4th IFAC Nonlinear Model Predictive Control Conference
International Federation of Automatic Control
Noordwijkerhout, NL. August 23-27, 2012

Self-triggered MPC for constrained linear systems and quadratic costs ⋆

J.D.J. Barradas Berglind * **T.M.P. Gommans** ** **W.P.M.H. Heemels** ***

Hybrid and Networked Systems group, Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands.

* *e-mail: j.d.j.barradas.berglind@student.tue.nl*
** *e-mail: t.m.p.gommans@tue.nl*
*** *e-mail: m.heemels@tue.nl*

Abstract: Self-triggered control is a recently proposed control paradigm that abandons the conventional periodic sampling of outputs and updating of control values with the objective to reduce the utilization of communication resources, while still guaranteeing desirable closed-loop behaviour. Self-triggered control consists of two elements, namely a feedback controller that computes the next control input, and a triggering mechanism that determines what the next control update time will be. In this paper we present a self-triggered MPC strategy based on quadratic costs, which applies to discrete-time linear systems subject to state and input constraints. This self-triggered MPC law possesses three important features. First of all, the control law is designed such that a priori chosen (sub)optimality levels are guaranteed next to asymptotic stability and constraint satisfaction. Secondly, it is one of the first approaches, which addresses the joint design of both the feedback controller and the triggering mechanism. Thirdly, it realizes significant reductions in the usage of network resources and avoids bursts in communication. These beneficial features and various implementation aspects of the proposed self-triggered MPC strategy will be discussed in detail.

1. INTRODUCTION

A current trend in control engineering is to no longer implement controllers on dedicated platforms with dedicated communication channels, but using (shared) communication networks. Since in such an environment the control task has to share the communication resources with other tasks, the availability of these resources is limited and might even change over time. In addition, also the energy-constrained nature of wireless sensor networks (WSNs) poses new challenges in control design; in particular the discharge of batteries of sensor nodes, which is mainly due to radio communication, must be taken into account. Despite the fact that network resources are scarce, controllers are typically still implemented in a time-triggered fashion, in which control tasks are executed periodically. This design choice is motivated by the fact that it enables the use of a well-developed theory on sampled-data systems to design controllers and analyse the resulting closed-loop systems. Unfortunately, this choice can lead to over-utilization of the available network resources and/or a limited lifetime of battery-powered devices. As it might not be necessary from a stability and performance point of view to execute the control task every period, several alternative control strategies have recently been proposed instead of periodic time-triggered implementations.

Two of such approaches are event-triggered control (ETC), see, e.g., Åström and Bernhardsson [1999], Arzén [1999], Heemels et al. [1999], Yook et al. [2002], Tabuada [2007], Heemels et al. [2008], Henningsson et al. [2008], Lunze and Lehmann [2010], and self-triggered control (STC), see, e.g., Velasco et al. [2003], Wang and Lemmon [2009], Mazo Jr. et al. [2010]. In event-triggered control and self-triggered control, the control law consists of two elements: namely, a feedback controller that computes the control input, and a triggering mechanism that determines when the control input has to be updated. The difference between event-triggered control and self-triggered control is that in the former the triggering mechanism uses current measurements and the triggering condition is verified continuously, while in the latter at a control update time the next update time is pre-computed based on predictions using previously sampled and transmitted data, and knowledge on the plant dynamics. Event-triggered control laws have been mostly developed for continuous-time systems, although some work has been done for discrete-time systems, see, e.g., Yook et al. [2002], Cogill [2009], Eqtami et al. [2010], Molin and Hirche [2010], Li and Lemmon [2011] and [Lehmann, 2011, Sec. 4.5]. In addition, in Arzén [1999], Henningsson et al. [2008], Heemels et al. [2008, 2011] so-

⋆ This work is supported by the Innovational Research Incentives Scheme under the VICI grant "Wireless control systems: A new frontier in automation" (no. 11382) awarded by NWO (Netherlands Organization for Scientific Research) and STW (Dutch Science Foundation), and by the European 7^{th} Framework Network of Excellence "Highly-complex and networked control systems (HYCON2)", under program number 257462. The first author is also partially supported by the Graduate Support Program, Complementary Grants of the SEP (Mexican Secretary of Public Education).

called periodic event-triggered control (PETC) strategies were proposed and analyzed for continuous-time plants.

