A Warm-start Interior-point Method for Predictive Control

Amir Shahzad * **Eric C. Kerrigan** **
George A. Constantinides *

* *Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, U.K.*
e-mail: a.shahzad07,g.constantinides@imperial.ac.uk.
** *Department of Aeronautics and Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, U.K.*
e-mail: e.kerrigan@imperial.ac.uk

Abstract: In predictive control, a quadratic program (QP) needs to be solved at each sampling instant. We present a new warm-start strategy to solve a QP with an interior-point method whose data is slightly perturbed from the previous QP. In this strategy, an initial guess of the unknown variables in the perturbed problem is determined from the computed solution of the previous problem. We demonstrate the effectiveness of our warm-start strategy to a number of online benchmark problems. Numerical results indicate that the proposed technique depends upon the size of perturbation and it leads to a reduction of 30–74% in floating point operations compared to a cold-start interior-point method.

Keywords: Optimization problems, Interior-point methods, Predictive Control.

1. INTRODUCTION

This paper describes a new warm-start strategy in an interior-point method (IPM) when solving a quadratic programming (QP) problem. In predictive control, at each sampling instant with the given state information, a QP problem is solved to obtain a sequence of inputs and only the first input is applied to the plant. This process is repeated at every sample instant with a new state estimate. We consider the scenario in which one QP problem has been solved by an IPM at the previous time instant, and we need to solve a slightly perturbed QP from the preceding QP at the current time instant. The strategy in which the information gained in solving the previous problem is used in choosing a starting point in an IPM is known as a warm-start strategy.

After the publication of the seminal paper by Karmarker (1984), interior-point methods have proven to be an efficient way of solving linear, quadratic, and nonlinear programming problems compared to active set methods, especially for large scale problems. However, they cannot easily exploit the solution of the preceding problem in warm-starting like active set methods can (Wright, 1997; Gondzio and Grothey, 2008). A warm-start of an IPM with the solution of the preceding problem, if it is close to the boundary of the feasible region, usually leads to the blocking of the search direction, which means that the steplength becomes very small, and the next iterate will be very close to the previous one. In the last decade, a number of attempts have been made to improve warm-start strategies in IPMs. Yildirim and Wright (2002) and John and Yildirim (2008) discuss warm-start strategies for linear programming (LP) problems. In Yildirim and Wright (2002), a worst case estimate on the number of IPM iterations required to converge to a solution of a perturbed LP from the warm-start starting point is determined. Their estimate mainly depends on the size of perturbation and on the conditioning of the original problem. John and Yildirim (2008) has implemented several warm-start strategies in IPMs for LP problems. They concluded that most of the strategies are effective in reducing the computational time for smaller perturbations. Recently, a new unblocking strategy was proposed by Gondzio and Grothey (2008). This is based on sensitivity analysis of the search direction with respect to the current point. Numerical tests show that, on average 50–60% of the computations can be saved on a range of LP and QP problems varying from small scale to large scale problems when this unblocking strategy is combined with other warm-start strategies.

In recent years, attempts have been made to use predictive control in fast processes with a short sampling time. To reduce the computational efforts new techniques have emerged. In Bemporad, Morari, Dua, and Pistikopoulos (2002), a large number of QPs are solved off-line for all possible initial states of the plant, and then an explicit function is formed using the solutions of the QPs. This approach is generally applicable to small-scale problems. Ferreau, Bock, and Diehl (2008) proposed a warm-start strategy based on an active set method that uses the ideas from parametric optimization. This strategy exploits the solution of the previous QP under the assumption that the active set does not change much from one QP to the next one. Furthermore, a premature termination is described

⋆ This work was funded by the EPSRC under grant number EP/G031576/1.

to meet the requirements of real-time implementations. However, in some cases this early termination may lead to infeasibility. In Wang and Boyd (2010), warm-starting and early termination of the QP problem is proposed. In warm-starting, the initialization of the QP problem is done using the predictions made in the previous step. The early termination significantly reduces the computations, but on the other hand it may lead to state equation violations.

In each iteration of an IPM, a linear system of equations needs to be solved to find the search direction. This linear system becomes increasingly ill-conditioned as the solution is approached. The linear system is either solved by direct or iterative methods. A well-conditioned approximation of this linear system is proposed by Shahzad, Kerrigan, and Constantinides (2010), in which an approximate search direction is calculated to reduce the computations by an iterative method. In this paper we have extended that work to design a warm-start strategy in an IPM. The main contribution of this paper is to present a new warm-start strategy in an IPM, which is not only computationally efficient, but also possesses the following important characteristics:

- the linear systems appearing in this approach are well-conditioned even at the later stages of the IPM,
- no matrix triple product is required to compute the Hessian matrix, and
- it is more effective than a cold-start IPM when there are few changes in active constraints from the preceding QP to the perturbed QP.

