A Prediction-Correction Method for Model Predictive Control

Santiago Paternain, Manfred Morari and Alejandro Ribeiro

Abstract—In this work we adapt a prediction-correction algorithm for continuous time-varying convex optimization problems to solve dynamic programs arising from Model Predictive Control (MPC). The algorithm proposed is computationally efficient since it finds an approximate solution of the problem with only two Hessian inversions per iteration. The error in the approximation as compared to solving the MPC problem exactly is of the order of the square of the sampling time. This computationally efficient solution allows to extend the applicability of Model Predictive Control to systems with faster dynamics and less computational power. Numerical examples where we consider a highly nonlinear system support the theoretical conclusions.

I. INTRODUCTION

Model Predictive Control is a technique with several advantages, for instance, the capability to deal with constraints [1] and hybrid systems [2], thus finding acceptance in the industry [3], [4]. Even in cases where the system is not required to satisfy explicit constraints, its dynamics impose equality constraints that relate the states of the system of consecutive times. Therefore at each control period, one needs to solve on-line a constrained optimization problem to select the next control input. Hence the applicability of model predictive control is limited by the ability to solve such problems. This explains why this technique found traction first in the control of chemical processes which typically have large time constants and where the computational resources are not limited. Improvements in computational algorithms of Model Predictive Control can expand its applications to systems with faster dynamics and limited computing power.

Several algorithms have been proposed to reduce the computational effort of solving optimization problems in the framework of model predictive control. Most of them rely on applying a Newton-like steps to solve approximations of the original problem. This can be done by linearizing the system along a fixed optimal trajectory [5], [6] or by performing successive iterations along approximately optimal trajectories [7], [8]. Other approaches include the use of predicted states to be used as seeds for the iterates to follow. These predictions are done by propagating the current solution through the dynamics of the system [9], [10].

Such predictions based only the dynamics of the system, do not necessarily track the sequence of the optimization problem’s solution. In a recent series of works, unconstrained [11], [12] and constrained [13], [14] time-varying convex optimization problems have been considered. The prediction step in these works is based on the Euler discretization of the time derivative of the optimal solution. In [11], [12] it is shown that the error in tracking the optimal solution of the problem by the prediction step is bounded by the square of the sampling time. The prediction step then ensures that the iterates lie in the quadratic convergence region of Newton’s method and thus a Newton correction step with step size one can be used to accurately solve the problem. In this work we draw inspiration from the last class of prediction-correction methods to solve optimization problems in the context of model predictive control. Although this problem is not convex, the same theoretical guarantees previously described can be established locally (Section IV). Because the error bounds in the approximation depend on the square of the sampling time, the result is of special interest for fast systems. An approximate solution for the MPC problem based on a sensitivity analysis proposed in [15] yields an update like the prediction step. However, no iterative method is proposed and the dependence of the error with the difference between the nominal and current state is quadratic. Hence limiting its application to regimes in which the states do no differ much from the nominal. In Section V we show that the control law defined by the proposed algorithm is Input to State Stable with respect to the errors that arise from the approximate solution of the optimization problem. Numerical examples in Section VI support the theoretical results.

II. PROBLEM FORMULATION

In this work we are interested in optimally controlling a nonlinear discrete-time dynamical system. Let us denote the time by a discrete variable \( k \in \mathbb{N} \cup \{0\} \) and let \( x \in \mathbb{R}^n \) denote the state of the system and \( u \in \mathbb{R}^p \) its control input. The dynamical system is described by the following difference equation

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

where \( f : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n \) is a continuous function that satisfies \( f(0, 0) = 0 \). The receding horizon problem consists in finding a sequence of inputs that minimizes a given performance metric from the current time until a horizon \( H \in \mathbb{N} \). Formally, let \( \ell : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R} \) and \( \ell_H : \mathbb{R}^n \rightarrow \mathbb{R} \) be continuous functions encoding the performance metric of interest. The problem of interest is then to select a sequence \( \{u(k), k \geq 0\} \) such that it minimizes the following cost (see e.g. [16, Chapter 2])

