Preparation of Papers for IEEE Sponsored Conferences & Symposia*

Hannes Heinemann[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 using primal dual methods to solve several optimization problems are described in literature, which give solutions with high accuracy. For application as MPC optimization often do not neccesariely has to find such exact solutions, therefor in [1] a primal barrier method with focus of getting fast sufficiently exact solutions of quadratic programs is described, where of the speciell MPC problem structure is made use. This papers contribution is manly based on the algorithm in [1], which is extended with some nummerical robustness aproaches and for application for more general optimization problems, especially QPQCs and 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. An Optimization problem using this type of constraints is called second order cone program. Constraints are described as a second order cone as following: {Description as Cone Definition K:=...} as following inequality:

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

D. Soft Constraints

IV. EXTENDED ALGORITHM

A. Generalized Constraints

In the described primal barrier first and second order derivative. A SOCC in the above mentioned form is not continuously differentiable. Therefor SOCC in Generalized form [2] can be used.

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

B. Extended Problem Statement

The algorithm of [1] shell still be used, therefor its general form is not changed. Only the constant matrix P and the vector h, also constant regarding 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 regarding z anymore.

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

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

Expending 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 $\hat{d}$.

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

*This work was not supported by any organization

[1]Hannes Heinemann is student hannes.heinemann@st.ovgu.de

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 $\phi(z)$.

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

C. Selecting κ

In [1] the use of a fixed kappa is proposed. But it is difficult to find one. Eventually it should be considered to calculate a new κ once every time step, not every inner step, proportional to the values in the optimization variable like in [guck nach] for linear programs. Adopted for quadratic programs κ can be estimated as follows

D. Numerical Improvements

V. RESULTS

A. Test QPs

Results of Solving Test QPs by [3]

B. Application Example

VI. CONCLUSIONS

APPENDIX

Appendixes should appear before the acknowledgment.

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] Weitere Literatur