

## Decomposition via ADMM for Scenario-Based Model Predictive Control

Kang, J.; Raghunathan, A.U.; Di Cairano, S.

TR2015-060 July 2015

### Abstract

We present a scenario-decomposition based Alternating Direction Method of Multipliers (ADMM) algorithm for the efficient solution of scenario-based Model Predictive Control (MPC) problems which arise for instance in the control of stochastic systems. We duplicate the variables involved in the non-anticipativity constraints which allows to develop an ADMM algorithm in which the computations scale linearly in the number of scenarios. Further, the decomposition allows for using different values of the ADMM stepsize parameter for each scenario. We provide convergence analysis and derive the optimal selection of the parameter for each scenario. The proposed approach outperforms the non-decomposed ADMM approach and compares favorably with Gurobi, a commercial QP solver, on a number of MPC problems derived from stopping control of a transportation system.

*2015 American Control Conference (ACC)*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# Decomposition via ADMM for Scenario-Based Model Predictive Control

Jia Kang<sup>1</sup>, Arvind U. Raghunathan<sup>2</sup> and Stefano Di Cairano<sup>2</sup>

**Abstract**—We present a scenario-decomposition based Alternating Direction Method of Multipliers (ADMM) algorithm for the efficient solution of scenario-based Model Predictive Control (MPC) problems which arise for instance in the control of stochastic systems. We duplicate the variables involved in the non-anticipativity constraints which allows to develop an ADMM algorithm in which the computations scale linearly in the number of scenarios. Further, the decomposition allows for using different values of the ADMM stepsize parameter for each scenario. We provide convergence analysis and derive the optimal selection of the parameter for each scenario. The proposed approach outperforms the non-decomposed ADMM approach and compares favorably with Gurobi, a commercial QP solver, on a number of MPC problems derived from stopping control of a transportation system.

## I. INTRODUCTION

Model predictive Control (MPC) [1] exploits a model of the plant to be controlled, that usually includes constraints on inputs and states, to predict the future plant behavior and to accordingly select the control action that provides the most desirable predicted response. The selection of the control action is performed by solving a finite horizon optimal control problem initialized at the current state of the system and retaining only the first portion of the solution as control action. In particular for plant models described by linear systems subject to polyhedral constraints, the resulting problem is a Quadratic Program (QP), which can be solved by many different algorithms, fast gradient [2], multiplicative dual update [3], and ADMM [4]–[7] being examples of methods that perform many simple iterations, and interior-point [8] and active-set [9] being examples of methods that perform fewer, more complex iterations.

Scenario-based MPC is a technique for model-based control of stochastic systems inspired by the scenario enumeration approach of stochastic programming [10]. In [11] a scenario-based stochastic MPC was proposed which results in a tree structure, where only the nodes that are more likely to realize are expanded. The approach in [11] was applied to problems from the automotive industry [12], [13], where the scenario-based MPC showed effective performance. However, the computational cost associated with solving the stochastic MPC increases cubically with the number of scenarios unless the problem structure is exploited. Furthermore, in [11]–[13] high performance solvers are used, which requires significant resources and have complex code and thus are challenging to implement in embedded systems due

to platform limitations, and code verification requirements. In this paper we address how to use simple algorithms that are suitable also for embedded control architectures and that use the scenario-based MPC problem structure to speed up the computation of the solution.

We propose a decomposition approach for scenario-based MPC problems. In particular, we consider the ADMM algorithm of [6], [7] wherein the optimal step-size parameter was derived. The formulation in [7] does not allow to exploit the structure inherent in scenario-based MPC problem. Here, we describe an ADMM formulation that involves an equality constrained quadratic program (QP) which decouples by scenario and a projection problem whose computation scales linearly in the number of scenarios. Our numerical results show that similar to IPMs, the ADMM algorithm converges in a constant number of iterations regardless of the number of scenarios and the length of the MPC horizon. This property combined with the simplicity of the iterations makes the algorithm particularly well-suited for embedded control applications. Further, we allow the ADMM parameter to be chosen differently for each scenario. We present convergence analysis and derive the optimal ADMM parameter for each scenario. While a similar decomposition has been proposed independently and almost concurrently in [14], here we exploit the results of [6], [7] to analyze the ADMM algorithm and to obtain the optimal value for the ADMM parameter.

The paper is organized as follows. Mathematical formulation of the scenario-based MPC is provided in Section II. The ADMM algorithm of [6], [7] is presented in Section III. Section IV describes the ADMM algorithm with scenario decomposition. Section V provides a short convergence analysis for the scenario decomposition based ADMM algorithm. Numerical results are provided in Section VI, followed by conclusions in Section VII.

*Notation:* We denote by  $\mathbb{R}, \mathbb{R}_+$  the set of reals and set of non-negative reals, respectively, and by  $\mathbb{S}^n$  the set of symmetric  $n \times n$  matrices. All vectors are assumed to be column vectors. For a vector  $x \in \mathbb{R}^n$ ,  $x^T$  denotes its transpose and for two vectors  $x, y$ ,  $(x, y) = [x^T \ y^T]^T$ . For a matrix  $A \in \mathbb{R}^{n \times n}$ ,  $\rho(A)$  denotes the spectral radius,  $\lambda_i(A)$  denotes the eigenvalues and  $\lambda_{\min}(A), \lambda_{\max}(A)$  denote the minimum and maximum eigenvalues. For a matrix  $A \in \mathbb{S}^n$ ,  $A \succ 0$  ( $A \succeq 0$ ) denotes positive (semi)definiteness. We denote by  $I_n \in \mathbb{R}^{n \times n}$  the identity matrix. The notation  $\lambda \perp x \in \mathcal{Y}$  denotes the inequality  $\lambda^T(x' - x) \geq 0, \forall x' \in \mathcal{Y}$ , which is also called the *variational inequality*. We use  $\|\cdot\|$  to denote the 2-norm for vectors and matrices. A sequence  $\{x^k\} \subset \mathbb{R}^n$  converging to  $x^*$  is said to converge Q-linear rate if  $\|x^{k+1} - x^*\| \leq r\|x^k - x^*\|$  where  $0 < r < 1$ . We

<sup>1</sup> J. Kang is with Texas A&M University. This work was performed while J. Kang was a research intern. kangjia1127@gmail.com

<sup>2</sup> A.U. Raghunathan and S. Di Cairano are with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139. raghunathan, dicairano@merl.com

denote by  $\{x^k\} \rightarrow \bar{x}$  the convergence of the sequence to  $\bar{x}$ .