\[
J(k) = \sum_{l=k}^{k+H} \ell(x(l), u(l)) + \ell_H(x(k+H+1)),
\]
The problem of minimizing (2) while satisfying (1) is a constrained optimization problem and it can be solved locally with iterative methods such as Newton’s method. To be more precise, define the vectors $\bar{x}_l \in \mathbb{R}^n$ for $l = 1 \ldots H + 1$ and $\bar{u}_l \in \mathbb{R}^p$ for $l = 1 \ldots H$ and their concatenation $\bar{x} \in \mathbb{R}^{n(H+1)}$ and $\bar{u} \in \mathbb{R}^{pH}$ as

$$\bar{x} = [x_1^T, x_2^T \ldots x_{H+1}^T]^T \quad \text{and} \quad \bar{u} = [u_1^T, u_2^T \ldots u_H^T]^T.$$  

(3)

With these definitions it is possible to write the loss (2) as

$$J(x(k), \bar{x}, \bar{u}) = \sum_{l=1}^{H} \ell(\bar{x}_l, \bar{u}_l) + \ell_H(\bar{x}_{H+1}),$$  

(4)

by imposing that the first variable of the vector $\bar{x}$ is equal to the state $x(k)$ at time $k$

$$c_1(x(k), \bar{x}, \bar{u}) = \bar{x}_1 - x(k) = 0,$$  

(5)

and the other variables satisfying the constraints imposed by the dynamics of the system

$$c_i(x(k), \bar{x}, \bar{u}) = \bar{x}_i - f(\bar{x}_{i-1}, \bar{u}_{i-1}) = 0, \forall i = 2 \ldots H + 1.$$  

(6)

The receding horizon problem can therefore be summarized as finding the sequence of control inputs – and states – that solves the following optimization problem

$$(\bar{x}^*_k, \bar{u}^*_k) := \arg\min_{\bar{x} \in \mathbb{R}^{n(H+1)}, \bar{u} \in \mathbb{R}^{pH}} J(x(k), \bar{x}, \bar{u})$$

subject to $c(x(k), \bar{x}, \bar{u}) = 0$.  

(7)

At each time the problem is solved and the action applied is the first component of the vector corresponding to the inputs, i.e., $u(k) = \bar{u}^*_k$, thus defining the control law

$$u(x(k)) = \kappa_{MPC}(x(k)) := \bar{u}^*_{k,1}.$$  

(8)

We define the closed loop system when the above input is applied to system (1) as

$$x(k+1) = f_{\kappa_{MPC}}(x(k)) := f(x(k), \kappa_{MPC}(x(k))).$$  

(9)

Under certain assumptions the origin of the closed loop dynamical system is globally asymptotically stable. We state these assumptions and the stability result for future reference.

Assumption 1 (Continuity of system and cost). The functions $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$, $\ell : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}_+$ and $\ell_H : \mathbb{R}^n \to \mathbb{R}_+$ are continuous, $f(0,0) = 0$, $\ell(0,0) = 0$ and $\ell_H(0) = 0$.

Assumption 2. There exist compact sets $\mathcal{X} \subset \mathbb{R}^{n(H+1)}$, $\mathcal{U} \subset \mathbb{R}^{pH}$ such that for all $k \geq 0$ the solution $(\bar{x}^*_k, \bar{u}^*_k)$ to the problem (7) satisfies $(\bar{x}^*_k, \bar{u}^*_k) \in \mathcal{X} \times \mathcal{U}$.

Assumption 3. The stage cost $\ell(\cdot)$ and the terminal cost $\ell_H(\cdot)$ satisfy

$$\min_{\bar{u}} \ell_H(f(x, \bar{u})) + \ell(x, \bar{u}) \leq \ell_H(x), \forall x \in \mathbb{R}^n,$$  

(10)

and

$$\ell(x, \bar{u}) \geq \alpha_1(||x||), \quad \ell_H(x) \leq \alpha_2(||x||),$$  

(11)

in which $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ are $\mathcal{K}_\infty$ functions.

**Theorem 1 (Theorem 2.24 [16]).** Suppose that Assumptions 1, 2 and 3 are satisfied. Then the origin is globally asymptotically stable for the system (6).

The usual numerical methods employed to solve the problem in (7) are based on Newton’s method. At each time $k$ one needs to solve a different optimization problem since the problem depends on the current state $x(k)$. The latter typically involves several Newton iterations because the constraints are not linear on state or the actions. Drawing inspiration from prediction-correction methods used to solve unconstrained time varying convex optimization problems [12] we present an algorithm to efficiently solve (7) while preserving stability of the system. We present the algorithm in the next section.

