Application Set Approximation in Optimal Input Design for Model Predictive Control

Afrooz Ebadat*, Mariette Annergren*, Christian A. Larsson*, Cristian R. Rojas*, Bo Wahlberg*, Hakan Hjalmarsson*, Mats Molander** and Johan Sjöberg**

***Abstract*— This contribution considers one central aspect of experiment design in system identification, namely application set approximation. When a control design is based on an estimated model, the achievable performance is related to the quality of the estimate. The degradation in control performance due to plant-modeling missmatch is quantified by an application cost function. A convex approximation of the set of models that satisfy the control specification is typically required in optimal input design. The standard approach is to use a quadratic approximation of the application cost function, where the main computational effort is to find the corresponding Hessian matrix. Our main contribution is an alternative approach for this problem, which uses the structure of the underlying optimal control problem to considerably reduce the computations needed to find the application set. This technique allows the use of applications oriented input design for MPC on much more complex plants. The approach is numerically evaluated on a distillation control problem.**

I. Introduction

System identification for control concerns the problem of using experimental data from a dynamical system to identify a model to be used for control design, see e.g. [1], [2], [3], [4], [5], [6], [7], [8]. The opportunity to also design the excitation input signal to be used in the experiment opens up for the possibility to connect the system identification experimental conditions to required control performance. One way is to formulate this as a convex optimization problem[9], [10], [11], [12]. We will study one aspect of the so-called applications oriented input design introduced in [13], specifically for model predictive control (MPC), namely application set approximation. The objective is to guarantee, with a given probability, that the estimated model belongs to the set of models that satisfies the control specifications. This objective can be stated mathematically as a set constraint where the set of all identified models corresponding to a particular level of confidence must lie inside the set of all models fulfilling the control specifications [13]. To ensure that the obtained optimization problem is convex, we generally must make a convex approximation of the set constraint. Two known approaches for doing this are the scenario approach, [14] and [15], and the ellipsoidal approach, [16]. The main drawback of these methods is the computational effort necessary to obtain a decent approximation. Both methods require several simulations to be made of the closed loop system with MPC. In this paper, we introduce a new method of approximating the set constraint with a convex one. The method is based on a perturbation analysis and only requires one simulation of the closed loop system. Thus, it is expected to be much faster than both the scenario and the ellipsoidal approach.

The outline of the paper is as follows. In Section II, we go-through the mathematical background necessary. We describe the scenario approach and the ellipsoidal approach in Section III, followed by description of the proposed new method in Section IV. In Section V, we illustrate the method in two numerical examples and in Section VI, some conclusions are stated.

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 257059, the 'Autoprofit' project (www.fp7-autoprofit.eu). This work was partially supported by the Swedish Research Council under contract 621-2009-4017.

*Automatic Control Lab and ACCESS, School of Electrical Engineering, KTH, SE-100 44 Stockholm, Sweden. (e-mail: {ebadat, marann, christian.larsson, crro, bo, hjalmars}@ee.kth.se).

**ABB AB. Corporate Resarch, SE-721 78 Västerås (e-mail: {mats.a.molander, johan.sjoberg}@se.abb.com).

II. Preliminaries

A. System and model

We consider a linear, time-invariant system which is described by an output error model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + e(t). \end{aligned} \tag{1}$$

Here, $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the input vector, $y(k) \in \mathbb{R}^{n_y}$ is the output vector, and $e(t)$ is white Gaussian measurement noise with covariance matrix $E\{e(t)^2\} = \Lambda_e$. A, B and C are the state space matrices of the system. In system identification, we want to find a model of (1). Here the model is parametrized with an unknown parameter $\theta \in \mathbb{R}^n$, that is,

$$\begin{aligned} x(t+1,\theta) &= A(\theta)x(t,\theta) + B(\theta)u(t,\theta), \\ y(t,\theta) &= C(\theta)x(t,\theta) + e(t). \end{aligned} \tag{2}$$

In addition, we assume that the true system can be described by the model (2) exactly when $\theta = \theta_o$. We call θ_o the true parameter vector. The objective of system identification is to estimate the values of θ that best describes the system. The estimated parameter vector, given N measurements in the identification experiment, is denoted $\hat{\theta}_N$.

B. Prediction error method

We use the prediction error method (PEM) with quadratic cost to estimate the unknown parameters of the considered system, $\theta \in \mathbb{R}^n$, from N available samples of input-output data, see [17]. A key asymptotic ($N \to \infty$) property of PEM, is that the estimated parameters lie in an *identification set* with a certain probability [18]. This set is then defined as

$$\mathcal{E}_{SI}(\alpha) = \left\{ \theta : [\theta - \theta_o]^T I_F(\theta_o)[\theta - \theta_o] \leq \frac{\chi^2_\alpha(n)}{N} \right\}, \tag{3}$$

where $\chi^2_\alpha(n)$ is the α-percentile of the χ^2-distribution with n degrees of freedom and I_F is the Fisher information matrix. We thus have that $\hat{\theta}_N \in \mathcal{E}_{SI}(\alpha)$ with probability α when $N \to \infty$. For more details, we refer the reader to [17].