## II. PROBLEM DESCRIPTION

Consider the uncertain discrete-time linear prediction model of a plant, possibly augmented with additional states for enforcing control specifications,

$$x(k+1) = A(d^p(k))x(k) + B(d^p(k))u(k) + Fd^a(k) \quad (1)$$

where  $x \in \mathbb{R}^{n_x}$  is the state vector,  $u \in \mathbb{R}^{n_u}$  is the control input vector,  $d^p(k) \in \mathbb{R}^{n_p}$  are uncertain parameters which can vary over the time-steps,  $d^a \in \mathbb{R}^{n_d}$  is an unmeasured disturbance vector, and  $A \in \mathbb{R}^{n_x \times n_x}$ ,  $B \in \mathbb{R}^{n_x \times n_u}$ ,  $F \in \mathbb{R}^{n_x \times n_d}$  are the state, control and disturbance transfer matrices, respectively.

At every discrete time step  $k \in \mathbb{Z}$ ,  $k \geq 0$ , given the current state  $x(k)$ , the scenario-based MPC solves the problem,

$$\begin{aligned} \min_{\{(x_r, u_r)\}_{r=1}^{N_r}} & \sum_{r=1}^{N_r} \alpha_r \phi(x_r, u_r) \\ \text{s.t. } & x_{r,t+1} = A_{r,t}x_{r,t} + B_{r,t}u_{r,t} + Fd_{r,t}^a \\ & (x_{r,t+1}, u_{r,t}) \in \mathcal{X}_{r,t+1} \times \mathcal{U}_{r,t} \\ & t = 0, \dots, N-1, r = 1, \dots, N_r \\ & x_{r,0} = x(k) \\ & u_{1,0} = \dots = u_{N_r,0} \end{aligned} \quad (2)$$

where  $\alpha_r \in [0, 1]$  is the probability for the particular scenario  $r$  and  $\sum_{r=1}^{N_r} \alpha_r = 1$ ,  $d_r = (d_{r,0}^a, \dots, d_{r,N-1}^a)$  for  $r = 1, \dots, N_r$  are  $N_r$  different realizations of the uncertain parameter and disturbance vector,  $u_r = (u_{r,0}, \dots, u_{r,N-1})$ , by  $x_r = (x_{r,1}, \dots, x_{r,N})$  denote the set of states and control trajectories for each of those realizations. The sets  $\mathcal{X}_{r,t}, \mathcal{U}_{r,t}$  are polyhedral sets defining the feasible region for the states and controls. The objective function in (2) is defined as,

$$\begin{aligned} \phi(x_r, u_r) := & \frac{1}{2} \sum_{t=0}^{N-1} (x_{r,t}^T Q_{x,r} x_{r,t} + u_{r,t}^T R_{u,r} u_{r,t}) \\ & + \frac{1}{2} x_{r,N}^T P_r x_{r,N} \end{aligned} \quad (3)$$

where  $Q_{x,r}, P_r \in \mathbb{R}^{n_x \times n_x}$  are stage and terminal cost matrices on the state, respectively,  $R_{u,r} \in \mathbb{R}^{n_u \times n_u}$  is the stage cost matrix for the controls. Typically,  $R_{u,r} \succ 0$  and  $Q_{x,r}, P_r \succeq 0$ . Note that (3) also allows to formulate a reference tracking objective by setting  $Q_{x,r} = C^T Q_{\varepsilon,r} C$ ,  $Q_{\varepsilon,r} \in \mathbb{R}^{n_e \times n_e}$ ,  $P_r = C^T P_{\varepsilon,r} C$ ,  $P_{\varepsilon,r} \in \mathbb{R}^{n_e \times n_e}$ , where  $\varepsilon_r = Cx_r$ ,  $\varepsilon \in \mathbb{R}^{n_e \times n_e}$ ,  $n_e < n_x$  models the tracking error for a reference model embedded in (1).

Stochastic MPC solves (2) to find the optimal input sequence  $\{u_r^*\}_{r=1}^{N_r}$ , and then applies to the system the control input  $u(k) = u_{r,0}^*$  for any  $r$  since the constraints in (2) ensure that the control for  $t = 0$  is identical across the scenarios. These constraints are called the non-anticipativity conditions [15] and result in: (i) computing one control action at time step  $k$  regardless of the manifestation of the uncertainty, and (ii) allowing to recompute a different control action at time step  $k+1$  once the uncertainty manifests at time step  $k$ . Thus, the formulation in (2) is an instance of

two-stage stochastic QP [15]. For simplicity of exposition, we restrict our attention to the two-stage formulation in (2) though the methods described here can be easily extended to multistage stochastic QPs. We introduce additional variables  $\bar{y}_r$  that are equated to the controls in the first time-step i.e.,

$$\bar{y}_r = u_{r,0} \quad \forall r = 1, \dots, N_r.$$

Through the introduction of  $\bar{y}_r$ , we pose (2) as,

$$\begin{aligned} \min_{\{\mathbf{y}_r\}_{r=1}^{N_r}} & \sum_{r=1}^{N_r} \left( \frac{1}{2} \mathbf{y}_r^T \mathbf{Q}_r \mathbf{y}_r + \mathbf{q}_r^T \mathbf{y}_r \right) \\ \text{s.t. } & \mathbf{A}_r \mathbf{y}_r = \mathbf{b}_r \quad \forall r = 1, \dots, N_r \\ & \mathbf{y}_r \in \mathcal{Y}_r \quad \forall r = 1, \dots, N_r \\ & \bar{y}_1 = \dots = \bar{y}_{N_r} \end{aligned} \quad (4)$$

where,

$$\begin{aligned} \mathbf{y}_r &= (y_r, \bar{y}_r), y_r = (x_r, u_r), \quad \mathbf{q}_r = 0 \\ \mathbf{Q}_r &= \alpha_r \begin{pmatrix} \mathbf{I}_{N-1} \otimes Q_{x,r} & 0 & 0 & 0 \\ 0 & P_r & 0 & 0 \\ 0 & 0 & \mathbf{I}_N \otimes R_{u,r} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{b}_r &= (A_{r,1}x(k) + F_r d_{r,0}^a, \dots, F_r d_{r,N-1}^a, 0) \\ \mathcal{Y}_r &= \underbrace{\mathcal{X}_{r,1} \times \dots \times \mathcal{X}_{r,N} \times \mathcal{U}_{r,0} \times \dots \times \mathcal{U}_{r,N-1}}_{=: \mathcal{Y}_r} \times \mathbb{R}^{n_u}. \end{aligned}$$

