Adaptive Model Predictive Control in the IPA-SQP Framework

Jing Sun, Hyeongjun Park, Ilya Kolmanovsky, and Richard Choroszucha

Abstract—In this paper, we propose an approach and a specific algorithm to integrate a parameter estimation with the receding horizon model predictive control. We derive this adaptive MPC algorithm based on the integrated perturbation analysis and sequential quadratic programming (IPA-SQP) framework. Previously, this approach was exploited for repeated constrained optimization in MPC when the initial conditions change. It is now shown that a similar algorithm can be derived to perform MPC updates when model parameters change. The detailed algorithm derivation is presented, along with discussions on the performance and implementation. An example based on the nonlinear dynamics of an inverted pendulum on a cart is included to demonstrate the effectiveness of the proposed algorithm.

I. INTRODUCTION

Compared to a substantial amount of research on conventional MPC, progress on adaptive MPC (AMPC) for systems with constraints has been limited. AMPC has been identified as a vital research direction in the plenary talk by M. Morari [1].

There are several challenges in the development of rigorous approaches to AMPC and in associated computations. In particular, the “separation principle” used for most certainty equivalence designs in indirect adaptive control may not hold [2, 3]. In addition, the control law does not have a closed-form solution and has to be computed numerically. Furthermore, since the parameter identification is intertwined with the feedback loop, the closed-loop performance of AMPC depends on both estimation and optimization, as well as their interactions. Finally, the predictive constraint enforcement based on an adaptive model requires special care.

Early developments in AMPC have been in the area of the generalized predictive control (GPC) [4–7], for which general state and output constraints have not been treated. More recent work on AMPC includes, for example, linear AMPC considered in [8], and AMPC using multiple linear models in [9–12]. For nonlinear systems, the development of AMPC solutions has generally proceeded in two directions. Several of the approaches rely on the argument of parameter convergence to apply the “separation principle.” For example, [13] shows that under appropriate assumptions, there must exist a finite time period when persistent excitation condition is satisfied (during which period the trajectories are assumed to be bounded) and therefore parameter convergence is achieved. In [8], additional constraints are included to ensure persistent excitation for online parameter identification. In [14], a different approach is taken to design an Input-to-State Stable (ISS) Control Lyapunov Function to provide robust stabilization for the closed-loop system without adaptation, and at the same time to ensure asymptotic stability with parameter adaptation for a class of unconstrained nonlinear systems. Another approach to integrating the adaptation into MPC is by ensuring sufficient “robustness” while the estimation is still in progress. This is the approach taken, for instance, by [2, 15–17], where a set valued description of parameter uncertainty is adapted, and a robust (min-max) MPC is implemented for the identified parameter set. While this approach can deal with constraints, the underlying min-max feedback MPC problem can be computationally prohibitive to solve, especially in real-time.

The approach pursued in this paper is motivated by the needs to integrate optimization-based control solutions with online parameter identification algorithms for hybrid propulsion systems arising in marine, aerospace and automotive domains. These systems involve multiple heterogeneous power plant and load networks, have stringent safety and self-sustainability requirements, and are expected to operate in a wide range of conditions over their long life cycle. Moreover, they require fast sampling given their fast dynamics, yet they have limited onboard memory and computing power due to cost and space constraints. Consequently, efficient AMPC algorithms are necessary to handle real-time optimization, physical and operational constraints, and online identification and adaptation.

Aimed at developing efficient algorithms to integrate adaptation and optimization, in this paper we propose an indirect AMPC algorithm based on the integrated perturbation analysis and sequential quadratic programming (IPA-SQP) [18] framework. Exploring the structure of the neighboring extremal solution, we treat parameter updates, together with updated initial conditions as the states are evolving, as perturbations in the MPC problem formulation and use IPA-SQP framework to derive an algorithm for fast computation/update of the control sequence.

