Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Mesh refinement for event-triggered nonlinear model predictive control ⋆

Omar J. Faqir * **Eric C. Kerrigan** *,**

* *Department of Electrical & Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mails: omar.faqir12@imperial.ac.uk;e.kerrigan@imperial.ac.uk.*
** *Department of Aeronautics, Imperial College London, SW7 2AZ London, U.K.*

Abstract: We consider the effect of using approximate system predictions in event-triggered control schemes. Such approximations often result from numerical transcription methods for solving continuous-time optimal control problems. Mesh refinement schemes guarantee upper bounds on the error in the differential equations used to model system dynamics. In particular, we show that with the accuracy guarantees of a mesh refinement scheme, then event-triggering schemes based on bounding the difference between predicted and measured state can be used with a guaranteed strictly positive inter-update time. We determine a lower bound for this time and show that additional knowledge of the employed transcription method and evaluation of the approximation errors may be used to obtain better online estimates of inter-update times. This is the first work to consider using the solution accuracy of an optimal control problem as a metric for triggering new control updates.

Keywords: Model-based control, Nonlinear control, Event-triggered control, Mesh refinement

1. INTRODUCTION

In most practical systems we do not have continuous state feedback. Instead, the system is measured and control recomputed at update times $\mathcal{T}_u \triangleq \{t_i^u\}_{i\in\mathbb{N}_0}$. Conventionally, the control engineer must determine a *single* update interval $\tau^u := t_{i+1} - t_i, \forall i \in \mathbb{R}^+$ to satisfy performance criteria for *all* possible disturbance realizations (Gommans et al., 2014). This 'worst-case' periodic design leads to over-allocation of communication and computation resources, without assurances of superior performance (Åström and Bernhardsson, 1999). Instead, in event-triggered and self-triggered schemes, the control is only updated when certain criteria are met, as formalized in Section 2.

(Nonlinear) model predictive control ((N)MPC) complements event- and self-triggered schemes. The predictions resulting from the optimization may result in longer open-loop run times than using zero-order hold state feedback controllers (Lucia et al., 2016). However, the computational burden of solving a nonlinear program (NLP) online is a key drawback. Aperiodic triggering schemes have the potential to increase the average available computation time between control updates. Results in Varutti et al. (2009) help bridge existing work in networked control to triggering of MPC schemes, and additionally uses predictions to compensate for delays or information losses.

Numerical methods for solving continuous-time optimal control problems only generate approximations of the continuous state and input trajectories. The computational burden of NMPC may potentially be reduced by either solving a less complex NLP, resulting in *lower quality* predictions, or by solving the NLP less frequently, resulting in a more *outdated* prediction. We investigate how solution accuracy and triggering conditions affect the inter-event times in an event-triggered NMPC framework. We derive a lower bound for the inter-update time (IUT), and further show how information concerning solution quality may be used online for less conservative IUT estimates.

In Section 2 we introduce the considered triggering schemes, detail the continuous-time optimization and explain high level principles of mesh refinement procedures. In Section 3 we derive sufficient conditions for a strictly positive lower bound on IUTs, which we improve on in Section 4 using additional error information provided by mesh refinement software. The introduced concepts are exemplified in Sections 5 and 6.

2. PRELIMINARIES

Consider the nonlinear dynamical system described by

$$\dot{x}(t) = f(t, x(t), u(t)) + w(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state of the system, $u(t) \in \mathbb{R}^m$ the control input. The bounded exogenous $w(t) \in \mathbb{R}^n$ input satisfies $|w(t)| \leq \hat{w}$, where $|\cdot|$ is an appropriately defined norm. Nominal system dynamics are described by $f(\cdot)$ and are Lipschitz in x with constant L_x, i.e. for all x_1, x_2, t,

$$|f(t, x_1, u) - f(t, x_2, u)| \leq L_x |x_1 - x_2|$$

2.1 Event- and Self-Triggering Schemes

In event-triggered control (ETC) (Heemels et al., 2012) the system is continuously measured but the control is only

⋆ The support of the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1) is gratefully acknowledged.

recomputed when a certain 'triggering' condition is met, e.g. the following, adapted from Heemels et al. (2012):

Problem 1. (**ETC**). Consider the dynamical system described by (1), and a time-varying state feedback law $u(t) = \mu(x(t), t)$ which renders the closed-loop system globally asymptotically stable (GAS). Identify a set of state- and input- dependent conditions $F_{\text{event}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, resulting in update times

$$t_{i+1}^u = \inf\{t \in \mathbb{R}_0^+ | t > t_i^u, F_{\text{event}}(x(t), u(t)) \geq 0\},$$

for which the closed-loop system with sampled-data implementation is GAS (and satisfies appropriately defined performance criteria).

