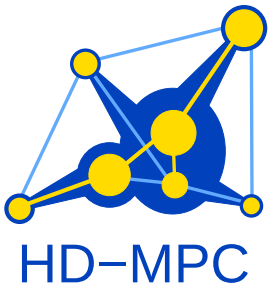
SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC



Deliverable Number:	D4.2.2
Deliverable Type:	Report
Contractual Date of Delivery:	September 1, 2010
Actual Date of Delivery:	**August 26, 2010**
Title of Deliverable:	**Report on redefinition of optimality criteria and generation of optimal solutions, and on analysis of sensitivity, scalability of solutions and computing cost**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP4
WP Leader:	Moritz Diehl
Partners:	TUD, KUL, USE, UWM
Author(s):	A. Kozma, M. Diehl

© Copyright by the HD-MPC Consortium

Table of contents

Project co-ordinator

Name:	Bart De Schutter
Address:	Delft Center for Systems and Control Delft University of Technology Mekelweg 2, 2628 Delft, The Netherlands
Phone Number:	+31-15-2785113
Fax Number:	+31-15-2786679
E-mail:	b.deschutter@tudelft.nl
Project web site:	`http://www.ict-hd-mpc.eu`

Executive summary

In this report we discuss optimality of optimization methods applied in distributed MPC, which do not only provide an optimal solution, but also find it with a fast convergence speed. We also show that in a certain class of algorithms there is no other algorithm that is more efficient than what we introduce here. We restrict our discussion to methods that make use of only first derivatives and give an application for solving strictly convex quadratic programs with a dual decomposition scheme. The performance of the scheme is demonstrated on an energy minimization problem. We also discuss scalability and sensitivity of the scheme.

Chapter 1

Optimality of first order methods

This section discusses first order methods and their performance on different problem classes based on [6], which discusses theoretical questions on this type of methods. The optimality conditions of a general nonlinear program are already established in optimization theory known as Karush-Kuhn-Tucker optimality conditions. The theory of the area is very well summarized in [1, 2, 9] and is not covered in this report. What we are more concerned with is the convergence speed of optimization methods. Especially in real-time applications not only obtaining an optimal solution is needed, but also fast convergence speed is a demand due to the lack of time.

In the following discussion a method is considered to be first-order if it generates a sequence of iterates x_k, where $x_k = x_0 + \sum_i \lambda_i \nabla f(x_i)$. First of all we make clear what kind of convergence rates we are interested in, then introduce gradient method and Nesterov's optimal gradient scheme. These schemes are optimal in the sense that no other first order method can perform better.

1.1 Convergence rates

In the following we define convergence rates that are interesting concerning our methods. We assume that $r_k \to 0$ and say r_k converges

- sublinearly if $r_k = O(\frac{1}{k^\alpha}), \alpha \in [0, \infty]$,
- linearly if $r_k = O(\alpha^k), \alpha \in [0, 1]$,
- superlinearly if $\limsup_{k\to\infty} \frac{r_{k+1}}{r_k} = 0$,
- quadratically if $r_k = O(\alpha^{2^k}), \alpha \in [0, 1]$.

The sequence r_k in the context of optimization is normally a distance measure from the optimal solution or optimal value. Note that sublinear convergence in practice means extremely slow performance, whereas linear convergence sometimes provides decent performance. Superlinear convergence is typically provided by Quasi-Newton methods. Quadratic performance would be desirable, but it turns out that with first-order methods this is not achievable, only exact Newton method has this rate.

1.2 Gradient Method

The gradient method – also known as steepest descent method – is widely known and applied in the area of optimization. The problem addressed is

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where f is a differentiable function. Gradient method generates iterates according to

$$x_{k+1} = x_k - t_k \nabla f(x_k),$$

where t_k is the step size. In terms of convergence rate the step size is crucial and can be adapted depending on what we know about f.

