Receding-Horizon Control and Scheduling of Systems with Uncertain Computation and Communication Delays

Sanad Al-Areqi, Daniel Görges, and Steven Liu

Abstract—This paper addresses robust control and scheduling codesign for networked embedded control systems (NECS) with uncertain but interval-bounded time-varying computation and communication delays. The NECS is modeled as a discrete-time switched linear system with polytopic uncertainty. A robust receding-horizon control and scheduling problem with a quadratic performance criterion is introduced and solved based on the concept of (relaxed) dynamic programming. Closed-loop stability is guaranteed a priori by imposing stability constraints formulated as linear matrix inequalities. The effectiveness of the proposed modeling and synthesis methods is evaluated for networked embedded control of a set of pendulums. Notably, the proposed strategy is generally applicable to discrete-time switched linear systems with polytopic uncertainty.

I. INTRODUCTION

Networked embedded control systems (NECSs) consist of controllers implemented on embedded processors and connected with sensors and actuators via a communication network. Such structures occur in various applications like vehicles, power systems, manufacturing and process control. Advantages include reduced wiring and maintenance costs as well as an increased flexibility and reconfigurability. Challenges result from computation and communication delays, packet loss, and medium access constraints due to the shared computation and communication resources. Therefore, both a robust control against the delays and an intelligent scheduling against the access constraints are indispensable.

Various approaches for joint control and scheduling design (codesign) have been proposed in recent years using offline scheduling, e.g. [1], [2], or using online scheduling, e.g. [3], [4], [5]. The approaches assume constant computation and transmission delays which are ensured by most real-time computation and communication systems but not guaranteed by many general-purpose computation and communication systems. Control and scheduling codesign strategies with robustness against uncertain time-varying computation and communication delays are therefore clearly required.

Recently, robust control and scheduling codesign for NECSs with uncertain but interval-bounded time-varying computation and communication delays has been addressed in [6]. The NECS is modeled as a discrete-time switched linear system with polytopic uncertainty where switching is employed to represent scheduling and the polytopic uncertainty to overapproximatively represent the uncertain time-varying computation and transmission delays. A periodic control and online scheduling (PCS\textsubscript{on}) strategy is proposed for robust control and scheduling codesign. Closed-loop stability is guaranteed inherently under the PCS\textsubscript{on} strategy. However, the computational complexity for determining the PCS\textsubscript{on} strategy can be critical. On the other hand, a (relaxed) receding-horizon control and scheduling ((R)RHCS) strategy which allows adjusting the computational complexity in a flexible way has been presented in [5], [7]. However, uncertainties have not been considered there.

In this paper a novel robust receding-horizon control and scheduling (RoRHCS) strategy for NECSs with uncertain but interval-bounded time-varying computation and communication delays is presented, extending and combining ideas from [5], [7] and [6]. The NECS is modeled as a discrete-time switched linear system with polytopic uncertainty as shown in Section II. Similar models have been considered in [8]. The RoRHCS strategy is presented in Sections III to V. The RoRHCS problem is formalized first and then solved based on dynamic programming. Relaxed dynamic programming is introduced to reduce the computational complexity, leading to a robust relaxed receding-horizon control and scheduling (RoRRHCS) strategy. An important contribution of this paper consists in incorporating polytopic uncertainties into (relaxed) dynamic programming. Another important contribution of this paper consists in guaranteeing the stability of NECSs under the Ro(R)RHCS strategy a priori, i.e. before solving the optimization problem, via appropriate stability conditions. All conditions can be formulated as linear matrix inequalities (LMIs) which can be solved efficiently. The effectiveness of the methods is illustrated for networked embedded control of a set of pendulums in Section VI. Conclusions are given in Section VII.

The following notation is used: $Q > 0$ ($\geq 0$) indicates a symmetric positive definite (positive semidefinite) matrix. Furthermore, a matrix $(A, B, C)$ represents a symmetric matrix $(A^T, B, C)$. Finally, the cardinality of a set $L$ is denoted as $|L|$.