Determining an appropriate $F_{\text{event}}(\cdot)$ is an open problem and depends on the control scheme under consideration. Many event-based controllers trigger on a measure of the difference between predicted and measured state. We explicitly denote triggering dependency on most recent state predictions $\tilde{x}(t)$ by defining and employing the triggering condition $\tilde{F}_{\text{event}}(x(t), u(t), \tilde{x}(t))$. The ETC scheme must satisfy the following properties. The IUTs, $\tau_i^u := t_{i+1}^u - t_i^u, \forall i \in \mathbb{R}_0^+$, must be strictly positive. To avoid 'Zeno' phenomena there must exist a lower bound $\underline{\tau}^u \leq \tau_i^u$ (Borgers and Heemels, 2014). The IUTs must also be upper bounded. This is easier to enforce, and in the case of MPC schemes is often restricted to τ_i^u being less than or equal to the prediction horizon. Feasibility of the MPC parametric optimization problem must be guaranteed $\forall t \in \mathcal{T}_u$.

In self-triggered control (STC) (Anta and Tabuada, 2010; Gommans et al., 2014; Velasco et al., 2003), the control is computed and IUTs explicitly calculated from the predicted system trajectory under this control, e.g. the following, which is adapted from Anta and Tabuada (2010):

Problem 2. (**STC**). Consider the dynamical system described by (1), and time-varying state feedback law $u(t) = \mu(x(t), t)$ which renders the closed-loop system GAS. Identify a set of state and input dependent conditions $F_{\text{self}} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}$, resulting in inter-sample periods

$$\tau_i^u = \inf\{\tau \in \mathbb{R}_0^+ | F_{\text{self}}(x(t_i^u), u(t), \tau) \geq 0\},$$

for which the closed-loop system is GAS (and satisfies appropriately-defined performance criteria).

Typically both $F_{\text{event}}(\cdot)$ and $F_{\text{self}}(\cdot)$ must ensure sufficient decrease in an appropriately defined Lyapunov function. STC lacks the inherent robustness of ETC, but doesn't require dedicated hardware for continuous operation. Furthermore, in STC t_{i+1}^u and τ_i^u are known in advance, which may be important for process scheduling on embedded platforms or dynamic communication resource allocation. Both STC and ETC conditions are based on emulation of an *a priori* known control law $\mu(\cdot)$.

2.2 Nonlinear Model Predictive Control

In MPC a finite time control sequence/trajectory is designed to optimize some performance metric of predicted plant evolution (Mayne et al., 2000). Classical MPC paradigms account for uncertainty by resolving the optimization problem — which is parametric in the most recent measurements — periodically. Only the first portion of the computed input is applied to the plant before the procedure is repeated. The complexity of performing optimization in real-time has historically restricted MPC to slow and/or simple (linear) dynamical systems. Greater computational resources and tailor-made optimization algorithms allow MPC of faster and more complex systems.

The continuous-time NMPC problem solved at time t_i^u may be cast in the general Bolza form

$$\min_{\hat{x},\hat{u}} \ \Phi(\hat{x}(t_0), \hat{x}(t_f)) + \int_{t_0}^{t_f} L(\hat{x}(t), \hat{u}(t), t) \, \mathrm{d}t \tag{Pa}$$

$$\text{s.t. } \forall t \in [t_0, t_f],$$

$$\dot{\hat{x}}(t) = f(\hat{x}(t), \hat{u}(t), t), \tag{Pb}$$

$$c(\hat{x}(t), \hat{u}(t), t) \leq 0, \tag{Pc}$$

$$\phi(\hat{x}(t_0), \hat{x}(t_f)) = 0, \tag{Pd}$$

$$\hat{x}(t_0) = x(t_i^u), \tag{Pe}$$

where $t_0 := t_i^u$. Optimization is performed over horizon $T := t_f - t_0 > 0$ on the internal variables $\hat{x}, \hat{u}$, representing the state and input trajectory. The cost (Pa) is composed of the stage (Lagrange) cost functional $L(\cdot)$ and boundary (Mayer) cost functional $\Phi(\cdot)$. The resulting state trajectory satisfies the nominal dynamics $f(\cdot)$, typically enforced as ordinary differential equations (Pb). States and controls must satisfy the path constraints (Pc) for all time. Key advantages of MPC are its use of a predictive model and guarantees of constraint satisfaction. With reference to the description of system (1), if we can determine the feasible set for problem (P), then we may relax L_x to be a local Lipschitz constant valid on the set of admissible states.

Although we will assume existence of a local minimizer to problem (P), we require no assumptions on the uniqueness of the minimizer. We refer to $x^*(\cdot), u^*(\cdot)$ as the *true* solution of the optimal control problem (P),

$$(x^*(\cdot), u^*(\cdot)) := \arg\min_{(\hat{x},\hat{u})} (\text{P}).$$

2.3 Direct Collocation & Mesh Refinement

The dynamic optimization (P) is continuous-time and infinite-dimensional. The problem is intractable or impossible to solve analytically in all but the simplest cases. Instead, direct collocation methods transcribe (P) into a finite-dimensional NLP, after which numerical optimization methods are employed. The NLP solution is then interpolated to reconstruct an approximate continuous-time solution $\tilde{x}(\cdot), u(\cdot)$ to (P) (Betts, 2010). System trajectories may be approximated as continuous piecewise polynomials using a combination of h- and p-methods (known as hp-methods) (Kelly, 2017), briefly detailed below. Define the interval $\Omega := (t_0, t_f)$, and $\overline{\Omega}$ as the closure of Ω.