The matrix  $\mathbf{A}_r$  is defined so that,

$$\begin{aligned} \mathbf{A}_r &= \begin{pmatrix} \hat{\mathbf{A}}_r & -\hat{\mathbf{B}}_r & 0 \\ 0 & -\mathbf{I}_{n_u} \otimes \mathbf{e}_1^T & \mathbf{I}_{n_u} \end{pmatrix} \\ \text{where, } \hat{\mathbf{A}}_r &= \begin{pmatrix} \mathbf{I}_{n_x} & 0 & \dots & 0 & 0 \\ -A_{r,1} & \mathbf{I}_{n_x} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{I}_{n_x} & 0 \\ 0 & 0 & \dots & -A_{r,N-1} & \mathbf{I}_{n_x} \end{pmatrix} \\ \text{and } \hat{\mathbf{B}}_r &= \begin{pmatrix} -B_{r,0} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -B_{r,N-1} \end{pmatrix} \end{aligned}$$

and  $\mathbf{e}_1 \in \mathbb{R}^N$  is the first vector of the standard basis of  $\mathbb{R}^N$ , i.e., the first component is 1 and the others are 0. The set  $\mathcal{Y}_r$  is assumed to be of the form  $\mathcal{Y}_r = \{\mathbf{y}_r \mid \mathbf{y}^{\min} \leq \mathbf{y}_r \leq \mathbf{y}^{\max}\}$  where  $-\infty \leq \mathbf{y}_i^{\min} < \mathbf{y}_i^{\max} \leq \infty$  for  $i = 1, \dots, n$ . The assumption on  $\mathcal{Y}_r$  is imposed for computational reasons, while the results of this paper can be easily extended to general convex sets. We show in the next section the ADMM formulation for the stochastic QP, where the structure of the two-stage stochastic QP formulation is not exploited.

## III. ADMM FOR THE FULL TWO-STAGE STOCHASTIC QP

Consider formulating the stochastic MPC in (4) as,

$$\begin{aligned} \min_{\mathbf{y}} & \frac{1}{2} \mathbf{y}^T \mathbf{Q} \mathbf{y} + \mathbf{q}^T \mathbf{y} \\ \text{s.t. } & \mathbf{A} \mathbf{y} = \mathbf{b} \\ & \mathbf{y} \in \mathcal{Y}_{\text{ND}} \end{aligned} \quad (5)$$

where  $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_{N_r})$ ,  $\mathcal{Y}_{\text{ND}} = \mathcal{Y}_1 \times \dots \times \mathcal{Y}_{N_r}$ , and the equality constraints  $\mathbf{A}\mathbf{y} = \mathbf{b}$  represent all the equality constraints defined in (4), i.e., the system dynamics for each scenario and the non-anticipativity constraints between scenarios. Consider the following modification of (5) as proposed in [6], [7],

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{w}} \quad & \frac{1}{2} \mathbf{y}^T \mathbf{Q} \mathbf{y} + \mathbf{q}^T \mathbf{y} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{y} = \mathbf{b}, \mathbf{w} \in \mathcal{Y}_{\text{ND}} \\ & \mathbf{y} = \mathbf{w} \end{aligned} \quad (6)$$

where  $\mathbf{w} \in \mathbb{R}^n$  are introduced and required to lie in the convex set  $\mathcal{Y}_{\text{ND}}$ . The steps in the ADMM algorithm are

$$\begin{aligned} \mathbf{y}^{l+1} &= \arg \min_{\mathbf{y}} L(\mathbf{y}, \mathbf{w}^l, \boldsymbol{\lambda}^l) \text{ s.t. } \mathbf{A} \mathbf{y} = \mathbf{b} \\ &= \mathbf{M}(\mathbf{w}^l + \boldsymbol{\lambda}^l - \tilde{\mathbf{q}}) + \mathbf{N} \mathbf{b} \end{aligned} \quad (7a)$$

$$\begin{aligned} \mathbf{w}^{l+1} &= \arg \min_{\mathbf{w}} L(\mathbf{y}^{l+1}, \mathbf{w}, \boldsymbol{\lambda}^l) \text{ s.t. } \mathbf{w} \in \mathcal{Y}_{\text{ND}} \\ &= \mathbb{P}_{\mathcal{Y}}(\mathbf{y}^{l+1} - \boldsymbol{\lambda}^l) \end{aligned} \quad (7b)$$

$$\boldsymbol{\lambda}^{l+1} = \boldsymbol{\lambda}^l + \mathbf{w}^{l+1} - \mathbf{y}^{l+1} \quad (7c)$$

where  $\beta \boldsymbol{\lambda}$  are multipliers for the constraint  $\mathbf{y} = \mathbf{w}$  in (6),  $\tilde{\mathbf{q}} = \mathbf{q}/\beta$ ,  $\mathbf{M} := \mathbf{Z} \left( \mathbf{Z}^T (\mathbf{Q}/\beta + \mathbf{I}_n) \mathbf{Z} \right)^{-1} \mathbf{Z}^T$ ,  $\mathbf{N} := (\mathbf{I}_n - \mathbf{M} \mathbf{Q}/\beta) \mathbf{R} (\mathbf{A} \mathbf{R})^{-1}$ , and  $\mathbf{R}, \mathbf{Z}$  denote an orthonormal basis for the range space of  $\mathbf{A}^T$  and null space of  $\mathbf{A}$ , respectively. The following result was shown in [7]. A key assumption in the proof is that the Linear Independence Constraint Qualification (LICQ) [16] holds at the solution.

*Theorem 1 (Theorem 1 & 2 [7]):* Suppose that the QP (5) has a solution, LICQ holds at the solution and  $\mathbf{Z}^T \mathbf{Q} \mathbf{Z} \succ 0$ . Then, the iterates  $\{(\mathbf{w}^l, \boldsymbol{\lambda}^l)\}$  generated by the ADMM algorithm (12) converge at a 2-step Q-linear rate to the solution of (5). Furthermore, the optimal step-size  $\beta^*$  that maximizes the rate of convergence is given by

$$\sqrt{\lambda_{\min}(\mathbf{Z}^T \mathbf{Q} \mathbf{Z}) \lambda_{\max}(\mathbf{Z}^T \mathbf{Q} \mathbf{Z})}. \quad (8)$$

The algorithm based on (7) performs a simple iteration that can be executed in an embedded microcontroller, but it still does not exploit the structure of the stochastic QP, and hence may have limited performance in terms of computing time. The next section presents an ADMM algorithm that scales linearly in the number of scenarios.