This paper is organized as follows. We start in Section 2 with an outline of the receding horizon regulator problem with quadratic objective and linear constraints using a discrete-time state space process. Further, a transformation to a QP is described. In Section 3, we describe the procedure of an IPM used to solve a QP problem. In Section 4, we describe a new warm-start strategy based on an IPM. The computational complexity involved in each IPM in terms of flops is presented in Section 5. In Section 6, four benchmark problems are presented to show the effectiveness of our proposed algorithm. Finally, some conclusions are drawn.

Notation: For a matrix, $A > 0$ (≥ 0) means that A is positive (semi-positive) definite and for a vector, $x < 0\,(\leq 0)$ means that each element of x is negative (non-positive). For vectors x, y, we denote $(x, y) := \begin{bmatrix} x^T & y^T \end{bmatrix}^T$. The symbol $\otimes$ represents the Kronecker product. $\mathbf{1}_n$ denotes a vector of ones of length n.

2. PROBLEM DESCRIPTION

Consider a discrete-time linear time-invariant dynamic system of the form

$$x_{i+1} = Ax_i + Bu_i, \tag{1}$$

where $x_i \in \mathbb{R}^n$ is the state vector and $u_i \in \mathbb{R}^m$ is the input vector at the i^{th} time instant. Let $\bar{x} = x_0 \in \mathbb{R}^n$ be the measurement or estimate of the state at the current time instant. The objective is to find, over a finite horizon of length N, a sequence of optimal control inputs $u_0, \ldots, u_{N-1}$ subject to equality constraints (1) and the inequality constraints

$$u_{\text{lb}} \leq u_i \leq u_{\text{ub}}, \quad i = 0, 1, \ldots, N-1, \tag{2a}$$

$$J_i x_i + E_i u_i \leq d_i, \quad i = 0, 1, \ldots, N-1, \tag{2b}$$

while minimizing the quadratic objective

$$x_N{}^T P x_N + \sum_{i=0}^{N-1} \left(x_i{}^T Q x_i + u_i^T R u_i + 2 x_i^T M u_i \right) \tag{3}$$

with $R > 0$, $Q - MR^{-1}M^T \geq 0$ and $P \geq 0$, $J_i \in \mathbb{R}^{l \times n}, E_i \in \mathbb{R}^{l \times m}, Q, P \in \mathbb{R}^{n \times n}, R \in \mathbb{R}^{m \times m}, M \in \mathbb{R}^{n \times m}$. Terminal constraints of the form $J_N x_N \leq d_N$ are easily included in (2b), because $x_N := Ax_{N-1} + Bu_{N-1}$.

To solve the above problem the vector of decision variables is defined as $\theta := \begin{bmatrix} u_0^T & u_1^T & \ldots & u_{N-1}^T \end{bmatrix}^T$.

The above problem can be converted to a QP problem, which can be written in the form

$$\min_{\theta} \mathcal{J}(\theta) := \frac{1}{2}\theta^T H \theta + \theta^T h(\bar{x}) \text{ s.t. } G\theta \leq g(\bar{x}), \tag{4}$$

where $\theta \in \mathbb{R}^{n_d}, H \in \mathbb{R}^{n_d \times n_d}, G \in \mathbb{R}^{n_i \times n_d}$ with $n_d := mN$ and $n_i := (2m + l)N$ is the number of inequality constraints. The matrices H, G and the vectors $h(\bar{x}), g(\bar{x})$ are defined in Appendix A. To simplify notation in subsequent sections the $(\bar{x})$ is omitted from vectors $h(\bar{x})$ and $g(\bar{x})$.

3. INFEASIBLE INTERIOR-POINT METHOD

In this section we review the ideas in interior point methods (Wright, 1997). The Karush-Kuhn-Tucker (KKT) conditions of (4) are

$$H\theta + G^T\phi + h = 0, \tag{5a}$$

$$G\theta - g + s = 0, \tag{5b}$$

$$\Phi S \mathbf{1}_{n_i} = 0, \quad \phi, s \geq 0, \tag{5c}$$

where $\phi \in \mathbb{R}^{n_i}$ is called a dual variable, $s \in \mathbb{R}^{n_i}$ is a vector of slack variables, $\mathbf{1}_{n_i} \in \mathbb{R}^{n_i}$ is a vector of ones and Φ, S are diagonal matrices defined by $\Phi := \text{diag}(\phi), S := \text{diag}(s)$.