In this paper we are interested in control strategies for constrained discrete-time linear systems in which measures are needed to reduce the usage of communication resources. In all of the above mentioned references on ETC and STC, the handling of constraints on inputs and states was not considered. Outside the scope of ETC and STC, one of the most widely used techniques for the control of constrained systems is model predictive control (MPC) and in fact, a few works are available that combine MPC strategies with ideas from ETC and STC. In Eqtami et al. [2010] the results of Tabuada [2007] are extended towards discrete-time systems. In addition, self-triggered implementations are provided and an event-triggered MPC setup for unconstrained systems was presented using tools from input-to-state stability (ISS). Extensions of this work were presented in Eqtami et al. [2011] in which event-triggered control schemes for constrained discrete-time systems were proposed. In particular, the optimal control sequence coming from the MPC optimization problem is applied in an open-loop fashion between the event times. ISS properties of the adopted (centralized and decentralized) nonlinear model predictive control (NMPC) schemes are exploited to choose the event-triggering mechanisms leading to ultimate boundedness guarantees. An alternative scheme also based on using the sequence of optimal controls obtained from the MPC optimization problem is presented in Bernardini and Bemporad [2012] focussing on linear plants. The event-triggering mechanism, positioned at the sensors, is chosen such that it detects if the true states deviate too much from the predicted states in the MPC problem, using absolute thresholds. Based on ideas from set-membership estimation and robust min-max MPC, an event-triggered control scheme is devised providing ultimate boundedness guarantees of the closed-loop system in the presence of persistent and bounded measurement and process noise. Although the idea of sending future predictions of the control values is appealing, one has to realize that this results in bursts of communications at certain points of time, which one may like to avoid. Event-triggered MPC approaches for discrete-time systems based on events generated by quantization of the state variable are provided in Grune and Muller [2009]. Besides these works focussing on discrete-time systems, also in the context of continuous-time NMPC event-triggered implementations have been proposed in Varutti et al. [2009] and Varutti and Findeisen [2011] even considering the presence of packet loss and transmission delays. Finally, in Sijs et al. [2010] a continuous-time event-based estimator providing periodic estimates of the states has been integrated with a robust MPC law.

In this paper, we propose a self-triggered MPC strategy. We focus on the control of discrete-time linear systems subject to state and input constraints using a quadratic cost function, although we envision the applicability of the main ideas in a much wider context. The MPC law we propose has three important features: (i) a priori closed-loop performance guarantees are provided next to asymptotic stability and constraint satisfaction, (ii) joint design of both the feedback law and the triggering condition is achieved, (iii) significant reduction in the usage of network resources is achieved and bursts in communications, as in the approaches sending sequences of predicted future values of inputs and/or states, are avoided. To elaborate on these features, we emphasize that the proposed self-triggered MPC strategy will not realize only stabilization towards a desired equilibrium, but it also results in guarantees on an infinite quadratic costs related to the MPC costs. In particular, one can choose a priori a desired sub-optimality level (compared to the optimal MPC costs) and the controller design will automatically take this into account, while at the same time aiming for a reduction of the network usage. Regarding (ii), we note that most of the existing design methods for event-triggered control and self-triggered control are emulation-based in the sense that the feedback controller is designed assuming a standard time-triggered implementation and thus not taking the event-triggered implementation into account in the first design stage. In the subsequent second stage the triggering mechanism is designed (in which the controller is already given). Since the feedback controller is designed before the triggering mechanism, it is difficult, if not impossible, to obtain an optimal design of the combined feedback controller and the triggering mechanism in the sense that the minimum number of controller executions is achieved while guaranteeing stability and a certain level of closed-loop performance. In this paper, we provide a synthesis technique that determines the next update time and the applied control value *simultaneously*, thereby choosing control values that are optimized for not being updated for a maximal number of steps. This is one of the few works in the literature approaching the co-design problem. Finally, we avoid bursts in communications as it is the objective of the control strategy to send just the current control value to the actuators and no sequence of predicted future values. In fact, this constitutes a fundamental difference with most of the mentioned event-triggered MPC approaches that still update the control values in a periodic fashion. As such, we aim at providing (partial) answers addressing a more *fundamental* question: what are the (minimally) required update times of the control signal in order to provide a certain performance guarantee, instead of just adopting a conventional periodic update pattern?

The remainder of the paper is organized as follows. After introducing the notation at the end of this section, in Section 2 the problem formulation is presented where a stabilizing MPC setup is described, which forms the basis for the development of the self-triggered MPC strategy. The proposed self-triggered approach is discussed in Section 3, while Section 4 contains the implementation considerations. Lastly, a discussion about the presented approach together with some concluding remarks are presented in Section 5.

1.1 Notation

The following notational conventions will be used. Let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real numbers and the set of non-negative integers, respectively. The notation $\mathbb{N}_{\geq s}$ and $\mathbb{N}_{[s,t)}$ are used to denote the sets $\{r \in \mathbb{N} \mid r \geq s\}$ and $\{r \in \mathbb{N} \mid s \leq r < t\}$, respectively, for some $s, t \in \mathbb{N}$. A polyhedral set is a convex set obtained as the intersection of a finite number of open and/or closed half-spaces. When inequalities $<, \leq, >$ and $\geq$ are applied to vectors, they should be interpreted element-wise. The inequalities $\prec$, $\preceq$, $\succ$ and $\succeq$ are used for matrices, e.g., for a square matrix $X \in \mathbb{R}^{n\times n}$ we write $X \prec 0, X \preceq 0, X \succ 0$ and $X \succeq 0$ when X is symmetric and, in addition, X is negative definite, negative semi-definite, positive definite and positive semi-definite, respectively. Sequences of vectors are indicated by bold letters, e.g. $\boldsymbol{u} \triangleq (u_0, u_1, \ldots, u_N)$ with $u_i \in \mathbb{R}^{n_u}, i = 0, 1, \ldots, N$, where N will be clear from the context.