#### IV. SCENARIO DECOMPOSITION-BASED ADMM

Consider the formulation of the stochastic MPC in (4),

$$\begin{aligned} \min_{\{\mathbf{y}_r\}_{r=1}^{N_r}} \quad & \sum_{r=1}^{N_r} \left( \frac{1}{2} \mathbf{y}_r^T \mathbf{Q}_r \mathbf{y}_r + \mathbf{q}_r^T \mathbf{y}_r \right) \\ \text{s.t.} \quad & \mathbf{A}_r \mathbf{y}_r = \mathbf{b}_r \quad \forall r = 1, \dots, N_r \\ & \mathbf{y} \in \mathcal{Y}_{\text{SD}} \end{aligned} \quad (9)$$

where  $\mathcal{Y}_{\text{SD}}$  is defined as

$$\mathcal{Y}_{\text{SD}} = \left\{ (\mathbf{y}_1, \dots, \mathbf{y}_{N_r}) \mid \begin{array}{l} \mathbf{y}_r \in \mathcal{Y}_r \quad \forall r = 1, \dots, N_r \\ \bar{\mathbf{y}}_1 = \dots = \bar{\mathbf{y}}_{N_r} \end{array} \right\}. \quad (10)$$

Observe that the non-anticipativity constraints are accounted for in the convex set  $\mathcal{Y}_{\text{SD}}$ . Consider the following modification of QP (9) based on [6], [7],

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{w}} \quad & \sum_{r=1}^{N_r} \left( \frac{1}{2} \mathbf{y}_r^T \mathbf{Q}_r \mathbf{y}_r + \mathbf{q}_r^T \mathbf{y}_r \right) \\ \text{s.t.} \quad & \mathbf{A}_r \mathbf{y}_r = \mathbf{b}_r \quad \forall r = 1, \dots, N_r \\ & \mathbf{y} = \mathbf{w}, \mathbf{w} \in \mathcal{Y}_{\text{SD}} \end{aligned} \quad (11)$$

where  $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_{N_r}) \in \mathbb{R}^n$  with  $n = N_r(N(n_x + n_u) + n_u)$ . The steps in the ADMM algorithm are

$$\begin{aligned} \mathbf{y}^{l+1} &= \arg \min_{\mathbf{y}} \sum_{r=1}^{N_r} L_r(\mathbf{y}_r, \mathbf{w}^l, \boldsymbol{\lambda}_r^l) \\ \text{s.t.} \quad & \mathbf{A}_r \mathbf{y}_r = \mathbf{b}_r \quad \forall r = 1, \dots, N_r \end{aligned} \quad (12a)$$

$$\mathbf{w}^{l+1} = \arg \min_{\mathbf{w} \in \mathcal{Y}_{\text{SD}}} \sum_{r=1}^{N_r} L_r(\mathbf{y}_r^{l+1}, \mathbf{w}, \boldsymbol{\lambda}_r^l) \quad (12b)$$

$$\boldsymbol{\lambda}^{l+1} = \boldsymbol{\lambda}^l + \mathbf{w}^{l+1} - \mathbf{y}^{l+1}. \quad (12c)$$

where  $L_r(\mathbf{y}_r, \mathbf{w}_r, \boldsymbol{\lambda}_r) := \frac{1}{2} \mathbf{y}_r^T \mathbf{Q}_r \mathbf{y}_r + \mathbf{q}_r^T \mathbf{y}_r + \frac{\beta_r}{2} \|\mathbf{y}_r - \mathbf{w}_r - \boldsymbol{\lambda}_r\|^2$ . The update step of  $\boldsymbol{\lambda}$  in (12c) scales linearly in the number of scenarios. The following shows that the same holds for (12a) and (12b).

*Proposition 1:* The computational cost for the update in (12) scales linearly in the number of scenarios  $N_r$ .

*Proof:* Observe that the objective functions and constraints in (12a) are decoupled by scenarios. Hence, the update can be rewritten as,

$$\mathbf{y}_r^{l+1} = \mathbf{M}_r(\mathbf{w}_r^l + \boldsymbol{\lambda}_r^l - \mathbf{q}_r/\beta_r) + \mathbf{N}_r \mathbf{b}_r$$

$$\text{where, } \mathbf{M}_r := \mathbf{Z}_r \left( \mathbf{Z}_r^T (\mathbf{Q}_r/\beta_r + \mathbf{I}_n) \mathbf{Z}_r \right)^{-1} \mathbf{Z}_r^T, \quad (13)$$

$$\mathbf{N}_r := (\mathbf{I}_n - \mathbf{M}_r \mathbf{Q}_r/\beta_r) \mathbf{R}_r (\mathbf{A}_r \mathbf{R}_r)^{-1},$$

and  $\mathbf{R}_r, \mathbf{Z}_r$  denote an orthonormal basis for the range space of  $\mathbf{A}_r^T$  and null space of  $\mathbf{A}_r$ , respectively. Thus, the update for  $\mathbf{y}$  decouples by scenario and scales linearly with the number of scenarios.

In (12b), the objective function is componentwise separable in  $\mathbf{w}$ . Consider the following decomposition of  $\mathbf{w}_r$  as  $\mathbf{w}_r = (\mathbf{w}_r, \bar{\mathbf{w}}_r)$  where  $\mathbf{w}_r$  corresponds to the variables  $\mathbf{y}_r$  in (4), and  $\bar{\mathbf{w}}_r$  corresponds to  $\bar{\mathbf{y}}_r$ . Using this decomposition the constraint  $\mathbf{w} \in \mathcal{Y}_{\text{SD}}$  can be posed as,

$$\mathbf{w}_r \in \mathcal{Y}_r \quad \forall r = 1, \dots, N_r \text{ and } \bar{\mathbf{w}}_1 = \dots = \bar{\mathbf{w}}_{N_r}. \quad (14)$$

Note that  $\bar{\mathbf{w}}_r$  are only constrained by the equality of the values across scenarios and are not limited by bounds. Hence, the update in (12b) can be obtained as

$$\begin{aligned} \mathbf{w}_r^{l+1} &= \mathbb{P}_{\mathcal{Y}_r}(\mathbf{y}_r^{l+1} - \boldsymbol{\lambda}_r^l) \\ \bar{\mathbf{w}}_r^{l+1} &= \frac{1}{\sum_{r=1}^{N_r} \beta_r} \sum_{r=1}^{N_r} \beta_r (\bar{\mathbf{y}}_r^{l+1} - \bar{\boldsymbol{\lambda}}_r^l) \end{aligned} \quad (15)$$

where we have used  $\boldsymbol{\lambda}_r = (\boldsymbol{\lambda}_r, \bar{\boldsymbol{\lambda}}_r)$ . Thus, the update for  $\mathbf{w}$  scales linearly in the number of scenarios.  $\blacksquare$

The next section provides a sketch of the convergence analysis for the ADMM algorithm in (12) and derives the optimal parameter values  $\beta_r^*$ .