Definition 1. (Mesh). The set $\mathcal{T}_h$ is called a **mesh** and consists of open intervals $T \subset \Omega$ satisfying conditions

(1) **Disjunction** $T_1 \cap T_2 = \emptyset \ \forall$ distinct $T_1, T_2 \in \mathcal{T}_h$,
(2) **Coverage** $\cup_{T \in \mathcal{T}_h} \overline{T} = \overline{\Omega}$,
(3) **Resolution** $\max_{T \in \mathcal{T}_h} |T| = h$,
(4) **Quasi-uniformity** $\min_{T_1, T_2 \in \mathcal{T}_h} \frac{|T_1|}{|T_2|} \geq \sigma > 0$,

where the constant σ must not depend on the mesh parameter h.

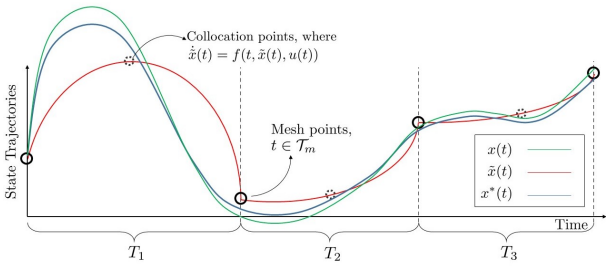


Fig. 1. A comparison of $x(\cdot)$, $x^*(\cdot)$ and the approximate solution $\tilde{x}(\cdot) \in \mathcal{X}_P$ defined on mesh $\mathcal{T}_h$.

We denote the number of mesh segments by $K := \operatorname{card} \mathcal{T}_h$ and refer to the k^{th} segment as $T_k, \forall k \in \mathcal{K}_h := \{1, \dots, K\}$. We define the ordered set of mesh points as $\mathcal{T}_m \triangleq \cup_{k \in \mathcal{K}} \inf \overline{T}_k$, where two polynomial segments are joined, and index them identically to the mesh segments. The approximate state trajectory $\tilde{x}(\cdot)$ defined on this mesh will then satisfy

$$\tilde{x} \in \mathcal{X}_p := \{x : \overline{\Omega} \to \mathbb{R}^n | x \in \mathcal{C}^0(\overline{\Omega}), x \in \mathcal{P}_p(T)^n, \forall T \in \mathcal{T}_h\},$$

where $\mathcal{P}_p(T)$ is the space of functions that are polynomials of maximum degree $p \in \mathbb{N}_0$ on interval T. In collocation methods the dynamic constraints (Pb) are enforced at a finite number of collocation points in each T_k, the location and number of which depend on the chosen method. Denote collocation points in T_k as $t_k^c \in \mathcal{T}_{k,c}$, and

$$\dot{\tilde{x}}(t^c) = f(\tilde{x}(t^c), u(t^c), t^c), \forall t_c \in \mathcal{T}_{k,c}, k \in \mathcal{K}.$$

In h-methods (e.g. Euler, Hermite-Simpson), a fixed-degree polynomial is used on each segment (Betts, 2010). For fixed p, the approximation quality may be improved by reducing h and increasing the number of segments. In p-methods the unknown trajectories are approximated as an interpolation of orthogonal basis functions (Fahroo and Ross, 2008). In global p-methods, a single interpolated function is used over the entire interval Ω. Approximation quality is adjusted by changing the polynomial order.

Mesh refinement schemes iteratively adjust the number of segments and/or polynomial order of the transcription. At each iteration of the refinement, problem (P) must be re-transcribed and the resulting NLP resolved. Such schemes guarantee a user-defined level of accuracy of the approximation, which is important in applications. We must therefore consider three different trajectories. The *true* solution of (P), $x^*(\cdot), u^*(\cdot)$, which is in general unknown. The *approximate* solution $\tilde{x}(\cdot), u(\cdot)$ which depends on the chosen transcription method and is computed numerically. Finally, the *actual* state trajectory $x(\cdot)$ of the plant resulting from evaluating (1) with the approximate input $u(\cdot)$. In general $\tilde{x}(t) \neq x^*(t) \neq x(t), \forall t \notin \mathcal{T}_u$. A pictorial representation of this is shown in Fig. 1.

Since $x^*(\cdot), u^*(\cdot)$ are not known, they cannot be used to evaluate the approximation accuracy. One possible error metric is the absolute local error at time t in the i^{th} state,

$$\varepsilon_i(t) := \dot{\tilde{x}}_i(t) - f_i(t, \tilde{x}(t), u(t)). \tag{3}$$

$\varepsilon(t) \triangleq [\varepsilon_1(t), \dots, \varepsilon_n(t)]'$ is the error in the solution of the differential equations (Pb) resulting from the transcription method, and is necessarily zero at collocation points. The *approximation error* in state i over mesh interval T_k is given by the quadrature

$$\eta_{k,i} := \int_{t_k}^{t_{k+1}} |\varepsilon_i(\tau)| \, \mathrm{d}\tau, \tag{4}$$

for $t_k, t_{k+1} \in \mathcal{T}_m$. Refinement schemes guarantee upper bounds on the error quadrature in each segment, $\eta_{k,i} \leq \hat{\eta}_i, \forall k \in \mathcal{K}, \ \forall i \in \{1, \dots, n\}$. Although it is common to use $\max_i \eta_{k,i}$ as a basis for refinement, some problems require states to be resolved to different accuracies. However, for our following analysis, we bound the sum error in T_k as

$$\eta_k := \int_{t_k}^{t_{k+1}} |\varepsilon(\tau)| \, \mathrm{d}\tau \leq \hat{\eta}. \tag{5}$$

This is by no means the only error metric used. Other examples include relative local error (Betts, 2010) or local error in dual variables (Paiva and Fontes, 2019).