C. Model predictive control

Model predictive control (MPC) is an optimization based control technique where a model is used to predict the behavior of the plant. At each control instant, MPC computes a sequence of optimal inputs by solving an optimization problem based on the predictions. The first input value is applied to the plant and the procedure is repeated at the next time step. A common formulation of the MPC procedure is

$$
\begin{aligned}
\min_{\{u(k,\theta)\}_{k=1}^{N_u}} \quad & J=\sum_{k=0}^{N_y} \|y(k+1,\theta)-r(k+1)\|_Q^2+\sum_{k=1}^{N_u} \|\Delta u(k,\theta)\|_R^2 \\
\text{s. t.} \quad & x(k+1,\theta)=A(\theta)x(k,\theta)+B(\theta)u(k,\theta), \\
& y(k+1,\theta)=C(\theta)x(k+1,\theta), \\
& x(1,\theta)=x^*(t,\theta), \\
& \Delta u(1,\theta)=u(1,\theta)-u^*(t-1,\theta), \\
& u_{min} \le u(k,\theta) \le u_{max}, k=1,...,N_u, \\
& y_{min} \le y(k+1,\theta) \le y_{max}, k=0,...,N_y.
\end{aligned} \tag{4}
$$

Here $r(k)$ is the reference trajectory, Q and R are weight matrices, N_u and N_y are control and prediction horizons, and $\Delta u(k,\theta)=u(k,\theta)-u(k-1,\theta)$. Note that $\Delta u(k,\theta)=0$ for $k>N_u$, $u^*(t-1,\theta)$ is the optimal input applied to the system at time $t-1$, and $x^*(t,\theta)$ is the estimated system state at time t, obtained by direct measurement or observer. This and other MPC formulations are discussed in more detail in [19].

D. Applications oriented input design

Since MPC uses a model in order to control a system, the control performance is potentially affected by plant–model mismatch. We use the concept of an application cost function to relate the plant–model mismatch to the performance degradation. We use a scalar function of θ as the application cost and denote it $V_{app}(\theta)$. The cost function is chosen such that its minimum value occurs at $\theta=\theta_o$. In particular, we assume without loss of generality that $V_{app}(\theta_o)=0$. Note that if $V_{app}(\theta)$ is twice differentiable in a neighborhood of θ_o, this implies that $V_{app}(\theta_o)=0$, $V'_{app}(\theta_o)=0$ and $V''_{app}(\theta_o)\succeq 0$, see [16]. There are many different possible choices for the application cost function with the above mentioned properties and the proper choice is highly application dependent.

We note that the constructed application cost in most cases requires knowledge of the true parameter values which seems to contrast our aim of designing an input signal to be used for identification of unknown parameters. This is a problem which appears in almost any optimal input design formulation and is typically solved in one of two ways. One possibility is to do an input design which is robust with respect to the unknown parameters, e.g., [20]. Another possibility is to use iterative input design and use the best available estimate of the parameters to update the input design, e.g., [21]. We believe that the ideas presented here are extendable to an iterative input design scheme.

For a given application, there is a limit on the maximum value of allowed performance degradation, that is, $V_{app}(\theta)\le \frac{1}{\gamma}$, where γ is a user-defined positive constant. Every parameter vector θ for which the performance degradation is less than $1/\gamma$ can be considered as an acceptable parameter from an application's point of view. Thus, we define the set of all acceptable parameters, the *application set*, as

$$\Theta(\gamma)=\left\{\theta : V_{app}(\theta)\le \frac{1}{\gamma}\right\}. \tag{5}$$

The application set (5) has been extensively used in applications oriented input design for system identification (see [16], [22] and [13]). The main objective of applications oriented input design is to provide a tool for designing the input signal to be used in the identification experiment such that the estimated model guarantees acceptable control performance when used in the control design, that is, we want $\hat{\theta}_N \in \Theta(\gamma)$ with high probability. This requirement can be formulated mathematically as the inclusion

$$\mathcal{E}_{SI}(\alpha)\subseteq \Theta(\gamma). \tag{6}$$

Therefore, the input design problem can be formulated as an optimization problem, where (6) plays the role of a constraint. However, one crucial issue is that while $\mathcal{E}_{SI}$ is an ellipsoidal set, the application set can be of any shape. Thus, the set constraint (6) may not be convex. Two known approaches to make a convex approximation of the constraint are discussed in the next section. Alternatives to constraint (6) can be found in [23].

III. Application set approximation

Two methods of approximating the set constraint with a convex one are the scenario approach, see [14], [15], and the ellipsoidal approach, see [16].

