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 and give solutions with high accuracy are described in literature. 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]

B. Quadratic Constrained Quadratic Program

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

$$z^T \Gamma z + \beta^T z \leq \alpha. \tag{1}$$

C. Second Order Cone Program

A more general form of constraints are second order cone constraints. The constraints of such second order cone program ca be formulated as following inequality:

$$\|Az + b\|_2 \leq c^T z + d. \tag{2}$$

*This work was not supported by any organization

[1]author hannes.heinemann@st.ovgu.de

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 additionlly introduc soft contraints. So the algorithm can violate some inequality constraints for the cost of some additional penalty. There are different ways to introduc soft constraints in an optimization algorithm. [5] proposes a method tailored for the algorithm we use without any additional typically needed slack variable.

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. Therefor SOCCs in Generalized form [2]

$$\|Az + b\|_2^2 \leq \left(c^T z + d\right)^2 \tag{3}$$

can be used.

B. Extended Problem Statement

The algorithm of [1] shell still be used, therefor the general form of the extended problem statement is not changed. Only the constant matrix P and the vector h, also constant with respect to the optimization variable z, are expended with p rows belonging to the p new quadratic constraints and q rows for the conic constraints. Different from the extended vector $\hat{h}$ the extended matrix $\hat{P}(z)$ is not constant with respect to z anymore

$$\hat{P}(z) = \begin{bmatrix} P \\ \beta_1^T + z^T \Gamma_1 \\ \vdots \\ \beta_p^T + z^T \Gamma_p \\ -\left(c_1^T z + 2d_1\right) c_1^T + \left(z^T A_1^T + 2b_1^T\right) A_1 \\ \vdots \\ -\left(c_q^T z + 2d_q\right) c_q^T + \left(z^T A_q^T + 2b_q^T\right) A_q \end{bmatrix}, \tag{4}$$

$$\hat{h} = \begin{bmatrix} h \\ \alpha_1^T \\ \vdots \\ \alpha_p^T \\ d_1^2 - b_1^T b_1 \\ \vdots \\ d_q^2 - b_q^T b_q \end{bmatrix}. \tag{5}$$

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 new $\hat{h}$ and $\hat{P}(z)$ the logarithmic barrier function looks like

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

where $\hat{p}_i^T(z)$ is the ith rows of $\hat{P}(z)$ depending on z. The gradient of the logarithmic barrier function $\nabla\phi(z)$ necessary to calculate the residual is derived simply by forming $\hat{P}$ with argument $2z$ multiplied by $\hat{d}$.

$$\nabla\phi(z) = \hat{P}^T(2z)\hat{d} \tag{7}$$

with

$$\hat{d}_i = \frac{1}{\hat{h}_i - \hat{p}_i^T(z)z} \tag{8}$$

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

$$\begin{aligned}\nabla^2\phi(z) =& \hat{P}(2z)\text{diag}(\hat{d})^2\hat{P}(2z) \\ &+ \sum_{i=lT+lf+1}^{lT+lf+p}\left(\hat{d}_i 2\Gamma_i\right) \\ &+ \sum_{j=lT+lf+p+1}^{lT+lf+p+q}\left(\hat{d}_j - 2\left(c_j c_j^T - A_j^T A_j\right)\right)\end{aligned} \tag{9}$$

C. Selecting κ

In [1] the use of a fixed κ is proposed. But it is difficult to find one. It should be considered to calculate a new κ once every time step, not every inner step, to have the effect of the barrier function to the cost in same magnitude as the effect of weighting terms like in [4] for linear programs. Adopted for quadratic programs a good κ can be estimated by

$$\kappa = \frac{z^T H z + g^T z}{T(n+m)} \tag{10}$$

D. Numerical Improvements

To solve the system of linear equations the most important step is to compute the Cholesky foctorizaion of every block in Phi. For Cholesky factorization the blocks have to be symmetric and positiv definite ([2]). To avoid possible positiv semidefinite blocks we introduce a regularization term in Phi. With

$$Phi := Phi + \epsilon I \tag{11}$$

we ensure, that all blocks are positive definite. With regularization we make a small error. If necessary we can improve our solution by iterative refinement described in [6]

$$\delta l = \tilde{K}^{-1}\left(r - Kl^{(k)}\right) \tag{12}$$

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] Weitere Literatur