DTU Library

A Trust-region-based Sequential Quadratic Programming Algorithm

Henriksen, Lars Christian; Poulsen, Niels Kjølstad

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

IMM - TECHNICAL REPORT - 2010-14

A Trust-region-based Sequential Quadratic Programming Algorithm

L. C. Henriksen[1] and N. K. Poulsen[2]

[1] Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, DK-4000 Roskilde, Denmark, larh@risoe.dtu.dk
[2] Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, nkp@imm.dtu.dk

ABSTRACT

This technical note documents the trust-region-based sequential quadratic programming algorithm used in other works by the authors. The algorithm seeks to minimize a convex nonlinear cost function subject to linear inequalty constraints and nonlinear equality constraints.

KEYWORDS
nonlinear programming problem, sequential quadratic programming, trust region

1 Introduction

A nonlinear cost function subject to nonlinear equality and inequalty constraints constitutes a nonlinear programming problem (NLP). Multiple methods for solving such problems exists, best known are sequential quadratic programming problem (SQP) [1, 2] algorithms and interior point (IP) algorithms [3, 4] or hybrids of the two [5]. The SQP is typically augmented with a trust region inequality constraint, which can be tightened of loosened depending on the progress made by the current search direction. More advanced versions of these algorithms are able to cope with nonlinear inequality constraints, a feature not yet tested with the algorithm presented in this paper. This paper describes a trust-region-based sequential quadratic programming (TRSQP) algorithm with a general framework which can be used for different problems with adaptations suited for the particular structure of the problem. Such a particular structure is seen in Nonlinear Model Predictive Control [6, 7, 8].

2 Trust-region-based sequential quadratic programming

A NLP of the general form

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}), \text{ s.t. } \boldsymbol{c}(\boldsymbol{x}) = 0 \text{ and } \boldsymbol{d}(\boldsymbol{x}) \leq= 0 \tag{1}$$

has the Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\nu}^T \boldsymbol{c}(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{d}(\boldsymbol{x}) \tag{2}$$

leading to the Karush-Kuhn-Tucker (KKT) conditions for optimality [3, 4]

$$\nabla\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = \nabla\boldsymbol{f}(\boldsymbol{x}) + \nabla\boldsymbol{c}(\boldsymbol{x})\boldsymbol{\nu} + \nabla\boldsymbol{d}(\boldsymbol{x})\boldsymbol{\lambda} = 0 \tag{3}$$

$$\boldsymbol{c}(\boldsymbol{x}) = 0 \tag{4}$$

$$\boldsymbol{d}(\boldsymbol{x}) \leq 0 \tag{5}$$

$$\mathrm{diag}(\boldsymbol{d}(\boldsymbol{x}))\mathrm{diag}(\boldsymbol{\lambda}) = 0 \tag{6}$$

$$\boldsymbol{\lambda} \geq 0 \tag{7}$$

the KKT conditions will not be investigated further here as they more relevant for the IP methods. The NLP can be solved via an iterative procedure known as sequential quadratic programming (SQP) where in each iteration the local approximated problem is solved as a quadratic programming problem (QP).

$$\min_{\Delta \boldsymbol{x}} \boldsymbol{m}(\Delta \boldsymbol{x}), \ \boldsymbol{m}(\Delta \boldsymbol{x}) = \tfrac{1}{2}\Delta \boldsymbol{x}^T \mathbf{H} \Delta \boldsymbol{x} + \nabla \boldsymbol{f}(\boldsymbol{x})^T \Delta \boldsymbol{x} \tag{8a}$$

subject to

$$\boldsymbol{c}(\boldsymbol{x}) + \nabla \boldsymbol{c}(\boldsymbol{x})^T \Delta \boldsymbol{x} = 0 \tag{8b}$$

$$\boldsymbol{d}(\boldsymbol{x}) + \nabla \boldsymbol{d}(\boldsymbol{x})^T \Delta \boldsymbol{x} \leq 0 \tag{8c}$$

$$\|\mathbf{D}\Delta \boldsymbol{x}\|_p \leq \delta \tag{8d}$$

The optimization variable is updated by the iterative progress

$$(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})^+ = (\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) + \Delta(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) \tag{9}$$

if certain step acceptance criteria are met. The extra inequality constraint (8d), known as a trust-region, where δ is the trust region radius, can be added to aid the convergence. Iterations continue either untill the maximum number of allowed iterations are met, if termination criteria are met or until the algorithm fails due e.g. bad numerical handling of the problem solving or because the problem was ill-posed to begin with. The outline of the TRSQP can be seen in Alg. 1.