In an IPM the optimal solution is obtained by solving the nonlinear optimality conditions (5). The classical algorithm to solve such equations is Newton's method. This is an iterative method in which at each iteration k, the solution of a linear system of the following form is required to find the search direction $(\Delta\theta^k, \Delta\phi^k, \Delta s^k)$:

$$\underbrace{\begin{bmatrix} H & G^T \\ G & -W^k \end{bmatrix}}_{\mathcal{A}^k} \begin{bmatrix} \Delta\theta^k \\ \Delta\phi^k \end{bmatrix} = \underbrace{\begin{bmatrix} r_H^k \\ r_L^k \end{bmatrix}}_{b^k}, \tag{6}$$

or equivalently,

$$\left(H + G^T (W^k)^{-1} G\right) \Delta\theta^k = r_H^k - G^T (W^k)^{-1} r_L^k \tag{7}$$

where $W^k, \Delta s^k, \Delta\phi^k, r_H^k, r_L^k$ are defined in Appendix A. We can split the matrix triple product in (7) according to

$$H + G^T (W^k)^{-1} G := H + (W_b^k)^{-1} + G_c^T (W_c^k)^{-1} G_c, \tag{8}$$

where $W_b^k \in \mathbb{R}^{n_b \times n_b}, W_c^k \in \mathbb{R}^{n_c \times n_c}$, $n_b := 2mN$ and $n_c := lN$. W_b^k, W_c^k, G_b, G_c are defined in Appendix A. The above form of computing the Hessian matrix indicates that there is no need to compute the matrix triple product if we only have input bounds.

An IPM in which an initial guess of θ satisfies the inequality constraints defined in (4) and $\phi, s > 0$ is called a *feasible*

Algorithm 1 Infeasible Interior Point Method

Input:
- H, G, h, g
- Initial guess $\theta^0, \phi^0 > 0, s^0 > 0$
- Tolerance $\epsilon > 0$

Output: Optimal θ.

Algorithm:

1: Set $k = 0$ and compute $\mu^0 := \frac{(\phi^0)^T s^0}{n_i}$, $e^0_{tol} := \|b^0\|_\infty$.
2: **while** $\mu^k > \epsilon$ and $e^k_{tol} > \epsilon$ **do**
3: Solve (7) for $\Delta\theta^k$ with Cholesky factorization. Compute $\Delta\phi^k$ and Δs^k from (A.1) and (A.2).
4: Choose α^k as the largest value in $(0, 1]$ such that the following conditions hold:

$$(\theta^k, \phi^k, s^k) + \alpha^k(\Delta\theta^k, \Delta\phi^k, \Delta s^k) \in \mathcal{N}_{-\infty}(\gamma, \beta),$$
$$\frac{(\phi^k + \alpha^k\Delta\phi^k)(s^k + \alpha^k\Delta s^k)}{n_i} \leq (1 - 0.01\alpha^k)\mu^k.$$

5: Compute $(\theta^{k+1}, \phi^{k+1}, s^{k+1}) := (\theta^k, \phi^k, s^k) + \alpha^k(\Delta\theta^k, \Delta\phi^k, \Delta s^k)$.
6: Compute $\mu^{k+1} := \frac{(\phi^{k+1})^T s^{k+1}}{n_i}$, $e^{k+1}_{tol} := \|b^{k+1}\|_\infty$.
7: Increment k by 1.
8: **end while**

IPM. An IPM for which only $\phi, s > 0$ is called an *infeasible* IPM. An infeasible IPM is described in Algorithm 1 for a QP problem. This is an extension of the infeasible-path-following IPM (Algorithm IPF) of (Wright, 1997, p. 110), which was designed for a linear programming problem. The central path neighbourhood $\mathcal{N}_{-\infty}(\gamma, \beta)$ is defined in (Wright, 1997, p. 109).

4. δ–ACTIVE SET BASED INFEASIBLE INTERIOR-POINT METHOD

In this subsection, we will briefly describe the δ–active set based infeasible IPM proposed by Shahzad et al. (2010), and extend that work to propose a warm-start strategy. Note that if $s^k_i \to 0$ as $k \to \infty$, then (5b) indicates that the i^{th} inequality constraint is active. Let $\mathcal{N} := \{1, 2, \ldots, n_i\}$ and define a δ-active set $\mathcal{N}^k_A(\delta)$, depending upon the parameter $\delta > 0$, as

$$\mathcal{N}^k_A(\delta) := \{i \in \mathcal{N} \mid 0 < w^k_i < \delta\}, \tag{9}$$

where $w^k_i := \frac{s^k_i}{\phi^k_i}$ and a δ-inactive set as $\mathcal{N}^k_A(\delta) := \mathcal{N} \setminus \mathcal{N}^k_I(\delta)$. As the iteration number k of Algorithm 1 increases, the values corresponding to the $\delta-$inactive constraints of the diagonal matrix W^k become very large, while others become very small. Note that if $s^k_i \not\to 0$ as $k \to \infty$ then $\phi^k_i \to 0$, because $\mu^k = \frac{(\phi^k)^T s^k}{n_i} \to 0$ as $k \to \infty$. Therefore, $w^k_i \to \infty$ as $k \to \infty$ for the constraints that are inactive at the solution. As a result, the condition number of $\mathcal{A}^k$ and the coefficient matrix of (7) becomes very large.