The remainder of the paper is organized as follows: Problem formulation is given in Section II, together with a brief introduction to IPA-SQP. In Section III, we present the AMPC algorithm derived within IPA-SQP framework to facilitate fast updates of the control sequence when model parameters change. Along with the derivation of the algorithm, some properties of the proposed AMPC are established. An example of applying the algorithm to an inverted pendulum on a cart is presented in Section IV, followed by conclusions in Section V.
II. Problem Formulation and IPA-SQP Framework

A. Problem Formulation

Consider a typical adaptive control problem where the system state equations and constraints are described by

\[ x(k + 1) = f(x(k), u(k), \theta), \]

and

\[ C(x(k), u(k), \theta) \leq 0, \]

with \( x \in \mathbb{R}^n, u \in \mathbb{R}^m, \theta \in \mathbb{R}^p \) being the state, input, and parameter vectors, respectively. The functions \( f(\cdot, \cdot, \cdot) \) and \( C(\cdot, \cdot, \cdot) \) are assumed to be twice continuously differentiable with respect to all of their arguments. The optimization problem \( L(\cdot, \cdot, \cdot) \) and terminal cost function \( \Phi(\cdot) \) are assumed to be twice continuously differentiable with respect to all of their arguments. Note that both system dynamics and constraints can depend on plant parameter, \( \theta \), which capture changes and uncertainties in the physical system and in the constraints. If the parameter \( \theta \) is known, a receding horizon optimization problem can be formulated to minimize a general cost function

\[ J(x(t), u(\cdot), \theta) = \sum_{k=t}^{t+N-1} L(x(k), u(k), \theta) + \Phi(x(t + N)) \]

with respect to the control sequence \( u(\cdot) \) (where \( u(\cdot) = \{u(t), u(t + 1), \ldots, u(t + N - 1)\} \) over the prediction horizon \([t, t + N]\), subject to the dynamic equation (1) and the mixed state and input constraints (2). Here \( t \) is the current time instant, and \( N \) is the prediction horizon. The control, prediction, and constraint horizons are assumed to be the same in this paper to simplify the exposition. The incremental cost function \( L(\cdot, \cdot, \cdot) \) and terminal cost function \( \Phi(\cdot) \) are assumed to be twice continuously differentiable with respect to all of their arguments. The optimization problem is solved repeatedly at each sampling time instant \( t \), with the state \( x(t) \) being updated, to calculate the control sequence; then \( u(t) \), the first element of the control sequence \( u(\cdot) \), is applied.

When the plant parameters \( \theta \) are unknown, they are assumed to be estimated with appropriately designed adaptation algorithms, leading to AMPC.

B. IPA-SQP Framework

We seek to develop computationally fast AMPC algorithms that can effectively integrate adaptation and constrained dynamic optimization. Towards this end, we extend to the adaptive case a recently developed computational framework for nonadaptive, nonlinear MPC, referred to as the Integrated Perturbation Analysis-Sequential Quadratic Programming (IPA-SQP), see [18, 19]. For a self-contained presentation, we include an overview of the IPA-SQP in this subsection.

The underlying mechanism behind the IPA-SQP approach to nonlinear MPC relies on a predictor-corrector solution to a constrained discrete-time optimal control problem that minimizes the cost function

\[ J[u(\cdot), x(t)] = \sum_{k=t}^{t+N-1} L(x(k), u(k)) + \Phi(x(t + N)), \]

subject to

\[ x(k + 1) = f(x(k), u(k)), \]

\[ x(t) = \text{current state of the system}, \]

and state and control constraints

\[ C(x(k), u(k)) \leq 0, \]

for all \( k \in [t, t + N - 1] \). Following the procedure of deriving a discrete-time minimum principle based on Karush-Kuhn-Tucker (KKT) conditions, we define the Hamiltonian as

\[ H(x(k), u(k), \lambda(k + 1), \mu(k)) = L(x(k), u(k)) + \lambda^T(k + 1)f(x(k), u(k)) + \mu^T(k)C(x(k), u(k)), \]