### III. Prediction-Correction Newton

Let us define a multiplier $\lambda \in \mathbb{R}^{n(H+1)}$ and the Lagrangian associated to the problem (7) as

$$L(x(k), \bar{x}, \bar{u}, \lambda) = J(x(k), \bar{x}, \bar{u}) + \lambda^\top c(x(k), \bar{x}, \bar{u}).$$  

(12)

To simplify the notation consider the vector $z = [\bar{x}^T, \bar{u}^T, \lambda]^T \in \mathbb{R}^{H(2n+p)+2n}$. Observe that $z^*_k$, the solutions of the problem (7), are given by the points in which the partial derivative of the Lagrangian with respect to $z$ is zero, this is

$$\nabla_z L(x(k), z^*_k) = 0.$$  

(13)

Hence we will focus on finding a solution of the above equation. Note that other stationary points – not only local minima – are solutions to (13). Thus our algorithm does not provide a global guarantee that the solution obtained is in fact a solution of (7) and hence local results will be proved in Section IV. As stated before the goal is an algorithm that is able to compute efficiently a solution of (13). Newton’s method converges quadratically – in logarithmic scale – when the iterates are located in the quadratic convergence zone. Formally, for any $\varepsilon > 0$, in $N$ steps, the iterate $z^N_k$ satisfies $\|\nabla_z L(x(k), z^N_k)\| < \varepsilon$ with $N = O(\log \log(1/\varepsilon))$ (see e.g. [17, Section 9.5.3]) as long as the initial guess $z^0_k$ is inside the quadratic convergence region. The latter is a neighborhood of the critical point that depends on the eigenvalues of the Hessian of the Lagrangian and on its Lipschitz constant. Outside this region, the convergence of Newton’s method is only linear – in logarithmic scale – and hence this regime is the bottle neck in optimization problems. Ensuring that the iterates remain in the quadratic convergence region becomes crucial in this setting due to the time varying nature of the problem of finding a solution to (13). In particular, after having found a solution for the state at time $k$ the systems evolves and it is not guaranteed that said point remains inside the quadratic convergence region at time $k+1$. To avoid this scenario, we introduce a prediction step based on the dependence of $z^*_k$ on $x(k)$. This dependence can be understood as a consequence of the Implicit Function Theorem applied to the first order optimality condition (13). As long as the second derivative of $\nabla^2_z L(x, z^*(x))$ is
in invertible, there exists a neighborhood of \((x, z^*(x))\) in which we have that
\[
\nabla_x z^*(x) = -\nabla^{2}_{zz} L(x, z^*(x))^{-1} \nabla_{zx} L(x, z^*(x)).
\] (14)

The previous result has also been used in [15] to solve the MPC problem approximately around a nominal state in a similar form as the prediction step that we define next. This step behaves as if the iterate \(z_k\) were the optimal solution for the state \(x(k)\) and considers a first order approximation, based on the previous expression, for the optimal solution at time \(k+1\). Using the chain rule, the variation of the optimal solution is the product of the derivative of the optimal solution with respect to the state (cf., (14)) times the state variation. Using this idea, the prediction step is defined as
\[
z_{k+1|k} = z_k - \nabla^{2}_{zz} L(x(k), z_k)^{-1} \nabla_{zx} L(x(k), z_k) (x(k+1) - x(k)).
\] (15)

Assuming that a solution at time \(k\) has been computed and that the corresponding input \(u(k)\) has been applied at time \(k\), the system state will evolve to \(x(k+1)\). Hence, at time \(k+1\) all the elements needed to compute \(z_{k+1|k}\) are available. The effect of the prediction is that of keeping the iterate at a distance of the optimal solution at time \(k+1\) that is \(O(T_s^2)\), where \(T_s\) is the sampling time of the system. This is, if at time \(k\) we have \(z_k = z_k^*\), then it holds that \(\|z_{k+1|k} - z_k^*\| < O(T_s^2)\). The latter is not surprising since the prediction step can be interpreted as the Euler discretization of the continuous time dynamics (see [11], [12]) and the error per step in the Euler discretization is of order \(T_s^2\). Likewise, if \(z_k \neq z_k^*\) the prediction is such that it does not increase the error with respect to the optimal solution by more than \(O(T_s^2)\) (cf., Proposition 1). In particular, the dependence with the square of the sampling time ensures that the predicted iterate will remain in the quadratic convergence region of the problem at time \(k+1\) for small enough sampling time. This prepares the ground for the following correction step, where \(z_{k+1|k}\) is used at time \(k+1\) as the starting position for a classic Newton iteration with step size one
\[
z_{k+1} = z_{k+1|k} - \nabla^{2}_{zz} L(x(k+1), z_{k+1|k})^{-1} \nabla_{zx} L(x(k+1), z_{k+1|k}).
\] (16)