II. MODELING

A. NECS Architecture

Consider the NECS shown in Fig. 1. The NECS consists of a set of plants $P = \{P_i, i = 1, \ldots, N\}$ which are controlled by a set of control tasks $T = \{T_i, i = 1, \ldots, N\}$ over a shared communication network using a single processor. Each plant $P_i$ together with the actuator $A_i$ and the sensor $S_i$ is described by a continuous-time state equation

$$\dot{x}_{ci}(t) = A_{ci}x_{ci}(t) + B_{ci}u_i(t - \tau_i(t)), \quad x_{ci}(0) = x_{ci0} \quad (1)$$
where \( A_{ci} \in \mathbb{R}^{n_i \times n_i} \) is the system matrix, \( B_{ci} \in \mathbb{R}^{n_i \times m_i} \) is the input matrix, \( x_{ci}(t) \in \mathbb{R}^{n_i} \) is the state vector and \( u_i(t - \tau_i(t)) \in \mathbb{R}^{m_i} \) is the control vector with uncertain time-varying input delay \( \tau_i(t) \in \mathbb{I}_c = [\underline{\tau}_i, \overline{\tau}_i] \) with known lower bound \( \underline{\tau}_i \) and upper bound \( \overline{\tau}_i \). The input delay \( \tau_i(t) \) includes the computation delay \( \tau_c(t) \) and the communication delays \( \tau_{Sc}(t) \) and \( \tau_{Ca}(t) \), i.e. \( \tau_i(t) = \tau_{Sc}(t) + \tau_c(t) + \tau_{Ca}(t) \). Each plant \( P_i \) is further assigned a continuous-time cost function

\[
J_i = \int_0^\infty \begin{pmatrix} x_{ci}(t) \\ u_i(t - \tau_i(t)) \end{pmatrix}^T \begin{pmatrix} Q_{ci} & 0 \\ 0 & R_{ci} \end{pmatrix} \begin{pmatrix} x_{ci}(t) \\ u_i(t - \tau_i(t)) \end{pmatrix} dt
\]

with the symmetric and positive definite weighting matrices \( Q_{ci} \in \mathbb{R}^{n_i \times n_i} \) and \( R_{ci} \in \mathbb{R}^{m_i \times m_i} \). The main objective consists in minimizing the overall cost \( J = \sum_{i=1}^{N} J_i \) robustly with respect to the uncertain time-varying input delay.

The access to the network is decided by a medium access control (MAC) which is predefined by the utilized network. The access to the processor is determined by a scheduler (S) which is realized on the processor. The scheduling decision is taken based on the states. The states are sampled at time instant \( t_k \) and transmitted to the scheduler. The scheduler then determines the control task \( T_{ij}(k) \) to be executed from a state feedback switching law \( j(k) : \mathbb{R}^n \rightarrow \mathcal{J} = \{1, \ldots, N\} \) with \( n = n_1 + \cdots + n_N \). The design of the switching law will be detailed in Sections III to V. The computed control vector \( u_{ij}(k)(t) \) is finally transmitted to plant \( P_{j}(k) \) and updated at time instant \( t_k + \tau_{ij}(k)(t) \). For the other plants \( P_i \neq P_{j}(k) \) the control vector is not updated.

The sampling period \( h(k) \) is determined by the upper bound of the input delay \( \tau_{ij}(k) \), i.e. \( h(k) = \tau_{ij}(k) \). Because \( \tau_{ij}(k) \) can be different for each task \( T_{ij}(k) \), the sampling period \( h(k) \) is generally time-varying but certain. The NECS timing is illustrated in Fig. 2. For notational convenience, the time dependency indication of the switching index \( j(k) \) will be omitted in the following when no ambiguity arises.

**Remark 1:** If \( \tau_{ij}(k) < \tau_{ij}(k) \), the network and processor are idle and can be utilized for other non-control tasks.

**Remark 2:** Taking the scheduling decision based on the states certainly induces considerable scheduling overhead. Reducing this overhead will be addressed in future work. The main factor determining the overhead is the communication delay due to sending the states of all plants for decision making which is, however, not necessary in ECSs where \( \tau_{Sc}(t) = \tau_{Ca}(t) = 0 \) and NECSs where sensors and controllers are not connected over a network, i.e. \( \tau_{Sc}(t) = 0 \).

