
Interior-point algorithms for nonlinear model predictive control

Adrian G. Wills¹ and William P. Heath²

¹ School of Elec. Eng. & Comp. Sci., Uni. of Newcastle, Australia
Adrian.Wills@newcastle.edu.au

² Control Systems Centre, School of Electrical and Electronic Engineering, The University of Manchester, United Kingdom William.Heath@manchester.ac.uk

Summary. In this contribution we present two interior-point path-following algorithms that solve the convex optimisation problem that arises in recentred barrier function model predictive control (MPC), which includes standard MPC as a limiting case. However the optimisation problem that arises in nonlinear MPC may not be convex. In this case we propose sequential convex programming (SCP) as an alternative to sequential quadratic programming. The algorithms are appropriate for the convex program that arises at each iteration of such an SCP.

1 Introduction

It is often the case in nonlinear model predictive control (NMPC) that the system dynamics are nonlinear in the states and inputs while the constraint sets for both state and input sequences are assumed to be convex [6]. The nonlinear dynamics can, and often do, make the associated control optimisation problem non-convex, and hence more difficult to solve.

One popular strategy for solving non-convex problems is sequential quadratic programming (SQP) where an iterative search procedure is used and the search directions are computed via a quadratic program [11]. Specific adaptations of this SQP approach have been developed for nonlinear model predictive control (NMPC) which take advantage of the sparse structure typical to these problems [4].

An emergent alternative approach for solving non-convex optimisation problems, particularly favoured in topology optimisation, is sequential convex programming (SCP) [16]. SCP is similar in concept to SQP but the search direction is obtained by solving a more general convex programme (CP) in place of the less general quadratic programme (QP). While it has been recognised that this generalisation pays dividends for some mechanical engineering problems [16], the benefits of using SCP for NMPC are as yet undetermined. A thorough evaluation of the relative merits of SQP and SCP for nonlinear MPC is beyond the scope of this contribution.

Rather, in this contribution we consider two interior-point algorithms useful for solving quite general convex programming problems, for example, of the type that arise at each iteration of an SCP approach. These algorithms are almost standard except that they are geared towards solving convex optimisation problems with a weighted barrier function appearing in the cost. This slight generalisation allows a parsimonious treatment of both barrier function based model predictive control [14] and “standard” model predictive control, which can be identified as a special

limiting case. The benefit of including a weighted barrier function is that iterations stay strictly inside the boundary and fewer iterations are needed to converge. This barrier approach is called r-MPC and has been successfully applied to an industrial edible oil refining process as discussed in [15].

Note that due to page limitations all proofs have been omitted and can be found in [13].

2 Nonlinear Model Predictive Control

In what follows we describe NMPC and formulate an optimisation problem which is convex except for the nonlinear equality constraints that represent the system dynamics. This motivates a very brief discussion of SCP which leads to the main theme of this contribution being the two algorithms in Sections 3 and 4. The problem may be formulated as follows. Consider the following discrete-time system with integer k representing the current discrete time event,

$$x(k+1) = f(x(k), u(k)). \quad (1)$$

In the above, $u(k) \in \mathbb{R}^m$ is the system input and $x(k) \in \mathbb{R}^n$ is the system state. The mapping f is assumed to be differentiable and to satisfy $f(0,0) = 0$. Given some positive integer N let \mathbf{u} denote a sequence of control moves given by $\mathbf{u} = \{u(0), u(1), \dots, u(N-1)\}$ and let \mathbf{x} denote a state sequence given by $\mathbf{x} = \{x(0), x(1), \dots, x(N)\}$.

For the purposes of this contribution we require that the input *sequence* \mathbf{u} should lie within a compact and convex set \mathbb{U} while the state *sequence* \mathbf{x} should lie in the closed and convex set \mathbb{X} . Let $V_N(\mathbf{x}, \mathbf{u})$ denote the objective function associated with prediction horizon N . We assume that V_N is a convex function. The control strategy for NMPC may be described as follows: at each time interval k , given the state $x(k)$, compute the following and apply the first control move to the system.

$$(\mathcal{MPC}) : \min_{\mathbf{x}, \mathbf{u}} V_N(\mathbf{x}, \mathbf{u}), \text{ s.t. } x_0 = x(k), \quad x_{i+1} = f(x_i, u_i), \quad \mathbf{x} \in \mathbb{X}, \quad \mathbf{u} \in \mathbb{U}.$$