2. PROBLEM FORMULATION

In this paper we consider a discrete-time linear time-invariant (LTI) system described by

$$x_{t+1} = Ax_t + Bu_t, \tag{1}$$

where $x_t \in \mathbb{R}^{n_x}$ is the state and $u_t \in \mathbb{R}^{n_u}$ is the input, at time $t \in \mathbb{N}$. We assume that (A, B) is a stabilizable pair and that the system in (1) is subject to input and state constraints given by

$$u_t \in \mathbb{U} \text{ and } x_t \in \mathbb{X},\ t \in \mathbb{N}, \tag{2}$$

where $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ and $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ are closed polyhedral sets containing the origin in their interiors.

2.1 Stabilizing MPC Setup

For the discrete-time LTI system described by (1), we now consider first the MPC setup given by the following optimization problem:

Problem 2.1. Given state $x_t = x \in \mathbb{X}$ at time $t \in \mathbb{N}$, minimize

$$\begin{aligned} & x_N^\top P x_N + \sum_{k=0}^{N-1} x_k^\top Q x_k + u_k^\top R u_k \\ \text{over} \quad & \boldsymbol{u} = (u_0, u_1, \ldots, u_{N-1}) \in \mathbb{U}^N \\ \text{subject to} \quad & \begin{cases} x_0 = x, \\ x_k \in \mathbb{X},\ k = 0, 1, \ldots, N-1, \\ x_N \in \mathbb{X}_T, \\ x_{k+1} = Ax_k + Bu_k,\ k = 0, 1, \ldots, N-1. \end{cases} \end{aligned} \tag{3}$$

Here, N denotes the prediction horizon and $\mathbb{X}_T$ is the terminal set which is assumed to be polyhedral and to contain the origin in its interior. In addition, $Q = Q^\top \succ 0$ and $R = R^\top \succ 0$ are the weighting matrices of the running cost, and $P = P^\top \succ 0$ determines the terminal cost.

A state $x \in \mathbb{X}$ is called feasible for this optimization problem if $\mathcal{U}_N(x) \neq \emptyset$, where

$$\mathcal{U}_N(x) \triangleq \Big\{ \boldsymbol{u} \in \mathbb{U}^{N-1} \Big| x_k(x, \boldsymbol{u}) \in \mathbb{X}, k = 1, \ldots, N-1, \text{ and } x_N(x, \boldsymbol{u}) \in \mathbb{X}_T \Big\}, \tag{4}$$

where $x_k(x, \boldsymbol{u})$ denotes the solution to (1) at time $k \in \mathbb{N}_{[0,N]}$ initialized at $x_0 = x$ with control inputs in $\boldsymbol{u} = (u_0, u_1, \ldots, u_{N-1})$. The set of feasible states is denoted by $\mathbb{X}_f$ and for $x \in \mathbb{X}_f$, $V(x)$ denotes the corresponding minimum value for Problem 2.1. Hence, $V : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ is the MPC value function given by

$$V(x) = \min_{\boldsymbol{u} \in \mathcal{U}_N(x)} x_N^\top P x_N + \sum_{k=0}^{N-1} x_k^\top Q x_k + u_k^\top R u_k, \tag{5}$$

where $x_k = x_k(x, \boldsymbol{u}),\ k \in \mathbb{N}_{[0,N]}$.

Under the given conditions, there is a unique optimal sequence of control updates $\boldsymbol{u}^* = (u_0^*, \ldots, u_{N-1}^*)$ for Problem 2.1 for any state $x \in \mathbb{X}_f$. The resulting MPC law $u^{mpc} : \mathbb{X}_f \to \mathbb{U}$ is now defined as

$$u^{mpc}(x) \triangleq u_0^*, \tag{6}$$

which is implemented in a receding horizon fashion as

$$u_t = u^{mpc}(x_t),\ t \in \mathbb{N}. \tag{7}$$

To guarantee recursive feasibility and closed-loop stability we will use the terminal set and cost method, which typically assumes for the linear case considered here, that there is a $K \in \mathbb{R}^{n_u \times n_x}$ such that:

(I) $(A + BK)^\top P (A + BK) - P \preceq -K^\top R K - Q$,

(II) $(A + BK)\mathbb{X}_T \subseteq \mathbb{X}_T$,

(III) $K\mathbb{X}_T \subseteq \mathbb{U}$ and $\mathbb{X}_T \subseteq \mathbb{X}$.

Under these assumptions, recursive feasibility and closed-loop asymptotic stability of (1) and (7) in $\mathbb{X}_f$ are guaranteed while satisfying the input/state constraints (2) along closed-loop trajectories, see Mayne et al. [2000] for details.

As a special case, taking P as the solution of the discrete algebraic Ricatti equation (DARE)

$$P = A^\top PA - (A^\top PB)(R + B^\top PB)^{-1}(B^\top PA) + Q, \tag{8}$$

together with

$$K = -(R + B^\top PB)^{-1} B^\top PA, \tag{9}$$

will satisfy (I) with equality.