If f is convex and has Lipschitz gradient (L) the optimal step size is given by $t_k = \frac{1}{L}$ and

$$f(x_k) - f^* \leq \frac{2L\|x_0 - x^*\|^2}{k+4} \tag{1.2}$$

holds, which results in sublinear convergence. If f is in addition strongly convex with convexity parameter μ, then

$$\|x_k - x^*\| \leq \left(\frac{Q-1}{Q+1}\right)^k \|x_0 - x^*\| \tag{1.3}$$

$$f(x_k) - f^* \leq \frac{L}{2}\left(\frac{Q-1}{Q+1}\right)^{2k} \|x_0 - x^*\|^2 \tag{1.4}$$

holds for $\forall k$ with $Q = \frac{L}{\mu} \geq 1$. This means that gradient method provides linear convergence on convex functions that have both lower and upper bound on the curvature.

1.3 Nesterov's Optimal Gradient Scheme

General lower bounds both for the class of convex functions and for strongly convex functions are proven in [6]. Gradient method is not optimal in a sense that the upper bounds in (1.2), (1.3) and in (1.4) are not proportional to general lower bounds, which gives freedom to decrease them. Now we assume that f is convex, has Lipschitz gradient and strongly convex with convexity parameter $\mu \geq 0$. Note that this problem class contains the set of convex functions as well in case of $\mu = 0$. Nesterov's optimal gradient scheme proceeds as follows.

Given: $x_0 \in \mathbb{R}^n, \alpha_0 \in (0,1)$, set $y_0 \leftarrow x_0, q \leftarrow \frac{\mu}{L}$
while(no convergence)
  $x_{k+1} \leftarrow y_k - \frac{1}{L}\nabla f(y_k)$
  Compute α_{k+1} from $\alpha_{k+1}^2 = (1-\alpha_{k+1})\,\alpha_k^2 + q\alpha_{k+1}$
  $\beta_k \leftarrow \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2+\alpha_{k+1}}$
  $y_{k+1} \leftarrow x_{k+1} + \beta_k(x_{k+1} - x_k)$
end

(1.5)

Note that only one evaluation of the gradient is needed, whereas the rest consists of vector operations. This slight computational overhead pays off with respect to gradient method as the following theoretical results show.

If f is convex and has Lipschitz gradient (L) then for the k^{th} iterate of Nesterov's optimal gradient scheme holds that

$$f(x_k) - f^* \leq L\frac{4}{(k+2)^2}\|x_0 - x^*\|^2. \tag{1.6}$$

If f is in addition strongly convex with convexity parameter $\mu > 0$, then

$$f(x_k) - f^* \leq L\left(1 - \sqrt{\frac{1}{Q}}\right)^k \|x_0 - x^*\|^2 \tag{1.7}$$

holds. Note that $\frac{4L}{(k+2)^2}\|x_0 - x^*\|^2 \leq \frac{2L}{k+4}\|x_0 - x^*\|^2$ for $\forall k > K$, i.e. we gave a tighter upper bound and thus to reach $f(x_k) - f^* \leq \varepsilon$ one needs to make $O(\frac{1}{\sqrt{\varepsilon}})$ steps, which is $O(\frac{1}{\varepsilon})$ in case of gradient method. In Figure 1.1 gradient method and Nesterov's optimal gradient scheme are compared in terms of convergence speed. The optimization problem we solved here is an instance of the problem that we will define in Section 2. It can be seen that the requested tolerance is reached earlier by the optimal method, as the supporting theory states.