We can associate with the sets \mathbb{X} and \mathbb{U} , respectively, *gradient recentred self-concordant barrier functions* B_x and B_u [14]. This allows (\mathcal{MPC}) to be expressed as the limiting case when $\mu \rightarrow 0$ of the following class of optimisation problems [3].

$$(\mathcal{MPC}_\mu) : \min_{\mathbf{x}, \mathbf{u}} V_N(\mathbf{x}, \mathbf{u}) + \mu B_x(\mathbf{x}) + \mu B_u(\mathbf{u}) \text{ s.t. } x_0 = x(k), \quad x_{i+1} = f(x_i, u_i).$$

The above class of optimisation problems (\mathcal{MPC}_μ) have, by construction, a convex cost function and nonlinear equality constraints. If these equality constraints are modelled locally by a linear approximation, then the resulting problem is convex and more readily soluble. This is the impetus for using SCP; at each iteration of the method a local linear approximation to the nonlinear equality constraints is formed and the corresponding CP is solved and the solution provides a search direction, which is then used in a simple line-search method to reduce a merit function. It is beyond the scope of this contribution to provide a detailed SCP algorithm, but standard texts on SQP offer the main themes (see e.g. Chapter 18 from [11]).

We turn our attention to solving (\mathcal{MPC}_μ) where the nonlinear equalities have been linearised, thus resulting in a convex optimisation problem. To this end we present two algorithms in Sections 3 and 4, based on path-following interior-point methods. The first algorithm is a two stage long-step path-following algorithm for the case where the convex constraint set is closed and bounded with non-empty interior. The second algorithm is based on a primal-dual path-following method which is less general but more efficient since it is suitable for the case where the convex constraint set is a self-scaled cone with non-empty interior.

3 Barrier generated path-following algorithm

Disregarding previous notation, consider the following convex optimisation problem (\mathcal{P}) and its closely related class of barrier generated problems (\mathcal{P}_μ) .

$$(\mathcal{P}) : \min_x f(x) \quad \text{s.t.} \quad x \in G, \quad (\mathcal{P}_\mu) : \min_x \frac{1}{\mu} f(x) + F(x).$$

In the above, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is assumed to be a twice continuously differentiable convex function, G is a closed and bounded convex subset in \mathbb{R}^n with non-empty relative interior denoted G° , and F is a ν -self-concordant barrier function for G [7].

Restricting the feasible domain G to be closed and bounded means that the barrier function F is strictly convex and attains its unique minimum over the interior of G (see Proposition 2.3.2 in [7]). Hence (\mathcal{P}_μ) has a unique minimiser for all $\mu > 0$, which we denote by $x(\mu)$, and the set $\mathcal{C} \triangleq \{x(\mu) : \mu > 0\}$ of solutions to (\mathcal{P}_μ) is known as the central-path of (\mathcal{P}) . In fact, the solution $x(\mu)$ coincides with the solution to (\mathcal{P}) in the limit as $\mu \rightarrow 0$.

However, we are interested in solving (\mathcal{P}_μ) for the case where μ may be chosen as a fixed and relatively large positive number (e.g. for $\mu = 0.1$). This means that standard interior-point algorithms are not directly applicable since they are geared towards solving (\mathcal{P}_μ) for $\mu \rightarrow 0$. Nevertheless, as demonstrated in this section, straightforward generalisations of standard algorithms allow for the case of solving (\mathcal{P}_μ) with $\mu \gg 0$.

In terms of the algorithm structure itself, the intuition is as follows. At the k 'th iteration, given a point x_k and value $\mu_k > 0$ such that x_k is close to $x(\mu_k)$ the algorithm selects $\mu_{k+1} < \mu_k$ and generates a new point x_{k+1} that is close to $x(\mu_{k+1})$. This process is repeated until the barrier weighting μ_k converges to some prescribed constant value $\mu_c > 0$ with the corresponding x_k 's converging to the point $x(\mu_c)$. A damped Newton method is used to generate the new point x_{k+1} , and, we employ a long-step approach, so that aggressive reductions in μ_k are allowed. This has the consequence of increasing theoretical complexity bounds but tends to be favoured for practical algorithms since these bounds are usually conservative.

With this in mind, it remains to find an initial point x_0 and value μ_0 such that x_0 is close to $x(\mu_0)$. This is the subject of Section 3.1 which discusses an initialisation algorithm.