In the scenario approach, the application set is described by a number, N_k, of samples (or scenarios) which are randomly chosen from the set. The constraint (6) is then replaced by a set of inequalities,

$$[\theta-\theta_0]^T I_F(\theta_0)[\theta-\theta_0] \ge \frac{\gamma\chi^2_\alpha(n)}{N} V_{app}(\theta_k), k=1,\dots,N_k. \tag{7}$$

However, in order to have a good approximation of the application set, the number of scenarios must be large enough (see e.g. [24] for the minimum required number of scenarios). For high dimensional and complex plants using controllers where it is not possible to find analytic expressions for V_{app}, which is the case for MPC, this requires a large number of often highly time-consuming and costly simulations.

The ellipsoidal approach is based on a second order Taylor expansion of $V_{app}(\theta)$ around θ_o, that is,

$$
\begin{aligned}
V_{app}(\theta) &\approx V_{app}(\theta_o)+V'_{app}(\theta_o)[\theta-\theta_o] \\
&\quad +\frac{1}{2}[\theta-\theta_o]^T V''_{app}(\theta_o)[\theta-\theta_o] \\
&= 0+0+\frac{1}{2}[\theta-\theta_o]^T V''_{app}(\theta_o)[\theta-\theta_o].
\end{aligned} \tag{8}
$$

The application set can thus be approximated by

$$\mathcal{E}_{app}(\gamma)=\left\{\theta : [\theta-\theta_o]^T V''_{app}(\theta_o)[\theta-\theta_o]\le \frac{2}{\gamma}\right\}. \tag{9}$$

The quality of the approximation not only depends on the application cost but also on the value of γ. For sufficiently large values of γ, $\mathcal{E}_{app}$ gives an acceptable approximation. However, the calculation of the Hessian matrix is a challenging task. In many problems it is not possible to analytically determine the Hessian of the application function due to nonlinearities in the controllers that are being used. Therefore, numerical approximations are used. Numerical methods, such as finite difference approximation, are not applicable in many cases because of the large number of variables involved. In the rest of this paper, we propose a method for convex approximation of the application set (5) for constrained MPC, where the main focus is to find an analytic method to calculate the Hessian.

IV. Application set approximation for MPC

MPC has drawn much attention in control fields, thanks to its ability to cope with system constraints. Using MPC, we can deal with both input and output constraints explicitly during the controller design and implementation. However, the resulting explicit solutions for MPC are difficult to deal with due to these constraints, which makes it unavoidable to use numerical calculations for the approximation of the application cost [22]. In this section we present a new approach based on analytical methods for the application cost approximation for MPC. The proposed approach leads to faster convex estimations of the application sets for MPC.

A. Application Cost Function for MPC

The application cost function measures the amount of performance degradation that stems from plant–model mismatch. One reasonable choice of this function for MPC is the difference between the measured output when the controller is based on the true parameters, θ_o, and when it is based on perturbed parameters θ, that is,

$$V_{app}(\theta) = \frac{1}{M}\sum_{t=1}^{M} \|y(t,\theta_o) - y(t,\theta)\|^2, \tag{10}$$

where M is the number of measurements used, t is time, and the second argument of y is the parameters used in the MPC.

The application cost (10) is one of the common cost functions that has been used in the literature, see e.g. [14], [16]. Therefore, in this paper we focus on an application set approximation for (10), however, the proposed approach is applicable for other choices of application costs.

B. The Unknown True System

As noted before, the application cost may depend on the true parameter values, and for the case (10), it does. Therefore, the following simulation based approximation of the application cost is suggested

$$\widehat{V}_{app}(\theta) = \frac{1}{M}\sum_{t=1}^{M} \|y(t,\hat\theta,\hat\theta) - y(t,\theta,\hat\theta)\|^2, \tag{11}$$

where $\hat\theta$ is the best available estimate of θ_o and the values of y come from simulations; the second argument of y is the parameter used in the MPC formulation (4) while the third argument of y is the parameter used in lieu of θ_o. The values of $\hat\theta$ may be updated as better estimates are available.

C. Application Function Approximation

In order to obtain a convex approximation of the application set, we start by estimating $y(t,\theta,\hat\theta)$ in (11). Using a Taylor expansion of $y(t,\theta,\hat\theta)$, we can write

$$\begin{aligned} y(\theta) = y(\hat\theta) &+ \sum_{i=1}^{n}\frac{\partial y(\hat\theta)}{\partial\theta_i}\delta\theta_i \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2 y(\hat\theta)}{\partial\theta_i\partial\theta_j}\delta\theta_i\delta\theta_j + H.O.T., \end{aligned} \tag{12}$$

where θ_i are the elements of θ and by H.O.T. we mean higher order terms in the Taylor expansion. Here, the first and third arguments of $y(t,\theta,\hat\theta)$, are omitted for the sake of simplicity. In order to find the derivatives in (12), we need to find the derivatives of the input signal generated by MPC with respect to θ. However, this is a challenging problem since the solution of MPC is not simple enough when there are inequality constraints on input and output signals. The proposed solution here is to notice that, when θ is a small perturbation of $\hat\theta$, the active constraints are the same as when the MPC is based on $\hat\theta$. Thus, the main idea is to let MPC run based on $\hat\theta$ at each time instance t, and determine the optimal value of the input signal $u(t,\hat\theta)$. We assume that the active constraints remain the same for small enough perturbations of $\hat\theta$, see [25]. This fact holds whenever the dual variables associated with the last two set of constraints in (4), which are active, are all non-zero. This in turn hold with the probability 1. In addition, under this condition y is differentiable with respect to its second argument.