2.2 Problem Formulation

Using the above MPC setup as a basis, we are interested in deriving control strategies that lead to performance guarantees in terms of the standard infinite horizon quadratic cost given by

$$\sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t, \tag{10}$$

where the matrices Q and R are the same as above. It is not hard to show and, in fact, it follows as a special case of Theorem 2 and Theorem 3 below, that the optimal MPC law as in (7) results in a closed-loop system (1) and (7) which

(i) is asymptotically stable in $\mathbb{X}_f$,

(ii) satisfies input and state constraints, i.e., (2),

(iii) leads to performance guarantees of the form

$$\sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t \leq V(x_0). \tag{11}$$

Clearly for all $x_0 \in \mathbb{X}_f$, the implementation of (7) requires the computation of the control value $u^{mpc}(x_t)$ and the communication of state measurements and updated control values at each time instant $t \in \mathbb{N}$.

The main focus of this paper is to synthesize control laws which require less communication between controller, sensors and actuators but still guarantee certain performance in terms of the quadratic cost function in (10), next to closed-loop stability in $\mathbb{X}_f$ as in (i) and constraint satisfaction as in (ii). In particular, regarding the former, we aim at reducing the number of times the control input has to be updated and communicated while still fulfilling the sub-optimality criterion

$$\sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t \leq \beta V(x_0), \tag{12}$$

where a performance degradation index $\beta \geq 1$ is introduced to balance sub-optimality and the obtained saving in communication resources.

3. SELF-TRIGGERED SOLUTION

3.1 Approach

In this section we present a self-triggered approach to solve the problem formulated in Section 2.2. To introduce this approach consider the update times $\{t_l \mid l \in \mathbb{N}\} \subseteq \mathbb{N}$ that satisfy $t_{l+1} > t_l$, for all $l \in \mathbb{N}$. As such,

$$u_t = \overline{u}_l, \;\; t \in \mathbb{N}_{[t_l, t_{l+1})}. \tag{13}$$

At an update time t_l the goal is now to choose both the next control value $\overline{u}_l$ and the next update time $t_{l+1} > t_l$ such that t_{l+1} is as large as possible while still guaranteeing (12). Note that at times $t_l, t_l+1, \ldots, t_{l+1}-1$ the same control value $\overline{u}_l$ is used and hence, no communication nor computation are needed on times $t_l+1, t_l+2, \ldots, t_{l+1}-1$. Instrumental to guarantee (12) will be the usage of the dissipation-like inequality

$$\sum_{t=t_l}^{t_{l+1}-1} (x_t^\top Q x_t + \overline{u}_l^\top R \overline{u}_l) \leq \beta \Big[V(x_{t_l}) - V(x_{t_{l+1}})\Big] \\ s.t. \begin{cases} \overline{u}_l \in \mathbb{U}, \\ x_t \in \mathbb{X}, \;\; t = t_l, t_l+1, \ldots, t_{l+1}-1, \\ x_{t_{l+1}} \in \mathbb{X}_f, \end{cases} \tag{14}$$

as we will see.

In particular, the aim will now be, at update time t_l and state x_{t_l} to maximize t_{l+1} such that there exists a $\overline{u}_l$ for which inequality (14) is feasible. To formalize this setup, the notation

$$\overline{\mathcal{U}}_M(x) \triangleq \Big\{\boldsymbol{u} \in \mathbb{U}^M \Big| \exists\, \overline{u} \in \mathbb{U}\; \forall i = 0, \ldots, M-1 \\ \big(\Pi_i \boldsymbol{u} = \overline{u} \text{ and } x_i(x, \boldsymbol{u}) \in \mathbb{X}\big) \text{ and} \\ x_M(x, \boldsymbol{u}) \in \mathbb{X}_f\Big\} \tag{15}$$

is introduced, in which the projection operators $\Pi_i : \mathbb{R}^{n_u M} \to \mathbb{R}^M$ are given by $\Pi_i \boldsymbol{u} = u_i$ for $i = 0, 1, \ldots, M-1$ where $\boldsymbol{u} = (u_0, u_1, \ldots, u_{M-1})$ with $u_i \in \mathbb{R}^{n_u}$ for $i = 0, 1, \ldots, M-1$. Hence, the constraints in (15) specify, amongst others, that $\boldsymbol{u}$ is of the form $(\overline{u}, \overline{u}, \ldots, \overline{u})$ and $(u_0, u_0, \ldots, u_0)$. Using the notation in (15) and letting $M_l \triangleq t_{l+1} - t_l$ and $x_{t_l} = x$, inequality (14) can be rewritten as

$$\sum_{k=0}^{M_l-1} \big(x_k^\top(x, \boldsymbol{u}) Q x_k(x, \boldsymbol{u}) + u_0^\top R u_0\big) \\ \leq \beta \Big[V(x) - V\big(x_{M_l}(x, \boldsymbol{u})\big)\Big] \\ s.t. \;\; \boldsymbol{u} \in \overline{\mathcal{U}}_{M_l}(x), \tag{16}$$