Further to this, however, is a minor technical issue. The algorithm presented in this section requires that the objective function f is linear, but we are interested in a more general choice of f (typically quadratic). Nevertheless, it is straightforward to embed problem (\mathcal{P}) into a slightly larger formulation which does have a linear objective. In fact, we do this now using the epigraphic form [2].

$$(\mathcal{A}) : \min_{(t,x)} t \quad \text{s.t.} \quad f(x) \leq t, \quad x \in G, \quad (\mathcal{A}_\mu) : \min_{(t,x)} \frac{1}{\mu} t - \ln(t - f(x)) + F(x),$$

where $t \in \mathbb{R}$ and $-\ln(t - f(x))$ is assumed to be a ν_f -self-concordant barrier function for $f(x) \leq t$ (e.g. in the case where f is quadratic then this assumption is satisfied with $\nu_f = 2$). Note that (\mathcal{A}_μ) has a unique minimiser, denoted by $(t(\mu), x(\mu))$. The following lemma shows that if $(t(\mu), x(\mu))$ minimises (\mathcal{A}_μ) then $x(\mu)$ minimises (\mathcal{P}_μ) and it therefore appears reasonable to solve problem (\mathcal{A}_μ) and obtain the minimiser for problem (\mathcal{P}_μ) directly.

Lemma 1. *Let $(t(\mu), x(\mu))$ denote the unique minimiser of problem (\mathcal{A}_μ) for some $\mu > 0$, then $x(\mu)$ also minimises problem (\mathcal{P}_μ) for the same value of μ .*

3.1 Initialisation stage

The purpose of an initialisation stage is to generate a point (t_0, x_0) and a value μ_0 such that (t_0, x_0) is close to $(t(\mu_0), x(\mu_0))$ (the minimiser of (\mathcal{A}_{μ})), which enables the main stage algorithm to progress as described above. The usual approach is to minimise the barrier function $F_{\mathcal{A}}(t, x) \triangleq -\ln(t - f(x)) + F(x)$, which in turn provides a point at the ‘‘centre’’ of the constraint set and close to the central-path of (\mathcal{A}) for μ large enough.

However, by construction $F_{\mathcal{A}}(t, x)$ is unbounded below as $t \rightarrow \infty$. This problem is overcome, in the literature, by including a further barrier on the maximum value of t resulting in the combined barrier function

$$F_{\mathcal{B}}(t, x) \triangleq -\ln(R - t) + F_{\mathcal{A}}(t, x), \quad F_{\mathcal{A}}(t, x) \triangleq -\ln(t - f(x)) + F(x). \quad (2)$$

The unique minimiser of the barrier function $F_{\mathcal{B}}$ and the central path of problem (\mathcal{A}) are related in the following important way.

Lemma 2. *Let (t^*, x^*) denote the unique minimiser of $F_{\mathcal{B}}(t, x)$. Then (t^*, x^*) coincides with a point on the central path of problem (\mathcal{A}) identified by $\mu = R - t^*$.*

Therefore, minimising the barrier function $F_{\mathcal{B}}$ actually provides a point on the central path of (\mathcal{A}) , which is precisely the goal of the initialisation stage. Furthermore, Nesterov and Nemirovskii’s initialisation algorithm (with associated complexity bounds – see Section 3.2.3 in [7]) is directly applicable in this case. Their approach may be described as follows.

Let $\lambda(h, z)$ denote the Newton decrement for a 1-strongly self-concordant non-degenerate function h at the point z defined as (see Section 2.2.1 in [7]),

$$\lambda(h, z) = \left(\nabla^T h(z) [\nabla^2 h(z)]^{-1} \nabla h(z) \right)^{1/2}.$$

Algorithm 1 *Given $x \in G^\circ$ and $f(x) < t < R$, let $z_0 = (t, x)$ and choose $\kappa \in (0, 1)$, $\gamma \in (0, 1)$ and $\beta \in (2 - 3^{1/2}, 1)$. Define the function $\phi_\eta(z) \triangleq \eta(-\nabla F_{\mathcal{B}}(z_0), z) + F_{\mathcal{B}}(z)$. Let $\eta = 1$ and iterate the following steps.*

1. *If $\lambda(F_{\mathcal{B}}, z) \leq \beta$ then let $z^* = z$ and stop.*
2. *If $\lambda(\phi_\eta, z) \leq \gamma$ then update η via $\eta \leftarrow \kappa\eta$.*
3. *Update z according to damped Newton step $z \leftarrow z - \frac{1}{1 + \lambda(\phi_\eta, z)} [\nabla^2 \phi_\eta(z)]^{-1} \nabla \phi_\eta(z)$.*

The following proposition shows that (t^*, x^*) combined with a careful choice of initial weighting parameter provide suitable initial conditions for the main stage algorithm.