It is important to point out that the computations in (15) and (16) are performed at time \(k+1\), that is, given a state \(x(k)\) and a control input \(u(k)\) we let the system evolve one step to obtain \(x(k+1)\) and then we perform the operations described in (15) and (16). In that sense, the prediction step is not truly a prediction but a compensation for the change in the optimization problem due to its dependence on the current state. Recall that the vector \(z\) includes the variables corresponding to the control inputs \(u\). Specifically \(\bar{u}_{k-1}\) refers to the optimal action that should be performed at time \(k\) according to the optimization problem (7). Thus, the prediction–correction scheme (15)–(16) defines a control law
\[
u(k) = \kappa(x(k), x(k-1), z_{k-1}) := \bar{u}_{k-1}, \quad \forall k > 0.
\] (17)

The initial control input \(u(0)\) can be computed by solving (7) with classic Newton steps. Since this computation can be done offline before the system starts evolving, the complexity required to ensure an accurate solution of (7) is affordable. In what follows we work towards showing that the the system (1) with the control law defined in (17), based on the prediction-correction scheme (15)–(16) with \(z_0\) being close to the optimal solution of (7) at time \(k = 0\) is input state stable. To do so, we first need to bound the error of solving (7) via the prediction-correction method proposed in (15)–(16) (Section IV). With said bound and leveraging the fact that the nominal control law defined (8) is asymptotically stable (cf., Theorem 1), we show in Section V input to state stability of the system (1) with respect to the approximation errors, when the feedback control used is the one defined in (17). Before moving to Section IV we introduce the following standard assumption in the analysis of prediction-correction optimization methods.

**Assumption 4.** Let \(z^* : \mathbb{R}^n \rightarrow \mathbb{R}^{(n+p+2)H}\) be a function such that
\[
\nabla_x L(x, z^*(x)) = 0.
\] (18)

For all \(x \in \mathbb{R}^n\), denote by \(\lambda_i \left( \nabla^{2}_{zz} L(x, z^*(x)) \right)\) the eigenvalues of the second derivative of the Lagrangian with respect to \(z\) at the point \((x, z^*(x))\). There exists a uniform bound \(m_0 > 0\) for all \(x\) such that for all we have
\[
\min_{i=1,\ldots,(n+p+2)H} |\lambda_i (\nabla^{2}_{zz} L(x, z^*(x)))| > 2m.
\] (19)

**Assumption 5.** The function \(L(x, z)\) is sufficiently smooth both in \(x \in \mathbb{R}^n\) and \(z \in \mathbb{R}^{(n+p+2)}\). In particular it has bounded second and third order derivatives with respect to \(x\) and \(z\) as
\[
\begin{align*}
\|\nabla_{zz} L(x, z)\| &\leq C_0, \quad \|\nabla_{zx} L(x, z)\| \leq C_1, \\
\|\nabla_{xx} L(x, z)\| &\leq C_2, \quad \|\nabla_{xx} L(x, z)\| \leq C_3.
\end{align*}
\] (20)

If the solution of (7) satisfies the Sufficient Second Order Conditions and the Linear Independence Constraint Qualification, then the matrix \(\nabla_{zz} L(x, z^*(x))\) is invertible for all \(x\) [18]. Assumption 4 is a more stringent condition since it imposes a uniform bound on the absolute value of the minimum eigenvalues of that matrix.