## V. CONVERGENCE ANALYSIS

We begin by showing that the solutions of (9) are the only fixed points of ADMM iterations (12). We first state the stationarity conditions of (9). Suppose  $\mathbf{y}^*$  solves (9), then there exist multipliers  $\boldsymbol{\xi}_r^*$  for the equality constraints  $\mathbf{A}_r \mathbf{y}_r = \mathbf{b}_r$  and multipliers  $\boldsymbol{\lambda}_r$  for the set inclusion constraints satisfying the stationary conditions:

$$\begin{aligned} \mathbf{Q}_r \mathbf{y}_r^* + \mathbf{A}_r^T \boldsymbol{\xi}_r^* - \boldsymbol{\lambda}_r^* &= -\mathbf{q}_r \\ \mathbf{A}_r \mathbf{y}_r^* &= \mathbf{b}_r \\ (\boldsymbol{\lambda}_1^*, \dots, \boldsymbol{\lambda}_{N_r}^*) &\perp \mathbf{y}_r^* \in \mathcal{Y}_{\text{SD}}. \end{aligned} \quad (16)$$

The variational inequality in (16) can be written as,

$$\boldsymbol{\lambda}_r^* \perp \mathbf{y}_r^* \in \mathcal{Y}_r, \quad \sum_{r=1}^{N_r} \bar{\boldsymbol{\lambda}}_r^* = \mathbf{0}. \quad (17)$$

The following characterizes the relation between  $\mathbf{w}^{k+1}, \boldsymbol{\lambda}^{k+1}$ .

*Lemma 1:* At every iteration of the ADMM algorithm  $\mathbf{w}^{l+1}, \boldsymbol{\lambda}^{l+1}$  in (12) satisfy: (i)  $\boldsymbol{\lambda}_r^{l+1} \perp \mathbf{w}_r^{l+1} \in \mathcal{Y}_r$  and (ii)  $\sum_{r=1}^{N_r} \beta_r \bar{\boldsymbol{\lambda}}_r^{l+1} = \mathbf{0}$ .

*Proof:* The proof of (i) follows along the same lines as in Lemma 2 [6] and is omitted. To show (ii), consider

$$\sum_{r=1}^{N_r} \beta_r \bar{\boldsymbol{\lambda}}_r^{l+1} = \sum_{r=1}^{N_r} \beta_r (\bar{\boldsymbol{\lambda}}_r^l + \bar{\mathbf{w}}_r^{l+1} - \bar{\mathbf{y}}_r^{l+1}) = \mathbf{0}$$

where the first equality follows from multiplying the update for each scenario in (12c) by  $\beta_r$  and summing, the second equality from  $\bar{\mathbf{w}}_1^{l+1} = \dots = \bar{\mathbf{w}}_{N_r}^{l+1}$  and (15).  $\blacksquare$

Using the above relation we can state the equivalence between fixed points of (12) and the optimal solutions to (9). For convenience,  $\beta \boldsymbol{\lambda}$  will denote  $(\beta_1 \boldsymbol{\lambda}_1, \dots, \beta_{N_r} \boldsymbol{\lambda}_{N_r})$  and  $\boldsymbol{\lambda}/\beta$  will denote  $(\boldsymbol{\lambda}_1/\beta_1, \dots, \boldsymbol{\lambda}_{N_r}/\beta_{N_r})$ .

*Theorem 2:* Suppose (9) is feasible. Then, if  $(\mathbf{y}^\circ, \mathbf{w}^\circ, \boldsymbol{\lambda}^\circ)$  is a fixed point of (12),  $(\mathbf{y}^\circ, \boldsymbol{\xi}^\circ, \beta \boldsymbol{\lambda}^\circ)$  is a KKT point for (9), where  $\boldsymbol{\xi}^\circ$  is the multiplier for the equalities in the subproblem for  $\mathbf{y}$  in (12a). Conversely, if  $(\mathbf{y}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*)$  is a KKT point of (9),  $(\mathbf{y}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*/\beta)$  is a fixed point of (12).

*Proof:* Suppose that  $(\mathbf{y}^\circ, \mathbf{w}^\circ, \boldsymbol{\lambda}^\circ)$  is a fixed point of (12). From the update for  $\boldsymbol{\lambda}$  in (12c) we obtain,

$$\boldsymbol{\lambda}^\circ = \boldsymbol{\lambda}^\circ + \mathbf{w}^\circ - \mathbf{y}^\circ \implies \mathbf{0} = \mathbf{w}^\circ - \mathbf{y}^\circ \implies \mathbf{y}^\circ \in \mathcal{Y}_{\text{SD}},$$

where the second implication follows from (12b). From Lemma 1,  $\mathbf{w}^\circ$  and  $\beta \boldsymbol{\lambda}^\circ, \mathbf{w}^\circ$  satisfy (17) and hence, the variational inequality  $\beta \boldsymbol{\lambda}^\circ \perp \mathbf{y}^\circ \in \mathcal{Y}_{\text{SD}}$  holds. From the update of  $\mathbf{y}$  in (12), there exist  $\boldsymbol{\xi}^\circ$  such that

$$\begin{pmatrix} \mathbf{Q}_r + \beta_r \mathbf{I} & \mathbf{A}_r^T \\ \mathbf{A}_r & \mathbf{0} \end{pmatrix} \begin{bmatrix} \mathbf{y}_r^\circ \\ \boldsymbol{\xi}_r^\circ \end{bmatrix} = \begin{bmatrix} \beta_r \mathbf{w}_r^\circ + \beta_r \boldsymbol{\lambda}_r^\circ - \mathbf{q}_r \\ \mathbf{b}_r \end{bmatrix},$$

i.e., the first order optimality conditions are satisfied. Substituting  $\beta_r \mathbf{y}_r^\circ$  for  $\beta_r \mathbf{w}_r^\circ$  and simplifying, we obtain that  $\mathbf{y}_r^\circ, \boldsymbol{\xi}_r^\circ, \beta_r \boldsymbol{\lambda}_r^\circ$  satisfy the first two constraints in (16). Hence,

$(\mathbf{x}^\circ, \boldsymbol{\xi}^\circ, \beta \boldsymbol{\lambda}^\circ)$  is a KKT point of (9) which implies that  $\mathbf{y}^\circ$  is a minimizer of (9). Thus, the first claim holds.