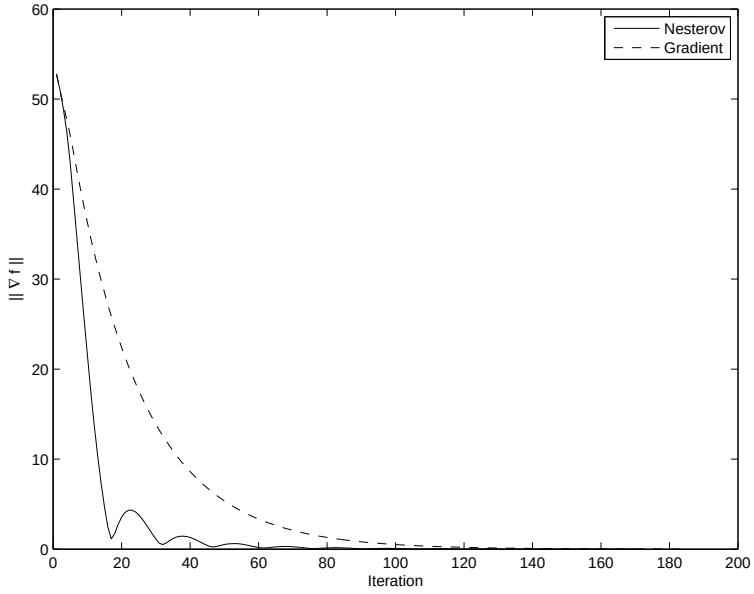


Figure 1.1: Convergence of Nesterov's optimal gradient scheme compared to classical gradient method on a sparse, strictly convex QP.

In the next chapter we give an application, where this scheme is used to solve convex QPs in a distributed setting.

Chapter 2

Optimal Solution of Quadratic Programs

In the present chapter we investigate a dual decomposition scheme applied to generate optimal solutions of structured quadratic programs (QP) in a distributed manner. We also discuss scalability and sensitivity of the scheme.

2.1 Dual decomposition scheme

The basic problem that we are concerned with has the form

$$\begin{aligned} \min_{x_1,\ldots,x_N} \quad & \sum_{i=1}^{N} \frac{1}{2} x_i^T Q_i x_i + c_i^T x_i \\ \text{s.t.} \quad & H_i x_i \leq d_i \quad i = 1,\ldots,m \\ & \sum_{i=1}^{N} A_i x_i = b, \end{aligned} \tag{2.1}$$

where $x_i \in \mathbb{R}^n$, $Q_i \in \mathbb{S}^n_{++}$ ($\mathbb{S}^n_{++}$ denotes positive definite matrix cone), $c_i \in \mathbb{R}^n$, $H_i \in \mathbb{R}^{p\times n}$, $d_i \in \mathbb{R}^p$, $A_i \in \mathbb{R}^{q\times n}$, $b \in \mathbb{R}^q$. Note that the i$^{\text{th}}$ term in the objective function only depends on x_i, hence the variables are not coupled via the objective function, but only through the equality constraints. Such problems arise in distributed linear MPC. We propose an algorithm that solves the dual problem by using Active Set Strategy and Nesterov's optimal gradient scheme. We reformulate (2.1) by adding the equality constraints to the objective with Lagrange weights and by taking the dual problem, resulting in

$$\max_{\lambda} \quad \sum_{i=1}^{N} \underbrace{\left(\begin{array}{cc} \min\limits_{x_i} & \left(\frac{1}{2} x_i^T Q_i x_i + \left(c_i^T + \lambda^T A_i \right) x_i - \lambda^T \frac{b}{N} \right) \\ \text{s.t.} & H_i x_i \leq d_i \end{array} \right)}_{P_i(\lambda)}. \tag{2.2}$$

Note that in this problem the parametric QP denoted by $P_i(\lambda)$ is a strictly convex QP in x_i with a fixed λ and thus can be solved by Online Active Set Strategy [3], which makes use of previous matrix factorizations and hence turns out to be very fast in practice. Moreover, these subproblems can be solved simultaneously e.g. on nodes of a computer cluster. The optimization of the dual variables takes places by using Nesterov's optimal gradient scheme [7] for non-smooth functions.