**IV. PREDICTION–CORRECTION ERROR**

Let us define the closed loop dynamical system resulting from applying the control law (17) to the system (1)
\[
\begin{align*}
x(k+1) &= f(x(k), x(k-1), z_{k-1}) \\
&:= f(x(k), \kappa(x(k), x(k-1), z_{k-1})),
\end{align*}
\] (21)

and let us denote its solution by \(\phi^*_k(x, k, 0), z_0)\), where \(z_0\) is an approximate solution of (7) at time \(k = 0\) with accuracy \(\varepsilon > 0\). This solution is computed offline via Newton’s method for instance. In Proposition 4 we establish a bound
on the error of the prediction step with respect to the optimal solution at the next iterate based on the distance at the current time. This bound will allow to establish that the predicted iterate lies in the quadratic convergence region. In order to state such result, we need a bound on the variation of the states between consecutive times. We formally state this assumption next.

Assumption 6. Let $z^*_k$ be the solution of (7) at time $k = 0$. Denote by $z_0$ an approximate solution of said problem computed offline up to a desired accuracy $\varepsilon > 0$, i.e., $\|z_0 - z^*_0\| < \varepsilon$. Denote by $k$ the time index, $x(0)$ the initial condition of the system and $\phi_k(k, x(0), z_0)$ the solution of the dynamical system (21). Then, for all $k \in \mathbb{N}$ and for all $x(0) \in \mathbb{R}^n$ it holds that

$$\|\phi_k(k + 1, x(0), z_0) - \phi(k, x(0), z_0)\|^2 \leq BT^2_k \leq \frac{m^2}{C_1} \left(\frac{C_1 C_0^2}{2m^2} + \frac{C_2 C_0}{m} + \frac{C_1}{2}\right) .$$

(22)

The first inequality in the previous assumption imposes a bound on the evolution of the dynamical system. In particular it cannot differ more than $\sqrt{BT}$ between to consecutive time indices. This can be interpreted as if the continuous time counterpart of the dynamical system had a derivative bounded by $\sqrt{B}$. The second inequality imposes a restriction on the largest sampling time that one can consider with respect to the constants of the optimization problem.

Proposition 1. Let $\phi_k(k, x(0), z_0)$ be the solution of the dynamical system (21). Let Assumptions 4, 5 hold and define

$$\delta_1 = \frac{C_1 C_0^2}{2m^2} + \frac{C_2 C_0}{m} + \frac{C_3}{m} \quad \text{and} \quad \delta_2 = \frac{C_1}{m^2} + \frac{C_2}{m} .$$

(23)

Let $\{z^*_k, k \geq 0\}$ be the sequence of solutions of problem (7) when the evolution of the dynamical state is defined by (21). If $\|\nabla x L(\phi(k, x(0), z_0), z_k)\| < m^2/C_1$ for any $k \geq 0$ we have that

$$\|z_{k+1} - z^*_k\| \leq \delta_1 BT^2_k + \left(1 + \delta_2 \sqrt{BT_k}\right)\|z_k - z^*_k\| .$$

(24)

With the bound of the norm of the difference between the predicted iterate and the optimum at time $k + 1$ from the previous proposition, we proceed to bound the error $\|z_{k+1} - z^*_k\|$. In particular, we show that if $\|z_0 - z^*_0\|$ is bounded by a function of the order of $T^2_k$, this bound is preserved for every $k > 0$. We formalize this result next.

Theorem 2. Let $\phi_k(k, x(0), z_0)$ be the solution of the dynamical system (21). Let Assumptions 4, 5 hold. Let $\{z^*_k, k \geq 0\}$ be the sequence of solutions of problem (7) when the evolution of the dynamical state is defined by (21). Let $\eta > 0$ and let $T^*_k$ satisfy the following inequality

$$BT^2_k \leq \min \left\{1, \frac{m\eta}{C_1 \left((1 + \delta_2)\eta + \delta_1\right)^2}\right\} ,$$

(25)

where $B$ is the constant defined in Assumption 6 and $\delta_1$ and $\delta_2$ are the constants defined in (23). Then, if

$$\|z_0 - z^*_0\| < \eta BT^2_k$$

(26)

the following holds for all $k \geq 0$

$$\|z_k - z^*_k\| \leq \eta BT^2_k .$$

(27)

The implication of the previous result is that the prediction–correction scheme yields a control law (17) that is not very different from exact MPC, i.e., solving (7) exactly. In particular the difference between these two control laws is bounded by $\eta BT^2_k$ for all $k$. The latter bound can be reduced arbitrarily by reducing $T^*_k$ or $\eta$ which entails a reduction on $T_k$ (cf., (25)). The fact that for practical reasons it is not desirable to have a sampling time that is too small imposes a limit on the accuracy of the solution. In the next section we use the fact that the control law (17) is similar to exact MPC to show that the closed loop system (21) remains stable.