3. MINIMUM IUT

Define the *prediction error* as $\epsilon(t) := \tilde{x}(t) - x(t)$, where $\epsilon(t) = 0, \ \forall t \in \mathcal{T}_u$, and the associated prediction error dynamics $\dot{\epsilon}(t) = \dot{\tilde{x}}(t) - \dot{x}(t)$. We propose employing a commonly used triggering condition (Li and Shi, 2014)

$$\tilde{F}_{\text{event}}(x(t), u(t), \tilde{x}(t)) := \hat{\epsilon} - |\epsilon(t)|, \tag{6}$$

for some $\hat{\epsilon} > 0$. Problem (P) is resolved whenever the error magnitude passes a certain threshold.

Importantly, triggering is not based on the true solution of (P), but rather on the approximation $\tilde{x}(t)$. Even if there are no exogenous inputs or unmodelled dynamics, this error metric may be non-zero for any $t \notin \mathcal{T}_u$. We seek an expression to relate the prediction error $|\epsilon(t)|$ to the maximum allowable approximation error $\hat{\eta}$, and thereby guarantee an IUT $\underline{\tau}^u$ when triggering is based on a (possibly bad) approximate solution of (P).

Theorem 1. For the ETC resulting from solving (P) to satisfy (5) at times $\mathcal{T}_u$, implicitly defined through triggering condition (6), the minimum IUT

$$\underline{\tau}^u = \frac{\hat{\epsilon} - \hat{\eta}}{\frac{\hat{\eta}}{\sigma h} + L_x \hat{\epsilon} + \hat{w}} \tag{7}$$

is guaranteed to be strictly positive for $\hat{\epsilon} > \hat{\eta}$.

Proof 1. For succinctness we will write $f(t, x, u)$ as $f(x)$. Without loss of generality, assume problem (P) was solved at $t_i^u = 0$, parametrized as $\tilde{x}(0) = x(0)$. Then,

$$\epsilon(t) = \int_0^t \dot{\epsilon}(\tau) \mathrm{d}\tau = \int_0^t \dot{\tilde{x}}(\tau) - f(x(\tau)) - w(\tau) \mathrm{d}\tau. \tag{8}$$

Rearranging (3) and subsequently substituting $\dot{\tilde{x}}(t) = \varepsilon(t) + f(\tilde{x}(t))$ into (8) results in

$$\epsilon(t) = \int_0^t \varepsilon(\tau) + f(\tilde{x}(\tau)) - f(x(\tau)) - w(\tau) \mathrm{d}\tau.$$

Through application of the triangle inequality, the prediction error magnitude $|\epsilon(t)|$ is bounded from above by

$$\int_0^t |\varepsilon(\tau)| \, \mathrm{d}\tau + \int_0^t |f(\tilde{x}(\tau)) - f(x(\tau))| \, \mathrm{d}\tau + \int_0^t |-w(\tau)| \, \mathrm{d}\tau. \tag{9}$$

Now consider the set $\mathcal{T}_t \subseteq \mathcal{T}_h$ of open intervals of T, i.e. $\mathcal{T}_t := \{T \in \mathcal{T}_h | T \cap \overline{(0, t)} \neq \emptyset\}$, which is also a mesh on $\Omega_t = (0, t)$, and by construction satisfies the properties of

Definition 1. Since the mesh refinement scheme guarantees $\eta_k \leq \hat{\eta}$, the first integral in (9) is bounded from above as

$$\int_0^t |\varepsilon(\tau)| \, \mathrm{d}\tau \leq \sum_{T \in \mathcal{T}_t} \int_T |\varepsilon(\tau)| \, \mathrm{d}\tau \leq \sum_{T \in \mathcal{T}_t} \hat{\eta}.$$

Through application of the resolution and quasi-uniformity properties of mesh $\mathcal{T}_t$, the smallest interval satisfies $\min_{T \in \mathcal{T}_t} |T| \geq \sigma h := \underline{h}$. Using $\lceil \cdot \rceil$ to denote the ceiling function, we bound the maximum number of segments in interval $(0, t)$ as $\lceil t \underline{h}^{-1} \rceil$, from which it follows that

$$\int_0^t |\varepsilon(\tau)| \, \mathrm{d}\tau \leq \left\lceil \frac{t}{\underline{h}} \right\rceil \hat{\eta} \leq \left(\frac{t}{\underline{h}} + 1 \right) \hat{\eta}.$$

From the assumption of Lipzchitz continuity of $f(\cdot)$ and application of the proposed triggering condition (6), the second integral in (9) is bounded as

$$\int_0^t |f(\tilde{x}(\tau)) - f(x(\tau))| \, \mathrm{d}\tau \leq t L_x \hat{\epsilon}.$$