Algorithm 1: Trust region based sequential quadratic programming solver

Set initial values for $(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})$, $\delta = \delta_{max}$ and $\mathbf{H} = \nabla^2 f$;
for *ITER from 1 to IMAX* **do**
- Calc. trust-region scaling matrix $\mathbf{D}$;
- Solve QP to obtain $\Delta(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})$;
- Set trial variables $(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})^+ = (\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) + \Delta(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})$;
- Calc. $\boldsymbol{f}(\boldsymbol{x}^+)$, $\boldsymbol{c}(\boldsymbol{x}^+)$ and $\boldsymbol{d}(\boldsymbol{x}^+)$ and their Jacobians;
- Calc. progress measures ρ and γ;
- Update trust region radius δ;
- Update Quasi-Newton approximation of Hessian of Lagrangian $\mathbf{H}$;
- **if** *Step accepted* **then**
  - Set $(\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = (\boldsymbol{x}, \boldsymbol{\nu}, \boldsymbol{\lambda})^+$;
- **if** *Convergence* **then**
  - Terminate algorithm;

The different points of the algorithm are elaborated in the following sections.

2.1 Step acceptance

Step acceptance is depending on two measures of progress as well as inequality feasibility. The actual to predicted cost reduction ratio

$$\rho = \frac{\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^+)}{-\boldsymbol{m}(\Delta \boldsymbol{x})} \tag{10}$$

provides a measure of how well the QP subproblem resembles the properties of the NLP at the current point. For $\rho \approx 1$, the QP and the NLP are in good agreement. For $\rho > 1$, a greater decrease in cost function than predicted by the QP has occured and for $0 < \rho < 1$ the actual decrease in cost function was not as good as predicted by the QP. For $\rho < 0$, the NLP and the QP are not in agreement of whether the cost function was decreased or increased with the current step. The relative improvement of the cost function

$$\gamma = \frac{\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^+)}{\boldsymbol{f}(\boldsymbol{x})} \tag{11}$$

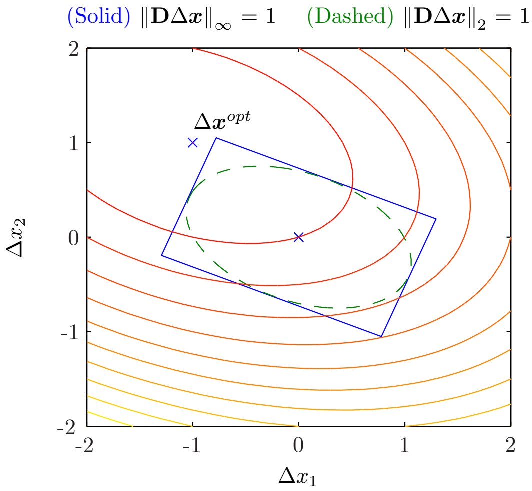

Figure 1: Two trust regions are shown in this figure: The box is given by the ∞-norm and the ellipsiod is given by the 2-norm. Both norms have scaling matrices based on the Hessian.

provides another measure of progress. Different strategies can be taken. In the present work the cost function is allowed to increasee as long as ρ or γ are positive and inequalties are feasible. This enables a search where the algorithm is able to move around. The step is accepted if the QP was able to find a feasible solution and if

$$\max(\rho, \gamma) > 0 \tag{12}$$

$$\max(\boldsymbol{d}(\boldsymbol{x}^+)) \leq \tau_{d(x)} \tag{13}$$

furthermore steps are not accepted in the first iteration as experience has shown better performance by letting the algorithm start up and generate its first Quasi-Newton Hessian approximiation etc.

