\ell(k_\ell) = \{&k_\ell + I_\ell(k_\ell),\ k_\ell + I_\ell(k_\ell) + p_\ell, \\ &k_\ell + I_\ell(k_\ell) + 2p_\ell,\ k_\ell + I_\ell(k_\ell) + 3p_\ell, \ldots\}. \end{aligned} \quad (12)$$

Hence, this sampling pattern must be reserved for use by sensor $\mathcal{S}_\ell$. We state the following to give conditions for when this is possible.

Lemma 5: Let loop ℓ choose its set $\mathcal{I}_\ell(k_\ell)$ of feasible times to wait until the next sample to be

$$\mathcal{I}_\ell(k_\ell) = \left\{ i \in \mathcal{I}_\ell^0 | i \neq k_q^{\text{next}} - k_\ell + n \cdot p_q - m \cdot p_\ell, \; m, n \in \mathbb{N},\ q \in \mathbb{L} \setminus \{\ell\} \right\}$$

where k_q^{next} is the next transmission time of sensor $\mathcal{S}_q$. Then it is possible to reserve the needed sampling pattern $S_\ell(k_\ell)$ in (12) at time $k = k_\ell$.

Proof: The proof is given in Appendix F. □

Constructing $\mathcal{I}_\ell(k_\ell)$ as above, we are not guaranteed that $\mathcal{I}_\ell(k_\ell) \neq \emptyset$. To guarantee this, we make the following assumption.

Assumption 2: Assume that for every loop ℓ on the network $\mathcal{I}_\ell^0 = \mathcal{I}^0$ and $p_\ell = p$. Further assume that $\mathbb{L} \subseteq \mathcal{I}^0$ and $\max \mathbb{L} \leq p$.

Theorem 4: Let Assumption 2 hold. If every loop ℓ chooses

$$\mathcal{I}_\ell(k_\ell) = \{ i \in \mathcal{I}^0 | i \neq k_q^{\text{next}} - k_\ell + r \cdot p, \quad r \in \mathbb{Z}, \quad q \in \mathbb{L} \setminus \{\ell\} \}$$

all transmissions on the network will be conflict free and

it will always be possible to reserve the needed sampling pattern $S_\ell(k_\ell)$ in (12).

Proof: The proof is given in Appendix G. □

Remark 9: The result in Lemma 5 requires the reservation of an infinite sequence. This is no longer required in Theorem 4 as all loops cooperate when choosing the set of feasible times to wait. In fact, loop ℓ only needs to know the current time k_ℓ, the period p, and the times when the other loops will transmit next $k_q^{\text{next}}\ \forall q \in \mathbb{L} \setminus \{\ell\}$ to find its own value $\mathcal{I}_\ell(k_\ell)$.

Remark 10: If Assumption 2 holds and every loop on the network chooses $\mathcal{I}_\ell(k_\ell)$ according to Theorem 4, then it is guaranteed that at time k_ℓ, we can reserve (12) and that no other loop can make conflicting reservations. Hence, at time $k_\ell + I_\ell(k_\ell)$, the sequence

$$S_\ell(k_\ell + I_\ell(k_\ell)) = \{k_\ell + I_\ell(k_\ell) + p, \\ k_\ell + I_\ell(k_\ell) + 2p,\ k_\ell + I_\ell(k_\ell) + 3p, \ldots\}$$

is guaranteed to be available. Thus, $p \in \mathcal{I}_\ell(k_\ell + I_\ell(k_\ell))$.

D. Stability

We continue with giving conditions for when the multiple loop receding horizon control law described in Algorithm 2 is stabilizing. Extending the theory developed in Section V to the multiple loop case, we may state the following.

Theorem 5: Assume $0 < Q_\ell$ and $0 < R_\ell$, and that (A_ℓ, B_ℓ) is controllable. Further let Assumption 2 hold. If we then choose $i \in \mathcal{I}_\ell(k) \subseteq \mathcal{I}^0 \subset \mathbb{N}^+$, with $\mathcal{I}_\ell(k)$ chosen as in Theorem 4, and $p = p^*$ given by

$$p^* = \max\{i | i \in \mathcal{I}^0, \forall\, \ell\ \forall \lambda \in \lambda(A_\ell)\ \lambda^i \neq 1 \text{ if } \lambda \neq 1\} \quad (13)$$

and apply Algorithm 2, then as $k \to \infty$

$$\frac{\alpha_\ell}{\gamma} \leq \min_{i \in \mathcal{I}_\ell(k)} \left(\frac{\alpha_\ell}{i} + \|x_\ell(k)\|^2_{P_\ell^{(i)}} \right) \leq \frac{\alpha_\ell}{\epsilon_\ell} \left(\frac{1}{p^*} - (1 - \epsilon_\ell)\frac{1}{\gamma} \right)$$

where $\gamma = \max \mathcal{I}^0$ and ϵ_ℓ is the largest value in the interval $(0, 1]$, which $\forall i \in \mathcal{I}^0$ fulfills

$$\left(A_\ell^{(i)} - B_\ell^{(i)} L_\ell^{(i)}\right)^T P_\ell^{(p^*)} \left(A_\ell^{(i)} - B_\ell^{(i)} L_\ell^{(i)}\right) \leq (1 - \epsilon_\ell) P_\ell^{(i)}.$$

Proof: The proof is given in Appendix H. □

Corollary 2: Assume $0 < Q_\ell$ and $0 < R_\ell$, and that (A_ℓ, B_ℓ) is controllable. Further let Assumption 2 hold. In addition, let $\mathcal{I}^0$ be chosen so that the resulting $\gamma = \max \mathcal{I}^0$ guarantees that Assumption 1 holds for every loop ℓ or alternatively let $\alpha_\ell = 0$ for every loop ℓ. If we then choose $i \in \mathcal{I}_\ell(k) \subseteq \mathcal{I}^0 \subset \mathbb{N}^+$, with $\mathcal{I}_\ell(k)$ chosen as in Theorem 4, and $p = p^*$ given by (13) and apply Algorithm 2, it holds that

$$\lim_{k \to \infty} x_\ell(k) = 0.$$

Proof: The proof follows from the results in Theorem 5 analogous to the proof of Corollary 1. □

TABLE I
PRECOMPUTED CONTROL LAWS WITH RELATED COST FUNCTIONS AND INTERMEDIATE VARIABLES

i	$A^{(i)}$	$B^{(i)}$	$Q^{(i)}$	$R^{(i)}$	$N^{(i)}$	$L^{(i)}$	$P^{(i)}$	$V^{(i)}(x)$
1	1	1	1	1	0	0.70	1.70	$\alpha/1 + P^{(1)} \cdot x^2$
2	1	2	2	3	1	0.46	1.73	$\alpha/2 + P^{(2)} \cdot x^2$
3	1	3	3	8	3	0.35	1.89	$\alpha/3 + P^{(3)} \cdot x^2$
4	1	4	4	18	6	0.28	2.08	$\alpha/4 + P^{(4)} \cdot x^2$
5=p^*	1	5	5	35	10	0.23	2.30	$\alpha/5 + P^{(5)} \cdot x^2$

VIII. SIMULATION STUDIES

To illustrate the proposed theory, we now continue with giving simulations. First, we show how the control law works when a single loop is controlled over the network and focus on the loop-specific mechanisms of the controller. Second, we illustrate how the controller works when multiple loops are present on the network and focus on how the controller allocates network access to different loops.