The final integrand in (9) is bounded as $|-w(t)| \leq \hat{w}$ by assumption. Therefore,

$$|\epsilon(t)| \leq \left(\frac{t}{\underline{h}} + 1 \right) \hat{\eta} + t \left(L_x \hat{\epsilon} + \hat{w} \right).$$

We guarantee $\underline{\tau}^u > 0$ by showing that $\exists \, t > 0$ such that

$$\left(\frac{t}{\underline{h}} + 1 \right) \hat{\eta} + t \left(L_x \hat{\epsilon} + \hat{w} \right) \leq \hat{\epsilon}.$$

Through manipulation, this inequality is satisfied when

$$t \leq \frac{\hat{\epsilon} - \hat{\eta}}{\frac{\hat{\eta}}{\sigma h} + L_x \hat{\epsilon} + \hat{w}}.$$

Since all parameters are in $\mathbb{R}_0^+$, the RHS is guaranteed to be strictly positive as long as $\hat{\epsilon} > \hat{\eta}$. □

The bound will be finite even when $\hat{w} = 0$ and doesn't use knowledge of the discretization/interpolation schemes, in which case the bound may be improved. Mesh refinement schemes warm-start and solve the NLP resulting from (P) with each new, increasingly dense, mesh. Betts (2010) suggests an initial relatively sparse mesh, with the refinement adding in points where necessary, as opposed to an unnecessarily dense starting mesh. For small $\hat{\eta}$ this may result in more NLPs being solved for a single approximation $\tilde{x}(\cdot), u(\cdot)$. Being a decreasing function of $\hat{\eta}$, (7) relates a trade-off between solving more NLPs per control update — yielding a higher accuracy solution — but less frequently, to solving fewer NLPs at each update, but more frequently.

4. ONLINE ESTIMATION OF IUT

The IUT bound (7) can be improved with information provided by solving (P). This is the first explicit definition of IUTs accounting for both problem data and the solution accuracy of (P). Although the following conditions resemble STC, we make no claims about stability or performance. Instead, we simply provide estimates of IUTs, which may be useful for computation and communication resource scheduling. First, we determine a lower bound on τ_i^u, not dependent on η_k, through using knowledge of the collocation constraints.

Theorem 2. (Collocation Triggering (CT)). Let $L_{p,k}$ be the Lipschitz constant of the approximating polynomial used in segment T_k. The IUT

$$\tau_i^{\mathrm{CT}} = \sup \left\{ \tau \in \mathbb{R}^+ \middle| \sum_{k \in \mathcal{K}_t} (t - t_i^u) \left(L_x \hat{\epsilon} + \hat{w} \right) + L_x \int_{T_k} |\tilde{x}(\tau) - \tilde{x}(t_k^c)| \, \mathrm{d}\tau + L_{p,k} \int_{T_k} |\tau - t_k^c| \, \mathrm{d}\tau \leq \hat{\epsilon} \right\} \tag{10}$$

calculated from the approximate solution $\tilde{x}(\cdot)$ to problem (P) at time t_i^u satisfies $|\epsilon(t_i^u + \tau)| \leq \hat{\epsilon}, \forall \tau \leq \tau_i^{\mathrm{CT}}$.

Proof 2. We begin by noting that

$$\int_0^t |\varepsilon(t)| \, \mathrm{d}\tau \leq \sum_{k \in \mathcal{K}_t} \int_{T_k} \left| \dot{\tilde{x}}(t) - f(\tilde{x}(\tau)) \right| \mathrm{d}\tau, \tag{11}$$

where the inequality follows from $\sup T_k \geq t$. As (Pb) are satisfied exactly at points t_k^{col}, we may write (11) as

$$\begin{aligned} &\sum_{k \in \mathcal{K}_t} \int_{T_k} \left| \dot{\tilde{x}}(\tau) - \dot{\tilde{x}}(t_k^{\mathrm{col}}) + f(\tilde{x}(t_k^{\mathrm{col}})) - f(\tilde{x}(\tau)) \right| \mathrm{d}\tau \\ &\leq \sum_{k \in \mathcal{K}_t} \int_{T_k} \left| \dot{\tilde{x}}(\tau) - \dot{\tilde{x}}(t_k^{\mathrm{col}}) \right| + \left| f(\tilde{x}(t_k^{\mathrm{col}})) - f(\tilde{x}(\tau)) \right| \mathrm{d}\tau \\ &\leq \sum_{k \in \mathcal{K}_t} \int_{T_k} L_{p,k} \left| \tau - t_k^{\mathrm{col}} \right| + L_x \left| x(t_k^{\mathrm{col}}) - \tilde{x}(\tau) \right| \mathrm{d}\tau, \end{aligned}$$

where the second inequality results from noting that all polynomials in $\mathcal{P}_p(T_k)$ defined on bounded set T_k are Lipschitz, with Lipschitz constant denoted as $L_{p,k}$. Substituting the quadrature error bound into (9) and restricting the RHS to $\leq \hat{\epsilon}$ yields the explicit IUT (10). □

For lower-order h methods, in particular, it may be easy to evaluate $L_{p,k}$ online for each interval T_k.