2.2 Trust region

A trust region with the general form

$$\|\mathbf{D}\Delta\boldsymbol{x}\|_p \leq \delta \tag{14}$$

where $\mathbf{D}$ is the scaling matrix, p is the number of the norm, e.g. 1,2 or ∞, and δ is the trust region radius, can be imposed on the QP. The first choice of a suitable scaling matrix $\mathbf{D}$ might be the identity matrix. An even better choice takes the Hessian $\mathbf{H}$ into account as the problem might not be equally sensitive to changes of $\boldsymbol{x}$ in all directions. The 2-norm yields a quadratic constraint $\Delta\boldsymbol{x}^T\mathbf{D}^T\mathbf{D}\Delta\boldsymbol{x} \leq \delta^2$ which can be chosen to be indentical to the Hessian $\Delta\boldsymbol{x}^T\mathbf{H}\Delta\boldsymbol{x} \leq \delta^2$. The quadratic constraints would make the approximated problem a quadratic constrained quadratic cost (QCQC) problem which does not fit into the normal QP framework used by the SQP, where only linear constraints occur. The ∞-norm results in linear inequalty constraints

$$-\delta \leq \mathbf{D}\Delta\boldsymbol{x} \leq \delta \tag{15}$$

and is thus suitable for the description of a trust region which can be included in standard QP. Inspired by the quadratic constraints the Hessian can be decomposed by e.g. singular value decomposition

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{U}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{1/2}\mathbf{V}^T = \mathbf{D}^T\mathbf{D} \tag{16}$$

giving a multidimensional box circumscribing the ellipsoid of the quadratic constraint as tight as possible. Fig. 1 shows how the ∞-norm and the 2-norm trust regions resemble the Hessian and ensure that steps are constrained along the dimensions in a fashion scaled by the Hessian.

2.3 Trust region radius

The trust region radius should be increased or decreased according to a set of rules: If ρ is small indicating poor agreement between the NLP and QP or if γ is negative indicating an increase in cost function or if the previous step has been rejected for some reason, then the trust region radius should be decreased. If the QP failed, it is most likely due to a too restrictive trust region radius and the radius should be increased. If ρ was high, indicating good agreement between the NLP and the QP, then the trust region radius should be increased.

$$\delta = \begin{cases} \delta/2 & \text{for} \quad \rho < 1/4 \text{ or } \gamma < 0 \text{ or Step Rejected} \\ \min(3\delta, \delta_{max}) & \text{for} \quad \text{QP failed} \\ \min(3\min(\delta, \|\mathbf{D}\Delta\boldsymbol{x}\|_p), \delta_{max}) & \text{for} \quad \rho > 3/4 \end{cases} \tag{17}$$

2.4 Quasi-Newton approximations of Hessian of Lagrangian

The quadratic cost in the QP should idealy by the Hessian of the Lagrangian of the NLP

$$\nabla^2\mathcal{L} = \nabla^2\boldsymbol{f}(\boldsymbol{x}) + \sum_i^{n_c} \nabla^2\boldsymbol{c}_i(\boldsymbol{x})\nu_i + \sum_i^{n_d} \nabla^2\boldsymbol{d}_i(\boldsymbol{x})\lambda_i \tag{18}$$

this would computational expensive and if no analytic second derivatives of the cost function and constraints are available, those would have to be determimed via finite differences. Commonly used alternatives are different Quasi-Newton approximations such as dBFGS (Alg. 2), SR1 (Alg. 3) and SR1pos (Alg. 4). They are all calculations based on the variable step $\boldsymbol{s}$ and and with the difference in the gradients of the Lagrangian $\boldsymbol{y}$

$$\boldsymbol{s} = \boldsymbol{x}^+ - \boldsymbol{x} \tag{19}$$

$$\boldsymbol{y} = \nabla\mathcal{L}(\boldsymbol{x}^+, \boldsymbol{\nu}^+, \boldsymbol{\lambda}^+) - \nabla\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}^+, \boldsymbol{\lambda}^+) \tag{20}$$