A. Single Loop

Let us exemplify and discuss how the controller handles the control performance versus communication rate tradeoff in an individual loop. We do this by studying the case with a single system on the network. The system we study is the single integrator system, which we discretize using sample and hold with sampling time $T_s = 1$ s giving us $x(k+1) = Ax(k) + Bu(k)$ with $(A, B) = (1, 1)$. Since we want the resulting self-triggered MPC described in Algorithm 1 to be stabilizing, we need to make sure that our design fulfills the conditions of Theorem 2. If we further want it to be asymptotically stabilizing, we in addition need it to fulfill the conditions of Corollary 1.

The design procedure is then as follows. First, we note that the system (A, B) is controllable. The next step is to decide the weights $0 < Q$ and $0 < R$ in the quadratic cost function (3). This is done in the same way as in classical linear quadratic control [36]. Here, we for simplicity choose $Q = 1$ and $R = 1$. We note that the system only has the eigenvalue $\lambda = 1$, fulfilling Assumption 1, so that (11) in Theorem 2 gives $p^* = \max \mathcal{I}^0$. Hence, Corollary 1 and thus Theorem 2, will hold for every choice of $\mathcal{I}^0$. This means that we may choose the elements in $\mathcal{I}^0$, i.e., the possible down sampling rates freely. A natural way to choose them is to decide on a maximum allowed down sampling rate and then choose $\mathcal{I}^0$ to contain all rates from 1 up to this number. Let us say that we here want the system to be sampled at least every $5 \cdot T_s$ s, then a good choice is $\mathcal{I}^0 = \{1, 2, 3, 4, 5\}$, giving $p^* = \max \mathcal{I}^0 = 5$.

Now having guaranteed that the conditions of Theorem 2 and Corollary 1 hold, we have also guaranteed that the conditions of Theorem 1 are fulfilled. Hence, we may use it to compute the state feedback gains and cost function matrices that are used in Algorithm 1. The results from these computations are shown in Table I, together with some of the intermediate variables from Definition 1. Here, we see that the cost functions are quadratic functions in the state x, where the coefficients $P^{(i)}$ are functions of Q and R. We also see that the cost to sample α/i enters linearly and as we change it,

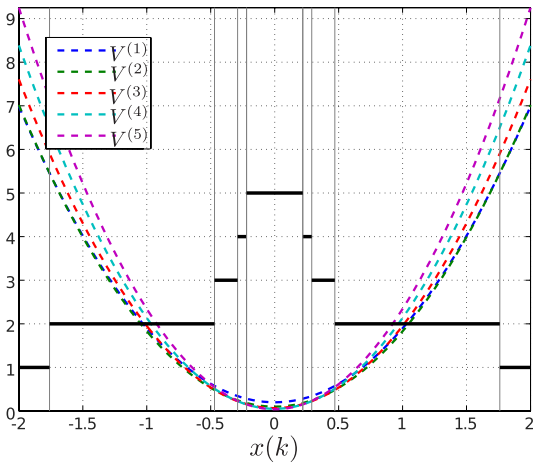

Fig. 3. Cost functions $V^{(i)}(x(k))$ (dashed line) together with the partitioning of the state space and the time to wait $I(k) = \arg\min_{\mathcal{I}^0} V^{(i)}(x(k))$ (solid line).

we will change the offset level of the curves and thereby their values related to each other. However, it will not affect the state feedback gains.

A graphical illustration of the cost functions in Table I, for the choice $\alpha = 0.2$, is shown in Fig. 3 together with the curve $I(k) = \arg\min_{\mathcal{I}^0} V^{(i)}(x(k))$, i.e., the index of the cost function which has the lowest value for a given state $x(k)$. This is the partitioning of the state space that the self-triggered MPC controller will use to choose which of the state feedback gains to apply and how long to wait before sampling again.

Applying our self-triggered MPC described in Algorithm 1 using the results in Table I to our integrator system when initialized in $x(0) = 2$, we get the response shown in Fig. 4(a). Note here that the system will converge to the fixed sampling rate p^* as the state converges.

It may now appear as it is sufficient to use periodic control and sample the system every $p^* \cdot T_s$ s to get good control performance. To compare the performance of this periodic sampling strategy with the self-triggered strategy above, we apply the control that minimizes the same cost function (3) as above with the exception that the system now may only be sampled every $p^* \cdot T_s$ s. This is in fact the same as using the receding horizon control above while restricting the controller to choose $i = p^*$ every time. The resulting simulations are shown in Fig. 4(b). As seen, there is a large degradation of the performance in the transient while the stationary behavior is almost the same. By this, we can conclude that it is not sufficient to sample the system every $p^* \cdot T_s$ s if we want to achieve the same transient performance as with the self-triggered sampling.

In the initial transient response, the self-triggered MPC controller sampled after one time instant. This indicates that there is performance to gain by sampling every time instant. To investigate this, we apply the control that minimizes the same cost function (3), now with the exception that the system may be sampled every T_s s, i.e., classical unconstrained linear quadratic control. Now simulating the system, we get the response shown in Fig. 4(c). As expected, we get slightly better transient performance in this case compared with our self-triggered sampling scheme; however, it is comparable.

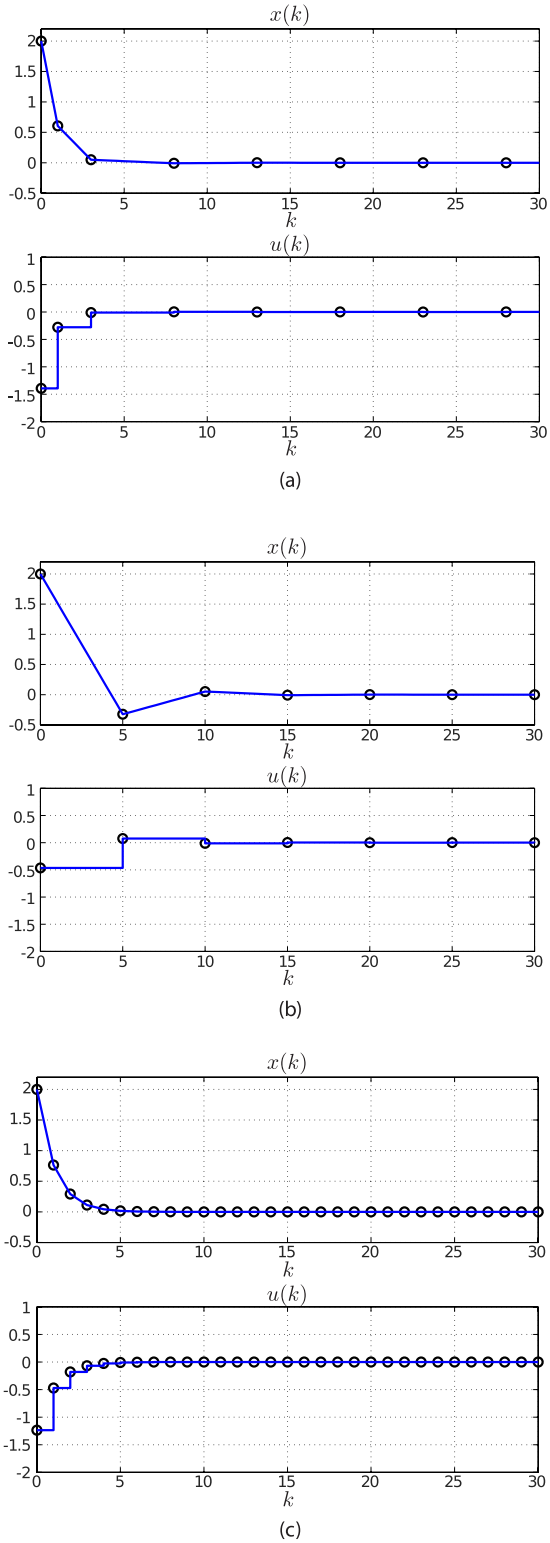

Fig. 4. Comparison of control performance for different sampling policies. (a) System response of the integrator system when minimizing the cost using our single-loop self-triggered MPC. (b) System response of the integrator system when minimizing the cost by sampling every fifth second. (c) System response of the integrator system when minimizing the cost by sampling every second.