Therefore, at time step t, we are able to find an explicit solution of the optimization problem in MPC for $\theta = \hat\theta + \delta\theta$ by considering active constraints as equality constraints. We can analyze the effects of perturbing the parameters when $\delta\theta$ is small enough. In the rest of this section, we briefly describe the explicit solution of MPC when we are considering only active constraints, then we provide insights into the perturbation analysis for the MPC solution. Finally, we show how these concepts can be used to find the derivatives in (12) and compute the application cost function.

1) Explicit Solution of constrained MPC: Consider the MPC formulation (4) at time instant t. For simplicity it is assumed that $N_u = N_y$. Now we seek to rewrite the MPC formulation as a quadratic program where we are considering only active constraints obtained by solving MPC for $\hat\theta$, which are equality constraints. Introducing

$$\begin{aligned} &X(t,\theta) = \\ &[x(t+N_u,\theta)^T,\ldots,x(t,\theta)^T,u(t+N_u-1,\theta)^T,\ldots,u(t,\theta)^T]^T, \\ &\mathcal{H} = [r(N_u+t)^T,\ldots,r(t)^T,0,\ldots,0,u^*(t-1,\theta)^T]^T, \end{aligned}$$

$$\mathcal{Q} = \begin{bmatrix} I_{N_u+1}\otimes Q & 0 \\ 0 & I_{N_u}\otimes R \end{bmatrix},$$

$$\Delta = \begin{bmatrix} I & -I & \cdot & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdot & I & -I \\ 0 & \cdot & 0 & I \end{bmatrix}, \Upsilon(\theta) = \begin{bmatrix} I_{N_u+1}\otimes C(\theta) & 0 \\ 0 & I_{N_u}\otimes\Delta \end{bmatrix},$$

where by $I_m \otimes M$, we mean the Kronecker products of I_m and M [26], we can rewrite the cost function J in (4) as:

$$J = (\Upsilon(\theta)X(t,\theta) - \mathcal{H})\mathcal{Q}(\Upsilon(\theta)X(t,\theta) - \mathcal{H})^T. \quad (13)$$

Moreover, the system dynamics and the first equality constraint in (4), give that $\mathcal{C}(\theta)X(t,\theta) = \mathcal{D}(\theta)$, with

$$\mathcal{C}=\begin{bmatrix} I & -A(\theta) \dots 0 & 0 & -B(\theta) \dots & 0 \\ \vdots & \vdots \ \ddots \vdots & \vdots & \vdots \ \ddots & \vdots \\ 0 & 0 \ \dots I & -A(\theta) & 0 & \dots -B(\theta) \\ 0 & 0 \ \dots 0 & I & 0 & \dots \ 0 \end{bmatrix}, \mathcal{D}=\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hat{x}(t,\theta) \end{bmatrix}.$$

Consider the inequality constraints in (4). They can be rewritten as

$$\begin{bmatrix} I_{N_u+1} \otimes C(\theta) & 0 \\ -I_{N_u+1} \otimes C(\theta) & 0 \\ 0 & I \\ 0 & -I \end{bmatrix} X(\theta) \leq \begin{bmatrix} I_{N_u+1} \otimes y_{max} \\ -I_{N_u+1} \otimes y_{min} \\ I_{N_u} \otimes u_{max} \\ -I_{N_u} \otimes u_{min} \end{bmatrix}. \quad (14)$$

Let Ξ be a diagonal matrix, where each diagonal element corresponds to one of the inequality constraints in (14). A diagonal element is zero if its corresponding constraint is inactive when the problem is solved for $\hat{\theta}$ and it is one for active constraints. Multiplying (14) by Ξ and introducing

$$\Xi_a = \Xi \begin{bmatrix} I_{N_u+1} \otimes C(\theta) & 0 \\ -I_{N_u+1} \otimes C(\theta) & 0 \\ 0 & I \\ 0 & -I \end{bmatrix}, \rho = \Xi \begin{bmatrix} I_{N_u+1} \otimes y_{max} \\ -I_{N_u+1} \otimes y_{min} \\ I_{N_u} \otimes u_{max} \\ -I_{N_u} \otimes u_{min} \end{bmatrix},$$

we obtain $\Xi_a = \rho$, which represents those inequality constraints that are active at time instance t. Then we can rewrite the entire set of constraints as $\mathcal{A}(\theta)X(t,\theta) = \mathcal{B}(\theta)$, where

$$\mathcal{A}(\theta) = \begin{bmatrix} \mathcal{C}(\theta) \\ \Xi_a \end{bmatrix}, \ \mathcal{B}(\theta) = \begin{bmatrix} \mathcal{D}(\theta) \\ \rho \end{bmatrix}.$$

