Optimization Algorithm Tailored for Embedded Model Predictive Control

author[1]

***Abstract*— Write an abstract here.**

I. INTRODUCTION

Write the introduction here.

II. RELATED WORK

A. Model predictive control

List some literature with information about model predictive control (MPC).

B. Solving optimization problems

Many Algorithms which use primal dual methods to solve several optimization problems are described in literature. Most of them focus on finding solutions with high accuracy. For application as MPC, optimization often do not necessarily has to find such exact solutions. Therefore [1] describes a primal barrier method with focus of getting fast sufficiently exact solutions of quadratic programs, making use of the special MPC problem structure. This paper's contribution is manly based on the algorithm in [1], which has been extended by some numerical robustness approaches. Regularization [6] and iterative refinement if necessary, soft constraints [5]. For application to more general optimization problems, we extend the algorithm to handle quadratic Constrained quadratic programs (QCQPs) and second-order cone programs (SOCPs).

III. PROBLEM STATEMENT

A. Description of a Quadratic Program

Introduce the unchanged problem statement of [1]
The control variables $u(t), \dots, u(t+T-1)$ and the presented states $x(t+1), \dots, x(t+T)$ up to the planning horizon T are rewritten to one optimization variable

$$z = (u(t), x(t+1), \dots, u(t+T-1), x(t+T)) \in \mathbb{R}^{T(m+n)}.$$

The quadratic program (QP) can be formulated as

$$\begin{aligned} \min_z \quad & z^T H z + g^T z \\ \text{s.t.} \quad & Pz \le h, \quad Cz = b \end{aligned} \tag{1}$$

For the use of an infeasible start primal barrier method a logarithmic barrier function

$$\phi(z) = \sum_{i=1}^{lT+l_f} -\log\left(h_i - p_i^T z\right) \tag{2}$$

is introduced the the cost function to handle the inequality constraints.

B. Quadratically Constrained Quadratic Program

{Vorteile von QCQP nennen} If linear constraints as bounds for state and control variable are not sufficient it can be necessary to use nonlinear constraints such as quadratic constraints

$$z^T \Gamma_i z + \beta_i^T z \le \alpha_i, \quad i = 1, \dots, p \tag{3}$$

with positive semidefinite matrices $P_1, \dots, P_p \in \mathbb{R}^{n\times n}$ ([7]).

C. Second-Order Cone Program

A more general form of optimization problems are second-order cone programs (SOCP). We can formulate a lot of different nonlinear convex optimization problems as SOCP ([7]). The constraints of dimension k_i of such SOCP ca be formulated as following inequality:

$$\|A_i z + b_i\|_2 \le c_i^T z + d_i \quad i = 1, \dots, q,$$

where $A_1, \dots, A_q \in \mathbb{R}^{(k_i-1)\times n}$, $b_1, \dots, b_q \in \mathbb{R}^{k_i-1}$, $c_1, \dots, c_q \in \mathbb{R}^n$ and $d_1, \dots, d_q \in \mathbb{R}$

D. Soft Constraints

To use the algorithm in [1] it is necessary that z strictly satisfies the inequality constraints all the time. If the system is moving near its constraints, in case of disturbances it is not guaranteed that its state x is remaining in the feasible area. To avoid such feasibility problems and to make the algorithm more robust we additionally introduce soft constraints. So the algorithm can violate some inequality constraints for the cost of some additional penalty. There are different ways to introduce soft constraints in an optimization algorithm. [5] proposes a method tailored for the algorithm we use without any typically needed additional slack variable.
exact penalty formulation: solution is the same as in the unmodified case. Only for if there is no feasible point that stis

IV. EXTENDED ALGORITHM

A. Generalized Constraints

In the described primal barrier method the gradient and the Hessian of the logarithmic barrier function are necessary. SOCCs in the above mentioned form are not continuously differentiable. Therefore SOCCs in Generalized form [2]

$$\|A_i z + b_i\|_2^2 \le \left(c_i^T z + d_i\right)^2 \quad i = 1, \dots, q \tag{4}$$

can be used.

*This work was not supported by any organization
[1]author hannes.heinemann@st.ovgu.de

B. Extended Problem Statement

The algorithm of [1] shell still be used, therefore the general form of the extended problem statement is not changed. We can resolve the norm in (4) and reorder the terms to

$$\left[-\left(c_i^T z + 2d_i\right) c_i^T + \left(z^T A_i^T + 2b_i^T\right) A_i\right] z \leq d_i^2 - b_i^T b_i \quad (5)$$

Only the constant matrix P and the vector h, also constant with respect to the optimization variable z, are expanded with p rows belonging to the p new quadratic constraints (3) and q rows for the conic constraints (5) to

$$\hat{P}(z) = \begin{bmatrix} P \\ \beta_1^T + z^T \Gamma_1 \\ \vdots \\ \beta_p^T + z^T \Gamma_p \\ -\left(c_1^T z + 2d_1\right) c_1^T + \left(z^T A_1^T + 2b_1^T\right) A_1 \\ \vdots \\ -\left(c_q^T z + 2d_q\right) c_q^T + \left(z^T A_q^T + 2b_q^T\right) A_q \end{bmatrix}$$

and

$$\hat{h} = \begin{bmatrix} h \\ \alpha_1^T \\ \vdots \\ \alpha_p^T \\ d_1^2 - b_1^T b_1 \\ \vdots \\ d_q^2 - b_q^T b_q \end{bmatrix}.$$