Note, however, that this improvement comes at the cost of a drastically increased communication need, which may not be suitable for systems where multiple loops share the same wireless medium.

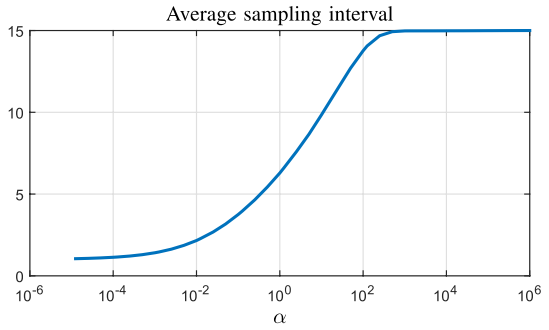

Fig. 5. Average sampling interval for different values of α for the integrator system.

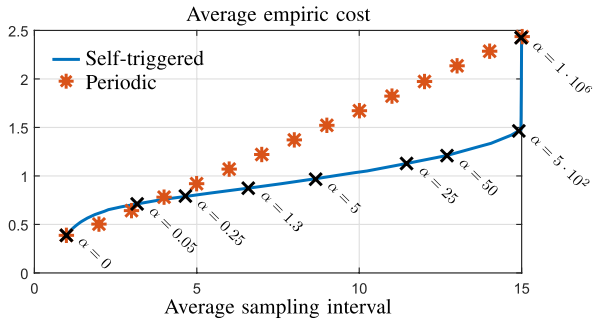

Fig. 6. Performance of the integrator using the self-triggered algorithm compared with the simple periodic algorithm for different sampling intervals. Some values of α are marked with the black crosses.

From the above, we may conclude that our self-triggered MPC combines the low communication rate in stationarity of the slow periodic controller with the quick transient response of the fast periodic sampling. In fact, we may, using our method, recover the transient behavior of fast periodic sampling at the communication cost of one extra sample compared with slow periodic sampling. The reason for this is that the fast sampling rate only is needed in the transient while we in stationarity can obtain sufficient performance with a lower rate.

B. Comparison With Periodic Control

To make a more extensive comparison between the self-triggered algorithm and periodic sampling, we sweep the value of the design parameter α in the cost function and run 100 simulations of 10 000 samples for each value. To add randomness to the process, the model (1) is extended to include disturbances such that for process ℓ

$$x_\ell(k+1) = A_\ell x_\ell(k) + B_\ell u_\ell(k) + E_\ell \omega_\ell(k) \tag{14}$$

where $\omega_\ell(k) \sim \mathcal{N}(0, \sigma_\ell^2)$ is zero-mean normal distributed with variance $\sigma_\ell^2 = 0.1$ and $E_\ell = 1$. We increase the maximum allowed sampling interval to $p^* = \max \mathcal{I}^0 = 15$. This though has the effect, that $P^{(1)} = 1.83$ whereas $P^{(2)} = 1.74$, such that it is always more favorable to have a sample interval of two samples instead of one, even when $\alpha = 0$. We therefore changed the cost of control to $R = 0.1$ such that we allow for larger control signals and force the process to sample at every time instance if $\alpha = 0$ is chosen. The initial state is randomly generated as $x_\ell(0) \sim \mathcal{N}(0, \sigma_{x_0,\ell}^2)$, where $\sigma_{x_0,\ell}^2 = 2\,500$.

Fig. 5 shows the average sampling interval during the simulations for different choices of α for the integrator system. The relation between the value of α and the average sampling interval is in general monotonic but highly affected by the system to be controlled and the statistics of the disturbances in the system.

Since we now have the range of values in which we want to sweep α, we present a simple periodic MPC cost to which we compare our algorithm. This cost is given by

$$J_\ell(x_\ell(k), \mathcal{U}_\ell) = \sum_{r=0}^{\infty} \sum_{l=0}^{T_s} \left(\|x_\ell(k+l+rT_s)\|_{Q_\ell}^2 + \|u_\ell(k+l+rT_s)\|_{R_\ell}^2 \right)$$

which using Definition 1 and Lemmas 1 and 2 can be rewritten as

$$J_\ell(x_\ell(k), \mathcal{U}_\ell) = \sum_{r=0}^{\infty} \|x_\ell(k+rT_s)\|_{Q_\ell^{(T_s)}}^2 + \|u_\ell(k+rT_s)\|_{R_\ell^{(T_s)}}^2 + 2x_\ell(k+rT_s)^T N_\ell^{(i)} u_\ell(k+rT_s) \tag{15}$$

for sampling period T_s where

$$x_\ell(k+T_s) = A_\ell^{(T_s)} x_\ell(k) + B_\ell^{(T_s)} u_\ell(k).$$

When $T_s = 1$, (15) reduces to classical unconstrained linear quadratic control.

The empiric cost for each simulation is calculated by

$$\frac{1}{T} \sum_{k=0}^{T-1} x_\ell(k)^T Q_\ell x_\ell(k) + u_\ell(k)^T R_\ell u_\ell(k) \tag{16}$$

where $T = 10\,000$ is the length of the simulation.

Fig. 6 shows the average performance that is obtained for different averaged sampling intervals obtained by sweeping α from 0 to 10^6. Fig. 6 shows that the self-triggered algorithm performs significantly better than periodic sampling, especially when communication costs are medium to high. However, at sampling intervals of 2 and 3 time steps, the self-triggered algorithm performs slightly worse than the periodic algorithm. This is caused by the fact that if the state of the system is close to zero when sampled, the cost of sampling is much higher than the cost of the state error and control, hence the time until the next sample is taken is large. To avoid this phenomenon, in future work, one could consider the statistics of the process noise in the cost function. It can further be noted that when $\alpha \to \infty$, the performance of the self-triggered algorithm is very close to the cost of the periodic algorithm. This is as expected, since the cost of α will be greater than the cost of the state and control, which will result in sample intervals of $i = p^* = 15$. This reduces the self-triggered algorithm to periodic control.