Proposition 1. *Let $(t_0, x_0) = (t^*, x^*)$ and $\mu_0 = -(e^T P e) / (e^T P g)$ where $g \triangleq \nabla F_{\mathcal{A}}(z^*)$ and $P \triangleq [\nabla^2 f_\tau^{\mathcal{A}}(z^*)]^{-1}$ and $e \triangleq [1, 0, \dots, 0]^T \in \mathbb{R}^{n+1}$. Then (t_0, x_0) and μ_0 are suitable initial conditions for the main stage algorithm.*

3.2 Main stage

Consider the point on the central path of (\mathcal{A}) corresponding to some $\mu_0 > 0$, denoted as usual by $(t(\mu_0), x(\mu_0))$. Given some pair (t_0, x_0) close to $(t(\mu_0), x(\mu_0))$, the main stage algorithm follows $(t(\mu), x(\mu))$ towards the point $(t(\mu_c), x(\mu_c))$, where the constant $\mu_c > 0$ is pre-specified by the user.

Algorithm 2 Given (t_0, x_0) and μ_0 such that $\lambda(f_{\mu_0}, (t_0, x_0)) \leq \beta$ (where β is chosen in Algorithm 1), then choose $\epsilon \in (0, 2 - 3^{1/2})$ and $\kappa \in (0, 1)$ and let $\mu = \mu_0$ and $z = (t_0, x_0)$ and iterate the following steps.

1. If $\mu = \mu_c$ and $\lambda(f_\mu, z) \leq \epsilon$ then stop.
2. If $\mu_c \leq \mu$ then let $\mu \leftarrow \max\{\mu_c, \kappa\mu\}$, otherwise let $\mu \leftarrow \min\{\mu_c, \mu/\kappa\}$.
3. If $\mu = \mu_c$ then let $\beta = \epsilon$.
4. Iterate $z \leftarrow z - \frac{1}{1+\lambda(f_\mu, z)}[\nabla^2 f_\mu(z)]^{-1}\nabla f_\mu(z)$ until $\lambda(f_\mu, z) \leq \beta$.

We are interested in the total number of Newton iterations taken in Step 4 in the above algorithm. In the following proposition we bound this number using results from [7].

Proposition 2. *The total number of Newton iterations required in Algorithm 2 is bounded from above by*

$$\mathcal{N}_T = \lceil \log_\kappa(\mu_c/\mu_0) \rceil \mathcal{N}(\beta) + \mathcal{N}(\lambda_*) + \lceil \log_2((2 - 3^{1/2})/\epsilon) \rceil.$$

where $\mathcal{N}(\beta) \triangleq \lceil O(1)(1 + |\kappa - 1|\nu_{\mathcal{A}}^{1/2} + \nu_{\mathcal{A}}(\kappa - 1 - \ln \kappa)) \rceil$ and $\nu_{\mathcal{A}} \triangleq \nu + \nu_f$ and $O(1)$ depends on β only.

4 Self-scaled Cones

The conic form optimisation problem introduced by [7] involves minimisation of a linear objective function over the intersection of an affine subspace and a closed and pointed convex cone. [8] define a class of cones and associated barrier functions which are called *self-scaled*. This class has special properties which allow for efficient interior-point algorithms to be developed [9, 10, 12]. In this section, three types of self-scaled cones are treated, namely the standard non-negative orthant, the second-order cone, and the cone of positive semi-definite symmetric matrices.