Suppose that  $(\mathbf{x}^*, \boldsymbol{\xi}^*, \boldsymbol{\lambda}^*)$  solves (9). Hence,

$$\begin{pmatrix} \mathbf{Q}_r + \beta_r \mathbf{I} & \mathbf{A}_r^T \\ \mathbf{A}_r & \mathbf{0} \end{pmatrix} \begin{bmatrix} \mathbf{y}_r^* \\ \boldsymbol{\xi}_r^* \end{bmatrix} = \begin{bmatrix} \beta_r \mathbf{y}_r^* + \boldsymbol{\lambda}_r^* - \mathbf{q}_r \\ \mathbf{b}_r \end{bmatrix}$$

which is obtained from (16) by rearranging and adding  $\beta_r \mathbf{y}_r^*$  to the left and right hand sides of the first equation, which is the fixed point of the update step of  $\mathbf{y}$  in (12) with  $\mathbf{y}^{k+1} = \mathbf{w}^k = \mathbf{y}^*, \boldsymbol{\lambda}^k = \boldsymbol{\lambda}^*/\beta$ . Furthermore, from (17),  $\boldsymbol{\lambda}_r^*/\beta_r \perp \mathbf{y}_r^*$  for all  $\beta_r > 0$ , which implies

$$\begin{aligned} (\boldsymbol{\lambda}_r^*/\beta_r)^T (\mathbf{v}' - \mathbf{y}^*) &\geq 0, \forall \mathbf{v}' \in \mathcal{Y}_r \\ \implies (\mathbf{y}_r^* - \mathbf{y}_r^* + \boldsymbol{\lambda}_r^*/\beta_r)^T (\mathbf{v}' - \mathbf{y}_r^*) &\geq 0, \forall \mathbf{v}' \in \mathcal{Y}_r. \end{aligned}$$

Thus,  $\mathbf{y}_r^* = \mathbb{P}_{\mathcal{Y}_r}(\mathbf{y}_r^* - \boldsymbol{\lambda}_r^*/\beta_r)$ . Further,

$$\sum_{r=1}^{N_r} \beta_r (\bar{\boldsymbol{\lambda}}_r^*/\beta_r) = \sum_{r=1}^{N_r} \bar{\boldsymbol{\lambda}}_r^* = \mathbf{0} \implies \bar{\mathbf{w}}_r^* = \bar{\mathbf{y}}_r^*,$$

where the second equality follows from (17) and the implication from the update step in (15). Consequently,  $(\mathbf{y}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*/\beta)$  is a fixed point of the update step for  $\mathbf{w}$  in (12). The fixed point of the update equation in  $\boldsymbol{\lambda}$  holds trivially, and thus also the second claim holds.  $\blacksquare$

Substituting for  $\mathbf{y}^{k+1}$  in (12) and simplifying we obtain,

$$\begin{aligned} \mathbf{w}^{k+1} &= \mathbb{P}_{\mathcal{Y}_{\text{SD}}}(\mathbf{v}^k) \\ \boldsymbol{\lambda}^{k+1} &= (\mathbb{P}_{\mathcal{Y}_{\text{SD}}} - \mathbf{I}_n)(\mathbf{v}^k) \end{aligned} \quad (18)$$

where  $\mathbb{P}_{\mathcal{Y}_{\text{SD}}}(\cdot)$  is as defined in (15) and  $\mathbf{v}^k = (\mathbf{v}_1, \dots, \mathbf{v}_{N_r})$ ,

$$\mathbf{v}_r^k = \mathbf{M}_r(\mathbf{w}_r^k + \boldsymbol{\lambda}_r^k) - \boldsymbol{\lambda}_r^k - \mathbf{M}_r \tilde{\mathbf{q}}_r + \mathbf{N}_r \mathbf{b}_r. \quad (19)$$

Using  $\mathbf{v}^k$ , the ADMM iteration can be equivalently cast as,

$$\begin{aligned} \mathbf{v}^{k+1} &= \mathbf{M}_{\text{SD}} \mathbb{P}_{\mathcal{Y}_{\text{SD}}}(\mathbf{v}^k) + (\mathbf{M}_{\text{SD}} - \mathbf{I}_n)(\mathbb{P}_{\mathcal{Y}_{\text{SD}}} - \mathbf{I}_n)(\mathbf{v}^k) \\ &\quad - \mathbf{M}_{\text{SD}} \tilde{\mathbf{q}}_{\text{SD}} + \mathbf{N}_{\text{SD}} \mathbf{b}_{\text{SD}} \end{aligned} \quad (20)$$

where  $\tilde{\mathbf{q}}_{\text{SD}} = (\tilde{\mathbf{q}}_1, \dots, \tilde{\mathbf{q}}_{N_r})$ ,  $\mathbf{b}_{\text{SD}} = (\mathbf{b}_1, \dots, \mathbf{b}_{N_r})$  and

$$\mathbf{M}_{\text{SD}} = \begin{pmatrix} \mathbf{M}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{M}_{N_r} \end{pmatrix}, \mathbf{N}_{\text{SD}} = \begin{pmatrix} \mathbf{N}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{N}_{N_r} \end{pmatrix}.$$

The above formulation is precisely the form analyzed in [7] and hence the analysis in [7] can be used verbatim after noting the following result.

*Lemma 2:* the orthonormal basis for the range and null space of equality constraints,  $\mathbf{R}_{\text{SD}}, \mathbf{Z}_{\text{SD}}$  in (9), are given by

$$\mathbf{R}_{\text{SD}} = \begin{pmatrix} \mathbf{R}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{R}_{N_r} \end{pmatrix}, \mathbf{Z}_{\text{SD}} = \begin{pmatrix} \mathbf{Z}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Z}_{N_r} \end{pmatrix}.$$

*Proof:* The proof follows by noting that the constraints in (9) decouple by scenario and hence,  $\mathbf{R}_{\text{SD}}, \mathbf{Z}_{\text{SD}}$  have the above block-diagonal structure.  $\blacksquare$

We can now state the main result on the optimal value of the parameter  $\beta_r$  that maximizes the rate of convergence. The results follow directly from the results in [7].