To avoid this ill-conditioning in the later stage of an IPM, the following scheme is proposed by Shahzad et al. (2010). Permute the matrix $\mathcal{A}^k$ according to the δ–active and $\delta-$inactive constraints as

$$\left[\begin{array}{ccc|c} H & (G^k_1)^T & (G^k_2)^T \\ G^k_1 & -W^k_1 & 0 \\ \hline G^k_2 & 0 & -W^k_2 \end{array}\right] \left[\begin{array}{c} \Delta\theta^k \\ \Delta\phi^k_1 \\ \hline \Delta\phi^k_2 \end{array}\right] = \left[\begin{array}{c} r^k_H \\ r^k_{L_1} \\ \hline r^k_{L_2} \end{array}\right], \tag{10}$$

Algorithm 2 δ–active Infeasible Interior Point Method

Input:
- H, G, h, g
- Initial guess $\theta^0, \phi^0 > 0, s^0 > 0, k_{\max}$
- Select $\delta, \alpha_{\min}$
- Tolerance $\epsilon > 0$

Output: Optimal θ.

Algorithm:

1: Set $k = 0$ and compute $\mu^0 := \frac{(\phi^0)^T s^0}{n_i}$, $e^0_{tol} := \|b^0\|_\infty$.
2: Compute $V_0 := GH^{-1}, V := V_0 G^T$.
3: **while** $\mu^k > \epsilon$ and $e^k_{tol} > \epsilon$ **do**
4: Compute δ–active set $\mathcal{N}^k_A(\delta)$.
5: Solve (11) for $\Delta\hat{\phi}^k_1$ with Cholesky factorization. Compute $\Delta\hat{\theta}^k$ from (12). Set $\Delta\theta^k := \Delta\hat{\theta}^k$, then compute $\Delta\phi^k$ and Δs^k from (A.1) and (A.2).
6: Choose α^k as mentioned in Algorithm 1.
7: **if** ($\alpha^k < \alpha_{\min}$ or $k > k_{\max}$) **then**
8: Replace δ with 1.5δ and go to step 1.
9: **end if**
10: Compute $(\theta^{k+1}, \phi^{k+1}, s^{k+1}) := (\theta^k, \phi^k, s^k) + \alpha^k(\Delta\theta^k, \Delta\phi^k, \Delta s^k)$.
11: Compute $\mu^{k+1} := \frac{(\phi^{k+1})^T s^{k+1}}{n_i}$, $e^{k+1}_{tol} := \|b^{k+1}\|_\infty$.
12: Increment k by 1.
13: **end while**

where $\begin{bmatrix} G^k_1 \\ G^k_2 \end{bmatrix} = U^k G$, $\begin{bmatrix} \Delta\phi^k_1 \\ \Delta\phi^k_2 \end{bmatrix} = U^k\Delta\phi^k$, $\begin{bmatrix} r^k_{L_1} \\ r^k_{L_2} \end{bmatrix} = U^k r^k_L$, $U^k \in \mathbb{R}^{n_i \times n_i}$ is a permutation matrix, $W^k_1 \in \mathbb{R}^{n^k_a \times n^k_a}, W^k_2 \in \mathbb{R}^{(n_i - n^k_a)\times(n_i - n^k_a)}, \|W^k_1\|_\infty < \delta$, and n^k_a represents the number of δ–active constraints.

Rather than solving (10), it is proposed that the following approximate system be solved

$$\left(U^k_1 V (U^k_1)^T + W^k_1\right)\Delta\hat{\phi}^k_1 = U^k_1 V_0 r^k_H - r^k_{L_1}, \tag{11}$$

where

$$\Delta\hat{\theta}^k = H^{-1}(r^k_H - (G^k_1)^T\Delta\hat{\phi}^k_1), \tag{12}$$

$V := GH^{-1}G^T, V_0 := GH^{-1}, U^k := \begin{bmatrix} U^k_1 \\ U^k_2 \end{bmatrix}$ and $U^k_1 \in \mathbb{R}^{n^k_a \times n_i}$. An IPM in which (11) is solved instead of (7) is called a δ–active based IPM, and it is described in Algorithm 2. It can be shown (Shahzad et al., 2010) that the error in solving (11) converges to zero for large values of k. Note that V and V_0 are computed outside the loop. In the LHS of (11), the matrix $U^k_1 V (U^k_1)^T$ can be formed by just picking the rows and columns of V according to U^k_1. In the RHS of (11), $U^k_1 V_0$ can be computed in a similar fashion. This means that no matrix triple product is involved in computing the Hessian matrix in this formulation.