### B. Discretization

For discretizing the continuous-time state equation \( (1) \) using zero order hold (ZOH), the task \( T_i \) considered for discretization and the task \( T_j \) executed within the sampling interval \( t_k \leq t < t_{k+1} \) must be distinguished. If the considered task is the executed task \( (T_i = T_j) \), then the control signal is updated, i.e.

\[
u_i(t) = \begin{cases} u_i(t_{k-1}) & \text{for } t_k \leq t < t_k + \tau_i(k) \\ u_i(t_k) & \text{for } t_k + \tau_i(k) \leq t < t_{k+1} \end{cases}
\]

with \( h(k) = t_{k+1} - t_k + \tau_i(k) \) and \( \tau_i(k) \leq h(k) \). For \( (T_i \neq T_j) \), the control signal is not updated, i.e.

\[
u_i(t) = u_i(t_{k-1}) \quad \text{for } t_k \leq t < t_{k+1}.
\]

In the following, the distinction between the two cases mentioned above is represented by the logical variable

\[
\delta_{ij} = \begin{cases} 1 & \text{if } T_i = T_j \\ 0 & \text{if } T_i \neq T_j \end{cases}
\]

Considering this specific ZOH, an augmented discrete-time state equation corresponding to \( (1) \) can be formulated as

\[
x_i(k+1) = A_{ij}(k)x_i(k) + B_{ij}(k)u_i(k) \quad (6)
\]

with

\[
x_i(k) = \begin{pmatrix} x_{i,j}(k) \\ u_i^T(k-1) \end{pmatrix}^T \in \mathbb{R}^{n_i + m_i}
\]

\[
A_{ij}(k) = \begin{pmatrix} \Phi_i(h(k)) & \Gamma_i(h(k)) - \Gamma_i(h(k) - \delta_{ij}(k)) \\ 0_{m_i \times n_i} & 0_{m_i \times m_i} \end{pmatrix}
\]

\[
B_{ij}(k) = \begin{pmatrix} \Gamma_i(h(k) - \delta_{ij}(k)) \\ \delta_{ij}I_{m_i \times m_i} \end{pmatrix} \in \mathbb{R}^{(n_i + m_i) \times m_i}
\]

\[
\delta_{ij} = \delta_{ij}(k) + (1 - \delta_{ij})h(k)
\]

where \( \Phi_i(t) = e^{A_{ij}t} \) and \( \Gamma_i = \int_0^t \Phi_i(s)dsB_{ij}(k) \). The specific ZOH is illustrated in Fig. 2. Note that the control signal \( u_{ij}(k)(t) \) is held over the current and subsequent sampling interval and then reset unless the control task \( T_{ij}(k) \) is executed again in the subsequent sampling interval. Using a block-diagonal structure, cf. [6], the overall system can be written as a discrete-time switched linear system

\[
x(k+1) = A_j(k)x(k) + B_j(k)u(k).
\]

\[
\int_0^\infty \begin{pmatrix} x_{ci}(t) \\ u_i(t - \tau_i(t)) \end{pmatrix}^T \begin{pmatrix} Q_{ci} & 0 \\ 0 & R_{ci} \end{pmatrix} \begin{pmatrix} x_{ci}(t) \\ u_i(t - \tau_i(t)) \end{pmatrix} dt
\]
The discretized cost function associated with plant $P_i$ for a time-varying sampling period $h(k)$ and an uncertain time-varying time delay $\tau_j(k) \in \mathcal{I}_j$ is then given by

$$J_i = \sum_{k=0}^{\infty} \left( \begin{array}{c} x_i(k) \\ u_i(k-1) \end{array} \right)^T \begin{pmatrix} Q_{11ij} & Q_{12ij} & Q_{13ij} \\ * & Q_{22ij} & * \\ * & * & Q_{33ij} \end{pmatrix} \left( \begin{array}{c} x_i(k) \\ u_i(k-1) \end{array} \right)$$

(8)

with the discretized weighting matrices as defined in [6]. The overall cost function can be written, cf. [6], as the sum of the individual cost functions

$$J = \sum_{i=0}^{N} J_i = \sum_{k=0}^{\infty} \left( \begin{array}{c} x(k) \\ u(k) \end{array} \right)^T \begin{pmatrix} Q(k) + H(k) \\ Q(k) \end{pmatrix} \left( \begin{array}{c} x(k) \\ u(k) \end{array} \right)$$

(9)

Note that in case $\delta_{ij} = 1$ the uncertain time-varying time delay $\tau_j(k) \in \mathcal{I}_j$ affects the system and input matrices in (6) as well as the weighting matrices in (8) in a nonlinear manner. This nonlinearity is handled using a polytopic overapproximation. Various methods have been proposed during the last years to obtain a polytopic overapproximation, see [9] for a recent survey and an assessment of the complexity and conservatism of the proposed methods. Due to space limitations the polytopic formulation is not further addressed.