Different from the extended vector $\hat{h}$ the extended matrix $\hat{P}(z)$ is not constant with respect to z anymore. Despite that expanding P and h does not change the structure of Φ exploited in [1]. So we have not to worry go on with its algorithm. With the new inequality constraint $\hat{P}(z)z \leq \hat{h}$ the logarithmic barrier function looks like

$$\phi(z) = \sum_{i=1}^{lT+l_f+p+q} -\log\left(\hat{h}_i - \hat{p}_i^T(z)z\right) \quad (6)$$

where $\hat{p}_i^T(z)$ is the ith rows of $\hat{P}(z)$ depending on z. The gradient of the logarithmic barrier function $\nabla\phi(z)$ is necessary to calculate the residual r. Because of first order derivatives of the functions associated with the inequality constraints

$$\nabla f_i(z) = \beta_i^T + 2z^T \Gamma_i \quad (7)$$

and

$$\nabla f_i(z) = \left(2\left(-\left(c_i^T z + d_i\right) c_i + \left(z^T A_i^T + b_i z^T\right) A_i\right)\right),$$

the gradient of the logarithmic barrier function is simply derived by forming $\hat{P}$ with argument $2z$ multiplied by $\hat{d}$.

$$\nabla\phi(z) = \hat{P}^T(2z)\hat{d} \quad (8)$$

with

$$\hat{d}_i = \frac{1}{\hat{h}_i - \hat{p}_i^T(z)z} \quad (9)$$

The second order derivatives of the functions associated with the inequality constraints are

$$\nabla^2 f_i(z) = 2\Gamma_i \quad (10)$$

and

$$\nabla^2 f_i(z) = -2\left(c_i c_i^T - A_i^T A_i\right), \quad (11)$$

of which we obtain two additional terms, have to be added to the Hessian of $\phi(z)$, to form Φ in the resulting system of linear equations.

$$\begin{aligned} \nabla^2\phi(z) =& \hat{P}(2z)\mathrm{diag}(\hat{d})^2\hat{P}(2z) \\ &+ \sum_{i=lT+lf+1}^{lT+lf+p} \left(2\hat{d}_i\Gamma_i\right) \\ &+ \sum_{j=lT+lf+p+1}^{lT+lf+p+q} \left(-2\hat{d}_j\left(c_j c_j^T - A_j^T A_j\right)\right) \end{aligned} \quad (12)$$

C. Selecting κ

In [1] the use of a fixed κ is proposed. But it is difficult to find one κ which provides fast convergence of the algorithm against the optimal solution for all feasible states of a controlled system . It should be considered to calculate a new κ once every time step of the MPC. To have the effect of the barrier function to the whole cost in same magnitude as the effect of weighting terms, we can calculate κ similar as in [4] for linear programs. Several tests have shown that a good κ can be estimated by adopting the suggested procedure of [4] for quadratic programs as

$$\kappa = \frac{z^T H z + g^T z}{T(n+m)}$$

D. Regularization

To solve the system of linear equations the most important step is to compute the Cholesky factorization of every block in Φ. For Cholesky factorization the blocks have to be symmetric and positive definite ([2]). To have all blocks positive definite and so ensure the Cholesky factorization of every block in Φ exists, we introduce a regularization term in the system of linear equations. We now solve

$$\begin{bmatrix} \Phi + \epsilon I & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta v \end{bmatrix} = -\begin{bmatrix} r_d \\ r_p \end{bmatrix} \quad (13)$$

and tolerate a small error we make instead of solving the original system of linear equations.

E. Iterative Refinement

As application for solving the optimization problem of a MPC it is often sufficient to use the solution of the linear equations with regularization term. If it is necessary to get the solution of the original linear equations we can compensate the error of regularization by iterative refinement as described in [6]. Let

$$Kl = r$$

the original system of linear equations and

$$\tilde{K}l = r$$

the system with regularization we can solve easily. A correction δl can be approximated by solving

$$\tilde{K}\delta l^{(k)} = \left(r - Kl^{(k)}\right)$$

and updating

$$l^{(k+1)} = l^{(k)} + \delta l^{(k)}.$$

V. RESULTS

A. Test QPs

Results of Solving Test QPs by [3]

B. Application Example

VI. CONCLUSIONS

What are the conclusions?

ACKNOWLEDGMENT

Acknowledgment.

REFERENCES

[1] Y. Wang and S. Boyd, Fast Model predictive control using online optimization, IEEE Transactions on Control Systems Technology, vol. 18, no. 2, pp. 267-278, March 2010.
[2] S. Boyd and L. Vandenberghe, Convex Optimization
[3] I. Maros and C. Mszros, A Repository of Convex Quadratic ...
[4] A. Marxen, Primal Barrier Methods for Linear Programming...
[5] A. Richards, Fast Model Predictive Control with Soft Constraints
[6] J. Mattingley, S. Boyd, CVXGEN: a code generator for embedded convex optimization
[7] M. S. Lobo and L. Vandenberghe, Application pf second-order cone programming
[8] Weitere Literatur