where \( \lambda \) and \( \mu \) denote the vectors of Lagrange multipliers. Under appropriate assumptions and given the initial condition \( x^*(t) \) at the time instant \( t \), the optimal state, control, and Lagrange multiplier sequences, denoted by \( x^*(\cdot), u^*(\cdot), \lambda^*(\cdot), \mu^*(\cdot) \), satisfy

\[ \nabla_u H(x^*(k), u^*(k), \lambda^*(k + 1), \mu^*(k)) = 0, \]

for all \( k \in [t, t + N - 1] \). If the initial state is different from \( x^*(t) \), i.e., \( x(t) = x^*(t) + \delta x_t \), and the set of active constraints remains the same, a linear correction to the optimal control sequence,

\[ \delta u(k) = K^*_1(k)\delta x(k), \]

can be constructed in such a way that the stationarity condition (8) is maintained to the first order for all \( k \in [t, t + N - 1] \) when \( x^*(\cdot), u^*(\cdot), \lambda^*(\cdot), \mu^*(\cdot) \) in (8) are replaced by \( x^*(\cdot) + \delta x^*(\cdot), u^*(\cdot) + \delta u^*(\cdot), \lambda^*(\cdot) + \delta \lambda^*(\cdot), \mu^*(\cdot) + \delta \mu^*(\cdot) \). The \( \delta x(k) \) used in (9) is generated by the perturbed system

\[ \delta x(k + 1) = f_x(k)\delta x(k) + f_u(k)\delta u(k), \]

for \( k \in [t, t + N - 1] \) with the initial condition \( \delta x(t) = \delta x_t \), where \( f_x, f_u \) are partial derivatives of \( f \) in (5) with respect to \( x, u \) along the optimal trajectory \( x^*(\cdot), u^*(\cdot) \). The matrix gain \( K^*_1(k) \) is constructed using a set of backward-in-time recursive relations resembling Riccati updates in finite-horizon discrete-time LQ problems.

Large state perturbations, which change the set of active constraints, are handled by breaking them down into smaller perturbations and repeatedly applying the above correction strategy. This is essentially the approach of Neighboring Extremals (NE) [20] (see also [21, 22] for several more recent extensions) that has been generalized to the case of discrete-time systems with state and control constraints in [18, 19, 23, 24].

For MPC applications, a modification of (9) is required since the nominal optimal sequences \( x^*(k), u^*(k), \lambda^*(k), \mu^*(k) \) are identified with the numerical (and possibly suboptimal) solution available from the previous time instant. In particular, (8) may not hold. The modified update compensates for the non-zero \( \nabla_u H(k) \) and has the following predictor-corrector form

\[ \delta u(k) = K^*_1(k)\delta x(k) + K^*_2(k)\nabla_u H(k), \]

with appropriately constructed gain matrices \( K^*_1(k), K^*_2(k) \) [18, 19]. Note that in [19], (10) is updated in multiple iterations for each sampling time step until a termination criterion in terms of \( \nabla_u H(k) \approx 0 \) is satisfied. This modification, when applied in multiple iterations, is equivalent to the sequential quadratic programming (SQP), and therefore (10) constitutes the IPA-SQP algorithm. Other approaches that exploit predictor-corrector type iterations for MPC have been reported in [25–28].
It is shown in [18] that (10) yields local quadratic convergence rates similar to the conventional SQP. At the same time, due to integration of the NE-based prediction, the computing time and effort are typically similar to the NE algorithm. For instance, they increase linearly rather than cubically in the horizon length, $N$. Examples considered in [18], [19] and references therein confirm that the method is computationally faster than the conventional SQP algorithm for discrete-time optimal control problems. In particular, a comprehensive computational benchmark analysis for a ship steering problem with a rather long prediction horizon ($N = 140$) was reported in [29], and substantial computation time reduction, without compromising performance, was demonstrated compared to SQP algorithm. Experimental results of IPA-SQP MPC were obtained at the sampling time of $100 \, \mu s$ for a full bridge DC/DC converter in [30].