4.1 Selection of δ

The computational cost of Algorithm 2 depends upon δ. Furthermore, a small value of δ might lead to blocking of the search direction, which means that α^k becomes too small, and the next iterate will be close to previous one. To determine an appropriate value of δ, and to avoid blocking, a scheme is proposed at line 8 of Algorithm 2. Any small value of $\alpha_{\min}$ can be used. We have used $\alpha_{\min} := 10^{-4}, k_{\max} := 30$ in the numerical tests presented

Table 1. Computational cost of an IPM per iteration.

Algorithm	Flops
Algorithm 1	$\frac{1}{3}n_d^3 + \frac{1}{2}n_c n_d^2 + \frac{11}{2}n_d^2 + 4n_c n_d$
Algorithm 2	$\frac{1}{3}(n_a^k)^3 + 2(2n_d + n_c + n_i + n_a^k)n_d + \frac{5}{2}(n_a^k)^2$

in Section 6. δ computed for the first QP is used as an initial guess for all other QPs. It is observed from numerical experiments that δ selected in this way does not change for all other QPs in the same benchmark problem. This means that we are using one fixed value of δ in all QPs in a benchmark problem. Note that in the computation of $\mathcal{N}_A^k(\delta)$ if no such w_i^k exists that satisfies (9) i.e. $\mathcal{N}_A^k(\delta) = \emptyset$, then we solve the unconstrained problem $H\theta = -h$ to get the optimal θ.

4.2 Warm-start

We consider the situation in which a QP (4) has been solved by Algorithm 2 at the previous time instant for a state estimate x^{old}, and we have an optimal solution $\theta^* := \left[(u_0^*)^T\ (u_1^*)^T\ \dots\ (u_{N-1}^*)^T\right]^T$. At the current time instant, we have to solve a similar QP but with a different state estimate x^{new}. This changes $g(x^{\text{old}})$ to $g(x^{\text{new}})$ and $h(x^{\text{old}})$ to $h(x^{\text{new}})$, while H, G remains constant in the new QP. In predictive control, we compute the future inputs along with the current input based on the plant model. So we initialize $\theta^0 := \left[(u_1^*)^T\ (u_2^*)^T\ \dots\ (u_{N-1}^*)^T\ (u_{N-1}^*)^T\right]^T$. for the new QP. Inspired from the observation that the active set does not change much from one QP to the next in active set methods for short sampling periods (Ferreau et al., 2007), we initialize ϕ^0 and s^0 such that the δ–active constraints at the solution of the previous QP remains δ–active in the first iteration of the new QP. Secondly, to make the starting point well-centred and away from the boundary of constraints to avoid blocking, we suggest to make the pairwise products $s_i^0\phi_i^0$ equal to each other.

Let $\mathcal{N}_A^*(\delta)$ denote the set of δ–active set at the solution of the previous QP. Then w^0 for the new QP is calculated as

$$w_i^0 := \begin{cases} k_1\,\delta & \text{for all } i \in \mathcal{N}_A^*(\delta) \\ \dfrac{1}{k_1}\,\delta & \text{otherwise,} \end{cases} \tag{13}$$

for a given $0 < k_1 < 1$. For a given μ^0, we want

$$s_i^0\phi_i^0 = \mu^0. \tag{14}$$

Since $s_i^0 = \phi_i^0 w_i^0$, from (14) we have

$$s_i^0 := \sqrt{w_i^0\mu^0}, \quad \phi_i^0 := \sqrt{\frac{\mu^0}{w_i^0}}. \tag{15}$$

5. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational complexity of Algorithms 1 and 2 are measured in flops. The cost of each IPM per iteration is given in Table 1. Note that only high-order terms are mentioned in Table 1. However, all terms, including the linear ones, are taken into account in the numerical results presented in Section 6.

6. NUMERICAL RESULTS

In this section, we compare the performance of our proposed Algorithm 2 with warm-start over the conventional IPM as described in Algorithm 1 with cold-start. To evaluate the performance, we consider four benchmark problems, which are available online (Diehl and Ferreau, 2008). Each problem provides a sequence of neighbouring QPs. The data corresponding to the size of each problem is given in Table 2, where nQP denotes the number of QPs and Δt is the sampling time

6.1 Problem 1

This problem considers a system of nine masses connected by springs. The objective is to regulate the system from a perturbed state to a steady-state position. One end of the chain is fixed to a wall, while the other end is free, and its three velocity components are used as inputs. The inputs are bounded below and above. Initially, a perturbation is exerted to the chain by moving the free end with a given constant velocity for 3 seconds. See Ferreau et al. (2008) for more details.

Fig. 1(a) shows the performance of our proposed warm-start strategy over conventional cold-start IPM. To see the effect of initial state x on the computational cost of the algorithm used and on the number of δ–active constraints, we have plotted $\|h(\bar{x})\|_2$ in Fig. 1(b), because data for x was not available in the collection of benchmark problems. Note that $h(\bar{x}) := D\bar{x}$, which implies $\|h(\bar{x})\|_2 \propto \|\bar{x}\|_2$. The number of δ–active constraints, and the change in the number of δ–active constraints are shown in Fig. 1(c) and Fig. 1(d) respectively.