The important point is that the switched system matrices $A_j(k), B_j(k)$ and the switched weighting matrix $Q_j(k)$ finally belong to the convex hulls

$$(A_j(k), B_j(k)) \in \Omega_j = \text{Co}\{(A_{j1}, B_{j1}), \ldots, (A_{jM}, B_{jM})\}$$

$$Q_j(k) \in \Psi_j = \text{Co}\{Q_{j1}, \ldots, Q_{jM}\}$$

where $(A_{jm}, B_{jm})$ and $Q_{jm}$ with $m \in \mathcal{M} = \{1, \ldots, M\}$ represent the vertices of the corresponding convex polytope.

Minimizing the infinite-horizon cost function (9) subject to the switched system (7) is in general a very difficult problem. Solutions are only known for special cases [6], [10]. For instance, in [6] the solution is found by imposing periodicity on the switching sequence. Alternative to the method proposed in [6], a receding horizon can be utilized as an approximation of an infinite horizon without imposing periodicity. This is addressed in the following sections.

III. PROBLEM FORMULATION

Consider the finite-horizon cost function with polytopic uncertainty

$$J_H(k) = x^T(k + H)Q_0x(k + H) + \sum_{i=1}^{H-1} \ell(k + i)$$

(10)

with the step cost $\ell(i) = \left( \begin{array}{c} x(i) \\ u(i) \end{array} \right)^T Q_j(i) \left( \begin{array}{c} x(i) \\ u(i) \end{array} \right) \geq 0$, the switched weighting matrix $Q_j(i) > 0$ defined in (9), and a terminal weighting matrix $Q_0 \succeq 0$. $H$ indicates the prediction horizon and $i$ indicates the time instant within the prediction horizon.

**Problem 3:** For the switched system (7) and the current state $x(k)$ find a control sequence $u^*(k), \ldots, u^*(k + H - 1)$ and a switching sequence $j^*(k), \ldots, j^*(k + H - 1)$ such that the cost function (10) is robustly minimized, i.e.

$$\min_{u(k), \ldots, u(k + H - 1)} \max_{\Omega_j(k), \ldots, \Omega_j(k + H - 1), \Psi_j(k), \ldots, \Psi_j(k + H - 1)} J_H(k)$$

subject to (7).

The first element $u^*(k)$ of the control sequence and the first element $j^*(k)$ of the switching sequence are then applied at the current time instant $k$. Repeating this procedure at each time instant results in a robust receding-horizon control and scheduling (RoRHCS) strategy.

Problem 3 requires the solution of a robust finite-horizon control and scheduling (RoFHCS) problem at each time instant $k$. The RoFHCS problem remains unchanged over time. Only the current state $x(k)$ changes. Therefore, it is sufficient to determine the solution parameterized in $x(k)$ for $k = 0$ only and adapt the current state at each time instant $k$. The RoFHCS problem is solved based on dynamic programming by determining the worst-case cost among the polytopic uncertainty. The resulting worst-case cost is then minimized by finding an optimal control sequence and switching sequence as shown in the following section.

IV. FINITE-HORIZON CONTROL AND SCHEDULING

A. Dynamic Programming Solution

Consider the iterated optimization problem

$$\min_{K_{l(i)}(i), P_{l(i)}(i)} \text{tr}(P_{l(i)})$$

subject to

$$A_{l(i)m}(i)P_{l(i+1)m}(i) + \left( \begin{array}{c} I \\ K_{l(i)}^T \end{array} \right) Q_{j(i)m} \left( \begin{array}{c} I \\ K_{l(i)} \end{array} \right) < P_{l(i)}$$

for all $m \in \mathcal{M}$ with $A_{l(i)m} = (A_{j(i)m} + B_{j(i)m}K_{l(i)})$, the terminal condition $P_{l(H)} = Q_0$ and $\text{tr}(\cdot)$ denoting the trace. The controller index $l(i) \in \mathcal{L}_i = \{1, \ldots, N^{H-1}\}$ indexes the solutions $P_{l(i)}$ and the corresponding switched feedback matrices $K_{l(i)}$ at time instant $i$. The matrices $P_{l(i)}$ result from the matrices $P_{l(i+1)}$ by evaluating (11) for all combinations of the current switching index $j(i) \in \mathcal{J}$ and the subsequent controller index $l(i + 1) \in \mathcal{L}_{i+1}$. This is illustrated by the switching tree in Fig. 3.