III. INDIRECT ADAPTIVE MPC IN IPA-SQP FRAMEWORK

The IPA-SQP described in the previous section provides a general framework for efficient computation of optimal solutions when some of the problem parameters, such as the initial condition, are changed. This problem formulation naturally applies to AMPC where the parameters are updated through an online identification mechanism. In this section, we develop the AMPC algorithm using the IPA-SQP framework, and explore the properties of the resulting AMPC scheme.

We consider the indirect AMPC where the adaptive control scheme is designed using the certainty equivalence principle. Namely, the adaptation is performed by combining a separate estimation algorithm that provides an estimate of the matrices used in (18)-(20). Consider the “certainty equivalence” optimization problem defined by (11)-(13). The time-varying gains $K^*_1(k)$ and $K^*_\theta(k)$ are defined as:

$$K^*_1(k) = -[I, 0]K_0(k) \begin{bmatrix} Z_{21}(k) \\ C^a_\theta(k) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

$$K^*_\theta(k) = -[I, 0]K_0(k) \begin{bmatrix} Z_{23}(k) \\ C^a_\theta(k) \end{bmatrix} \in \mathbb{R}^{m \times p},$$

where $Z_{21}, i = 1, 2, 3$ are matrices calculated through a backward-in-time iterative procedure, as defined in the next subsection in the derivation of the algorithm.

Remark 3.1: It should be noted that the assumptions of $Z_{22}$ being positive definite and $C^a_\theta$ being full row rank are made for the AMPC algorithm. These assumptions are satisfied if local optimum is achieved at the time instant $(t - 1)$ and the constraints involve the control input $u$ explicitly. For state-only constraints where $C^a_\theta = 0$, $K_0$ in (18) and (19) will be singular and the algorithm cannot be applied directly. The constraint backpropagation algorithms presented in [31] can be used to modify the AMPC algorithm and avoid singularity in (20), at the expense of additional computational complexity.

B. Derivation of AMPC

We now formally derive the AMPC algorithm and define the matrices used in (18)-(20). Consider the “certainty equivalence” optimization problem defined by (11)-(13). The Hamiltonian for the optimization problem is defined as:

$$H(x, u, \lambda, \mu, \hat{\theta}_t) = L(x, u, \hat{\theta}_t) + \lambda^T(k + 1)f(x, u, \hat{\theta}_t) + \mu^T C^a_\theta(x, u, \hat{\theta}_t).$$

with $\lambda$ and $\mu$ denoting the Lagrange multipliers, and the time argument $k$ for all variables is omitted except for $\lambda(k + 1)$. Let $x^*(\cdot) = x_{t-1}(\cdot)$, $u^*(\cdot) = u_{t-1}(\cdot)$, $\lambda^*(\cdot) = \lambda_{t-1}(\cdot)$, $\mu^*(\cdot) = \mu_{t-1}(\cdot)$ be the optimal solution obtained.
at \((t-1)\). Under appropriate conditions, the KKT optimality assumptions can be expressed as:

\[
\lambda^*(k) = \nabla_x H(x^*(k), u^*(k), \lambda^*(k+1), \mu^*(k), \hat{\theta}_{t-1}),
\]

\[
\nabla_u H(x^*(k), u^*(k), \lambda^*(k+1), \mu^*(k), \hat{\theta}_{t-1}) = 0,
\]

for all \(k \in [t, t+N-1]\) and

\[
\lambda(t+N) = \nabla_x \Phi(x^*(t+N)).
\]

Considering the change in the initial condition \(\delta x_t\) and the parameter estimate \(\delta \hat{\theta}_t\) as the perturbations in the optimization problem formulation, we can write (22)-(24) in the perturbation form:

\[
\begin{align*}
\delta \lambda(k) &= H_{xx}(k) \delta x(k) + H_{xu}(k) \delta u(k) + f_{x}(k) \delta \lambda(k+1) \\
&\quad + C_x^T(k) \delta \mu(k) + H_{x\theta}(k) \delta \hat{\theta}_t,
\end{align*}
\]

\[
H_{uu}(k) \delta x(k) + H_{uu}(k) \delta u(k) + f_{u}(k) \delta \lambda(k+1) \\
+ C_u^T(k) \delta \mu(k) + H_{u\theta}(k) \delta \hat{\theta}_t = 0,
\]

for \(k \in [t, t+N-1]\) and

\[
\delta \lambda(t+N) = \Phi_{x\lambda}(t+N) \delta x(t+N).
\]