The primal-dual path-following algorithm presented below in Section 4.3 is based on [8, 9]. The algorithm finds the point on the primal-dual central path corresponding to the positive real number μ_c . The sections preceding this (namely Sections 4.1 and 4.2) give background material and definitions used in association with the algorithm and notation is as follows. Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and let K denote a self-scaled cone in E . The standard primal and dual conic form optimisation problems considered in the remainder of this section are given by

$$(\mathcal{PC}) : \min_x \langle c, x \rangle, \text{ s.t. } Ax = b, x \in K, \quad (\mathcal{DC}) : \max_{y,s} \langle b, y \rangle, \text{ s.t. } A^*y + s = c, s \in K_*.$$

In the above, $c \in E$, $b \in \mathbb{R}^m$ and $A : E \rightarrow \mathbb{R}^m$ is a surjective linear operator and K_* denotes the cone dual to K (which is K itself). Let K° denote the interior of K .

We are interested in finding the point on the central path of (\mathcal{PC}) corresponding to the positive scalar μ_c . This may be achieved by solving the following primal-dual central path minimisation problem for $\mu = \mu_c$.

$$(\mathcal{PD}_\mu) : \min_{x,y,s} \frac{1}{\mu} \langle s, x \rangle + F(x) + F_*(s), \quad \text{s.t. } Ax = b, A^*y + s = c$$

In the above, F is a ν -self-scaled barrier function for the cone K and F_* is a sign modified Fenchel conjugate of F for the cone K_* – see [8]. Denote the set of minimisers of (\mathcal{PD}_μ) for $\mu \in (0, \infty)$ by \mathcal{CPD} ; this set is typically called the primal-dual

central path. Let $S^\circ(\mathcal{PD})$ denote the set of strictly feasible primal-dual points for (\mathcal{PD}) given by

$$S^\circ(\mathcal{PD}) = \{(x, s, y) \in E \times E \times \mathbb{R}^m : Ax = b, A^*y + s = c, x \in K^\circ, s \in K^\circ\}.$$

The algorithm presented in section 4.3 uses a predictor-corrector path-following strategy. Such strategies typically start from a point close to the primal-dual central path and take a predictor step which aims for $\mu = 0$. This direction is followed until the new points violate some proximity measure of the central path, at which time a series of centring steps are taken. These steps aim for the “closest” point on the central path and cease when suitable proximity is restored.

There exist many proximity measures for the central path, but [9] use a so-called functional proximity measure which is a global measure in the sense that it has meaning everywhere on $S^\circ(\mathcal{PD})$, and is defined as

$$\gamma(x, s) = F(x) + F_*(s) + \nu \ln(\mu(x, s)) + \nu, \quad \mu(x, s) = \frac{1}{\nu} \langle s, x \rangle.$$

A region of the central path, denoted $\mathcal{F}(\beta)$, that uses this measure is defined by

$$\mathcal{F}(\beta) = \{(x, y, s) \in S^\circ(\mathcal{PD}) : \gamma(x, s) \leq \beta\}.$$

4.1 Centring direction

Given a strictly feasible point $(x, s, y) \in S^\circ(\mathcal{PD})$ and $\omega \in K^\circ$ such that $\nabla^2 F(\omega)x = s$, the centring direction (d_x, d_s, d_y) is defined as the solution to the following.

$$\nabla^2 F(\omega)d_x + d_s = -\frac{1}{\mu(s, x)}s - \nabla F(x), \quad Ad_x = 0, \quad A^*d_y + d_s = 0. \quad (3)$$

Let $u \in E$ and $v \in K^\circ$. Define $\sigma_v(u) = \frac{1}{\alpha}$, where $\alpha > 0$ is the maximum possible value such that $v + \alpha u \in K$. It is convenient to define a centring algorithm [9].

Algorithm 3 *Given a strictly feasible initial point $(x_0, s_0, y_0) \in S^\circ(\mathcal{PD})$, let $(x, s, y) = (x_0, s_0, y_0)$ and define the Newton iterates as follows.*

1. *If termination conditions are satisfied then stop.*
2. *Form (d_x, d_y, d_s) by solving (3) with (x, y, s) .*
3. *Update (x, y, s) according to $x \leftarrow x + \alpha d_x$, $s \leftarrow s + \alpha d_s$, $y \leftarrow y + \alpha d_y$, where α is given at each iteration by $\alpha = (\mu(x, s) \sigma_s(\nabla F(x)) + \bar{\sigma})^{-1}$ and $\bar{\sigma} \triangleq \max\{\sigma_x(d_x), \sigma_s(d_s)\}$.*

4.2 Affine scaling direction

Given a strictly feasible point $(x, s, y) \in S^\circ(\mathcal{PD})$ and $\omega \in K^\circ$ such that $\nabla^2 F(\omega)x = s$, the affine scaling direction (p_x, p_s, p_y) is defined as the solution to the following.