6.2 Problem 2

This is the same problem as Problem 1 with the only difference that it has state constraints as well. The state constraints are imposed in a way that the chain does not hit the vertical wall. See Ferreau et al. (2008) for more details. Results are shown in Fig. 2.

6.3 Problem 3

In this problem the objective is to control a direct injection turbo charged Diesel engine. Inputs are the slew rates of the exhaust gas recirculation (EGR) valve and the position of the variable geometry turbocharger (VGT). The inputs are bounded below and above, and states constraints are also imposed. See Ferreau et al. (2007) for more details. Results are shown in Fig. 3, which indicate that a sharp increase in the initial state (Fig. 3(b)) leads to a sharp increase in the change in number of δ–active constraints (Fig. 3(d)), which results in a sudden increase in the computational cost of Algorithm 1 and Algorithm 2 (Fig. 3(a)).

6.4 Problem 4

The problem of trajectory generation for the load movement of a boom crane is considered. The inputs are bounded and constraints on states are also considered. See Arnold et al. (2007) for more details. Results are shown in Fig. 4.

The computational advantage of our proposed warm-start strategy (Algorithm 2) over conventional IPM (Algorithm 1) is summarised in Table 4. Let $\theta_1^*(t), \theta_2^*(t)$ be

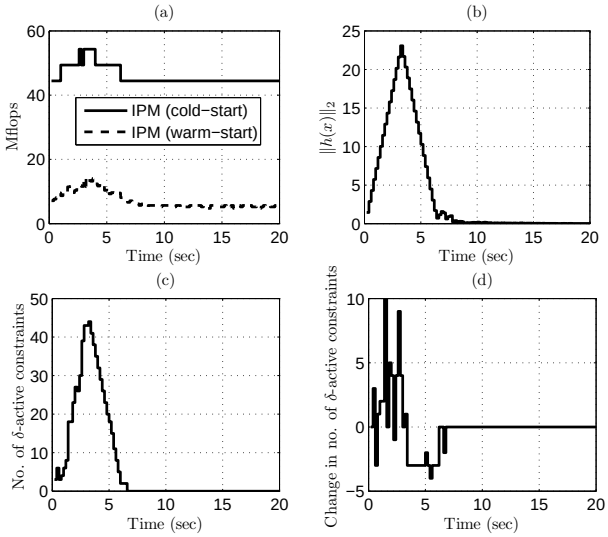

Fig. 1. (a) Comparison of computational cost of Algorithm 1 with cold start and Algorithm 2 with warm-start. (b) $\|h(\bar{x})\|_2$ indicates a measure corresponding to $\bar{x}$, because $h(\bar{x}) := D\bar{x}$. (c) Number of δ–active constraints in the last iteration of Algorithm 2 for each QP. (d) Change in number of δ–active constraints in the last iteration of Algorithm 2 for each QP.

Table 2. Data for benchmark problems.

Problem	nQP	Δt	n	m	N	n_d	n_b	n_c
1	101	0.2	57	3	80	240	480	0
2	101	0.2	57	3	80	240	480	709
3	600	0.05	2	2	10	20	40	40
4	921	0.1	4	1	57	57	72	160

Table 3. Parameters for Algorithm 2 for benchmark problems.

Problem	δ	μ^0	k_1
1	10^2	0.3	0.01
2	10^2	0.3	0.01
3	10^4	0.3	0.01
4	10^2	0.3	0.01

the solution of a QP at time t using Algorithms 1 and 2, respectively. We define the maximum relative difference between the solution of Algorithms 1 and 2 as

$$\Delta\mathcal{J}^* := \max_{t\in[0,t_f]} \frac{|\mathcal{J}(\theta_1^*(t)) - \mathcal{J}(\theta_2^*(t))|}{|\mathcal{J}(\theta_1^*(t))|} \times 100, \qquad (16)$$

where t_f is the final time in simulation, and it is shown in Table 4. This new strategy saves about 30–74% in computations in the worst case, when we have a sudden large change in the current state. The % in worst case is calculated as $\frac{f_1-f_2}{f_1} \times 100$, where $f_i := \max_{t\in[0,t_f]} \text{Mflops}_{\text{Algo}_i}(t)$ for $i = 1, 2$, $\text{Mflops}_{\text{Algo}_1}$ and $\text{Mflops}_{\text{Algo}_2}$ represent the computational cost of Algorithm 1 and 2, respectively.

7. CONCLUSIONS

We have presented a new warm-start strategy based on an interior-point method to solve a sequence of QPs arising in predictive control. The proposed scheme is implemented on a number of benchmark problems, in which each new QP data is perturbed from the previous QP data.