Writing the model equations and constraints in the perturbation form, we have

\[
\Delta x(k+1) = f_x(k) \Delta x(k) + f_u(k) \Delta u(k) + f_{\theta}(k) \Delta \theta_t,
\]

\[
C_x(k) \Delta x(k) + C_u(k) \Delta u(k) + C_{\theta}(k) \Delta \theta_t = 0,
\]

Assume that \(\delta \lambda(k+1)\) takes the form

\[
\delta \lambda(k+1) = S(k+1) \delta x(k+1) + T(k+1) \delta \hat{\theta}_t,
\]

we have, from \(\delta \lambda(t+N) = \Phi_{x\lambda}(t+N) \delta x(t+N),\) that

\[
S(t+N) = \Phi_{x\lambda}(t+N), \quad T(t+N) = 0.
\]

Using (30) in (25), we have

\[
\begin{align*}
\Delta \lambda(k) &= Z_{11}(k) \Delta x + Z_{12}(k) \Delta u(k) + Z_{13}(k) \Delta \theta_t + C_x^T(k) \Delta \mu(k),
\end{align*}
\]

where

\[
\begin{align*}
Z_{11}(k) &= H_{xx}(k) + f_{x}(k) S(k+1) f_x(k),
Z_{12}(k) &= H_{xu}(k) + f_{u}(k) S(k+1) f_u(k),
Z_{13}(k) &= H_{x\theta}(k) + f_{\theta}(k) S(k+1) f_{\theta}(k) T(k+1),
\end{align*}
\]

for all \(k \in [t, t+N-1].\) Similarly, we can express (26) as

\[
Z_{21}(k) \Delta x(k) + Z_{22}(k) \Delta u(k) + Z_{23}(k) \Delta \theta_t + C_u^T(k) \Delta \mu(k) = 0,
\]

where

\[
\begin{align*}
Z_{21}(k) &= Z_{12}^T(k),
Z_{22}(k) &= H_{uu}(k) + f_{u}(k) S(k+1) f_u(k),
Z_{23}(k) &= H_{u\theta}(k) + f_{\theta}(k) S(k+1) f_{\theta}(k) T(k+1).
\end{align*}
\]

Note that from (33) and (29)

\[
\begin{align*}
Z_{21}(k) \Delta x(k) + Z_{22}(k) \Delta u(k) + Z_{23}(k) \Delta \theta_t + C_u^T(k) \Delta \mu(k) &= 0,
C_x(k) \Delta x(k) + C_u(k) \Delta u(k) + C_{\theta}(k) \Delta \theta_t &= 0.
\end{align*}
\]

Assuming \(Z_{22}(k) > 0\) and \(C_u(k)\) is of full rank, and defining \(K_0(k)\) in (20), we have

\[
\begin{bmatrix}
\delta u(k) \\
\delta \mu(k)
\end{bmatrix} = -K_0(k) \begin{bmatrix}
Z_{21}(k) \\
C_u(k)
\end{bmatrix} \Delta x(k) + \begin{bmatrix}
Z_{23}(k) \\
C_{\theta}(k)
\end{bmatrix} \Delta \theta_t.
\]

Using (36) in (31), we have

\[
\delta \lambda(k) = \begin{bmatrix}
Z_{11}(k) - [Z_{12}(k) C_x(k)] K_0(k) \\
Z_{13}(k) - [Z_{12}(k) C_u(k)] K_0(k)
\end{bmatrix} \begin{bmatrix}
\delta x(k) \\
\delta \theta_t
\end{bmatrix}
\]

\[
+ \begin{bmatrix}
Z_{23}(k) - [Z_{22}(k) C_u(k)] K_0(k) \\
Z_{23}(k) - [Z_{22}(k) C_{\theta}(k)] K_0(k)
\end{bmatrix} \begin{bmatrix}
\delta x(k) \\
\delta \theta_t
\end{bmatrix}
\]

