Proceedings of the
2005 IEEE Conference on Control Applications
Toronto, Canada, August 28-31, 2005

Dynamic Safety Margin in Fault-Tolerant Predictive Controller

M. Abdel-Geliel, E. Badreddin, A. Gambier

Automation Lab, University of Mannheim, Germany
elgeliel@ti.uni-mannheim.de, badreddin@ti.uni-mannheim.de, gambier@ti.uni-mannheim.de

***Abstract* — Dynamic safety margin (DSM) is a new performance index used to measure the distance between a predefined safety boundary in the state space and the system trajectory as it evolves. Controller design based on DSM is important to maintain a predefined margin of safety during the transient and in the presence of large disturbances particularly in safety-critical systems. In this paper, a fault tolerant control design, using predictive controller based on DSM, to recover system performance after some system faults is discussed. In addition, real-time results of a control system, which was implemented in a two-tank system, are presented to demonstrate the fruitfulness of this design.**

I. Introduction

The evaluation of a control system depends mainly on the difference between the desired performance according to the given specifications and the actual performance. Safety is one of the most important specifications for the controlled system. Safety control problem requires moving the system from a given set of initial states in its state space to a predetermined safe region. Dynamic Safety Margin (DSM) is a new performance index for the control system design, which was introduced in [1], [2] and [16]. This index measures how far the system state trajectory is from a predefined safety boundary in the state space. The state variables of interest have to be inside that boundary region in normal operation and in case of uncertainties and/or disturbances. Thus, controller design based on DSM permits to maintain a predefined margin of safety during transient and steady state of safety-critical systems. Moreover, it can be used in Fault Tolerant Control system design (FTC) [3], [4] in order to speed up performance recovery in some cases of faults.

Some methods for introducing DSM into controller design are stated in [1] and [16]. In those contributions, adapting parameters of PID controllers, switching controller and/or optimal control are highlighted. An algorithm for the computation of DSM and the use of DSM in fault diagnosis and isolation is discussed in [2]. Model-based Predictive Control (MPC) belongs to a class of approaches that determines the optimal control profile according to a prediction of the system behavior over a receding time horizon, i.e. a sequence of future control actions is chosen in order to predict the evolution of the system and it is applied to the plant until new measurements are available. Then, a new sequence is determined, which replace the previous one [5]-[7]. DSM can be introduced in MPC as either a hard constraint or an additional term in the performance index (soft constraints). The way to do this, the analysis for FTC and the practical application in real-time is the main contribution of this work.

The paper is organized as follow: First, DSM for safety-critical system and the FTC problem are explained. Next, state-space predictive controller design based on DSM to maintain a predefined margin of safety during the system operation is developed and simulation results are presented. It follows an illustrative real-time example on a two-tank laboratory prototype with industrial components. Finally, conclusions and future work are drawn.

II. Definition of FTC and DSM

An FTC system is a control system that can accommodate components faults and is able to maintain stability and acceptable degree of performance when not only the system is fault-free but also when there are component malfunctions. The FTC prevents faults in a subsystem from developing into failures at system level. The design of FTC systems can be classified as passive as well as active (PFTC and AFTC). In PFTC, a system may tolerate only a limited number of faults, which are assumed to be known prior to the design of the controller. Once the controller is designed, it can compensate anticipated faults without any access of on-line fault information. AFTC compensates the effect of faults either by selecting a pre-computed control law, or by synthesizing a new control low on-line in real-time.

In the following, the general idea will briefly be explained (see [1] for details). Let X be the state space in $\Re^n$, and consider that a subspace $\Phi \subseteq X$, which defines the safe operation region for some subset of state variables, $\mathbf{x} \in \Re^m$ in the state subspace Φ, can be specified by a set of inequalities $\{\phi_i(\mathbf{x}) \le 0, i=1,...q\}$, where ϕ_i: $\Re^m \to \Re$. $\phi_i(\mathbf{x}) > 0$ indicates unsafe operation (Fig. 1). It is assumed that the system is stable in the sense of Lyapunov and that the safe region is fully contained in the stability region. Starting with the initial condition $\mathbf{x}_0$, the system trajectory will evolve to the operating point $\mathbf{x}_s$ traversing the state space with varying distance to the safety boundary. DSM is defined as the instantaneous shortest distance $\delta(t)$, between the state variables of interest and a predefined boundaries $\{\phi_i(\mathbf{x}) = 0, i=1,...q\}$ in this subspace of state variables. When the system reaches the operating point $d\delta(t)/dt = 0$ and $\delta(\cdot)$ reaches a constant value indicating the Stationary Safety Margin (SSM). Most industrial

designs are done trying to satisfy specified values of SSM.