Algorithm 2: damped Broyden-Fletcher-Goldfarb-Shanno (dBFGS) update

if $\boldsymbol{s}^T\boldsymbol{y} < 0.2\boldsymbol{s}^T\mathbf{H}\boldsymbol{s}$ **then**
- $\theta = 0.8\boldsymbol{s}^T\mathbf{H}\boldsymbol{s}(\boldsymbol{s}^T\mathbf{H}\boldsymbol{s} - \boldsymbol{s}^T\boldsymbol{y})^{-1}$;
- $\boldsymbol{y} = \theta\boldsymbol{y} + (1-\theta)\mathbf{H}\boldsymbol{s}$;

$\mathbf{H} = \mathbf{H} - \mathbf{H}\boldsymbol{s}\boldsymbol{s}^T\mathbf{H}(\boldsymbol{s}^T\mathbf{H}\boldsymbol{s})^{-1} + \boldsymbol{y}\boldsymbol{y}^T(\boldsymbol{y}^T\boldsymbol{s})^{-1}$;

Algorithm 3: Symmetric rank-1 (SR1) update

if $(\boldsymbol{y} - \mathbf{H}\boldsymbol{s}) < 10^{-6}(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})\boldsymbol{s}$ **then**
- $\mathbf{H} = \mathbf{H} + (\boldsymbol{y} - \mathbf{H}\boldsymbol{s})(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T((\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T\boldsymbol{s})^{-1}$;

Algorithm 4: Positive definite symmetric rank-1 (SR1pos) update

if $(\boldsymbol{y} - \mathbf{H}\boldsymbol{s}) < 10^{-6}(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})\boldsymbol{s}$ **then**
- $\mathbf{H} = \mathbf{H} + (\boldsymbol{y} - \mathbf{H}\boldsymbol{s})(\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T((\boldsymbol{y} - \mathbf{H}\boldsymbol{s})^T\boldsymbol{s})^{-1}$;
- $\mathbf{\Lambda V} = \mathbf{HV}$;
- $\mathbf{H} = \mathbf{V\Lambda}^+\mathbf{V}^{-1}$;

The dBFGS and SR1pos both maintain a positive definite Hessian, making the QP easier to solve. The SR1 might be a better fit if the NLP is not positive definite.

2.5 Termination

The algorithm terminates if steps toward the optimum are becomming to small and if the last accepted step was feasible

$$\|\Delta \boldsymbol{x}\|_2 \leq \tau_{\Delta x} \tag{21}$$

$$\max(\boldsymbol{d}(\boldsymbol{x}^+)) \leq \tau_{d(x)} \tag{22}$$

$$\left\|\boldsymbol{c}(\boldsymbol{x}^+)\right\|_\infty \leq \tau_{c(x)} \tag{23}$$

3 Discussion and future work

The presented algorithm has been tested with Nonlinear Model Predictive Control, with nonlinear equality constraints [6]. It has not been tested with nonlinear inequality constraints and performance with nonlinear inequality constraints remains to be investigated.

References

[1] Richard H. Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):–, 2000.

[2] Stephen J. Wright and Matthew J. Tenny. A feasible trust-region sequential quadratic programming algorithm. *SIAM journal on optimization*, 14(4):1074–1105, 2004.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] R.A. Waltz, J. Nocedal, D. Orban, and J.L. Morales. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.*, 107(3):391–408, 2006.

[6] L. C. Henriksen, N. K. Poulsen, and M. H. Hansen. Nonlinear model predictive control of a simplified wind turbine. *Submitted for Proc. World Congr. Int. Fed. Autom. Control., IFAC*, pages –, 2011.

[7] L. C. Henriksen and N. K. Poulsen. An online re-linearization scheme suited for model predictive or linear quadratic control. IMM-Technical Report 2010-13, Dept. of Informatics and Mathematical Modelling, Technical University of Denmark, 2010.

[8] Matthew J. Tenny, Stephen J. Wright, and James B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, 2004.