(37)

Therefore, by defining \(S(k)\) and \(T(k)\) as

\[
\begin{align*}
S(k) &= Z_{11}(k) - [Z_{12}(k) C_x(k)] K_0(k), \\
T(k) &= Z_{13}(k) - [Z_{12}(k) C_u(k)] K_0(k),
\end{align*}
\]

we can compute all the matrices defined earlier in (18)-(20) and formulate the gains in the predictor-corrector controller form specified in (18) and (19).

The AMPC algorithm described in Section III (A) is derived.

C. AMPC with IPA-SQP

Similar to the non-adaptive case, successive application of the NE solution requires the assumption that \(H_t = 0\) at the previous time step, which cannot be guaranteed with the NE solution alone. The IPA-SQP algorithm thus incorporates additional iterations at each time step. After the first iteration at time \(t\) using the extended NE solution, since no new update of \(\theta_t\) is expected before the next sampling time \((t+1)\), the same IPA-SQP updates as delineated in [18] can be applied.

Remark 3.2. Note that the NE solution is derived under the assumption that active constraint set is not changed by the perturbations. If the perturbations of \(\delta x_t\) and \(\delta \theta_t\) are large enough to result in change of active constraint set, these cases can be treated with the same augmented algorithm as developed in [19].

D. Properties of AMPC

As with all other indirect adaptive schemes [32], the behavior and properties of the closed-loop adaptive control system depend on both the control and parameter adaptation algorithms. The certainty equivalence principle based design assumes the adaptation law \(\delta \theta_t\) is “well-behaved” in the sense that the measured observation error is converging and the parameter error is reducing. Examples of such “well-behaved” adaptive algorithms include gradient, projection, and least squares algorithms [33].

For general nonlinear AMPC problems with constraints, since closed-form solutions for the optimization are in general not attainable, closed-loop stability and performance analysis is difficult. This is also the case for the AMPC algorithm derived in Section III. However, in a special case where the system is linear, the cost is quadratic, and constraints are not imposed, we can obtain the following result for the AMPC:

**Theorem 1:** Consider a linear system described by the transfer function

\[
y = G_p(z)u
\]

where

\[
G_p(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}
\]
Fig. 1. An inverted pendulum on a cart.

\( a_i, b_j, i = 0, \ldots, n; j = 0, \ldots, m; n > m \) are unknown parameters to be estimated using gradient or least squares algorithms. If the AMPC algorithm (16)-(19) is used to minimize the quadratic cost function

\[
J = \sum_{k=1}^{t+N-1} \{g^2(k) + y^2(k)\} + x^T(t + N)S_f x(t + N)
\]

where \( S_f \) satisfies the discrete Riccati equation for the estimated system \((\hat{A}_f, \hat{B}_f, C)\) with \( Q = CC^T \) and \( R = \gamma \), then, the closed-loop system is stable while \( y(k) \to 0 \) as \( k \to \infty \). Here \((\hat{A}_f, \hat{B}_f, C)\) is the controller canonical state-space realization of the transfer function with estimated parameters.

Sketch of the proof: The proof of the theorem follows the steps: (i) With the definition of \( S_f \), the solution of the finite horizon optimization problem in MPC can be shown to be the same as that of the infinite horizon LQ problem. (ii) Assuming \((A_k, B_k, C)\) is the controller canonical realization of the transfer function \( G(z) \) with estimated parameters, stability of the closed-loop system can be shown by following the same arguments in [34] used for continuous time system, and using the facts that (a) the closed-loop system is stable for frozen parameters, (b) the parameter variations will converge to zero, and (c) the difference between the frozen parameter system and the time varying parameter (due to parameter adaptation) can be shown to be approaching to zero using the swapping lemma of [32].