Although the mesh refinement algorithm may not return the approximation error over arbitrary intervals, software such as `ICLOCS2` (Nie et al., 2018) does return η_k corresponding to T_k. However, depending on how the integral (4) is evaluated, we do not necessarily know the distribution of $\varepsilon(t)$ over this interval.

Corollary 1. (Quadrature Error Triggering (QET)). The IUT

$$\tau_i^{\mathrm{QET}} = \sup \left\{ \tau \in \mathbb{R}^+ \middle| \sum_{k \in \mathcal{K}_t} \eta_k + t \left(L_x \hat{\epsilon} + \hat{w} \right) \leq \hat{\epsilon} \right\} \tag{12}$$

determined from $\tilde{x}(\cdot)$, η_k, which approximately solves problem (P) at time t_i^u, satisfies $|\epsilon(t_i^u + \tau)| \leq \hat{\epsilon}, \forall \tau \leq \tau_i^{\mathrm{QET}}$.

Proof 3. We begin by noting that $\int_0^t |\varepsilon(\tau)| \, \mathrm{d}\tau \leq \sum_{k \in \mathcal{K}_t} \eta_k$, where the inequality follows from $\sup |\mathcal{T}_t| \geq t$. Therefore we may bound the prediction error at $t \in [t_i^u, t_{i+1}^u)$ as

$$|\epsilon(t)| \leq \sum_{k \in \mathcal{K}_t} \eta_k + (t - t_i^u) \left(L_x \hat{\epsilon} + \hat{w} \right).$$

Substituting into (9), and restricting the RHS to be $\leq \hat{\epsilon}$ results in the explicitly defined IUT given in (12). □

If instead the refinement returns $\epsilon(t)$, then replacing $\sum \eta_k$ with $\int_0^t |\epsilon(t)|$ guarantees the ordering $\tau_k^{\mathrm{CT}} \leq \tau_k^{\mathrm{QET}}$.

5. EXAMPLE: LINEAR SYSTEM

Consider the LTI system described by

$$\dot{x}(t) = Ax(t) + E\hat{w}, \tag{13}$$

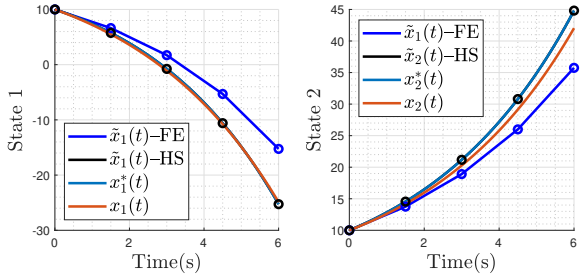
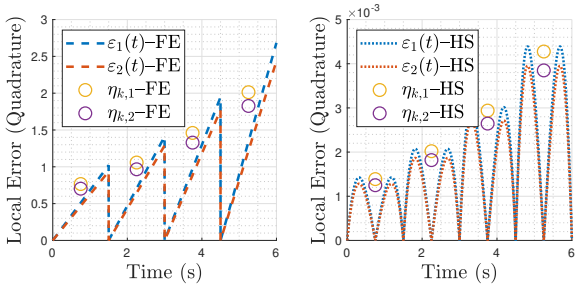
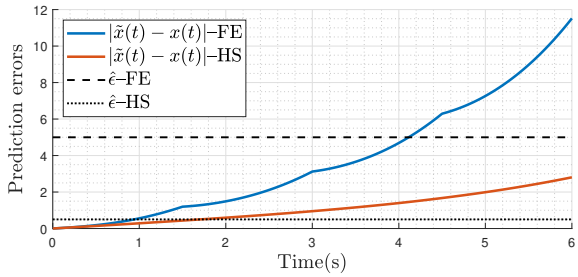


Fig. 2. A comparison of the true $x(t)$, the nominal $x^*(t)$ and the approximate $\tilde{x}(t) \in \mathcal{X}_1$ resulting from FE and HS collocation on a mesh with $h = 1.5$, $\sigma = 1$.



(a) Approximation errors and quadratures.



(b) Prediction errors of collocation schemes.

Fig. 3. Errors associated to the trajectories shown in Fig. 2.

$$x(t_0) = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \; A = \begin{bmatrix} 0.05 & 0.5 \\ 0 & -0.5 \end{bmatrix}, \; E = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

where $x \in \mathbb{R}^n$, simulated over $\Omega = 6\,\text{s}$. We discretize $x(t)$ on a coarse uniform mesh of $h = 1.5\,\text{s}$ using a Forward Euler(FE) or a third-order Hermite-Simpson (HS) scheme, detailed in Kelly (2017). Fig. 2 shows both approximations alongside the 'true' and 'nominal' trajectories $x(t), x^*(t)$ generated from Matlab's `linsim` function. Shown in Fig. 3a, the approximation error $\epsilon_i(t)$ and quadratures $\eta_{k,i}$ for the HS scheme are orders of magnitude lower than the FE. We choose $\hat{\eta} = \max_k \eta_k$ a posteriori, being 3.8420 in the case of FE and 0.0081 for HS. Since (13) is unstable, the states diverge and approximation errors grow.