Finally, the following optimization problem is obtained:

$$\begin{aligned} \min_{X(t,\theta)} \quad & (\Upsilon(\theta)X(t,\theta) - \mathcal{H})\mathcal{Q}(\Upsilon(\theta)X(t,\theta) - \mathcal{H})^T, \\ \text{s.t.} \quad & \mathcal{A}(\theta)X(t,\theta) = \mathcal{B}(\theta). \end{aligned} \quad (15)$$

Problem (15) is a quadratic optimization problem with equality constraints. The KKT conditions [27] for this problem are

$$\begin{aligned} & 2\Upsilon(\theta)^T \mathcal{Q}(\Upsilon(\theta)X(t,\theta) - \mathcal{H}) + \mathcal{A}^T(\theta)\lambda = 0, \\ & \mathcal{A}(\theta)X(t,\theta) = \mathcal{B}(\theta), \end{aligned}$$

where λ are Lagrange multipliers. This can be written as

$$\begin{bmatrix} 2\Upsilon(\theta)^T \mathcal{Q}\Upsilon(\theta) & \mathcal{A}(\theta)^T \\ \mathcal{A}(\theta) & 0 \end{bmatrix} \begin{bmatrix} X(t,\theta) \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\Upsilon(\theta)^T \mathcal{Q}\mathcal{H}(\theta) \\ \mathcal{B}(\theta) \end{bmatrix}, \quad (16)$$

or equivalently

$$\Psi(\theta) \begin{bmatrix} X(t,\theta) \\ \lambda \end{bmatrix} = \Lambda(\theta). \quad (17)$$

Since the block matrices inside $\Psi(\theta)$ are not invertible, (17) can be explicitly solved using the pseudoinverse and Schur complement of the resulting block matrix

$$\begin{bmatrix} X(t,\theta) \\ \lambda \end{bmatrix} = (\Psi(\theta)^T \Psi(\theta))^{-1} \Psi(\theta)^T \Lambda(\theta). \quad (18)$$

Solving (18), we can obtain an explicit solution for $X(t,\theta)$.

Notice that $\Psi(\theta)$ is usually invertible, however, its inverse does not have an explicit simple expression, since the block matrices inside $\Psi(\theta)$ are not necessarily invertible. In order to compute an explicit solution for $\frac{\partial X(t,\theta)}{\partial \theta}$, we use the pseudoinverse of $\Psi(\theta)$ in (18) instead of its inverse, since $\Psi(\theta)^T \Psi(\theta)$ does have a simple expression for its inverse.

2) Perturbation Analysis: The analysis in this section is based on the analysis techniques in [28] and [29]. Having the MPC solution at time step t, i.e. $X(t,\theta)$, our aim is to compute its derivatives with respect to θ, based on which the derivatives in (12) will be calculated. This can be obtained by linearizing $X(t,\theta)$ around $\hat{\theta}$, invoking the Taylor expansion

$$\begin{aligned} X(t,\theta) = X(t,\hat{\theta}) & + \sum_{i=1}^{n} \frac{\partial X(t,\hat{\theta})}{\partial \theta_i} \delta\theta_i \\ & + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 X(t,\hat{\theta})}{\partial \theta_i \partial \theta_j} \delta\theta_i \delta\theta_j + H.O.T \ , \end{aligned} \quad (19)$$

where $\theta = \hat{\theta} + \delta\theta$. The Taylor expansion of $X(t,\theta)$, i.e. (19), can be computed writing the Taylor expansions of $\mathcal{A}(\theta)$, $\mathcal{B}(\theta)$, $\Upsilon(\theta)$, and $\mathcal{H}(\theta)$, which in turn are derived by having the derivatives of $A(\theta)$, $B(\theta)$, $C(\theta)$[1], $\hat{x}(t,\theta)$, and $u^*(t-1,\theta)$. Note that the derivatives of $\hat{x}(t,\theta)$, and $u^*(t-1,\theta)$ are available from the Taylor expansion of $X(t-1,\theta)$ in the previous time instants [2]. Now, recall the plant linear model

$$\begin{aligned} x(t+1,\theta) &= A(\theta)x(t,\theta) + B(\theta)u^*(t,\theta), \\ y(t,\theta) &= C(\theta)x(t,\theta), \end{aligned} \quad (20)$$

where $u^*(t,\theta)$ is the optimal input designed by MPC. We aim to find the coefficients in (12). They can be calculated in a recursive manner by differentiating (20) with respect to θ, using the derivatives of $u(t,\theta)$, exploiting (19).