*Theorem 3:* Suppose (4) has a solution, LICQ holds at the solution and  $\mathbf{Z}_r^T \mathbf{Q}_r \mathbf{Z}_r \succ 0$ . Then, the iterates  $\{(\mathbf{w}^l, \boldsymbol{\lambda}^l)\}$  generated by the ADMM algorithm (12) converge at a 2-step Q-linear rate to the optimal solution of (4). Furthermore, the optimal step-size  $\beta_r^*$  that maximizes the rate of convergence is given by

$$\beta_r^* = \sqrt{\lambda_{\min}(\mathbf{Z}_r^T \mathbf{Q}_r \mathbf{Z}_r) \lambda_{\max}(\mathbf{Z}_r^T \mathbf{Q}_r \mathbf{Z}_r)}. \quad (21)$$

*Proof:* The analysis in Theorem 1 in [7] can be applied to show the two-step Q-linear convergence of  $\{(\mathbf{w}^l, \boldsymbol{\lambda}^l)\}$ . Further, from Theorem 2 in [7] we have that the maximal convergence rate can then be obtained by choosing  $\beta_r$  as

$$(\beta_1^*, \dots, \beta_{N_r}^*) = \arg \min_{\beta_1, \dots, \beta_{N_r}} \|2\mathbf{Z}_{\text{SD}}^T \mathbf{M}_{\text{SD}} \mathbf{Z}_{\text{SD}} - \mathbf{I}_{n-m}\|.$$

Since  $\mathbf{Z}_{\text{SD}}^T \mathbf{M}_{\text{SD}} \mathbf{Z}_{\text{SD}}$  is block-diagonal with  $\mathbf{Z}_r^T \mathbf{M}_r \mathbf{Z}_r$  on its diagonal and each  $\beta_r$  affects only  $\mathbf{M}_r$ , we have that

$$\beta_r^* = \arg \min_{\beta} \|2\mathbf{Z}_r^T \mathbf{M}_r \mathbf{Z}_r - \mathbf{I}_{n_u}\|.$$

This proves the claim.  $\blacksquare$

## VI. NUMERICAL RESULTS

To test the performance of ADMM [6] presented in Section III and scenario decomposition-based ADMM presented in Section IV on stochastic MPC problems, we implemented both algorithms in MATLAB R2010b in M-code and also the corresponding C-mex versions. OpenBLAS [17] was used for performing the matrix and vector operations. Also, we used the start-of-the-art solver Gurobi 5.6.2 [18] to benchmark the performance. All of the solvers have the same MATLAB interface with our stochastic MPC problems. The tested stochastic MPC problem comes from the stopping control of a transportation system [19] with dynamics

$$\ddot{d}(t) = \frac{1}{rm} \tau(t) - \frac{c_1}{m} \dot{d}(t) - \frac{c_0 g}{m} \mu, \quad (22)$$

where  $d[\text{m}]$  is the distance from the stop position,  $\dot{d}[\text{m/s}]$  is the velocity,  $\tau[\text{Nm}]$  is the tractive and braking torque,  $c_1$  and  $c_0$  are parameters determining the bearing friction and rolling resistance  $g = 9.81[\text{m/s}^2]$  is the gravity acceleration, and  $\mu$  is the friction with the ground. The states and inputs are subject to constraints  $d_{\min} \leq d \leq d_{\max}$ ,  $v_{\min} \leq \dot{d} \leq v_{\max}$ ,  $u_{\min} \leq u \leq u_{\max}$ , and we consider  $\mu$  as the uncertainty for each scenario. In our tests, the matrices  $\mathbf{Q}_r$ ,  $\mathbf{A}_r$  are identical across scenarios.

We present computational results for different number of scenarios ( $N_r = 2, 4, 8, 16$ ), and different lengths of the MPC prediction horizon ( $N = 25, 50, 100$ ). Table I shows the average (over 10 runs) elapsed time and iterations for all the three solvers. All timing was performed on a 3.20 GHz Intel(R) Core(TM) i7-3930K CPU processor with 32 GB RAM. In Table I ADMM denotes the algorithm presented in Section III while ADMM-d refers to the algorithm presented in Section IV. Note that we do not take advantage of parallelism in the current implementation other than those offered through MATLAB's and OpenBLAS's [17] multi-threaded execution of their subroutines. The termination error

for all three solvers was set to  $10^{-6}$ . The ADMM timing only reports the elapsed time in ADMM iterations since the optimal values for  $\beta_r$ , the matrix  $\mathbf{M}$  and the vector  $N\mathbf{b}$  are all precomputed.

Comparing the ADMM MATLAB version with the ADMM C-mex version, one can see that ADMM MATLAB is approximately one order of magnitude slower than ADMM mex. This is as expected since the C++ implementation is generally 10–20 times faster than the MATLAB implementation. Comparing ADMM mex with Gurobi, one can see that ADMM mex is comparable with Gurobi when the problem size is small. However, ADMM mex becomes less favorable compared to Gurobi as the problem size increases, since its timing increases quadratically with the problem size. On the other hand, the scenario decomposition-based ADMM (ADMM-d) scales much more favorably and comparably with Gurobi as the problem size increases. Note that the barrier method in Gurobi is a second-order Newton-based algorithm while ADMM is only a first-order algorithm. Consequently, the convergence of speed of ADMM is slower than Gurobi, and ADMM is expected to take far more iterations to reach convergence. Since the number of coupling constraints for stochastic MPC problems is generally very small, good speedup is possible [20].

Figure 1 visualizes the ratio of the CPU time for the ADMM algorithms to the CPU time for Gurobi as we increase the number of scenarios for 50 and 100 time steps. Again, we can clearly see that the performance of ADMM deteriorates compared with Gurobi and ADMM-d as the problem size increases. ADMM-d is quite comparable with Gurobi on most instances. We believe that we can further improve performance of ADMM-d through a parallel implementation, which will be considered in a subsequent study.