C. Multiple Loops

We now continue with performing a simulation study, where we control two systems over the same network. We will start by showing a simple example followed by a more extensive performance comparison of the Algorithm 2 with periodic control. We will keep the integrator system

from Section VIII-A now denoting it process $\mathcal{P}_1$ with dynamics $x_1(k+1) = A_1x_1(k) + B_1u_1(k)$ with $(A_1, B_1) = (1, 1)$ as before. In addition, we will the control process $\mathcal{P}_2$, which is a double integrator system, which we discretize using sample and hold with sampling time $T_s = 1$ s giving

$$\underbrace{\begin{pmatrix} x_2^1(k+1) \\ x_2^2(k+1) \end{pmatrix}}_{x_2(k+1)} = \underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{A_2} \underbrace{\begin{pmatrix} x_2^1(k) \\ x_2^2(k) \end{pmatrix}}_{x_2(k)} + \underbrace{\begin{pmatrix} 1 \\ 0.5 \end{pmatrix}}_{B_2} u_2(k).$$

We wish to control these processes using our proposed multiple-loop self-triggered MPC described in Algorithm 2. As we wish to stabilize these systems, we start by checking the conditions of Theorem 5 and Corollary 2. First, we may easily verify that both the pairs (A_1, B_1) and (A_2, B_2) are controllable. To use the stability results, we need Assumption 2 to hold, implying that we must choose $p_1 = p_2 = p$, $\mathcal{I}_1^0 = \mathcal{I}_2^0 = \mathcal{I}^0$, and choose $\mathcal{I}^0$ such that $\{1, 2\} \in \mathcal{I}^0$ and $2 \leq p$. For reasons of performance, we wish to guarantee that the systems are sampled at least every $5 \cdot T_s$ s and therefore choose $\mathcal{I}_1^0 = \mathcal{I}_2^0 = \mathcal{I}^0 = \{1, 2, 3, 4, 5\}$ fulfilling the requirement above. We also note that $\lambda(A_1) = \{1\}$ and $\lambda(A_2) = \{1, 1\}$ and that hence both system fulfill Assumption 1 for this choice of $\mathcal{I}^0$, implying that (13) in Theorem 5 gives $p^* = \max \mathcal{I}^0 = 5$. Thus, choosing $p = p^*$, as stated in Theorem 5, results in that Assumption 2 holds. What now remains to be decided are the weights α_ℓ, Q_ℓ, and R_ℓ.

For the integrator process $\mathcal{P}_1$, we keep the same tuning as in Section VIII-A with $Q_1 = R_1 = 1$. Having decided Q_1, R_1, $\mathcal{I}^0$, and p^*, we use Theorem 3 to compute the needed state feedback gains and cost function matrices $(P_1^{(i)}, L_1^{(i)})$ $\forall i \in \mathcal{I}^0$ needed by Algorithm 2. We also keep $\alpha_1 = 0.2$ as it gave a good communication versus performance tradeoff.

For the double integrator process $\mathcal{P}_2$, the weights are chosen to be $Q_2 = I$, as we consider both states equally important and $R_2 = (1/10)$ to favor control performance and allow for larger control signals. Having decided Q_2, R_2, $\mathcal{I}^0$, and p^*, we may use Theorem 3 to compute the needed state feedback gains and cost function matrices $(P_2^{(i)}, L_2^{(i)})$ $\forall i \in \mathcal{I}^0$ needed by Algorithm 2. The sampling cost is chosen to be $\alpha_2 = 1$, as this gives a good tradeoff between control performance and the number of samples.

We have now fulfilled all the assumptions of both Theorem 5 and Corollary 2. Hence, applying Algorithm 2, choosing $\mathcal{I}_\ell(k_\ell)$ according to Theorem 4 will asymptotically stabilize both process $\mathcal{P}_1$ and $\mathcal{P}_2$.

Controlling $\mathcal{P}_1$ and $\mathcal{P}_2$ using our multiple-loop self-triggered MPC described in Algorithm 2 with the above designed tuning, we get the result shown in Fig. 7. As expected, the behavior of the controller illustrated in Section VIII-A carries through also to the case when we have multiple loops on the network. In fact, comparing Fig. 4(a) by showing how the controller handles process $\mathcal{P}_1$ when controlling it by itself on the network and Fig. 7(a) that shows how $\mathcal{P}_1$ is handled in the multiple-loop case, we see that they are the same. Further we see that, as expected, in stationarity, the two loops controlling process $\mathcal{P}_1$ and $\mathcal{P}_2$ both converge to the sampling rate p^*.

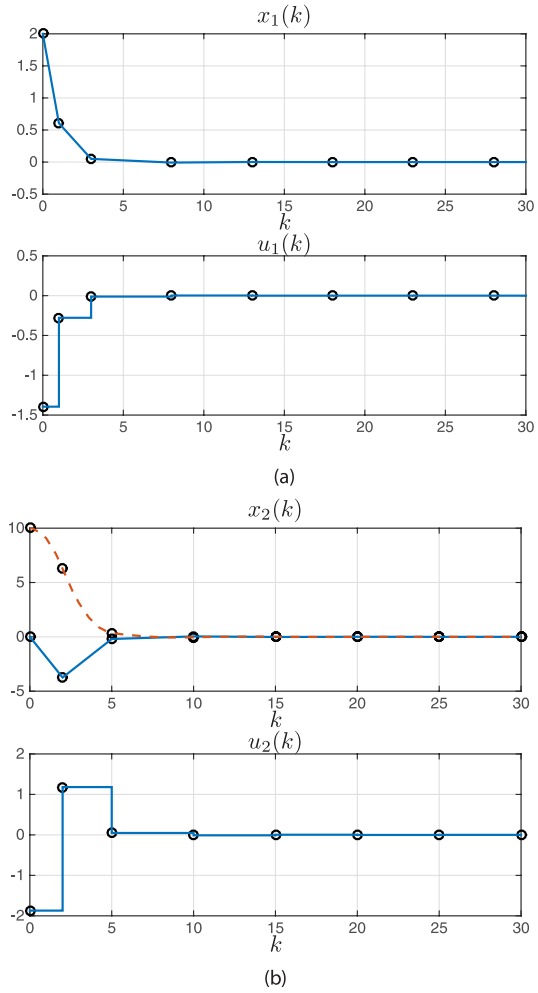

Fig. 7. Processes $\mathcal{P}_1$ and $\mathcal{P}_2$ controlled and scheduled on the same network using our multiple-loop self-triggered MPC. (a) System response for process $\mathcal{P}_1$. (b) System response for process $\mathcal{P}_2$.

As mentioned previously, the controller uses the mechanism in Theorem 4 to choose the set of feasible times to wait until the next sample. In Fig. 8, we can observe how the resulting sets $\mathcal{I}_\ell(k_\ell)$ look in detail. At time $k = 0$, loop 1 gets to run Algorithm 2 first. As sensor $\mathcal{S}_2$ is not scheduled for any transmissions, yet $\mathcal{I}_1(0) = \mathcal{I}^0$ from which the controller chooses $I_1(0) = 1$. Then loop 2 gets to run Algorithm 2 at time $k = 0$. As sensor $\mathcal{S}_1$ now is scheduled for transmission at time $k = 0 + I_1(0) = 1$, Theorem 4 gives $\mathcal{I}_2(0) = \mathcal{I}^0 \setminus \{1\}$ from which the controller chooses $I_2(0) = 2$. The process is then repeated every time, a sample is transmitted to the controller, giving the result in Fig. 8. As seen, both the set $\mathcal{I}_\ell(\cdot)$ and the optimal time to wait $I_\ell(\cdot)$ converges to some fixed value as the state of the corresponding process $\mathcal{P}_\ell$ converges to zero.

D. Comparison With Periodic Control

For a more thorough performance comparison, we simulate the systems using Algorithm 2 and compare them to the periodic algorithm that uses the cost function (15). The single and double integrator processes ($\mathcal{P}_1$ and $\mathcal{P}_2$) are simulated using the parameters mentioned in Section VIII. The variance of the disturbances for both processes are set to $\sigma_\ell^2 = 0.1$, $\forall \ell$,

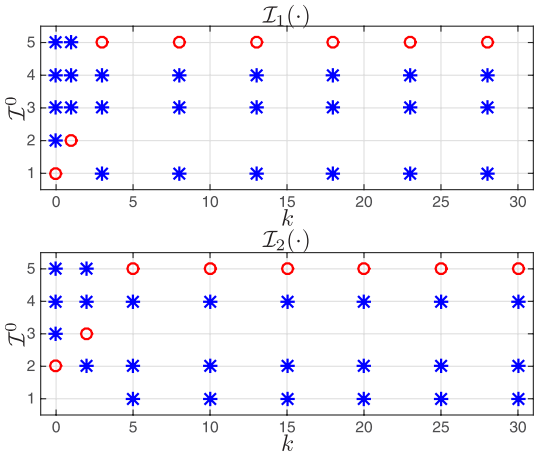

Fig. 8. Sets $\mathcal{I}_\ell(\cdot)$ of feasible times to wait until the next sample for loop 1 and loop 2. The optimal time to wait $I_\ell(\cdot)$ is marked by red circles, whereas the other feasible times are starred.