**Theorem 4:** Problem 3 is solved for $k = 0$ with minimum worst-case cost

$$J_H^* = \min_{l(0) \in \mathcal{L}_0} x^T(0)P_{l(0)}x(0)$$

(12)
by the optimal controller sequence starting with
\[ l^* (0) = \arg \min_{l(0) \in L_0} x^T (0) P_{l(0)} x(0), \] (13)
the optimal switching sequence starting with
\[ j^* (0) = \sigma(l^* (0)) \] (14)
and the optimal full state feedback control law
\[ u^*(i) = K_{l^*(i)} x(i). \] (15)

The remaining controller sequence \( l^*(i) \) and switching sequence \( j^*(i) = \sigma(l^*(i)) \) for \( i = 1, \ldots, H - 1 \) follow from iterating the switching tree backwards, where the switching function \( \sigma : L \rightarrow \mathbb{J} \) follows by construction.

**Proof:** Introduce the iterated value function
\[ V^*_H(i) = \min_{u(i)} \max_{j(i)} \{ \ell(i) + V^*_H(i+1)(i+1) \} \] (16)
with the terminal condition \( V^*_H(H) = x^T(H) Q_{0} x(H) \). Equation (16) describes the minimum worst-case cost-to-go from \( i \) to \( H \) with the subindex \( H - i \) specifying the steps-to-go. Following the principle of dynamic programming, the value function is iterated backwards from \( i = H \) to \( i = 0 \). For the last optimization step \( i = H \) the value function is
\[ V^*_H(H) = x^T(H) P_{l(H)} x(H) \] (17)
with \( P_{l(H)} = Q_{0} \) and \( l(H) \in L_H = \{1\} \).

For the second to last step \( i = H - 1 \) we have
\[ V^*_H(H - 1) = \min_{u(i)} \max_{j=(H-1)} \{ \ell(H-1) + V^*_H(H) \}. \] (18)
Substituting (17) and the state equation (7) with the state feedback control law (15) into (18) yields
\[ V^*_H(H - 1) = \min_{u(i)} \max_{j=(H-1)} \{ \ell(H-1) + V^*_H(H) \} \]
\[ = \kappa_{l(H-1)} \max_{j=(H-1)} x^T(H-1) \left( \tilde{Q}_{l(H-1)} + \tilde{P}_{l(H-1)} \right) x(H-1) \]
\[ < \min_{u(i)} \max_{j=(H-1)} x^T(H-1) P_{l(H-1)} x(H-1) \]
\[ = l(H-1) \in L_{H-1} \]
\[ \mathbf{A}_{l(H-1)} x(H-1) \] (19)

Equation (19) deserves some interpretation. The maximization with respect to \( \Omega_j(H - 1) \) and \( \Psi_j(H - 1) \) corresponds due to convexity to the maximization with respect to the vertices \( 
\tilde{P}_{l(H-1)m} \) and \( \tilde{Q}_{l(H-1)m} \), yielding the second equality in (19). This maximization can be resolved by introducing an upper bound characterized by the matrix \( P_{l(H-1)} \), yielding the inequality in (19). Requiring fulfillment of this inequality in (19) for an arbitrary state vector \( x(H - 1) \) finally yields the inequality in (11). The minimization of the upper bound with respect to \( K_{l(H-1)} \) is regarded in (11) using the fact that the trace of a matrix corresponds to the expected value of the associated quadratic form as shown in [11, p. 338]. The minimization of the upper bound with respect to \( J^*_H \) is regarded in (11) via the controller index \( l(H-1) \in L_{H-1} = \{1, \ldots, N\} \) which establishes an explicit enumeration of all \( j(H-1) \in \mathbb{J} \), yielding the last equality in (19).

For \( i = H - 2 \) the value function corresponds to (18) with shifted time instant. Continuing the backward iteration finally leads for \( i = 0 \) to (12) which completes the proof. \( \blacksquare \)

**Remark 5:** An equivalent LMI form of the optimization problem (11) is given by
\[ \begin{bmatrix} G_{l(i)}^T + G_{l(i)} - Z_{l(i)} & * & * \\ Q_{j(i)m}^1/2 \mathbf{A}_{j(i)m} G_{l(i)} & \mathbf{I} & * \end{bmatrix} > 0 \]
where \( P_{l(i)} = Z_{l(i)}^{-1} \) and \( K_{l(i)} = W_{l(i)} G_{l(i)}^{-1} \). For more details about the derivation of this LMI optimization problem see [6]. Note that this LMI form can also be adopted for the forthcoming Theorems.