Note that the theorem implies that \( S_f \) is time-varying and needs to be recomputed on-line from the discrete Riccati equation for the estimated system to preserve stability. We also note that the preceding stability result does not require parameter convergence. On the other hand, it does not address constraints. In fact, the algorithm proposed in Section III (A) enforces constraints in the certainty equivalence sense, and the effects of uncertainties and parameter errors on the constraints, namely, the possibility of loss of feasibility during adaptation, are not considered. Such extensions will be pursued in future work.

IV. AMPC For An Inverted Pendulum on A Cart

An example, based on the nonlinear dynamics model of the inverted pendulum on the cart in Figure 1, is now considered. The state of the dynamic model is \( x = [x_1, x_2, x_3, x_4]^T \), where \( x_1 \) and \( x_2 \) are the position and velocity of the cart, and \( x_3 \) and \( x_4 \) are the angle and angular velocity of the pendulum with respect to the vertical direction. Based on the model given in [35], we have:

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= \frac{g \sin x_4 \cos x_3 - l \omega x_3^2 \sin x_3 + u}{(M + \theta) - \theta \cos^2 x_3}, \\
\dot{x}_3 &= x_4, \\
\dot{x}_4 &= \frac{g(M + \theta) \sin x_3 - l \omega x_3^2 \sin x_3 \cos x_3 + u \cos x_3}{l(M + \theta) - l \theta \cos^2 x_3},
\end{align*}
\]

where \( u \) denotes the control force produced by an actuator in the horizontal direction. \( \theta \) represents the mass of the pendulum ball, \( l \) is the length of the massless rod, and \( M \) is the mass of the cart. The discrete-time model of the system is obtained by sampling (40) with 0.1 sec update period.

We consider a stabilization problem where the objective is to control the angle and angular velocity of the pendulum with the unknown mass \( \theta \) to zero subject to control constraints on the actuator. In the simulations, the values of \( \theta = 0.1 \) kg (unknown to the controller), \( M = 2 \) kg, and \( l = 0.5 \) m are used. The control force is constrained so that

\[-1.2 \leq u(t) \leq 1.2.\]

The incremental and terminal cost functions are defined as

\[
L = x^T Q x + u^T R u, \quad \Phi = x^T S_j x,
\]

where we choose \( Q = diag(0, 0, 1, 1), R = 10, \) and \( S_j = 10 \times Q \). The prediction horizon is chosen as 10 steps in this example.

The closed-loop response is shown in Figure 2 with \( \theta(0) = 0.35 \). As these plots demonstrate, AMPC is able to drive the angle and angular velocity of the inverted pendulum to the desired equilibrium, while the parameter estimate \( \theta \) converges. The control constraints are satisfied. Figure 3 compares the responses of three closed-loop systems: (i) MPC applied to the plant model with \( \theta = 0.1 \) when the parameter estimate \( \hat{\theta} = \theta = 0.1 \) is accurate (Test A); (ii) MPC applied to the plant model with \( \theta = 0.1 \) when the parameter estimate \( \hat{\theta} = 0.35 \) is inaccurate (Test B); and (iii) AMPC applied with the initial parameter estimate of \( \hat{\theta}(0) = 0.35 \) (Test C). The weight matrices and other parameters are the same between these three controllers. In all three cases,
the control constraints are satisfied. Even without adaptation, MPC is sufficiently robust to withstand the uncertainty in the parameter $\theta$. However, with the adaptation the response of AMPC is improved and it approaches that of the MPC with known $\theta$ even when the parameter is not converging due to the lack of persistent excitation.

V. Conclusions

In this paper, we developed an approach to incorporate changes in the model parameter estimates, resulting from online adaptation, in nonlinear MPC algorithm for control sequence updates. This leads to simple updates of the control sequence, e.g., proportionally to the change in the parameter estimate when perturbations are small and the active set does not change. When the active set does change, a computational procedure similar to that described in [19] can be used to account for such changes. This approach is promising for the implementation of AMPC, especially when efficient computation is an important consideration for real-time implementation. An illustrative example has been reported. While the present paper has addressed the computational updates, further study of stability and recursive feasibility will be pursued in future publications.

REFERENCES