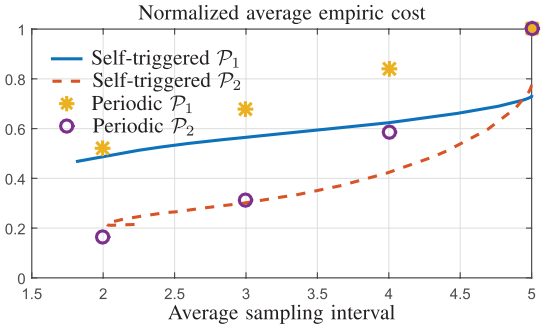

Fig. 9. Performance for both processes for the self-triggered algorithm compared with a simple periodic algorithm for different sampling intervals. The maximum sampling interval $p^* = 5$ done by varying α from 0 to $5 \cdot 10^4$ and calculating the average amount of samples for every value of α.

the variance of the initial state to $\sigma^2_{x_0,\ell} = 25, \forall \ell$, and $E_2 = [1 \;\; 1]^T$ in (14). The value of α is identical for both processes in each simulation, such that $\alpha_1 = \alpha_2$. Further, $R_1 = 0.1$. The simulation for both algorithms is initialized as described in Remark 8.

Fig 9 shows the performance for the processes, $\mathcal{P}_1$ and $\mathcal{P}_2$, calculated by averaging (16) over 100 simulations each of length 10 000 for different values of α when $p_1^* = p_2^* = 5$. Fig. 10 shows the performance when $p_1^* = p_2^* = 15$. The cost of each process is normalized with respect to its largest cost for easier viewing.

Both figures show that the self-triggered algorithm in general outperforms periodic sampling. The performance margin increases as the sampling interval increases. Fig. 9 shows that $\mathcal{P}_1$ when $\alpha = 0$ samples every 1.8 time steps on average, whereas $\mathcal{P}_2$ only samples every 2.2 time steps. When α increases, both processes sample almost every 2 time steps. The reason for this is that the cost of the state for the down sampled systems in some cases is lower.

The worse performance of the self-triggered algorithm at lower sampling intervals is more significant in Fig. 10, where the performance of process $\mathcal{P}_1$ shows a similar performance as in Fig. 6. The lowest average sampling interval for process $\mathcal{P}_2$ is 3.6 time steps when $\alpha = 0$. The self-triggered algorithm, though, significantly outperforms the periodic algorithm when the average sampling interval increases.

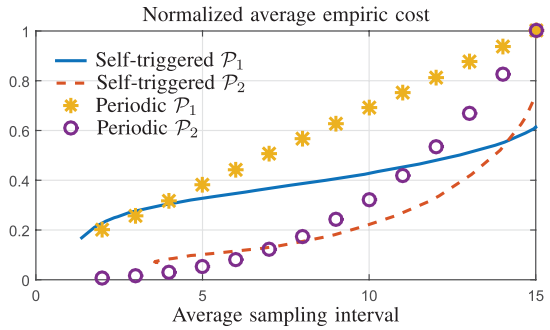

Fig. 10. Performance for both processes for the self-triggered algorithm compared with a simple periodic algorithm for different sampling intervals. The maximum sampling interval $p^* = 15$ done by varying α from 0 to 10^9 and calculating the average amount of samples for every value of α.

As $\alpha \to \infty$, the cost of sampling forces the self-triggered algorithm to behave similarly to the periodic sampled algorithm. Therefore, the performance gap between the periodic and self-triggered algorithms narrows, as the average sampling interval is close to p^*.

IX. Conclusion

We have studied the joint design of control and adaptive scheduling of multiple loops, and have presented a method that at every sampling instant computes the optimal control signal to be applied as well as the optimal time to wait before taking the next sample. It is shown that this control law may be realized using MPC and computed explicitly. The controller is also shown to be stabilizing under mild assumptions. The simulation results show that the use of the presented control law in most cases may help reducing the required amount of communication without almost any loss of performance compared with fast periodic sampling.

In the multiple-loop case, we have also presented an algorithm for guaranteeing conflict free transmissions. It is shown that under mild assumptions, there always exists a feasible schedule for the network. The complexity of the multiple-loop self-triggered MPC and the corresponding scheduling algorithm scales linearly in the number of loops.

An interesting topic for future research is to further investigate the complexity and possible performance increase for such an extended formulation.

Intuitively, additional performance gains can be achieved when the cost function considers process noise as well. Exploiting this could be of interest in future research.

Another topic for future research would be to apply the presented framework to constrained control problems.

Appendix

A. Proof of Lemma 2

Proof: Following Lemma 1, the problem is equivalent to

$$\min_{\mathcal{U}(i)} \sum_{r=0}^{\infty} \Big(x(k+i+r\cdot p)^T Q^{(p)} x(k+i+r\cdot p) + u(k+i+r\cdot p)^T R^{(p)} u(k+i+r\cdot p) + 2x(k+i+r\cdot p)^T N^{(p)} u(k+i+r\cdot p) \Big)$$

with

$$x(k+i+(r+1)\cdot p) = A^{(p)}x(k+i+r\cdot p)+B^{(p)}u(k+i+r\cdot p).$$

This problem has the known optimal solution [37], $\|x(k+i)\|^2_{P^{(p)}}$, where $P^{(p)}$ is given by the Riccati equation (6), which has a solution provided that $0 < R^{(p)}$, implied by $0 < R$, and $0 < Q^{(p)}$, implied by $0 < Q$, and that the pair $(A^{(p)}, B^{(p)})$ is controllable. It is exactly what is stated in the lemma. □

B. Proof of Theorem 1

Proof: From the Theorem, we have that $0 < Q$ and $0 < R$, and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Thus, we may use Lemma 2 to express the cost (4) as

$$J(x(k),i,\mathcal{U}(i)) = \frac{\alpha}{i}+\|x(k+i)\|^2_{P^{(p)}} + \sum_{l=0}^{i-1}\Big(\|x(k+l)\|^2_Q+\|u(k)\|^2_R\Big).$$

Now applying Lemma 1, we get

$$J(x(k),i,\mathcal{U}(i)) = \frac{\alpha}{i}+\|x(k+i)\|^2_{P^{(p)}} + x(k)^T Q^{(i)}x(k)+u(k)^T R^{(i)}u(k) + 2x(k)^T N^{(i)}u(k)$$

with $x(k+i) = A^{(i)}x(k)+B^{(i)}u(k)$. Minimizing $J(x(k),i,\mathcal{U}(i))$ now becomes a finite-horizon optimal control problem with one prediction step into the future. This problem has the well-defined solution (7) [37] given by iterating the Riccati equation (8). □