To guarantee $\hat{\epsilon} > \hat{\eta}$, we chose an ETC threshold $\hat{\epsilon} = 5$ for the FE and $\hat{\epsilon} = 0.5$ for the HS scheme. The ETC condition (6) is met at 4.12 s and 1.7 s, respectively. Fig. 3b shows the FE approximation error results in rapid growth in $\epsilon(t)$. The minimum IUT $\underline{\tau}^u$ and $\tau^{\text{QET}}, \tau^{\text{CT}}$ are detailed in Table 1, and the associated functions plotted in Fig. 4. The line segments of QET-FE(η) and QET-HS(η) in Fig. 4 have slope $L_x\hat{\epsilon} + \hat{w}$, and reflect state uncertainty due to $w(t)$. The discontinuities arise from approximation errors.

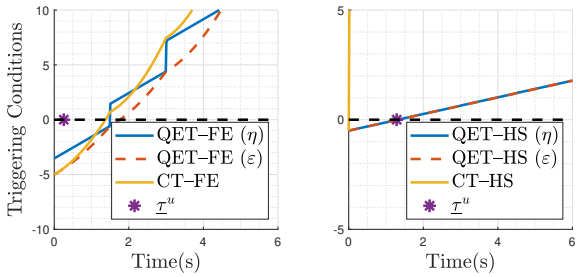


Fig. 4. A comparison of the online triggering estimations proposed in Section 4 for FE and HS Schemes.

Table 1. Comparison of estimated IUTs.

	$\hat{\eta}$	$\underline{\tau}^u$	$\tau^{\text{QET}}(\eta)$	$\tau^{\text{QET}}(\varepsilon)$	τ^{CT}
FE ($\hat{\epsilon} = 5$)	3.842	0.25	1.49	1.74	1.39
HS ($\hat{\epsilon} = 0.5$)	0.0081	1.22	1.27	1.28	0.005

For the FE scheme, $\underline{\tau}^u$ is significantly more conservative than $\tau^{\text{QET}}(\eta)$, whereas $\underline{\tau}^u \approx \tau^{\text{QET}}(\eta)$ for the HS. This is because $\underline{\tau}^u$ is estimated from the worst case absolute local error. Using relative local error, as suggested in Betts (2010, Ch. 5), may result in better agreement between $\underline{\tau}$ and $\tau^{\text{QET}}(\eta)$ for the FE scheme.

6. CLOSED-LOOP SIMULATION

We now consider decreasing-horizon closed-loop simulation of a cart-pole swing-up problem while minimizing the integral of the square of the control effort. The nonlinear dynamics $f(\cdot)$ and trapezoidal collocation scheme we employ are detailed by Kelly (2017, Sec. 6). This transcription scheme results in piecewise quadratic state reconstruction and piecewise linear inputs. Additive noise satisfying $\hat{w} \leq 10^{-1}$ has been added. In this example our mesh refinement strategy is similar to that presented by Betts (2010, Ch. 5), but acting upon the *absolute* local error. Our results do not depend on the proxy used for refinement, or the method of adding/removing new points – only on the termination criteria of the refinement scheme. The resulting NLP is solved using `IPOPT`. The mesh points at t_i^u are $\mathcal{T}_{m,i}$. At each control update `IPOPT` is warm-started with the primal variables from the previous solution $\tilde{x}(t), u(t)\ \forall t > t_i^u, t \in \mathcal{T}_{m,i-1}$.

Fig. 5 shows the system trajectory for a decreasing horizon NMPC, with an initial horizon length of 2 s. For the full problem definition see Kelly (2017, Sec. 6). In short, the cart must invert the pendulum and move a certain distance by time 2 s. We estimate $L_x = 37.9$ for the admissible set, resulting in a guaranteed IUT of $\underline{\tau}^u = 0.0207\,\text{s}$. From Fig. 5, the average IUT is actually significantly larger. We also observe a correspondence between the density of mesh points and update times. The bound (7) suggests a trade-off between the cost of each update, and update frequency. We now simulate the cart-pole swing-up problem in a receding horizon fashion, using a constant horizon of 2 s for 10 s. For this numerical analysis, the mesh refinement scheme terminates based on the *relative* local error (Betts, 2010, Ch. 5) — which we bound with $\hat{\eta}_{\text{rel}}$ — since this is more representative of the state-of-the-art. Owing to the benefits of warm-starting, the initial solve accounts for a large amount of the total computation time.

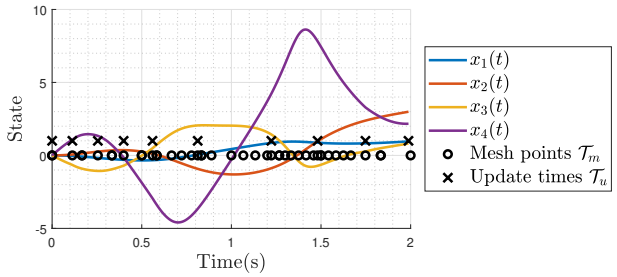


Fig. 5. An example closed-loop trajectory for the cart-pole swing-up problem with $\hat{\eta} = 0.01$ and $\hat{\epsilon} = 0.05$.