3) Application Cost Function: For the application function (11), we can calculate the Hessian matrix in terms of the obtained derivatives of y as follows:

$$\begin{aligned} \widehat{V}''_{app}(\theta) = & \frac{2}{M} \sum_{t=1}^{M} \left\{ \frac{\partial y(t,\hat{\theta})}{\partial \theta} \right\}^T \left\{ \frac{\partial y(t,\hat{\theta})}{\partial \theta} \right\} \\ & + \frac{2}{M} \sum_{t=1}^{M} \left\{ \frac{\partial^2 y(t,\hat{\theta})}{\partial \theta^2} \right\}^T \left\{ y(t,\hat{\theta},\hat{\theta}) - y(t,\hat{\theta},\hat{\theta}) \right\}. \end{aligned} \quad (21)$$

Note that the second term is zero. Substituting (21) into (9), we get a convex approximation of the application set.

The method provides a fast tool for convex approximation of application cost function. Many calculations in different time instants are the same and can be pre-computed. Moreover, the active constraints may not change often, thus, at each time instant a large number of the calculations can be skipped by re-using the results from previous time instants. Thus, the proposed approach is often faster than both the scenario-based and the ellipsoidal approximation method.

[1] The structure of the matrices $A(\theta)$, $B(\theta)$ and $C(\theta)$ may be either known a priori due to physical properties of the considered system (grey-box system identification), or unknown (black-box system identification). For the latter case, each element in these matrices is considered as an unknown parameter.

[2] The derivatives of $\hat{x}(t,\theta)$ depend on the observed states by the observer.

V. Numerical examples

In this section we evaluate the proposed method in Section IV with two numerical examples.

A. Example 1

Consider the following system:

$$\begin{aligned} x(t+1) &= \theta_2 x(t) + u(t), \\ y(t) &= \theta_1 x(t) + e(t). \end{aligned} \tag{22}$$

The true system is given by the parameter values $\theta_0 = [0.6 \;\; 0.9]^T$ and the measurement noise has the variance $\lambda_e^2 = 0.01$. The objective is to find the application set Θ, when MPC is used for reference tracking. We use the MPC formulation in (4), with the following settings: $N_u = N_y = 5$, $Q = 10$, $R = 1$, $u_{max} = -u_{min} = 1$, $y_{max} = -y_{min} = 2$. We set the length of the experiment to $N = 10$ samples and the reference trajectory is a series of unit steps over the samples. Note that we use the application cost function defined in (10). Now using the proposed approach, we can obtain the application ellipsoid. The level curves for the application set together with the approximation of the set based on the proposed approach and the uniformly distributed scenarios are in shown in Fig. 1. The results are studied for different values of accuracy, i.e γ.

In order to check the accuracy of the proposed method, we perform a number of scenarios with different values of θ which are generated randomly with a uniform distribution. The experiment has been done for different values of γ. The results show that from 400 generated points for $\gamma = 1000$, 122 points are satisfying the condition $V_{app}(\theta) < \frac{1}{\gamma}$. Among all accepted values of θ, 90% are completely inside or on the border of the approximated ellipsoid, which means that the estimated ellipsoid covers at least 90% of the acceptable points. This value increases to 100% when $\gamma = 10000$. This mainly stems from the fact that the Taylor approximation of application cost function around θ_o is more accurate when we are closer to θ_o.

Furthermore, the Hessian matrix is computed employing numerical methods, provided by DERIVESTsuite. The application set is then approximated using the ellipsoidal approach (9). As expected, the result is the same as when the proposed method is used. However, in the proposed method, we need only one complete simulation of the closed loop system with MPC, while in the numerical approximation of the Hessian, which is based on finite difference approximation, $O(6n^2)$ simulations are required depending on the selected accuracy. Therefore, the new approach is expected to be faster. While it takes 94 seconds for the numerical method to calculate the Hessian matrix in this example, the new method needs only 12 seconds to give the same approximation, which means that 87% of time is saved.

B. Example 2

In this example we illustrate the algorithm on a distillation column simulation example. The nonlinear system representation is taken from a benchmark process proposed by the Autoprofit project [30]. For a general description of distillation columns, we refer the reader to [31].

The plant is linearized around the steady state operating conditions and then, using model order reduction methods, the second order model

$$\begin{aligned} x(t+1) &= \begin{bmatrix} \theta_1 & \theta_2 \\ \theta_3 & \theta_4 \end{bmatrix} x(t) + \begin{bmatrix} \theta_5 & \theta_6 \\ \theta_7 & \theta_8 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} -0.8954 & 0.1421 \\ -0.2118 & -0.1360 \end{bmatrix} x(t) + e(t) \end{aligned} \tag{23}$$

is obtained, where $e(t)$ is a white measurement noise with variance $E\{e(t)^T e(t)\} = 0.001$. We assume that 1% performance degradation from the case when MPC is using the true parameters is allowed, that is, $\gamma = \frac{100}{V(\theta_0)}$, where $V(\theta_0) = \frac{1}{M}\sum_{t=1}^{M} \|y(t,\theta_0,\theta_0) - r(t)\|^2$, see [22]. Since MPC is used for tracking, the model is augmented with a constant output disturbance on each output to get integral action. This technique is presented in further detail in [19]. The proposed method has been employed to calculate the approximate application cost in (8). In order to evaluate the capability of the method, we run the process for 100 different values of θ, taken from a uniform distribution. Fig. 2 shows the real and approximated values of the application cost function for different values of θ.