V. INPUT TO STATE STABILITY

We work next towards establishing input to state stability of the system (21) with respect to the approximation errors. In the previous section we showed that the error in solving problem (7) by using the prediction-correction approach (15)–(16) is bounded by $\eta BT^2_k$ (cf., Theorem 2). Define then, the sequence of disturbances $\{d(k), k \geq 0\}$ as

$$d(k) = u_{MPC}(k) - u(k) ,$$

(28)

and write the dynamical system (21) as

$$x(k + 1) = f_{\eta}(x(k)) = f_{MPC}(x(k), d(k)) ,$$

(29)

where $f_{MPC}(x(k), d(k))$ is such that

$$f_{MPC}(x(k), 0) = f_{MPC}(x(k)) .$$

(30)

We define input to state stability of the closed loop system (21) where the input is the sequence of disturbances arising from (7) approximately.

Definition 1 (Input to State Stability). Let $\tilde{g} : \mathbb{R}^n \times \mathbb{R}^w \to \mathbb{R}^n$ and consider the following discrete time dynamical system

$$x(k + 1) = \tilde{g}(x(k), d(k)) ,$$

(31)

with solution denoted by $\phi_{\tilde{g}}(x(0))$. System (31) is input to state stable if there exists a $KL$-function $\beta$ and a $K$-function $\gamma$ such that for all initial state $x(0)$ and sequence of disturbances $\{d(k), k \geq 0\}$ satisfying

$$\|\phi_{\tilde{g}}(x(0))\| \leq \beta(\|x(0)\|, k) + \gamma \left(\max_{j \in \mathbb{N}} \|d(j)\|\right) .$$

(32)

We say that (31) is locally input to state stable if there exists constants $c_1$ and $c_2$ such that (32) holds for any initial state $\|x(0)\| \leq c_1$ and disturbances $\|d(k)\| \leq c_2$.

From Theorem 2 one has that $\|d(k)\| < \eta BT^2_k$ for all $k \geq 0$. The bounds on the disturbances along with the following results in [19] allows us to establish local input to state stability and input to state stability with stronger assumptions.

Theorem 3 (Theorem 2 [19]). Let $\tilde{g}(x, d)$ in (31) be absolutely continuous in $d$ for all $x \in \mathbb{R}^n$ and for all $d \in D \subset \mathbb{R}^w$, where $D$ is a compact set. If the nominal
dynamical system \( g(x(k)) = \dot{g}(x(k), \mathbf{0}) \) is asymptotically stable, the perturbed system (31) is input to state stable if one of the following conditions holds

- Function \( g(x) \) is absolutely continuous in \( x \in \mathbb{R}^n \)
- There exists a absolutely continuous Lyapunov function \( V(x) \) for the system \( x(k + 1) = g(x(k)) \).

**Corollary 1** (Corollary 1 [19]). Let \( \dot{g}(x, d) \) in (31) be continuous in a neighborhood of \( x = 0 \) and \( d = 0 \). If the nominal dynamical system \( g(x(k)) = \dot{g}(x(k), \mathbf{0}) \) is asymptotically stable, the perturbed system (31) is locally input to state stable if one of the following conditions holds

- Function \( g(x) \) is continuous in a neighborhood of \( x = 0 \)
- There exists a Lyapunov function \( V(x) \) for the system \( x(k + 1) = g(x(k)) \) which is continuous in a neighborhood of \( x = 0 \).

Notice that the assumptions on the continuity of the closed loop system are not trivially satisfied by all systems since the continuity of the optimizer in (7) is not generally guaranteed for multiparametric optimization problems. For instance some necessary conditions are linear equality constraints, continuous objective function and the solution of (7) being unique and compact, see e.g. [20, Chapter 6]. In the case here considered equality constraints are not linear, hence we need to assume these to establish input to state stability of the system (21) with respect to the disturbances.