Low-level optimization — Online Active Set Method

On the low-level we solve a set of QPs that has the form

$$\begin{aligned} \min_{x_i} \quad & \frac{1}{2} x_i^T Q_i x_i + \left(c_i^T + \hat{\lambda}^T A_i \right) x_i - \hat{\lambda}^T \frac{b}{N} \\ \text{s.t.} \quad & H_i x_i \leq d_i, \end{aligned} \tag{2.3}$$

with parameter $\hat{\lambda}$. The solution of an inequality constrained QP might be obtained by using active set strategy. This class of methods maintains a set of active constraints and iteratively solves the optimization problem with the actual active set. The method proceeds by adding and removing constraints and solving linear systems preserving feasibility. Online active set methods speed up the linear system phase by storing and reusing the factorization matrices. Consider if $\hat{\lambda}$ changes — coming from the high-level —, then only the first order term of the program varies, the quadratic term remains the same. This makes Online Active Set Strategy very efficient in practice.

High-level optimization — Nesterov's optimal gradient scheme

The dual problem on the high-level has the form

$$\max_{\lambda} \sum_{i=1}^{N} P_i(\lambda). \tag{2.4}$$

It is known that optimization in the dual space is a convex maximization problem. If we apply Nesterov's gradient scheme (1.5) boils down to a very simple scheme with which we can reach faster – moreover optimal – convergence with a bit of overhead. The procedure maintains two sequences in the dual space.

$$\lambda_x^{(k)} = \lambda_y^{(k-1)} - t \nabla f(\lambda_y^{(k-1)}) \tag{2.5}$$

$$\lambda_y^{(k)} = \lambda_x^{(k)} + \frac{k-1}{k+2} \left(\lambda_x^{(k)} - \lambda_x^{(k-1)} \right) \tag{2.6}$$

Here k is the loop variable, and a Lipschitz constant can be given by neglecting the inequalities in (2.3).

$$\frac{1}{t} = L = \sum_{i=1}^{N} \max_j \{ \lambda_{i,j} | \lambda_{i,j} \in \lambda(A_i Q_i^{-1} A_i^T) \}. \tag{2.7}$$

The gradient of the dual can be given by

$$\nabla \mathscr{L}(\lambda) = \sum_{i=1}^{N} (A_i x_i^*(\lambda)) - b. \tag{2.8}$$

Using Nesterov's optimal gradient scheme has an advantage in contrast to classical gradient method, namely, the iteration number that is needed to reach $\| \mathscr{L}(\lambda^{(k)}) - \mathscr{L}(\lambda^*) \| \leq \varepsilon$ with an arbitrary $\varepsilon > 0$ tolerance only $O(\frac{1}{\sqrt{\varepsilon}})$ steps are needed, which is $O(\frac{1}{\varepsilon})$ in case of classical gradient. One drawback of the present algorithm is that the optimization in the dual space has poor convergence properties, namely has sublinear rate. This performance decreases even more because the given Lipschitz constant is not tight.

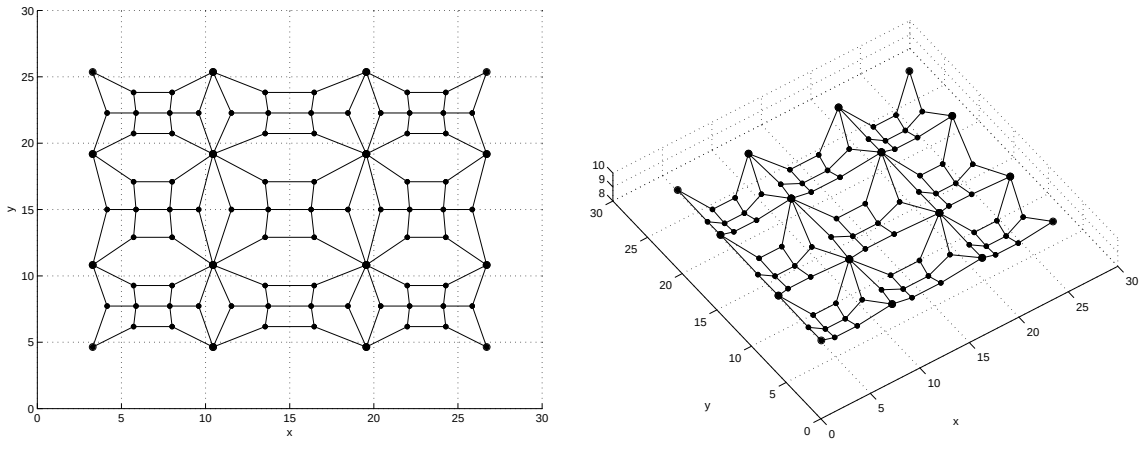
2.2 Demonstration on an example

In the present section we would like to show how our dual decomposition scheme performs also compare with gradient method applied in the dual space.