where we use the connection $u_0 = \overline{u}_l$ and also shift time (using time-invariance of (1) and (2)). Hence, at time t_l and state $x_{t_l} = x$, we aim at maximizing $M_l \in \mathbb{N}$ such that (16) is still feasible for some $\boldsymbol{u} \in \overline{\mathcal{U}}_{M_l}(x)$ and in (13) $\overline{u}_l$ is then taken as $\Pi_0 \boldsymbol{u}$, and $t_{l+1} = t_l + M_l$. Therefore, we define for $x \in \mathbb{X}_f$ and fixed $M_l = M$,

$$\mathcal{U}_M^{st}(x) \triangleq \Big\{\boldsymbol{u} \in \overline{\mathcal{U}}_M(x) \;\Big|\; (18) \text{ holds}\Big\}, \tag{17}$$

where

$$\sum_{k=0}^{M-1} \big(x_k^\top(x, \boldsymbol{u}) Q x_k(x, \boldsymbol{u}) + u_0^\top R u_0\big) \\ \leq \beta \Big[V(x) - V\big(x_M(x, \boldsymbol{u})\big)\Big]. \tag{18}$$

Based on the above considerations, the following self-triggered MPC algorithm is proposed.

Algorithm 1.
At update time $t_l \in \mathbb{N}$ with $l \in \mathbb{N}$, and corresponding state $x_{t_l} = x$, the next update time t_{l+1} and the corresponding control law as in (13) are given by

$$t_{l+1} = t_l + \mathcal{M}^{st}(x) \tag{19a}$$

$$u_t = \overline{u}_l \in \Pi_0 \mathcal{U}^{st}(x), \; t \in \mathbb{N}_{[t_l, t_{l+1})}, \tag{19b}$$

where

$$\mathcal{M}^{st}(x) \triangleq \sup\big\{M \in \mathbb{N} \;\big|\; \mathcal{U}_M^{st}(x) \neq \emptyset\big\} \tag{20}$$

and

$$\mathcal{U}^{st}(x) \triangleq \mathcal{U}^{st}_{\mathcal{M}^{st}(x)}(x). \tag{21}$$

3.2 Theoretical properties

The self-triggered MPC law given in (19) has the following properties.

Theorem 1. Recursive feasibility: Letting $\beta \geq 1$, the control law given by Algorithm 1 is well defined, i.e., for all $x \in \mathbb{X}_f$, $\mathcal{M}^{st}(x) \in \mathbb{N}_{\geq 1}$ and $\mathcal{U}^{st}(x) \neq \emptyset$.

Proof. The proof is based on showing that for each $x \in \mathbb{X}_f$,

$$x^\top Q x + u_0^\top R u_0 + \beta V\big(x_1(x, u_0)\big) \leq \beta V\big(x\big) \tag{22}$$

holds for some u_0. Observe that (22) is equivalent to condition (18) for $M = 1$. From the fact that u^{mpc} was constructed based on Problem 2.1 using the conditions (I), (II) and (III) related to the terminal set and cost method it follows that for all $x \in \mathbb{X}_f$

$$V\big(x_1(x, u^{mpc}(x))\big) - V\big(x\big) \leq -x^\top Q x - \big(u^{mpc}(x)\big)^\top R u^{mpc}(x). \tag{23}$$

See, e.g., Mayne et al. [2000] for this fact.
Since $\beta \geq 1$ we see that (23) implies (22) for $u_0 = u^{mpc}(x)$. Clearly, $u^{mpc}(x) \in \mathbb{U}$ and $x_1(x, u^{mpc}(x)) \in \mathbb{X}_f$, and thus $u^{mpc}(x) \in \overline{\mathcal{U}}_1(x)$ it follows that $u^{mpc}(x) \in \mathcal{U}_1^{st}(x)$. Hence, $\mathcal{U}^{st}(x) \neq \emptyset$ and $\mathcal{M}^{st}(x) \in \mathbb{N}_{\geq 1}$ for all $x \in \mathbb{X}_f$. □

Theorem 2. Sub-optimality and constraint satisfaction: Letting $\beta \geq 1$, the control law given by Algorithm 1 satisfies the sub-optimality criterion given in (12) as well as the constraints given in (2) for the closed-loop system (1) with (19) for all $x_0 \in \mathbb{X}_f$.

Proof. Considering condition (14) with the control law as in (19b), for $l \in \mathbb{N}$ yields

$$\sum_{t=t_l}^{t_{l+1}-1} (x_t^\top Q x_t + u_t^\top R u_t) \leq \beta\Big[V(x_{t_l}) - V(x_{t_l+1})\Big] \quad (24a)$$

and by definition of (15),

$$u_t \in \mathbb{U},\ x_t \in \mathbb{X} \text{ for } t \in \mathbb{N}_{[t_l, t_{l+1})},\ \text{and } x_{t_{l+1}} \in \mathbb{X}_f \subseteq \mathbb{X}. \quad (24b)$$