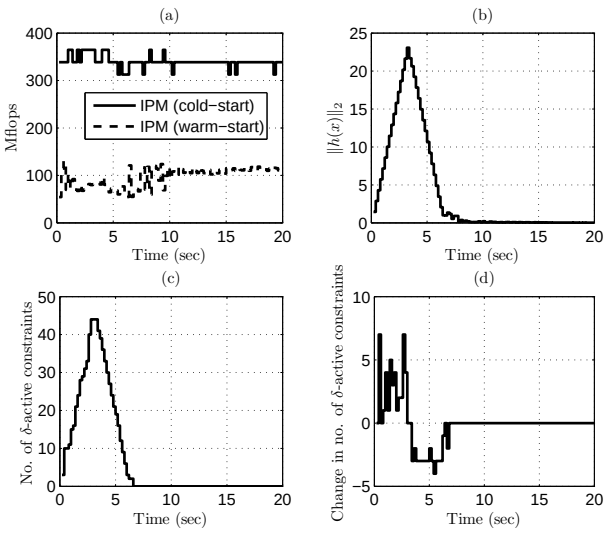

Fig. 2. (a) Comparison of computational cost of Algorithm 1 with cold start and Algorithm 2 with warm-start. (b) $\|h(\bar{x})\|_2$ indicates a measure corresponding to $\bar{x}$, because $h(\bar{x}) := D\bar{x}$. (c) Number of δ–active constraints in the last iteration of Algorithm 2 for each QP. (d) Change in number of δ–active constraints in the last iteration of Algorithm 2 for each QP.

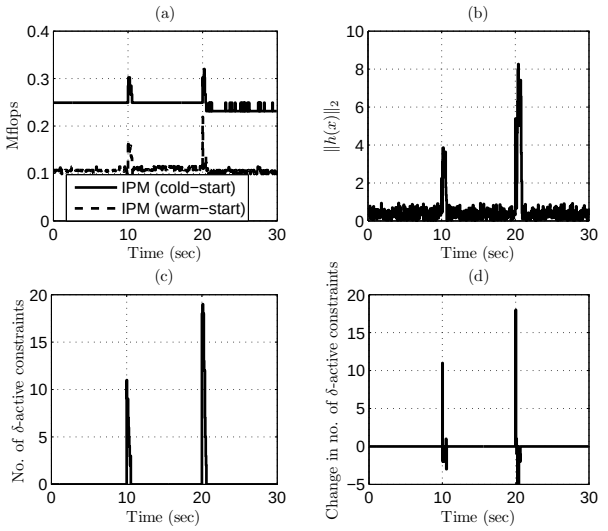

Fig. 3. (a) Comparison of computational cost of Algorithm 1 with cold start and Algorithm 2 with warm-start. (b) $\|h(\bar{x})\|_2$ indicates a measure corresponding to $\bar{x}$, because $h(\bar{x}) := D\bar{x}$. (c) Number of δ–active constraints in the last iteration of Algorithm 2 for each QP. (d) Change in number of δ–active constraints in the last iteration of Algorithm 2 for each QP.

Numerical results indicate that the proposed warm-start strategy depends upon the size of perturbation. We have achieved a reduction in computational cost of about 30–74% in the worst case, when compared to a cold-start interior-point method.

The proposed warm-start technique can also be used to a wider class of problems, in which it is necessary to solve a sequence of neighbouring QPs to get a solution.

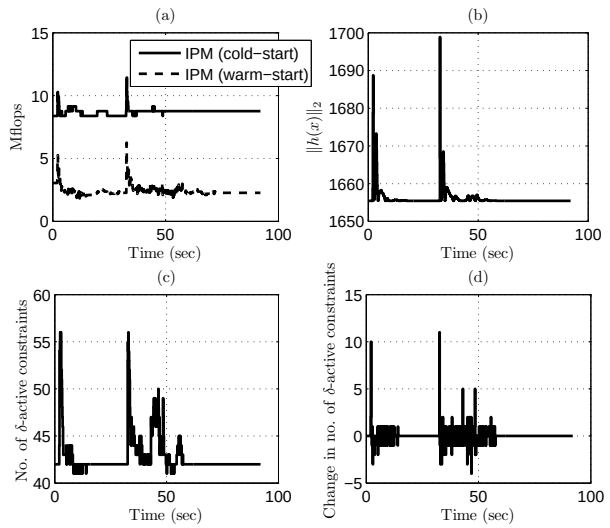

Fig. 4. (a) Comparison of computational cost of Algorithm 1 with cold start and Algorithm 2 with warm-start. (b) $\|h(\bar{x})\|_2$ indicates a measure corresponding to $\bar{x}$, because $h(\bar{x}) := D\bar{x}$. (c) Number of δ–active constraints in the last iteration of Algorithm 2 for each QP. (d) Change in number of δ–active constraints in the last iteration of Algorithm 2 for each QP.

Table 4. Reduction in computational cost of Algorithm 2 with warm-start over Algorithm 1 with cold start in solving benchmark problems.