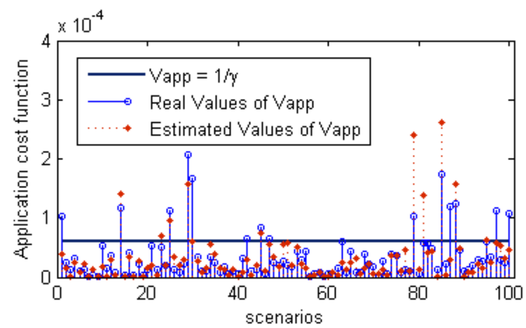

Fig. 2. Approximated ('o –') and real ('. - -') values of $V_{app}(\theta)$ for 100 different samples of θ taken form a uniform distribution.

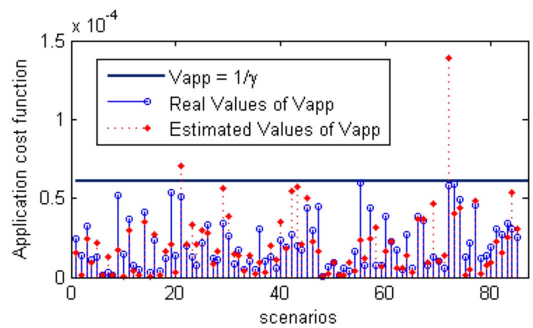

Fig. 3. Approximated ('o –') and real ('. - -') values of $V_{app}(\theta)$ inside the application set. 92% of the samples inside the region are classified as acceptable ones by the proposed method.

In order to have a better insight, the samples which are located inside the application set are illustrated in Fig. 3. It can be easily seen that the proposed method has a good performance inside the application set. Among 85 scenarios which result in an acceptable application cost, 83 scenarios are approximated as acceptable ones using the proposed method. The method classifies 6 points outside the region

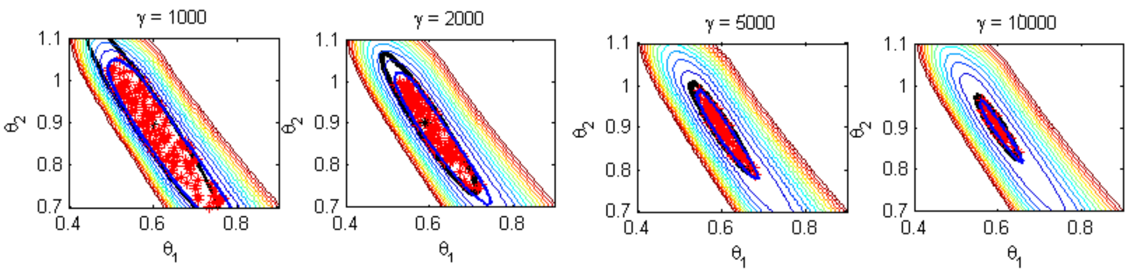

Fig. 1. Level curves ('——') for the application set defined in (10). The innermost curve corresponds to the required accuracy ('▬'). The approximated ε_{app} ('▬') is shown. The approximation is much better for larger values of γ. The accepted scenarios, i.e. $V_{app}(\theta) \leq \frac{1}{\gamma}$, for all cases are shown ('$*$') .

as acceptable ones. Therefore, the obtained accuracy of the proposed method is around 92%.

VI. Conclusions

In this paper we have introduced a general technique for the approximation of the application set, a structure required for the implementation of optimal input design schemes. In particular, we have focused on MPC, a control technique for which it is not possible to obtain the application set explicitly. Some simulation examples have been presented, which show the advantages of the new method with respect to previous techniques,in terms of speed.

The method is general enough to be applied to other controller strategies and application areas where it is not possible to derive the application set explicitly. Specifically, the method can be extended to MPC for nonlinear plants, with more complicated noise structures, and the derivation of expressions for higher order derivatives of the cost function could be used, in principle, to obtain better approximations of the application set using techniques such as the one in [32].

References

[1] H. Hjalmarsson, "From experiment design to closed loop control," *Automatica*, vol. 41, no. 3, pp. 393–438, Mar. 2005.

[2] U. Forssell and L. Ljung, "Some results on optimal experiment design," *Automatica*, vol. 36, no. 5, pp. 749–756, May 2000.

[3] G. Belforte and P. Gay, "Optimal experiment design for regression polynomial models identification," *International Journal of Control*, vol. 75, no. 15, pp. 1178–1189, 2002.

[4] C. Jauberthie, L. Denis-Vidal, P. Coton, and G. Joly-Blanchard, "An optimal input design procedure," *Automatica*, vol. 42, no. 5, pp. 881 – 884, 2006.