C. Proof of Theorem 2

Proof: By assumption, (A, B) is controllable. Together with the choice of p^* this, via Lemma 3, implies that $(A^{(p^*)}, B^{(p^*)})$ is controllable. Further, let $\hat{x}(k'|k)$ denote an estimate of $x(k')$, given all available measurements up until time k. Defining

$$\|\hat{x}(k|k)\|^2_{S^{(i)}} \triangleq \sum_{l=0}^{i-1}\big(\|\hat{x}(k+l|k)\|^2_Q+\|\hat{u}(k|k)\|^2_R\big) \quad (17)$$

we may, since by assumption $0 < Q$ and $0 < R$, use Lemma 2 and Theorem 1 to express V_k, the optimal value of the cost (4) at the current sampling instant k, as

$$\begin{aligned} V_k &\triangleq \min_{i\in\mathcal{I}^0,\hat{u}(k|k)} J(x(k),i,\hat{u}(k|k)) \\ &= \min_{i\in\mathcal{I}^0}\frac{\alpha}{i}+\|\hat{x}(k+i|k)\|^2_{P^{(p^*)}}+\|\hat{x}(k|k)\|^2_{S^{(i)}} \\ &= \min_{i\in\mathcal{I}^0}\frac{\alpha}{i}+\|\hat{x}(k|k)\|^2_{P^{(i)}}. \end{aligned} \quad (18)$$

We will use V_k as a Lyapunov-like function. Assume that V_{k+i} is the optimal cost at the next sampling instant $k+i$. Again using Theorem 1, we may express it as

$$\begin{aligned} V_{k+i} &\triangleq \min_{j\in\mathcal{I}^0,\hat{u}(k+i|k+i)} J(x(k+i),j,\hat{u}(k+i|k+i)) \\ &\le \min_{\hat{u}(k+i|k+i)} J(x(k+i),j=p^*,\hat{u}(k+i|k+i)) \\ &= \frac{\alpha}{p^*}+\|\hat{x}(k+i|k+i)\|^2_{P^{(p^*)}} \\ &= \frac{\alpha}{p^*}+\|\hat{x}(k+i|k)\|^2_{P^{(p^*)}} \end{aligned} \quad (19)$$

where the inequality comes from the fact that choosing $j = p^*$ is suboptimal. Taking the difference, we get

$$V_{k+i}-V_k \le \frac{\alpha}{p^*}-\frac{\alpha}{i}-\|\hat{x}(k|k)\|^2_{S^{(i)}}$$

which in general is not decreasing. However, we may use the following idea to bound this difference: assume that there $\exists\,\epsilon\in(0,1]$, and $\beta\in\mathbb{R}^+$ such that we may write

$$V_{k+i}-V_k \le -\epsilon V_k+\beta$$

for all $k, k+i \in \mathcal{D}$ (10), (18). Thus, at l sampling instances into the future, which happens at, let us say, time $k+l'$, we have that

$$V_{k+l'} \le (1-\epsilon)^l\cdot V_k+\beta\cdot\sum_{r=0}^{l-i}(1-\epsilon)^l.$$

Since $\epsilon\in(0,1]$, this is equivalent to

$$V_{k+l'} \le (1-\epsilon)^l\cdot V_k+\beta\cdot\frac{1-(1-\epsilon)^l}{1-(1-\epsilon)}$$

which as $l\to\infty$ gives us an upper bound on the cost function, $V_{k+l'}\le\beta/\epsilon$. Applying this idea on our setup, we should fulfill

$$\frac{\alpha}{p^*}-\frac{\alpha}{i}-\|\hat{x}(k|k)\|_{S^{(i)}} \le -\epsilon\frac{\alpha}{i}-\epsilon\|\hat{x}(k+i|k)\|^2_{P^{(p^*)}}-\epsilon\|\hat{x}(k|k)\|^2_{S^{(i)}}+\beta.$$

Choosing $\beta\triangleq\alpha/p^*-(1-\epsilon)\alpha/\gamma$, we have fulfillment if

$$\epsilon\Big(\|\hat{x}(k+i|k)\|^2_{P^{(p^*)}}+\|\hat{x}(k|k)\|^2_{S^{(i)}}\Big)\le\|\hat{x}(k|k)\|^2_{S^{(i)}}. \quad (20)$$

Clearly there $\exists\epsilon\in(0,1]$ such that the above relation is fulfilled if $0<\|\hat{x}(k|k)\|^2_{S^{(i)}}$. For the case $\|\hat{x}(k|k)\|^2_{S^{(i)}}=0$, we must, following the definition (17) and the assumption $0<Q$, have that $\hat{x}(k|k)=0$ and $\hat{x}(k+i|k)=0$, and hence the relation is fulfilled also in this case. Using the final step in (18), we may express (20) in easily computable quantities giving the condition

$$\|\hat{x}(k+i|k)\|^2_{P^{(p^*)}}\le(1-\epsilon)\|\hat{x}(k|k)\|^2_{P^{(i)}}.$$

As this should hold $\forall x$ and $\forall i\in\mathcal{I}^0$, we must fulfill

$$\big(A^{(i)}-B^{(i)}L^{(i)}\big)^T P^{(p^*)}\big(A^{(i)}-B^{(i)}L^{(i)}\big)\le(1-\epsilon)P^{(i)}$$

which is stated in the theorem. Summing up, we have

$$V_{k+l'}\le\frac{\alpha}{\epsilon}\left(\frac{1}{p^*}-(1-\epsilon)\frac{1}{\gamma}\right)$$

which is minimized by maximizing ϵ. From the definition of the cost (3), we may also conclude that $\alpha/\gamma\le V_{k+l'}$. With

$$V_{k+l'}=\min_{i\in\mathcal{I}^0}\left(\frac{\alpha}{i}+\|\hat{x}(k+l'|k+l')\|^2_{P^{(i)}}\right)$$

we may conclude that

$$\frac{\alpha}{\gamma} \leq \lim_{k\to\infty} \min_{i\in\mathcal{I}^0} \left(\frac{\alpha}{i} + \|\hat{x}(k|k)\|^2_{P^{(i)}}\right) \leq \frac{\alpha}{\epsilon}\left(\frac{1}{p^*} - (1-\epsilon)\frac{1}{\gamma}\right).$$

□

D. Proof of Lemma 4

Proof: From (11), it is clear that $p^* = \gamma$ if $\nexists \lambda \in \lambda(A)$ except $\lambda = 1$ such that $\lambda^\gamma = 1$. In polar coordinates, we have that $\lambda = |\lambda|\exp(j \cdot \angle\lambda)$ implying $\lambda^\gamma = |\lambda|^\gamma \exp(j \cdot \gamma \cdot \angle\lambda) = 1$ may only be fulfilled if $|\lambda| = 1$ and $\angle\lambda = (2\pi/\gamma) \cdot n$ for some $n \in \mathbb{N}^+$, which contradicts Assumption 1. □

E. Proof of Corollary 1

Proof: From Theorem 2, we have that as $k \to \infty$

$$\frac{\alpha}{\gamma} \leq \min_{i\in\mathcal{I}^0} \left(\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}}\right) \leq \frac{\alpha}{\epsilon}\left(\frac{1}{p^*} - (1-\epsilon)\frac{1}{\gamma}\right)$$

which as $p^* = \gamma$, given by Lemma 4, simplifies to

$$\frac{\alpha}{\gamma} \leq \min_{i\in\mathcal{I}^0} \left(\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}}\right) \leq \frac{\alpha}{\gamma}$$

independent of ϵ. Implying that as $k \to \infty$, we have that $i = \gamma$ and $\|x(k)\|^2_{P^{(i)}} = 0$, since $0 < P^{(i)}$, provided $0 < Q$, this implies that $x(k) = 0$ independent of α. In the case $\alpha = 0$, the bound from Theorem 2 simplifies to that as $k \to \infty$

$$0 \leq \min_{i\in\mathcal{I}^0} \|x(k)\|^2_{P^{(i)}} \leq 0$$

and hence for the optimal i, we have $\|x(k)\|^2_{P^{(i)}} = 0$ implying $x(k) = 0$ as above. □