## VII. CONCLUSIONS

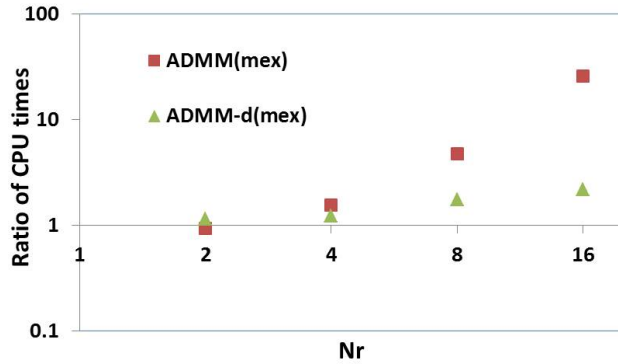
We present a scenario-decomposition based Alternating Direction Method of Multipliers (ADMM) algorithm for the efficient solution of scenario-based Model Predictive Control (MPC) problems. The algorithm results in an ADMM formulation involving an equality constrained quadratic program (QP) which decouples by scenario and a projection problem, the computation of which scales linearly in the number of scenarios. Numerical results show that the proposed approach compares favorably with Gurobi and the computational time scales well with the increase in scenarios and number of time-steps in MPC. However, we also note that in the context of ADMM algorithm if the scenarios vary from MPC problem to the next then, the overhead in computing some of the matrices involved in the iterations might outweigh the gains in the iteration simplicity. Future works will consider approaches for mitigating this computational burden.

## REFERENCES

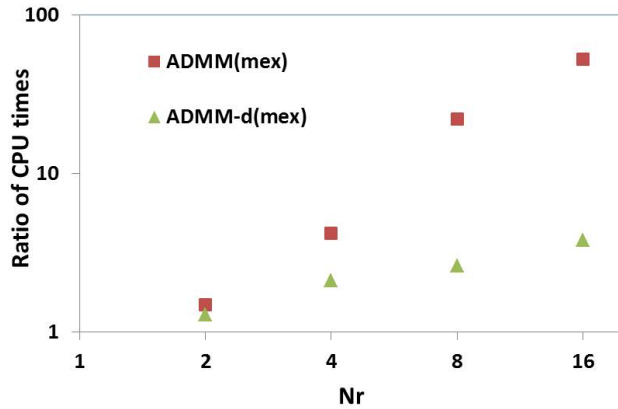
- [1] C. Garcia, D. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," *Automatica(Oxford)*, vol. 25, no. 3, pp. 335–348, 1989.

TABLE I  
TIMING RESULTS FOR STOCHASTIC MPC PROBLEM

N	# Scenarios	Gurobi		ADMM(MATLAB)		ADMM(mex)		ADMM-d(MATLAB)		ADMM-d(mex)	
		time(s)	iters	time(s)	iters	time(s)	iters	time(s)	iters	time(s)	iters
25	2	0.0090	15	0.0385	534	0.0032	534	0.2010	536	0.0036	536
	4	0.0086	15	0.0585	505	0.0075	505	0.3415	512	0.0063	512
	8	0.0103	15	0.1029	498	0.0207	498	0.5952	508	0.0142	508
	16	0.0163	15	0.3866	511	0.0848	511	1.1656	521	0.0303	521
50	2	0.0072	15	0.0545	441	0.0068	441	0.2142	442	0.0083	442
	4	0.0127	15	0.1000	472	0.0198	472	0.4062	486	0.0157	486
	8	0.0157	15	0.3722	490	0.0752	490	0.7946	501	0.0276	501
	16	0.0260	15	1.4740	488	0.6768	488	1.4978	498	0.0569	498
100	2	0.0112	16	0.0801	401	0.0166	401	0.3690	410	0.0144	410
	4	0.0183	16	0.3633	481	0.0768	481	0.5407	492	0.0385	492
	8	0.0283	16	1.3948	464	0.6269	464	1.0084	475	0.0744	475
	16	0.0524	16	5.5793	467	2.7446	467	2.1195	479	0.1993	479



(a)  $N = 50$  time steps



(b)  $N = 100$  time steps

Fig. 1. Ratio of computational times for ADMM algorithms to Gurobi for different  $N_r, N$ . Note that the y-axis is scaled logarithmically.

[2] S. Richter, C. Jones, and M. Morari, "Real-time input-constrained mpc using fast gradient methods," in *Proc. 48th IEEE Conf. on Dec. and Control*, Shanghai, China, 2009, pp. 7387–7393.

[3] S. Di Cairano, M. Brand, and S. Bortoff, "Projection-free parallel quadratic programming for linear model predictive control," *Int. J. Control*, 2013, in press, available online at [www.tandfonline.com](http://www.tandfonline.com).

[4] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal

parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," Tech. Rep. arXiv:1306.2454v1, 2013.

[5] A. U. Raghunathan and S. Di Cairano, "Alternating Direction Method of Multipliers (ADMM) for Strictly Convex Programs : Optimal Parameter Selection," in *Proceedings of American Control Conference*, 2014, pp. 4324–4329.

[6] —, "Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control," in *Proceedings of Symposium on Mathematical Theory of Networks and Systems*, 2014, pp. 807–814.

[7] —, "ADMM for Convex Quadratic Programs: Linear Convergence and Infeasibility Detection," arXiv, Tech. Rep. <http://arxiv.org/abs/1411.7288>, 2015.

[8] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Tr. Contr. Sys. Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[9] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Rob. Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[10] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer, 2011.

[11] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Trans. Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2012.

[12] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *Proc. 49th IEEE Conf. on Dec. and Control*, Atlanta, GA, 2010, pp. 6077–6082.

[13] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Trans. Contr. Systems Technology*, 2013, preprints available online.

[14] L. E. Sokoler, B. Dammann, H. Madsen, and J. B. Jrgensen, "A Decomposition Algorithm for Mean-Variance Based Economic Model Predictive Control," in *2014 IEEE Multi-Conference on Systems and Control (MSC)*, 2014, pp. 1086–1093.

[15] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.

[16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[17] (2014) OpenBLAS. [Online]. Available: <https://github.com/xianyi/OpenBLAS/wiki/faq>

[18] Gurobi Optimization, "Gurobi optimizer reference manual," 2014. [Online]. Available: <http://www.gurobi.com>

[19] S. Di Cairano, A. Ulusoy, and S. Haghghat, "Soft-landing control by control invariance and receding horizon control," in *American Control Conference*, 2014, p. 784789.

[20] J. Kang and J. D. Sirola and C. D. Laird, "Parallel Solution of Non-linear Contingency-constrained Network Problems," in *Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design*, 2014.