Problem	**Worst case** (%)	$\Delta\mathcal{J}^*$ (%)
1	73.6	1.6×10^{-2}
2	64.2	3.6×10^{-3}
3	31.6	8.8×10^{-4}
4	43.3	6.3×10^{-2}

REFERENCES

Arnold, E., Neupert, J., Sawodny, O., and Schneider, K. (2007). Trajectory tracking for boom cranes based on nonlinear control and optimal trajectory generation. In *16th IEEE International Conference on Control Applications*, 1444–1449. Singapore.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Diehl, M. and Ferreau, H.J. (2008). Online QP benchmark collection. URL `http://www.kuleuven.be/optec/software/onlineQP`.

Ferreau, H.J., Bock, H.G., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit MPC. *International Journal of robust and Nonlinear Control*, 18(8), 816–830.

Ferreau, H.J., Ortner, P., Langthaler, P., del Re, L., and Diehl, M. (2007). Predictive control of a real-world diesel engine using extended online active set strategy. *Annual Reviews in Control*, 31, 293–301.

Gondzio, J. and Grothey, A. (2008). A new unblocking technique to warmstart interior point methods based on sensitivity analysis. *SIAM J. Optim.*, 19(3), 1184–1210.

John, E. and Yildirim, E.A. (2008). Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, 41(2), 151–183.

Karmarker, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, (4), 373–395.

Shahzad, A., Kerrigan, E.C., and Constantinides, G.A. (2010). A fast well-conditioned interior point method for predictive control. In *49th IEEE Conference on Decision and Control*. Atlanta, Georgia, USA.

Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.

Wright, S.J. (1997). *Primal-Dual Interior Point Methods*. SIAM Publications.

Yildirim, E.A. and Wright, S.J. (2002). Warm-start strategies in interior-point methods for linear programming. *SIAM J. Optim.*, 12(3), 782–810.

Appendix A. DEFINITION OF MATRICES AND VECTORS

$$H := 2\left(\Psi + \Gamma^T \Omega \Gamma\right) + \Gamma^T \Pi + \Pi^T \Gamma,$$
$$h(\bar{x}) := D\bar{x},$$

where

$$\Psi := I_N \otimes R, D := 2\left[\left(\Gamma^T \Omega + \Pi^T\right)\Phi + \bar{M}\right],$$
$$\Phi := \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \Gamma := \begin{bmatrix} B & 0 & \cdots & 0 & 0 \\ AB & B & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & AB & B \end{bmatrix},$$
$$\Omega := \begin{bmatrix} I_{N-1} \otimes Q & 0 \\ 0 & P \end{bmatrix}, \Pi := \begin{bmatrix} 0 & I_{N-1} \otimes M \\ 0 & 0 \end{bmatrix},$$
$$\bar{M} := [M \ 0 \ \cdots \ 0] \in \mathbb{R}^{n \times n_d}.$$

$$G := \begin{bmatrix} G_b \\ G_c \end{bmatrix}, \ g(\bar{x}) := \begin{bmatrix} g_b \\ g_c(\bar{x}) \end{bmatrix}, W^k := \begin{bmatrix} W_b^k & 0 \\ 0 & W_c^k \end{bmatrix},$$

where

$$G_b := \begin{bmatrix} I_{n_d} \\ -I_{n_d} \end{bmatrix}, \ g_b := \begin{bmatrix} \mathbf{1}_{n_d} u_{\text{ub}} \\ -\mathbf{1}_{n_d} u_{\text{lb}} \end{bmatrix},$$
$$G_c := \begin{bmatrix} J_0 B + E_0 & 0 & \cdots & 0 \\ J_1 AB & J_1 B + E_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ J_{N-1} A^{N-1} B & J_{N-1} A^{N-2} B & \cdots & J_{N-1} B + E_{N-1} \end{bmatrix}$$
$$g_c(x) := \begin{bmatrix} d_0 - J_0 A x \\ \vdots \\ d_{N-1} - J_{N-1} A^N x \end{bmatrix}.$$

$$W^k := (\Phi^k)^{-1} S^k,$$
$$\Delta\phi^k := (S^k)^{-1}\left(\Phi^k r_G^k + r_S^k + \Phi^k G \Delta\theta^k\right), \quad \text{(A.1)}$$
$$\Delta s^k := (\Phi^k)^{-1}\left(r_S^k - S^k \Delta\phi^k\right), \quad \text{(A.2)}$$

where

$$r_G^k := G\theta^k - d + s^k, r_S^k := -\Phi^k S^k \mathbf{1}_{n_i},$$
$$r_H^k := -\left(H\theta^k + F^T \nu^k + G^T \phi^k\right),$$
$$r_L^k := r_G^k - (\Phi^k)^{-1}(r_S^k + \sigma\mu^k \mathbf{1}_{n_i}).$$