**Remark 6:** The notation \( i(i) \in \mathbb{L}_i \) will be abbreviated by the notation \( l(i) \in L_i \) if no ambiguity arises.

**Remark 7:** The number of solutions \( P_{l(i)} \), given by the number of branches in the switching tree \( |\mathbb{L}_i| = N^{H-i} \), grows exponentially with the time horizon \( H \). Therefore, the solution rapidly becomes computationally intractable. In the following, relaxed dynamic programming proposed in [12] for general dynamic programming problems is utilized to reduce the offline and therewith online computational complexity associated with the RoFHCS problem.

### B. Relaxed Dynamic Programming Solution

The main idea of relaxed dynamic programming consists in changing the objective function in Problem 3 such that the resulting cost function \( J_H(k) \) is bounded by
\[ J^*_H(k) \leq J_H(k) \leq \alpha J^*_H(k) \] (21)
instead of finding the minimum worst-case cost value \( J^*_H(k) \) where \( \alpha \geq 1 \) is a relaxation factor. This can be performed through replacing the iterated value function (16) by the relaxed iterated value function
\[ V_{H-i}(i) \leq V_{H-i}(i) \leq V_{H-i}(i) \] (22)
with the lower and upper bound
\[ V_{H-i}(i) = \min_{u(i)} \max_{j(i)} \{ \ell(i) + V_{H-i}(i + 1) \} \] (23a)
\[ V_{H-i}(i) = \min_{u(i)} \max_{j(i)} \{ \alpha \ell(i) + V_{H-i}(i + 1) \} \] (23b)

2657
and the terminal condition \( V_0(H) = x^T(H)Q_0x(H) \). The relaxed value function satisfies
\[
V^{\ast}_{H-i}(i) \leq V_{H-i}(i) \leq V^i_{H-i}(i) \leq \alpha V^i_{H-i}(i) \tag{24}
\]
Thus, the gap between the suboptimal value and the optimal value is bounded. Considering Theorem 4, an appropriate parameterization of the relaxed iterated value function is
\[
V_{H-i}(i) = \min_{l \in \mathbb{L}_i} x^T(i) P_l x(i)
\]
with \( l \in \mathbb{L}_i \subseteq \mathbb{L}_t \) as small as possible. To determine the matrices \( P_l, l \in \mathbb{L}_i \), the lower and upper bounds of the relaxed value function are used. From Theorem 4, these are computed as
\[
V_{H-i}(i) = \min_{l \in \mathbb{L}_i} x^T(i) P_l x(i), \quad \nabla V_{H-i}(i) = \min_{l \in \mathbb{L}_i} x^T(i) \nabla P_l x(i)
\]
where \( P_l(i) \) and \( \nabla P_l(i) \) are obtained by solving the following optimization problems
\[
\min_{\mathbb{K}_l(i)} \text{tr}(P_l(i)) \quad \text{subject to} \quad (26a)
\]
\[
\hat{A}_j(i) P_l(i+1) \hat{A}_j(i) + (I K_j(i)) Q_{j(i)m} (I K_j(i))^T < P_l(i)
\]
\[
\min_{\mathbb{K}_l(i)} \text{tr}(P_l(i)) \quad \text{subject to} \quad (26b)
\]
\[
\bar{A}_j(i) P_l(i+1) \bar{A}_j(i) + (I K_j(i)) Q_{j(i)m} (I K_j(i))^T < P_l(i)
\]
for all \( m \in \mathbb{M} \) where \( \hat{A}_j(i) = (A_j(i)m + B_j(i)m K_j(i)) \), \( \bar{A}_j(i) = (A_j(i)m + B_j(i)m K_j(i)) \) with \( j(i) = \sigma(l(i)) \) and \( P_l(H) = Q_0 \). A simple approach to obtain matrices \( P_l \) such that the relaxed iterated value inequality (22) is satisfied then consists in using some of the lower bound solutions \( P_l \) as the matrices \( P_l \); see [7, Proposition 2.3 and Algorithm 1].