Our demonstration problem is called “net of hammocks”. A hammock is seen as a 2-dimensional grid of mass points connected to each other with weightless, extendible and compressable strings. The four corner points of a hammock have fixed height, whereas the rest are free to move. We take some of such hammocks and connect them via the corner points, which have fixed height, resulting in a net of hammocks. The objective is to find the equilibrium point, i.e. to minimize the sum of the string lengths and the potential energy, while all the mass points should be above or on the ground. In order to make our QP strictly convex, we add a regularization term as well. The problem can formulated in the following way.

$$\begin{aligned}\min_{X_{1,1},\ldots,X_{N,N}} \quad & \sum_{i=1}^{N-1}\sum_{j=1}^{N-1}\left(\|X_{i,j}-X_{i+1,j}\|_2^2+\|X_{i,j}-X_{i,j+1}\|_2^2\right)+ \\ & \sum_{i=1}^{N-1}\left(\|X_{N,i}-X_{N,i+1}\|_2^2+\|X_{i,N}-X_{i+1,N}\|_2^2\right) \\ & +(0,0,1)gm\sum_{i=1}^{N}\sum_{j=1}^{N}X_{i,j}+\sum_{i,j\in\{1,N\}}\|X_{i,j}-r_{i,j}\|_2^2. \\ \text{s.t.} \quad & (0,0,1)X_{i,j}\geq h_{ground} \quad (i,j\in\{1,\ldots,N\}) \\ & (0,0,1)X_{i,j}=h_{corners} \quad (i,j\in\{1,N\})\end{aligned} \tag{2.9}$$

Here $X_{i,j}\in\mathbb{R}^3$ denotes the position of the mass point in the (i,j) vertex position, g is the gravitational constant, m is the mass of a point, h_{ground} and $h_{corners}$ are height constants. In order to get the form of (2.1) we introduce several hammocks and couple them via equality constraints. The optimal configuration of a 3×3 net with 4×3 mass points is plotted in Figure 2.1(a) and 2.1(b).



(a) (b)

Figure 2.1: Optimal configuration of a 3×3 net with 4×3 hammocks. The central mass point pair of each hammock is touching the ground, the coupling points are marked with bigger circles.

As a test problem, we took a 10×10 net, each with 10×10 mass points, resulting in 30000 decision variables, 1080 coupling variables in the centralized problem. Each QP had 300 decision variables, 1480 equality constraints and 100 inequality constraints. Nesterov's scheme and the classical gradient method — both with Lipschitz information — were compared, the stopping criteria was $\|\Delta\lambda\| \leq \delta$ (see Table 2.1).

Table 2.1: Comparison of the proposed approach with Nesterov's scheme and gradient method. The measurements have the form of `mm:ss` or `hh:mm:ss`.

(a) Runtime

δ	Nesterov	Gradient
10^{-3}	0:55	02:58
10^{-4}	1:55	03:59
10^{-5}	2:52	04:56
10^{-6}	3:29	05:52

(b) CPU-time and iterations

δ	Nesterov	Iterations	Gradient	Iterations
10^{-3}	1:05:15	949	03:34:32	3895
10^{-4}	2:12:45	2217	04:50:23	5190
10^{-5}	3:15:29	3037	05:56:55	6485
10^{-6}	4:09:30	4528	07:03:58	7781

To have a comparison we solved the centralized QP with a primal-dual interior-point solver (OOQP) [4], which took 12.2 seconds and 12 iterations with about 10^{-8} precision on a desktop computer.