$$\nabla^2 F(\omega)p_x + p_s = -s, \quad Ap_x = 0, \quad A^*p_y + p_s = 0. \quad (4)$$

Note that $\langle p_s, p_x \rangle = 0$ from the last two equations of (4). Furthermore, $\langle s, p_x \rangle + \langle p_s, x \rangle = \langle s, x \rangle$ since from the first equation in (4),

$$-\langle s, x \rangle = \langle \nabla^2 F(\omega)p_x + p_s, x \rangle = \langle s, p_x \rangle + \langle p_s, x \rangle.$$

Thus,

$$\langle s + \alpha p_s, x + \alpha p_x \rangle = \langle s, x \rangle + \alpha(\langle s, p_x \rangle + \langle p_s, x \rangle) + \alpha^2 \langle p_s, p_x \rangle = (1 - \alpha) \langle s, x \rangle.$$

Therefore, α can be chosen such that $\langle s + \alpha p_s, x + \alpha p_x \rangle = \nu \mu_c$, i.e. $\alpha = 1 - \mu_c / \mu(s, x)$. In general, it is not always possible to take a step in direction (p_x, p_s, p_y) with step size α calculated here, since this may result in an infeasible point. However, if it is possible to take the full step of size α then the duality gap is equal to $\nu \mu_c$.

4.3 Path-following algorithm

The structure of the algorithm is described as follows: starting from a strictly feasible initial point $(x_0, s_0, y_0) \in S^\circ(\mathcal{PD})$, then firstly a predictor step is computed which aims at reducing the distance between $\mu(x, s)$ and μ_c . The step size is computed in order to maintain iterates within a certain region of the central path. If it is possible to take a step that reduces the gap $|\mu(x, s) - \mu_c|$ to zero, whilst remaining inside the allowed region of the central path, then this step will be taken. After computing the intermediate *predictor* point, the algorithm proceeds to *correct* the iterates towards the central path until they are sufficiently close. This process sets the scene for a new predictor step in which further reduction of the gap $|\mu(x, s) - \mu_c|$ may be achieved.

Algorithm 4 Choose ϵ, β and Δ such that $\epsilon > 0$, $0 < \beta < 1 - \ln(2)$, $\beta < \Delta < \infty$. Given a problem in the form of (\mathcal{PD}) , an initial point $(x_0, s_0, y_0) \in \mathcal{F}(\beta)$ and a value $\mu_c > 0$, let $(x, s, y) = (x_0, s_0, y_0)$ and iterate the following steps.

1. While $\mu(s, x) \neq \mu_c$ and $\gamma(s, x) > \beta$ iterate the following steps.
 - a) Compute the affine scaling direction (p_x, p_s, p_y) by solving (4).
 - b) Update λ with $\lambda \leftarrow 1 - \mu_c/\mu(x, s)$. If $\lambda > 0$ then find $\eta > 0$ such that $\gamma(x + \eta p_x, s + \eta p_s) = \Delta$ and update α with $\alpha \leftarrow \min\{\eta, \lambda\}$.
Otherwise find $\eta < 0$ such that $\gamma(x + \eta p_x, s + \eta p_s) = \Delta$ and $\alpha \leftarrow \max\{\eta, \lambda\}$.
 - c) Update the predictor point (x^+, s^+, y^+) using $x^+ \leftarrow x + \alpha p_x$, $s^+ \leftarrow s + \alpha p_s$, $y^+ \leftarrow y + \alpha p_y$.
 - d) Update the iterate (x, s, y) using the Newton process 3 starting from (x^+, s^+, y^+) and stopping as soon as a point in $\mathcal{F}(\beta)$ is found.
2. Update the iterate (x, s, y) using Algorithm 3 starting from (x, s, y) and stopping as soon as a point in $\mathcal{F}(\epsilon)$ is found.

From Theorem 7.1 in [9], the number of corrector steps n_c in each iteration is bounded by

$$n_c \leq \Delta (\tau - \ln(1 + \tau)), \quad \tau = 0.5 (3\beta/(1 + \beta))^{1/2}. \quad (5)$$

The number of final corrector steps n_{fc} is given by the same relation but with $\beta = \epsilon$.