V. RECEDING-HORIZON CONTROL AND SCHEDULING

A. Explicit Solution

An explicit solution of Problem 3 can be formulated based on the finite-horizon solution. This can be summarized as

**Theorem 8:** The solution to Problem 3 is given by the piecewise linear (PWL) state feedback control law
\[
u^*(k) = K_t x(k) \quad \text{if} \quad x(k) \in X_t,
\]
and switching law
\[
j^*(k) = \sigma(l^*) \quad \text{if} \quad x(k) \in X_{l^*},
\]
where the regions \( X_t \) with \( \bigcup_{l \in \mathbb{L}_t} X_l = \mathbb{R}^n \) are described by
\[
X_{l^*} = \{ x(k) | x^T(k) P_l x(k) \leq x^T(k) P_{l^*} x(k) \forall l \in \mathbb{L}_l \}.
\]

**Proof:** According to Theorem 4 and in particular (13) the optimal controller index results for the current state \( x(k) \) as
\[
l^* = \arg \min_{l \in \mathbb{L}_t} x^T(k) P_l x(k).
\]
This corresponds to \( x^T(k) P_l x(k) \leq x^T(k) P_{l^*} x(k) \forall l \in \mathbb{L}_l \), which leads to the regions \( X_{l^*} \) and \( l^* \).

**Remark 9:** Substituting (27)/(28) into the switched system (7) leads to the autonomous PWL closed-loop system
\[
x(k+1) = \tilde{A}_l^i x(k) \quad \text{if} \quad x(k) \in X_{l^*},
\]
with \( \tilde{A}_l^i(k) = (A_j^i(k) + B_j^i(k) K_l^i) \).

**Remark 10:** The explicit PWL state feedback control and switching law (27)/(28) and PWL closed-loop system (29) can be formulated analogously using \( l \in \mathbb{L}_0 \) instead of \( l \in \mathbb{L}_t \), leading to a robust relaxed receding-horizon control and scheduling (RoRRHCS) strategy.

B. Stability

Global uniform asymptotic stability (GUAS) under the Ro(R)RHCS strategy is not guaranteed inherently. By adjusting the terminal weighting matrix \( Q_0 \) in the cost function (10) or the lower/upper bound (23), GUAS is guaranteed a priori.

1) Closed-Loop Stability under the RoRHCS Strategy

**Theorem 11:** If \( Q_0 > 0 \) in (10) is chosen such that
\[
\tilde{A}_{j(m)}^T Q_0 \tilde{A}_{j(m)} + \left( I K_j \right)^T Q_{j(m)} \left( I K_j \right) - Q_0 < 0 \tag{30}
\]
for all \( j \in \mathbb{J}, m \in \mathbb{M} \) with \( \tilde{A}_{j(m)} = (A_{j(m)} + B_{j(m)} K_j) \) and an arbitrary switched feedback matrix \( K_j \), then the switched system (7) under the RoRHCS strategy is GUAS.

**Proof:** The proof is based on using the value function
\[
V_{H}^i(k) = \min_{u(k)} \max_{\mathbb{K}_l(i)} \text{tr}(P_l(i)) \quad \text{subject to} \quad (31a)
\]
\[
\tilde{A}_j(i) P_l(i+1) \tilde{A}_j(i) - P_l(i) < 0
\]
\[
\bar{A}_j(i) P_l(i+1) \bar{A}_j(i) - P_l(i) < 0
\]
with \( \tilde{A}_j(i) = (A_j(i)m + B_j(i)m K_j) \), \( \bar{A}_j(i) = (A_j(i)m + B_j(i)m K_j) \) as a candidate Lyapunov function, cf. [7, Sec. III-A]. The value function is positive definite, decrescent and radially unbounded. Furthermore, the difference of the value function \( V_{H}^i(k) \) is negative definite if (30) is satisfied.

**Remark 12:** For the existence of a \( Q_0 > 0 \) and arbitrary \( K_j \) satisfying (30) the subsystems \( (A_j, B_j) \) of the switched system (7) must be stabilizable for all \( j \in \mathbb{J} \). Since only one plant can be connected with its controller at a time in an NECS, the stabilizability can not be guaranteed unless all subsystems are asymptotically stable. Hence, for (30) the asymptotic stability of all subsystems is necessary.

2) Closed-Loop Stability under the RoRRHCS Strategy

GUAS of the PWL closed-loop system (29) is guaranteed by adding the stability constraints
\[
\tilde{A}_j(i) P_l(i) \tilde{A}_j(i) - P_l(i) < 0 \quad \text{and} \quad \bar{A}_j(i) P_l(i) \bar{A}_j(i) - P_l(i) < 0
\]
to the optimization problems (26) which notably may increase the suboptimality bound given in (24). The relaxed value function \( V_{H}^i(k) = \min_{l \in \mathbb{L}_t} x^T(k) P_l x(k) \) with \( \mathbb{L}_0 \subseteq \mathbb{L}_0 \) can then be considered as Lyapunov function, proving GUAS of the PWL closed-loop system (29).