F. Proof of Lemma 5

Proof: When loop q was last sampled at time $k_q < k_\ell$, it was optimized over $\mathcal{I}_q(k_q)$ and found the optimal feasible time until the next sample $I_q(k_q)$. The loop then reserved the infinite sequence

$$S_q(k_q) = \{k_q + I_q(k_q),\ k_q + I_q(k) + p_q,\ k_q + I_q(k_q) + 2p_q,\ k_q + I_q(k_q) + 3p_q,\ \ldots\}.$$

When loop ℓ now should choose $I_\ell(k_\ell)$, it must be able to reserve (12). To ensure that this is true, it must choose $I_\ell(k_\ell)$ so that $S_\ell(k_\ell) \cap S_q(k_q) = \emptyset$, $\forall q \in \mathbb{L} \setminus \{\ell\}$. This holds if we have that

$$k_\ell + I_\ell(k_\ell) + m \cdot p_\ell \neq k_q + I_q(k_q) + n \cdot p_q \quad \forall m, n \in \mathbb{N} \quad \forall q \in \mathbb{L} \setminus \{\ell\}.$$

Simplifying the above condition, we get conditions on the feasible values of $I_\ell(k_\ell)$

$$I_\ell(k_\ell) \neq (k_q + I_q(k_q)) - k_\ell + n \cdot p_q - m \cdot p_\ell \quad \forall m, n \in \mathbb{N}, \quad \forall q \in \mathbb{L} \setminus \{\ell\}.$$

Noting that $(k_q + I_q(k_q))$ is the next transmission of loop q, we denote it by k_q^{next}, giving the statement in the lemma. □

G. Proof of Theorem 4

Proof: Assuming $p_\ell = p$ and $\mathcal{I}^0_\ell = \mathcal{I}^0$ for all loops, Lemma 5 gives $\mathcal{I}_\ell(k_\ell)$, as stated in the theorem. Since by assumption $\mathbb{L} \subseteq \mathcal{I}^0$, we know that

$$\mathcal{I}_\ell(k_\ell) \supseteq \{i \in \mathbb{L} | i \neq k_q^{\text{next}} - k_\ell + r \cdot p, \quad r \in \mathbb{Z}, \quad q \in \mathbb{L}\setminus\{\ell\}\}.$$

Further, since $\max \mathbb{L} \leq p$, we know that the set

$$\left\{i \in \mathbb{L} | i = k_q^{\text{next}} - k_\ell + r \cdot p,\ r \in \mathbb{Z}\right\}$$

contains at most one element for a given loop q. Thus

$$\{i \in \mathbb{L} | i \neq k_q^{\text{next}} - k_\ell + r \cdot p,\ r \in \mathbb{Z},\ q \in \mathbb{L}\setminus\{\ell\}\}$$

contains at least one element as $q \in \mathbb{L} \setminus \{\ell\} \subset \mathbb{L}$. Hence, $\mathcal{I}_\ell(k_\ell)$ always contains at least one element $\forall k_\ell$, and thus there always exists a feasible time to wait. □

H. Proof of Theorem 5

Proof: Direct application of Theorem 2 on each loop is done. The corresponding proof carries through as the choice of $\mathcal{I}_\ell(k)$ together with the remaining assumptions guarantees that we may apply Theorem 3 on the feasible values of i in every step, thus the expression for V_k in (18) always exists. The critical step is that the upper bound on V_{k+i} in (19) must exist, i.e., the choice $j = p^*$ must be feasible. This is also guaranteed by the choice of $\mathcal{I}_\ell(k)$ (Remark 10). □

REFERENCES

[1] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Trans. Ind. Informat.*, vol. 4, no. 2, pp. 102–124, May 2008.

[2] HART Communication Foundation. (2013). *WirelessHART*. [Online]. Available: http://www.hartcomm.org

[3] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard 802.15.4, 2011. [Online]. Available: http://standards.ieee.org/about/get/802/802.15.html

[4] M. Velasco, P. Martý, and J. M. Fuertes, "The self triggered task model for real-time control systems," in *Proc. 24th IEEE Work-Prog. Session Real-Time Syst. Symp. (RTSS)*, Cancun, Mexico, 2003. [Online]. Available: http://esaii.upc.edu/people/pmarti/RTSSWIPM2003.pdf

[5] E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson, "Self-triggered model predictive control for network scheduling and control," in *Proc. 8th IFAC Int. Symp. Adv. Control Chem. Process.*, Singapore, 2012, pp. 432–438.

[6] P. Antsaklis and J. Baillieul, "Special issue on technology of networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 5–8, Jan. 2007.

[7] K. J. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," in *Proc. 14th World Congr. IFAC*, Beijing, China, 1999, pp. 301–306.

[8] K.-E. Årzén, "A simple event-based PID controller," in *Proc. 14th World Congr. IFAC*, Beijing, China, 1999, pp. 423–428.

[9] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[10] W. P. M. H. Heemels, J. H. Sandee, and P. P. J. Van Den Bosch, "Analysis of event-driven controllers for linear systems," *Int. J. Control*, vol. 81, no. 4, pp. 571–590, 2008.

[11] A. Bemporad, S. Di Cairano, and J. Júlvez, "Event-based model predictive control and verification of integral continuous-time hybrid automata," in *Proc. 9th Int. Workshop Hybrid Syst., Comput. Control (HSCC)*, Santa Barbara, CA, USA, Mar. 2006, pp. 93–107. [Online]. Available: http://dx.doi.org/10.1007/11730637_10

[12] P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen, "Event-based model predictive control for networked control systems," in *Proc. 48th IEEE Conf. Decision Control, 28th Chin. Control Conf. (CDC/CCC)*, Dec. 2009, pp. 567–572.