Also from Theorem 7.1 in [9], if $\mu_0 \geq \mu_c$, then the number of predictor steps $n_{p,1}$ is bounded from above by

$$n_{p,1} \leq \left\lceil \frac{\ln(\mu_0/\mu_c)}{\ln(1/(1 - \delta_1))} \right\rceil, \quad \delta_1 = -\frac{1}{2\nu} c(\Delta, \beta) + \frac{1}{2} \left(\left(\frac{1}{\nu} c(\Delta, \beta) \right)^2 + 4 \frac{1}{\nu} c(\Delta, \beta) \right)^{1/2}, \quad (6)$$

where $c(\Delta, \beta)$ is a positive constant that depends on Δ and β only. Furthermore, in [10] it is noted that Theorem 7.1 from [9] holds for negative values of α . Hence, if $\mu_0 \leq \mu_c$, then the number of predictor steps $n_{p,2}$ is bounded from above by

$$n_{p,2} \leq \left\lceil \frac{\ln(\mu_c/\mu_0)}{\ln(1 - \delta_2)} \right\rceil, \quad \delta_2 = -\frac{1}{2\nu} c(\Delta, \beta) - \frac{1}{2} \left(\left(\frac{1}{\nu} c(\Delta, \beta) \right)^2 + 4 \frac{1}{\nu} c(\Delta, \beta) \right)^{1/2}. \quad (7)$$

Proposition 3. The total number of predictor and corrector steps n_{pc} in the above algorithm is bounded from above by

$$n_{pc} \leq \begin{cases} n_{p,1} n_c + n_{fc} & \text{if } \mu_0 \geq \mu_c, \\ n_{p,2} n_c + n_{fc} & \text{if } \mu_0 \leq \mu_c. \end{cases} \quad (8)$$

5 Conclusion

The two algorithms presented in this contribution are useful for solving convex programming problems where the solution is generalised to be a specific point on the central-path. This finds immediate application to r-MPC which employs a gradient recentred self-concordant barrier function directly into the cost. More generally however, it would appear that these algorithms are useful within a sequential convex programming approach for solving nonlinear model predictive control problems. This latter point is the subject of further research activity.

References

1. Albuquerque, J. and V. Gopal and G. Staus and L. T. Biegler and B. E. Ydstie. "Interior point SQP strategies for structured process optimization problems". *Computers & Chemical Engineering* **2**, S853–S859, (1997).
2. Boyd, S. and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge UK (2004).
3. Fiacco, A. V. and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons Inc. New York, (1968).
4. Gopal, V. and L. T. Biegler. "Large scale inequality constrained optimization and control". *IEEE Control Systems Magazine* **18**(6), 59–68, (1998).
5. Kailath, T. *Linear Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, (1980).
6. Mayne, D. Q., Rawlings, J. B., Rao, C. V. and Scaekaert, P. O. M., "Constrained Model Predictive Control: Stability and optimality", *Automatica.*, **36**, 789–814, (2000).
7. Nesterov, Y. and A. Nemirovskii. *Interior-point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, (1994).
8. Nesterov, Y. E. and M. J. Todd. "Self-scaled cones and interior-point methods in nonlinear programming". *Mathematics of Operations Research* **22**, 1–42, (1997).
9. Nesterov, Y. E. and M. J. Todd. "Primal-dual interior-point methods for self-scaled cones". *SIAM Journal on Optimization* **8**, 324–364, (1998).
10. Nesterov, Y. E., M. J. Todd and Y. Ye. "Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems". *Mathematical Programming* **84**, 227–267, (1999).
11. Nocedal, J. and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, (1999).
12. Renegar, J. *A Mathematical View of Interior-Point Methods in Convex Optimization*. SIAM, Philadelphia, (2001).
13. Wills, A. G. Barrier Function Based Model Predictive Control. PhD thesis. School of Electrical Engineering and Computer Science, University of Newcastle, Australia, (2003).
14. Wills, A. G., Heath, W. P., "Barrier function based model predictive control", *Automatica.*, **40**, 1415–1422, (2004).
15. Wills, A. G., Heath, W. P., "Application of barrier function based model predictive control to an edible oil refining process", *Journal of Process Control.*, **15**, 183–200, (2005).
16. Zillober, C., and K. Schittkowski and K. Moritzen. "Very large scale optimization by sequential convex programming". *Optimization Methods and Software* **19**(1), 103–120, (2004).