VI. EXAMPLE

Consider an NECS comprising three moving pendulums. The objective is to actively damp the pendulums. The linearized model of each pendulum \( P_i \), is given by
\[
\hat{\phi}_i(t) = \frac{1}{M_i\ell_i} \left( \frac{0}{\omega_{n_i}^2 M_i\ell_i} \right) \hat{\phi}_i(t) + \left( 0 \right) u_i(t - \tau_i)
\]
with \( u_i(t - \tau_i) = F_i(t - \tau_i) + F_{di}(t) \) where \( \phi_i \) is the pendulum angle, \( F_i \) is the force acting on the cart, \( F_{di} \) is a force disturbance and \( i = 1, 2, 3 \). The moving pendulums have the same pendulum mass \( m_i = 0.3 \text{ kg} \), cart mass \( M_i = 0.3 \text{ kg} \) and damping coefficient \( b_i = 0.1 \text{ kg/s} \), but different pendulum length \( l_i/2/3 = 0.136/0.242/0.545 \text{ m} \), leading to different natural frequencies \( \omega_i/2/3 = 12/9/6 \text{ s}^{-1} \). Moreover, \( g = 9.81 \text{ m/s}^2 \) is the gravitational acceleration. The input delay \( \tau_i \) is bounded on the interval \( I_i = [2.5, 3] \text{ ms/}\forall i \). Consider further the quadratic cost function (2) with the weighting matrices \( Q_{ci} = \text{diag}(100, 0.1) \) and \( R_{ci} = 0.1 \).

First, we determine \( Q_0 \) satisfying (30) by formulating the stability condition (30) into an equivalent LMI, see Remark 5. Then, by setting \( P_{i(I)} = Q_0 \) and prediction horizon \( H = 8 \), we solve the iterated optimization problem described in Theorem 4 and store the feedback matrices \( K_{i(I)} \) and the regions \( P_{i(I)} \). Finally, we apply online the RoRHS strategy proposed in Theorem 8. The results for random impulsive force disturbances \( F_{di} \in [-150, 150] \text{ N} \) and uniformly distributed random input delays \( \tau_i \in I_i \) are shown in Fig. 4. Obviously, a control task \( T_1 \) is executed preferably when the related pendulum \( P_i \) is subject to an impulsive force disturbance. E.g., \( T_1 \) is executed preferably for \( t \in [0, 0.3] \text{ s} \) since only \( P_1 \) is subject to an impulsive disturbance. The same behavior results for \( T_3 \) for \( t \in [0.3, 0.6] \text{ s} \) and for \( T_3 \) for \( t \in [0.6, 1.0] \text{ s} \). Computation and communication resources are thus distributed according to the needs of the pendulums.

A comparison between the Ro(R)RHS codesign strategies, periodic control and online scheduling (PCS_{on}) proposed in [6], and optimal pointer placement (OPP) proposed in [4] is summarized in Table I. Evidently the Ro(R)RHS strategy outperforms the others. However, no conclusion can be made about the superiority of the Ro(R)RHS strategy over the PCS_{on} strategy since a formal cost relation between these strategies is not known. Concerning the computational complexity (measured by \( |L_{o}| \) for the RoRHS strategy and by the number of admissible \( p \)-periodic schedules \( |P_{p,adm} | \) for the PCS_{on} strategy), the online complexity under the PCS_{on} strategy can be reduced by the OPP strategy as described in [6] while no systematic way is known yet to reduce the associated offline complexity. On the other hand, both the online and offline complexity of the RoRHS strategy can be adjusted by the relaxation factor \( \alpha \), leading to the RoRRHCS strategy at the expense of introducing suboptimality. Overall, the RoRRHCS strategy provides a systematic way to balance between complexity and performance.

VII. CONCLUSIONS AND FUTURE WORK

In this paper robust control and scheduling codesign for NECs based on a discrete-time switched system model with polytopic uncertainty and the receding horizon concept is studied. Future work will focus on reducing the scheduling overhead for NECs due to the dependency of the switching law on the states of the whole system and removing the necessary assumption that all subsystems must be asymptotically stable in order to satisfy the stability condition (30).

REFERENCES