Summing (24a) for $l \in \mathbb{N}$ and using that $\lim_{l\to\infty} V(x_{t_l}) \geq 0$ and $t_0 = 0$ leads to

$$\begin{array}{ll} l=0; & \sum_{t=t_0}^{t_1-1}(x_t^\top Q x_t + u_t^\top R u_t) \leq \beta V(x_{t_0}) - \beta V(x_{t_1}) \\ l=1; & \sum_{t=t_1}^{t_2-1}(x_t^\top Q x_t + u_t^\top R u_t) \leq \beta V(x_{t_1}) - \beta V(x_{t_2}) \\ l=2; & \sum_{t=t_2}^{t_3-1}(x_t^\top Q x_t + u_t^\top R u_t) \leq \beta V(x_{t_2}) - \beta V(x_{t_3}) \\ & \vdots \qquad\qquad\qquad\qquad \vdots \\ \hline + & \\ & \sum_{t=0}^{\infty}(x_t^\top Q x_t + u_t^\top R u_t) \leq \beta V(x_0), \end{array}$$

which is the sub-optimality criterion in (12). Since we have (24b) for all $l \in \mathbb{N}$, also the input and state constraints as in (2) are satisfied. □

Theorem 3. Stability: Letting $\beta \geq 1$, the control law given by Algorithm 1 renders the closed-loop system given by (1) and (19) asymptotically stable in $\mathbb{X}_f$.

Proof. From the proof of Theorem 2 and using $Q \succ 0$ we obtain for all $x_0 \in \mathbb{X}_f$ that

$$c_1 \sum_{t=0}^{\infty} \|x_t\|^2 \leq \sum_{t=0}^{\infty}(x_t^\top Q x_t + u_t^\top R u_t) \leq \beta V(x_0) \quad (25)$$

for some $c_1 > 0$ along trajectories of the closed-loop system (1) and (19). From (25) we immediately obtain that $\lim_{t\to\infty} x_t = 0$. In addition since we have $V(x) \leq c_2\|x\|^2$ for all $x \in \mathbb{X}_T$ also Lyapunov stability follows from (25). Therefore, the closed-loop system given by (1) and (19) is asymptotically stable in $\mathbb{X}_f$. □

4. IMPLEMENTATION CONSIDERATIONS

In the implementation of the control law (19), see (17), a crucial role is played by the evaluation of (18). Since the term $V\big(x_M(x,\boldsymbol{u})\big)$ in the right-hand side of (18) is unknown due to the fact that it depends on $\boldsymbol{u}$, which is yet to be designed, we rearrange terms in (18) leading to

$$\begin{aligned} &\beta V\big(x_M(x,\boldsymbol{u})\big) \\ &\quad + \sum_{k=0}^{M-1} \big(x_k^\top(x,\boldsymbol{u}) Q x_k(x,\boldsymbol{u}) + u_0^\top R u_0\big) \leq \beta V(x) \quad (26) \\ &s.t. \quad \boldsymbol{u} \in \overline{\mathcal{U}}_M(x). \end{aligned}$$

To implement (19), the objective is now for the given $x \in \mathbb{R}^{n_x}$ to find the largest $M \in \mathbb{N}$ for which (26) is feasible. For fixed $M \in \mathbb{N}$ one can check feasibility of (26) subject to $\boldsymbol{u} \in \overline{\mathcal{U}}_M(x)$ via minimization of the left-hand side over $\boldsymbol{u} \in \overline{\mathcal{U}}_M(x)$ and then verify if the obtained minimum is smaller than the right-hand side, i.e., smaller than $\beta V(x)$.

However, inspecting the left-hand side of (26) it can be observed that it is not expressed in a convenient form, as $V\big(x_M(x,\boldsymbol{u})\big)$ is a piecewise quadratic (PWQ) function, see, e.g., Bemporad et al. [2002]. To avoid solving the aforementioned minimization problem, which then becomes of the mixed integer quadratic programming (MIQP) type that is typically hard to solve, we replace $V\big(x_M(x,\boldsymbol{u})\big)$ by the corresponding optimization in (5) such that it can be transformed into a quadratic programming (QP). This effectively results in extending the prediction horizon.

To perform this replacement and to extend the prediction horizon, a similar notation to the one in (15) is introduced for the larger horizon $M+N$ leading to

$$\begin{aligned} \overline{\mathcal{U}}_{M,N}(x) \triangleq \Big\{\boldsymbol{u} \in \mathbb{U}^{M+N} \Big| &\big(\exists\, \overline{u} \in \mathbb{U}\ \forall i = 0,\ldots,M-1 \\ &\Pi_i \boldsymbol{u} = \overline{u}\big),\ \forall i = 0,\ldots,M+N-1\ \ x_i(x,\boldsymbol{u}) \in \mathbb{X}, \\ &\text{and } x_{M+N}(x,\boldsymbol{u}) \in \mathbb{X}_T\Big\}. \end{aligned} \quad (27)$$

Note that $\overline{\mathcal{U}}_M(x) = \overline{\mathcal{U}}_{M,0}(x)$.