**Proposition 2.** Let Assumptions 2, 3 hold. Then, if the system (29) is absolutely continuous in \( d \) for all \( x \in \mathbb{R}^n \) and for all \( d \) and (30) is absolutely continuous in \( x \) then (29) is input to state stable. If the system (29) is continuous in a neighborhood of \( x = 0 \) and \( d = 0 \) and (30) is continuous in a neighborhood of \( x = 0 \) then (29) is locally input to state stable.

**Proof.** Theorem 1 establishes that the nominal system \( f_{\text{MPC}}(x) \) is asymptotically stable. The rest of the proof follows from the fact that the disturbances \( d(k) \) are bounded (Theorem 2) and Theorem 3 and Corollary 1.

**VI. NUMERICAL EXPERIMENTS**

In this section we consider the problem of controlling the position of a point mass subject to nonlinear friction. Let us denote by \( M \) the mass and let \( x \) be its position, \( g \) the gravitational constant and \( F_a(x) \) the dynamic friction depending on the velocity of the mass and the control input force. Then, the dynamics of the system are

\[
M \ddot{x} = u - MgF_a(x).
\]

We discretize the previous dynamics using Euler’s forward method to obtain a system of the form (1). To be precise, define the state vector \( x \in \mathbb{R}^2 \) where its first and second components correspond to position and speed respectively. Then, for sampling time \( T_s \), the discretization yields

\[
x_1(k + 1) = x_1(k) + T_s x_2(k) \text{ for all } k \geq 0,
\]

\[
x_2(k + 1) = x_2(k) + T_s \left( \frac{u(k)}{M} - g F_a(x_2(k)) \right) \text{ for all } k \geq 0.
\]

We consider a quadratic cost \( \ell(x, u) = (x^T Q x + u^T R)^2 / 2 \), with \( R > 0 \) and \( Q \in \mathbb{R}^{2 \times 2} \). The specific values selected for the experiments are \( g = 9.81 \), \( M = 1 \), \( x(0) = [0.1, 0]^T \), and \( Q \) is diagonal with \( Q_{11} = 1 \times 10^3 \) and \( Q_{22} = 1.25 \), \( R = 1 \times 10^{-3} \). \( T_s = 0.2 \), \( H = 20 \) and dynamic friction with the following expression (see plot in Figure 1)

\[
F_a(x) = 0.25(\tanh(100\dot{x}) - \tanh(10\dot{x})) + 0.1 \tanh(50\dot{x}) + 0.01\dot{x}.
\]

In Figure 1, we observe the control input, position and velocity resulting by solving (7) with three different approaches. MPC – in green – is performing 50 Newton iterations at each control period, thus solving the problem almost exactly. In blue we observe the performance of the prediction correction method (15–16) and in red we observe the performance of running two Newton steps after computing a seed for time \( k+1 \) by propagating the solution at time \( k \) with the dynamics of the system. It is non surprising that the performance of MPC is better than that achieved with other methods, however in the case of Prediction-Correction and the state prediction only two Hessian inversions are needed. At time \( t = 0.2 \) the action selected by the controller based on propagating the solution with the dynamics, differs considerably from that taken by other two controllers which entails a slower convergence to the desired setpoint \([0, 0]^T\). The latter is due to the fact that between times \( t = 0 \) and \( t = 0.2 \) the speed varies considerably and thus the evolution of the system will differ from the naive Euler integration. In summary, prediction correction achieves a performance comparable to that of solving MPC “exactly” but reducing its computational cost to 4% of the exact solution and it outperforms an algorithm with the same computational cost.

**VII. CONCLUSION**

We considered a prediction-correction algorithm to solve approximately a time varying multiparameter optimization problem in the context of Model Predictive Control. In particular, the prediction-correction needs only two Hessian
inversions which makes it efficient as compared to classical algorithms that solve the sequence of optimization problems independently and ensures an error bounded by the square of the sampling time. In addition, under some assumption we showed that the closed loop system is input to state stable with respect to the approximation error.

Fig. 2: Control input, position and velocity of the system (33) for three different approaches of solving (7). The sampling time is \( T_s = 0.2 \) s, the horizon is \( H = 20 \) and MPC is running 50 Newton’s steps to solve the problem at each time instant.