[5] C. R. Rojas, J. S. Welsh, G. C. Goodwin, and A. Feuer, "Robust optimal experiment design for system identification," *Automatica*, vol. 43, no. 6, pp. 993–1008, June 2007.

[6] C. R. Rojas, J. C. Agüero, J. S. Welsh, and G. C. Goodwin, "On the equivalence of least costly and traditional experiment design for control," *Automatica*, vol. 44, no. 11, pp. 2706–2715, 2008.

[7] G. Franceschini and S. Macchietto, "Model-based design of experiments for parameter precision: State of the art," *Chemical Engineering Science*, vol. 63, no. 19, pp. 4846 – 4872, 2008.

[8] L. Pronzato, "Optimal experimental design and some related control problems," *Automatica*, vol. 44, no. 2, pp. 303–325, February 2008.

[9] R. Hildebrand and M. Gevers, "Identification for control: Optimal input design with respect to a worst-case ν-gap cost function," *SIAM J. Control Optim*, vol. 41, no. 5, pp. 1586–1608, 2003.

[10] M. Barenthin, H. Jansson, H. Hjalmarsson, J. Mårtensson, and B. Wahlberg, "A control perspective on optimal input design in system identification," in *Forever Ljung in System Identification*. Studentlitteratur, Sep. 2006, ch. 10.

[11] X. Bombois, G. Scorletti, M. Gevers, P. M. J. Van den Hof, and R. Hildebrand, "Least costly identification experiment for control," *Automatica*, vol. 42, no. 10, pp. 1651–1662, 2006.

[12] H. Hjalmarsson and H. Jansson, "Closed loop experiment design for linear time invariant dynamical systems via LMIs," *Automatica*, vol. 44, no. 3, pp. 623–636, 2008.

[13] H. Hjalmarsson, "System identification of complex and structured systems," *European Journal of Control*, pp. 275–2310, 2009.

[14] C. A. Larsson, C. R. Rojas, and H. Hjalmarsson, "MPC oriented experiment design," in *18th IFAC World Congress*, no. 1986, Milano (Italy) August, 2011, pp. 9966–9971.

[15] G. Calafiore and M. Campi, "The Scenario Approach to Robust Control Design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[16] B. Wahlberg, H. Hjalmarson, and M. Annergren, "On Optimal Input Design in System Identification for Control," in *49th IEEE Conference on Decision and Control*, Atlanta, GA, USA, 2010.

[17] L. Ljung, *System Identification - Theory for the user*, 2nd ed. New Jersey: Prentice Hall, Upper Saddle River, 1999.

[18] L. Ljung and B. Wahlberg, "Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra," *Advances in Applied Probability*, vol. 24, no. 2, pp. 412–440, 1992. [Online]. Available: http://www.jstor.org/stable/1427698

[19] J. M. Maciejowski, *Predictive Control with Constraints*. Edinburgh Gate, Harlow, Essex, England: Prentice Hall, 2002.

[20] C. R. Rojas, "Robust experiment design," Ph.D. dissertation, The University of Newcastle, June 2008.

[21] L. Gerencser, H. Hjalmarsson, and J. Martensson, "Identification of arx systems with non-stationary inputs - asymptotic analysis with application to adaptive input design," *Automatica*, vol. 45, no. 3, pp. 623 – 633, 2009.

[22] C. A. Larsson, M. Annergren, and H. Hjalmarsson, "On Optimal Input Design in System Identification for Model Predictive Control," in *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, 2011, pp. 805–810.

[23] C. R. Rojas, D. Katselis, H. Hjalmarsson, R. Hildebrand, and M. Bengtsson, "Chance constrained input design," *IEEE Conference on Decision and Control and European Control Conference*, no. 5, pp. 2957–2962, Dec. 2011.

[24] M. C. Campi and S. Garatti, "The Exact Feasibility of Randomized Solutions of Uncertain Convex Programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[25] S. Deshpande, B. Srinivasan, and D. Bonvin, "How Important is the Detection of Changes in Active Constraints in Real-Time Optimization," in *18th IFAC World Congress*, Milano (Italy) August, 2011.

[26] A. J. Laub, *Matrix Analysis for Scientists and Engineers*. USA: Society of Industrial and Applied Mathematics (SIAM), 2005.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2012.

[28] X.-R. Cao, "Stochastic learning and optimization : A sensitivity-based approach," *Annual Reviews in Control*, vol. 33, no. 1, pp. 11–24, 2009.

[29] Y. Ho, "Parametric Sensitivity of a Statistical Experiment," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 982–983, 1979.

[30] "Autoprofit." [Online]. Available: http://www.fp7-autoprofit.eu/home/

[31] S. Skogestad, "Dynamics and control of distillation columns: A tutorial introduction," *Chemical Engineering Research and Design*, vol. 75, no. 6, pp. 539 – 562, 1997.

[32] H. Hjalmarsson and F. Egebrand, "Input design using cylindrical algebraic decomposition," in *IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, Dec. 2011, pp. 811–817.