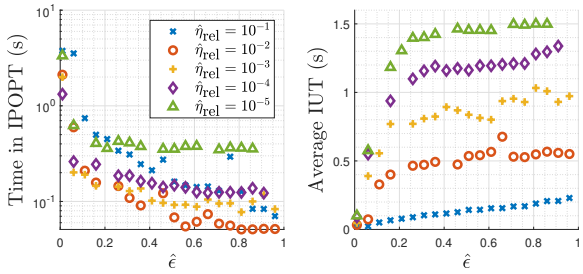


Fig. 6. Time in `IPOPT` — normalized by simulation termination times — and average IUTs for a range of $\hat{\eta}, \hat{\epsilon}$. Values of $\hat{\eta}, \hat{\epsilon}$ causing feasibility issues are omitted.

The total time in IPOPT and the average IUT is shown in Fig. 6 for a range of $\hat{\eta}_{\text{rel}}, \hat{\epsilon}$. As we reduce $\hat{\eta}_{\text{rel}}$, or increase $\hat{\epsilon}$, the average IUT increases, as expected. Although there is a marginal increase in average IUT time for a given $\hat{\eta}_{\text{rel}}$ by increasing $\hat{\epsilon}$ (due to physical system parameters such as $\hat{w}$ and L_x), we do see that even for relatively large allowable prediction errors, using a tighter ODE tolerance has significant effects on the IUT. Similarly, as $\hat{\epsilon}$ increases, the time in `IPOPT` tends to decreases, since the number of solves decrease. However, there is also an effect of $\hat{\eta}_{\text{rel}}$ on the computation time, because $\hat{\eta}_{\text{rel}}$ also affects the quality of our warm-start. If we write problem (P) as being parametric in the measured state, then the variable $\hat{\epsilon}$ determines the amount this problem is being perturbed at each update. For a large perturbation, our previous solution may not be an effective guess for warm-starting. In particular, note that for the largest ODE tolerance, the computational time is almost as much as for the smallest.

7. CONCLUSIONS

Event- and self-triggering schemes attempt to regulate performance loss due to the growth in uncertainty as the system evolves in open-loop. Typically, uncertainty arising in the physical world may be at best unmodelled and at worst unstructured. However, uncertainty arising due to the solution of (P) is highly structured – therefore it should not be treated equivalently to, say, $w(t)$. This is the first work that has considered using the accuracy of the solution of an optimal control problem as a metric for triggering times. We have considered the use of approximate system predictions arising through numerical direct collocation solutions of OCPs in event-triggered control schemes. A commonly used metric for refining the state approximations has allowed us to guarantee a minimum inter-update time $\underline{\tau}^u$ of the ETC. If additionally the direct collocation software returns the approximation error metric as part of the solution, we may achieve a better estimate of the inter-update interval. Further work may consider the use of path constraint violation as a metric for mesh refinement and/or triggering.

REFERENCES

Anta, A. and Tabuada, P. (2010). To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9), 2030–2042.

Åström, K.J. and Bernhardsson, B. (1999). Comparison of periodic and event based sampling for first-order stochastic systems. *IFAC Proceedings Volumes*, 32(2), 5006–5011.

Betts, J.T. (2010). *Practical methods for optimal control and estimation using nonlinear programming*, volume 19. Siam.

Borgers, D.N. and Heemels, W.M. (2014). Event-separation properties of event-triggered control systems. *IEEE Transactions on Automatic Control*, 59(10), 2644–2656.

Fahroo, F. and Ross, I.M. (2008). Advances in pseudospectral methods for optimal control. In *AIAA guidance, navigation and control conference and exhibit*, 7309.

Gommans, T., Antunes, D., Donkers, T., Tabuada, P., and Heemels, M. (2014). Self-triggered linear quadratic control. *Automatica*, 50(4), 1279–1287.

Heemels, W., Johansson, K.H., and Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 3270–3285. IEEE.

Kelly, M. (2017). An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4), 849–904.

Li, H. and Shi, Y. (2014). Event-triggered robust model predictive control of continuous-time nonlinear systems. *Automatica*, 50(5), 1507–1513.

Lucia, S., Kögel, M., Zometa, P., Quevedo, D.E., and Findeisen, R. (2016). Predictive control, embedded cyber-physical systems and systems of systems–a perspective. *Annual Reviews in Control*, 41, 193–207.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Nie, Y., Faqir, O., and Kerrigan, E.C. (2018). ICLOCS2: Try this optimal control problem solver before you try the rest. In *2018 UKACC 12th International Conference on Control (CONTROL)*, 336–336. IEEE.

Paiva, L.T. and Fontes, F.A. (2019). Sampled–data model predictive control: Adaptive time–mesh refinement algorithms and guarantees of stability. *Discrete & Continuous Dynamical Systems-B*, 24(5), 2335–2364.

Varutti, P., Kern, B., Faulwasser, T., and Findeisen, R. (2009). Event-based model predictive control for networked control systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 567–572. IEEE.

Velasco, M., Fuertes, J., and Marti, P. (2003). The self triggered task model for real-time control systems. In *Work-in-Progress Session of the 24th IEEE Real-Time Systems Symposium (RTSS03)*, volume 384.