Replacing $V\big(x_M(x,\boldsymbol{u})\big)$ in (26) by the corresponding optimization in (5) and incorporating the introduced notation in (27), the implementation of (19) can be realized as follows:

Problem 4.1. At update time t_l and state $x_{t_l} = x$ find the supremal M, denoted by M^*, such that

$$\begin{aligned} \min_{\boldsymbol{u} \in \overline{\mathcal{U}}_{M,N}(x)} &\Big(\beta\big(x_{M+N}^\top(x,\boldsymbol{u}) P x_{M+N}(x,\boldsymbol{u})\big) \\ &+ \sum_{k=M}^{M+N-1} \beta\big(x_k^\top(x,\boldsymbol{u}) Q x_k(x,\boldsymbol{u}) + u_k^\top R u_k\big) \\ &+ \sum_{k=0}^{M-1} \big(x_k^\top(x,\boldsymbol{u}) Q x_k(x,\boldsymbol{u}) + u_k^\top R u_k\big)\Big) \leq \beta V(x), \end{aligned} \quad (28)$$

holds and let $\boldsymbol{u}^* \in \overline{\mathcal{U}}_{M^*,N}(x)$ be the corresponding minimizer. Take the next update time $t_{l+1} = t_l + M^*$ and $\overline{u}_l = \Pi_0 \boldsymbol{u}^*$ in (13).

Note that in Problem 4.1, it has to be verified if the obtained minimum in the left-hand side is smaller than the right-hand side of the inequality (28), i.e. $\beta V(x)$, where $V(x)$ is the MPC value function given in (5). It is worth noting that indeed, the left-hand side of (28) is in the form of a QP with extended horizon $M+N$, which is to be compared with the solution of a QP with horizon N in the right-hand side of (28).

Although (28) results in an attractive QP-based check for a fixed $M \in \mathbb{N}$, computing the supremal M^* such that (28) holds is still difficult. For practical implementation, one therefore can consider one of the following schemes that still preserve the properties expressed in Theorems 1, 2 and 3.

Implementation Scheme 1.
The first scheme replaces $\mathcal{M}^{st}(x)$ in (19) by

$$\mathcal{M}^{st}_{max}(x) \triangleq \max\Big\{M \in \mathbb{N}_{[0,M_{max}]} \Big| \mathcal{U}^{st}(x) \neq \emptyset\Big\}, \quad (29a)$$

where an upper-bound $M_{max} \in \mathbb{N}$ is used. Clearly, $\mathcal{U}^{st}(x)$ in (19) becomes in this scheme

$$\mathcal{U}^{st}_{max}(x) = \mathcal{U}^{st}_{\mathcal{M}^{st}_{max}(x)}(x). \tag{29b}$$

Implementation Scheme 2.
A second scheme, which does not fix an a priori maximum on the inter-event time M, is obtained by replacing $\mathcal{M}^{st}(x)$ in (19) by

$$\mathcal{M}^{st}_{incr}(x) \triangleq \sup\left\{ M \in \mathbb{N} \,\middle|\, \mathcal{U}^{st}_{m}(x) \neq \emptyset, m \in \mathbb{N}_{[0,M]} \right\}, \tag{30a}$$

and $\mathcal{U}^{st}(x)$ by

$$\mathcal{U}^{st}_{incr}(x) = \mathcal{U}^{st}_{\mathcal{M}^{st}_{incr}(x)}(x). \tag{30b}$$

Hence, the scheme in (30) incrementally increases M until feasibility of (28) ceases to hold, and does not search for possibly larger values beyond $\mathcal{M}^{st}_{incr}(x)$ for which (28) can still be guaranteed.

5. DISCUSSION AND CONCLUDING REMARKS

Revisiting the optimal MPC setup described in Section 2.1, observe that at every time a QP has to be solved corresponding to an MPC problem of horizon N, and at each time the state information $x_t \in \mathbb{R}^{n_x}$ and the computed control law $u^*(x_t) \in \mathbb{R}^{n_u}$ have to be transmitted over a communication channel from sensors to controller and from controller to actuators, respectively.

Comparing this with the proposed self-triggered MPC scheme, thereby focussing on the practical implementations as discussed in (29) and (30), we see that communication is only needed at the update times $t_l, l \in \mathbb{N}$, while on times $t_l + 1, t_l + 2, \ldots, t_{l+1} - 1$ for each $l \in \mathbb{N}$ no computation/communication is required. However, at the update times more computations are needed, although the average computational load might be roughly the same as the optimal MPC setup.

In particular, to obtain $\mathcal{M}^{st}_{max}(x)$ for $M_{max} \in \mathbb{N}$ in the first implementation (29), it is necessary to compute $M_{max} + 1$ QP's corresponding to MPC problems with prediction horizons $N, N+1, \ldots, N+M_{max}$, respectively (having at most $N+1$ free control variables). Hence, for the first scheme, at time t_l and state $x_{t_l} = x$, one has to solve $M_{max} + 1$ QP's, and then the next $\mathcal{M}^{st}_{max}(x) - 1$ steps, no communication and computation is needed.