On this particular example our dual decomposition scheme turned out to be very slow compared to a centralized solver.

2.3 Sensitivity of Dual Decomposition

In this section we try to clarify how our dual decomposition scheme depends on the Lipschitz constant. Since this basically determines the convergence speed it is very important to choose an appropriate one.

There are several methods how to treat the Lipschitz constant. One possibility is to neglect low-level inequalities and give a global constant, which is never exact if any of the inequalities are active. Another possibility is to estimate it with an adaptive procedure [8]. The use of L can also be eliminated by doing an extra imprecise line-search [5].

In order to see how Nesterov's scheme performs we solved an unconstrained QP with different Lipschitz constants and fixed number of iterations (see Figure 2.2).

From our experiments it turned out that too small Lipschitz constant makes the scheme very unstable due to very long steps. If we choose L to be too large, the steps are getting short and thus in this region the result is less accurate. Surprisingly the exact Lipschitz constant does not give maximal precision.

2.4 Scalability of Dual Decomposition

Scalability of an optimization method is especially important if the target problem is large scale. Recall that in our scheme the calculation of Lipschitz constant is given by

$$L = \sum_{i=1}^{N} \max_j \{\lambda_{i,j} | \lambda_{i,j} \in \lambda(A_i Q_i^{-1} A_i^T)\}. \tag{2.10}$$

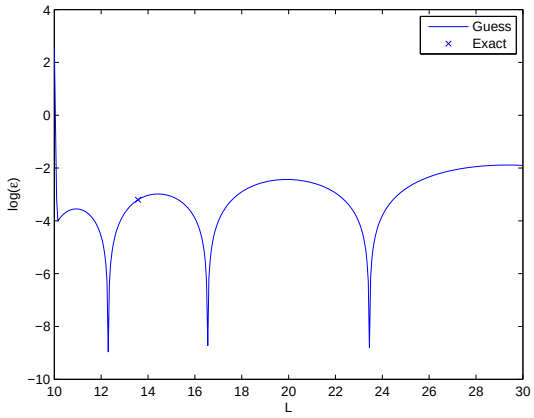


Figure 2.2: Accuracy reached by Nesterov's scheme after 200 iterations with different Lipschitz guesses.

If N becomes large, so does L, which results in very short steps in (2.5). Now let us consider how the number of iterations change if we increase N with respect to the accuracy in the objective (ε). Assume that our subproblems have the same quadratic terms, then $L = N\hat{L}$.

Firstly, suppose that ε is the requested absolute error. Note that often this type of error is considered as stopping criteria. Using the upper bound in (1.6) we get

$$N\hat{L}\frac{4}{(k+2)^2}\|x_0 - x^*\| \leq \varepsilon. \tag{2.11}$$

This implies that the number of iterations change with the order of $\sqrt{\frac{N}{\varepsilon}}$.

Secondly, if ε denotes the relative error of the objective and supposing that the objectives of subsystems have the same order, i.e. $\sum_i^N P_i(\lambda) = O(N)$, and $\varepsilon = \hat{\varepsilon}N$ then

$$\hat{L}\frac{4}{(k+2)^2}\|x_0 - x^*\| \leq \hat{\varepsilon} \tag{2.12}$$

holds, which implies that the number of subsystems does not affect the number of iterations with respect to relative error and has order of $\frac{1}{\sqrt{\varepsilon}}$.

Bibliography

[1] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[3] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy for fast parametric quadratic programming in MPC applications. In *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble*, pages 21–30, 2006.

[4] E.M. Gertz and S.J. Wright. Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29(1):58–81, 2003.

[5] C.C. Gonzaga and E.W. Karas. Optimal steepest descent algorithms for unconstrained convex problems: fine tuning Nesterov's method. Technical report, Federal University of Santa Catarina, 2008.

[6] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2003.

[7] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Programming*, 103(1):127–152, 2005.

[8] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion paper*, 76, 2007.

[9] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.