[13] D. Gross and O. Stursberg, "Distributed predictive control for a class of hybrid systems with event-based communication," in *Proc. 4th IFAC Workshop Distrib. Estimation Control Netw. Syst.*, 2013, pp. 383–388.
[14] A. Molin and S. Hirche, "A bi-level approach for the design of event-triggered control systems over a shared network," *Discrete Event Dyn. Syst.*, vol. 24, no. 2, pp. 153–171, 2014.
[15] R. Blind and F. Allgöwer, "On time-triggered and event-based control of integrator systems over a shared communication system," *Math. Control, Signals, Syst.*, vol. 25, no. 4, pp. 517–557, 2013.
[16] X. Wang and M. D. Lemmon, "Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 452–467, Mar. 2009.
[17] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2030–2042, Sep. 2010.
[18] M. Mazo, Jr., A. Anta, and P. Tabuada, "An ISS self-triggered implementation of linear controllers," *Automatica*, vol. 46, no. 8, pp. 1310–1314, 2010.
[19] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," in *Proc. 48th IEEE Int. Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 5832–5838.
[20] D. J. Antunes, W. P. M. H. Heemels, J. P. Hespanha, and C. J. Silvestre, "Scheduling measurements and controls over networks—Part II: Rollout strategies for simultaneous protocol and controller design," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, Jun. 2012, pp. 2042–2047.
[21] D. Antunes, W. P. M. H. Heemels, and P. Tabuada, "Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design," in *Proc. IEEE 51st Annu. Conf. Decision Control*, Maui, HI, USA, Dec. 2012, pp. 7212–7217.
[22] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 847–861, Apr. 2013.
[23] W. P. M. H. Heemels and M. C. F. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.
[24] A. Casavola, E. Mosca, and M. Papini, "Predictive teleoperation of constrained dynamic systems via internet-like channels," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 681–694, Jul. 2006.
[25] P. L. Tang and C. W. de Silva, "Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 707–718, Jul. 2006.
[26] G. P. Liu, J. X. Mu, D. Rees, and S. C. Chai, "Design and stability analysis of networked control systems with random communication time delay using the modified MPC," *Int. J. Control*, vol. 79, no. 4, pp. 288–297, 2006.
[27] D. E. Quevedo and D. Nešić, "Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses," *Automatica*, vol. 48, no. 8, pp. 1803–1811, 2012.
[28] D. E. Quevedo, G. C. Goodwin, and J. S. Welsh, "Minimizing downlink traffic in networked control systems via optimal control techniques," in *Proc. 42nd IEEE Conf. Decision Control*, Dec. 2003, pp. 1200–1205.
[29] M. Lješnjanin, D. E. Quevedo, and D. Nešić, "Packetized MPC with dynamic scheduling constraints and bounded packet dropouts," *Automatica*, vol. 50, no. 3, pp. 784–797, 2014.
[30] D. Bernardini and A. Bemporad, "Energy-aware robust model predictive control based on noisy wireless sensors," *Automatica*, vol. 48, no. 1, pp. 36–44, 2012.
[31] D. Lehmann, E. Henriksson, and K. H. Johansson, "Event-triggered model predictive control of discrete-time linear systems subject to disturbances," in *Proc. Eur. Control Conf.*, Zürich, Switzerland, Jul. 2013, pp. 1156–1161.
[32] J. D. J. Barradas Berglind, T. M. P. Gommans, and W. P. M. H. Heemels, "Self-triggered MPC for constrained linear systems and quadratic costs," in *Proc. 4th IFAC Conf. Nonlinear Model Predict. Control*, Noordwijkerhout, The Netherlands, 2012, pp. 342–348.
[33] A. Eqtami, S. Heshmati-Alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Self-triggered model predictive control for nonholonomic systems," in *Proc. Eur. Control Conf.*, Zürich, Switzerland, Jul. 2013, pp. 638–643.
[34] P. Colaneri, R. Scattolini, and N. Schiavoni, "Regulation of multirate sampled-data systems," *Control-Theory Adv. Technol.*, vol. 7, no. 3, pp. 429–441, 1991.
[35] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, Jun. 2001.
[36] J. M. Maciejowski, *Predictive Control With Constraints*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[37] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA, USA: Athena Scientific, 1995.

Erik Henriksson received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from the Royal Institute of Technology, Stockholm, Sweden, in 2007 and 2014, respectively.

He was a Visiting Researcher with the Universita degli Studi di Siena, Siena, Italy, in 2007, and The University of Newcastle, Callaghan, NSW, Australia, in 2010. He is currently with Öhlins Racing AB, Upplands Väsby, Sweden, where he is involved in semiactive suspension for hypersport motorcycles.

Daniel E. Quevedo (S'97–M'05–SM'14) received the Ingeniería Civil Electrónica and Magister en Ingeniería Electrónica degrees from the Universidad Técnica Federico Santa María, Valparaíso, Chile, in 2000, and the Ph.D. degree from The University of Newcastle, Callaghan, NSW, Australia, in 2005.

He is currently a Professor of Automatic Control at the University of Paderborn, Germany. He has been a Visiting Researcher at various institutions, including Uppsala University, Uppsala, Sweden, the Royal Institute of Technology, Stockholm, Sweden, Kyoto University, Kyoto, Japan, the Karlsruher Institut für Technologie, Karlsruhe, Germany, the University of Notre Dame, Notre Dame, IN, USA, INRIA, Grenoble, France, the Hong Kong University of Science and Technology, Hong Kong, Aalborg University, Aalborg, Denmark, and Nanyang Technological University, Singapore. His current research interests include automatic control, signal processing, and power electronics.

Prof. Quevedo was a recipient of the IEEE Conference on Decision and Control Best Student Paper Award in 2003, and was a finalist in 2002. He received a five-year Australian Research Fellowship in 2009. He was supported by the Full Scholarship from the Alumni Association during his time with the Universidad Técnica Federico Santa María, and received several university-wide prizes upon graduating. He serves as an Editor of the *International Journal of Robust and Nonlinear Control*.

Edwin G. W. Peters received the B.Sc. degree in electrical engineering and the M.Sc. degree in signal processing from Aalborg University, Aalborg, Denmark, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree in control systems with The University of Newcastle, Callaghan, NSW, Australia.

His current research interests include networked control, stochastic systems, and optimization.

Henrik Sandberg received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from Lund University, Lund, Sweden, in 1999 and 2004, respectively.

He was a Post-Doctoral Scholar with the California Institute of Technology, Pasadena, CA, USA, from 2005 to 2007. In 2013, he was a Visiting Scholar with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. He has held visiting appointments with Australian National University, Canberra, ACT, Australia, and the University of Melbourne, Melbourne, NSW, Australia. He is currently an Associate Professor with the Department of Automatic Control, Royal Institute of Technology, Stockholm, Sweden. His current research interests include secure networked control, power systems, model reduction, and fundamental limitations in control.

Dr. Sandberg was a recipient of the Best Student Paper Award from the IEEE Conference on Decision and Control in 2004 and the Ingvar Carlsson Award from the Swedish Foundation for Strategic Research in 2007. He is currently an Associate Editor of *IFAC Automatica* journal.

Karl Henrik Johansson (F'13) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University, Lund, Sweden.

He held visiting positions with the University of California at Berkeley, Berkeley, CA, USA, from 1998 to 2000, and the California Institute of Technology, Pasadena, CA, USA, from 2006 to 2007. He is currently the Director of the KTH ACCESS Linnaeus Centre, and a Professor with the School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. He is a Wallenberg Scholar, and has been a six-year Senior Researcher with the Swedish Research Council. He is the Director of the Stockholm Strategic Research Area ICT The Next Generation. His current research interests include networked control systems, hybrid and embedded system, and applications in smart transportation, energy, and automation systems.

Dr. Johansson was a recipient of the best paper award at the IEEE International Conference on Mobile Ad-hoc and Sensor Systems in 2009, the Individual Grant for the Advancement of Research Leaders from the Swedish Foundation for Strategic Research in 2005, the Triennial Young Author Prize from IFAC in 1996, the Peccei Award from the International Institute of System Analysis, Austria, in 1993, and the Young Researcher Awards from Scania in 1996 and from Ericsson in 1998 and 1999. He has been a member of the IEEE Control Systems Society Board of Governors and the Chair of the IFAC Technical Committee on Networked Systems. He has been on the Editorial Boards of several journals, including *Automatica*, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and *IET Control Theory and Applications*. He is currently on the Editorial Board of IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS and the *European Journal of Control*. He has been a Guest Editor for special issues, including two issues of IEEE TRANSACTIONS ON AUTOMATIC CONTROL: one on wireless sensor and actuator networks and one on control of cyber-physical systems. He was the General Chair of the ACM/IEEE Cyber-Physical Systems Week in Stockholm in 2010 and the IPC Chair of many conferences. He has served on the Executive Committees of several European research projects in the area of networked embedded systems. He received the Wallenberg Scholar, as one of the first 10 scholars from all sciences, by the Knut and Alice Wallenberg Foundation.