In obtaining $\mathcal{M}^{st}_{incr}(x)$, the second implementation scheme (30) implies incrementally increasing M until feasibility of (28) ceases to hold which translates to solving $\mathcal{M}^{st}_{incr}(x)+1$ QP's. Hence, $\mathcal{M}^{st}_{incr}(x)+1$ QP's are solved, while the next $\mathcal{M}^{st}_{incr}(x) - 1$ steps, no communication and computation is needed. This indicates for the second scheme that when $\mathcal{M}^{st}_{incr}(x_{t_l}), l \in \mathbb{N}$, is typically large, the average number of QP's to be solved is almost the same as the optimal MPC setup as in Problem 2.1. The same holds for the first implementation scheme if $\mathcal{M}^{st}_{max}(x_{t_l}), l \in \mathbb{N}$, is large and close to M_{max}.

Hence, less communication is realized through more computation at the update times. Accordingly, the proposed schemes are particularly relevant for practical control problems in which computation is cheap and communication is expensive (e.g., wireless communication via battery powered devices, where battery replacement is either costly or infeasible). Note though that the average computational load might not differ much from the optimal MPC scheme, as on the times between the updates no computations are needed for the self-triggered MPC scheme, while the optimal MPC needs to perform these for all times.

As a final remark, note that the sub-optimality parameter $\beta \geq 1$ can be used to trade the usage of network resources and the performance guarantees as in (12). As such, the proposed self-triggered MPC setup provides a viable control strategy to balance usage of network resources and control performance for constrained linear systems.

Future work will involve the validation of the presented ideas through simulation and experimental studies, the extension of the present case to the context of nonlinear systems and thus to self-triggered NMPC setups, the investigation of robustness properties and the comparison with other schemes.

REFERENCES

K.-E. Arzén. A simple event-based PID controller. In *Preprints IFAC World Conf.*, volume 18, pages 423–428, 1999.

K.J. Åström and B.M. Bernhardsson. Comparison of periodic and event based sampling for first order stochastic systems. In *Proc. IFAC World Conf.*, pages 301–306, 1999.

A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

D. Bernardini and A. Bemporad. Energy-aware robust model predictive control based on noisy wireless sensors. *Automatica*, 48:36–44, 2012.

R. Cogill. Event-based control using quadratic approximate value functions. In *Joint IEEE Conference on Decision and Control and Chinese Control Conference*, pages 5883–5888, Shanghai, China, 2009.

A. Eqtami, V. Dimarogonas, and K.J. Kyriakopoulos. Event-triggered control for discrete-time systems. In *Proc. American Control Conf.*, pages 4719–4724, 2010.

A. Eqtami, D.V. Dimarogonas, and K.J. Kyriakopoulos. Event-triggered strategies for decentralized model predictive controllers. In *Proc. IFAC World Conf.*, 2011.

L. Grune and F. Muller. An algorithm for event-based optimal feedback control. In *Proc. IEEE Conf. Decision & Control*, pages 5311–5316, 2009.

W.P.M.H. Heemels, R.J.A. Gorter, A. van Zijl, P.P.J. v.d. Bosch, S. Weiland, W.H.A. Hendrix, and M.R. Vonder. Asynchronous measurement and control: a case study on motor synchronisation. *Control Eng. Prac.*, 7:1467–1482, 1999.

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*, 81:571–590, 2008.

W.P.M.H. Heemels, M.C.F. Donkers, and A.R. Teel. Periodic event-triggered control based on state feedback. In *Proc. IEEE Conf. Decision & Control*, pages 2571–2576, 2011.

T. Henningsson, E. Johannesson, and A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44:2890–2895, 2008.

D. Lehmann. *Event-based state-feedback control.* Logos Verlag, Berlin, 2011.

L. Li and M. Lemmon. Weakly coupled event triggered output feedback system in wireless networked control systems. In *Allerton Conference on Communication, Control and Computing*, Urbana-Champaign, 2011.

J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46:211–215, 2010.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

M. Mazo Jr., A. Anta, and P. Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46:1310–1314, 2010.

A. Molin and S. Hirche. Structural characterization of optimal event-based controllers for linear stochastic systems. In *Proc. IEEE Conf. Decision and Control*, pages 3227–3233, 2010.

J. Sijs, M. Lazar, and W.P.M.H. Heemels. On integration of event-based estimation and robust MPC in a feedback loop. In *Proc. Conf. Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, pages 31–41. Springer-Verlag, 2010.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans. Autom. Control*, 52: 1680–1685, 2007.

P. Varutti and R. Findeisen. Event-based NMPC for networked control systems over UDP-like communication channels. In *Proc. American Control Conf.*, pages 3166–3171, 2011.

P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen. Event-based model predictive control for networked control systems. In *Proc. IEEE Conf. Decision & Control*, pages 567–572, 2009.

M. Velasco, J.M. Fuertes, and P. Marti. The self triggered task model for real-time control systems. In *Proc. IEEE Real-Time Systems Symposium*, pages 67–70, 2003.

X. Wang and M. Lemmon. Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability. *IEEE Trans. Autom. Control*, 45:452–467, 2009.

J.K. Yook, D.M. Tilbury, and N.R. Soparkar. Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators. *IEEE Trans. Control Systems Technology*, 10(4):503–518, 2002.