It is necessary to distinguish between safety boundary and individual state limits of amplitudes in time domain. Sometimes, some of the safety boundaries are defined by the state limits but not always. Fig. 2 shows the difference between amplitude bound of variables and safety boundary. It is clear from Fig. 2 that all state variables lays inside its individual amplitude boundary but some state vectors do not satisfy safety boundary constrains. In general, the safe-operation region Φ is defined by a set of inequalities given by

$$\Phi = \left\{ \phi_i(\mathbf{x}) < 0 \middle| i = 1, \ldots, q \right\} \tag{1}$$

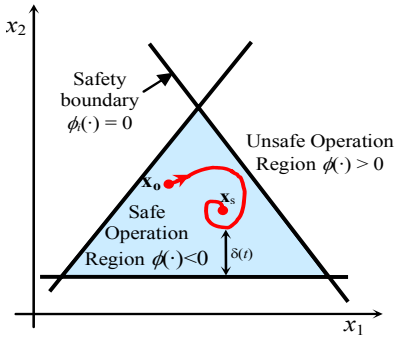

Fig. 1: DSM definition

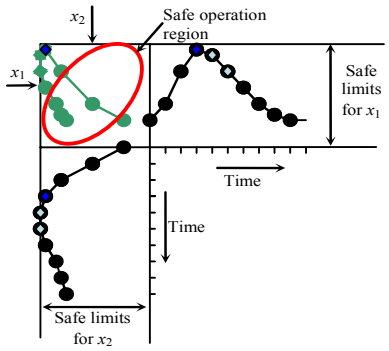

Fig. 2. DSM and state limits

and DSM is defined as

$$\delta(t) = \min_{1 \le i \le q} \delta_i(t) \tag{2}$$

$$\delta_i(t) = s(t) \cdot \left\| \mathbf{x}_i \big|_{\phi_i(\mathbf{x}_i)} - \mathbf{x} \right\|_{\min}, \ \mathbf{x} = [x_1, x_2, \ldots, x_m]^T \tag{3}$$

where $s(t)$ is given by

$$s(t) = \begin{cases} 1 & \text{if } \mathbf{x} \textbf{ inside} \text{ the safe operation region} \\ -1 & \text{if } \mathbf{x} \textbf{ outside} \text{ the safe operation region} \end{cases}$$

and $\|\cdot\|_{\min} \hat{=}$ shortest distance from $\mathbf{x}(t)$ to $\mathbf{x}_i$, respectively.

Variable q is the number of defined inequalities and m the number of state variables relevant to safety. Notice that $m \le n$ the dimension of the state-space.

In most cases, the safe operation region can be defined by a set of linear inequalities $\{\phi_i \le 0\}$. In case that the boundary function ϕ_i is nonlinear, it can be subdivided into two or more linear constraints (piecewise linear approximation).

If Φ is convex defined by linear boundary constraints and the variables of interest are given by the whole state vector, i.e. $m = n$, then the safe region Φ is polytope and defined by q linear inequalities in the form

$$\phi_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} - c_i \le 0 \tag{4}$$

where $\mathbf{a}_i^T \in \Re^n$, $c_i \in \Re$ is a constant and $\phi_i(\cdot) = 0$ is a subspace of state vector $\mathbf{x}_i \subset \Phi$ where $\mathbf{a}_i^T \mathbf{x}_i = c_i$. Thus, for the state vector $\mathbf{x}$, $\delta_i(\cdot)$ can be calculated [2] as

$$\delta i(t) = \frac{c_i - \mathbf{a}_i^T \mathbf{x}(t)}{\|\mathbf{a}i\|_2} \begin{cases} \ge 0 & \text{iff } \phi_i(\mathbf{x}) < 0 \\ < 0 & \text{iff } \phi_i(\mathbf{x}) > 0 \end{cases} \tag{5}$$

For all boundaries, the distance vector

$$\mathbf{d}(t) = [\delta_1(t), \delta_2(t), \ldots, \delta_q(t)]^T$$

can be obtained from

$$\mathbf{d}(t) = \mathbf{d}_c - \mathbf{D}_a \mathbf{x}(t) \in \Re^q \tag{6}$$

where $\mathbf{d}_c = \mathbf{D}_{ia} \mathbf{c}_c \in \Re^q$, $\mathbf{D}_a = \mathbf{D}_{ia} \mathbf{A}_c \in \Re^{q \times n}$

$$\mathbf{D}_{\text{ia}} = diag \left(\frac{1}{\|\mathbf{a}_1\|_2}, \frac{1}{\|\mathbf{a}_2\|_2}, \cdots, \frac{1}{\|\mathbf{a}_q\|_2} \right) \in \Re^{q \times q};$$

$$\mathbf{c}_c = [c_1 \quad c_2 \quad \cdots \quad c_q]^T \in \Re^q; \mathbf{A}_c = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_q]^T \in \Re^{q \times n}$$

$\delta(\cdot)$, DSM, is the minimum element in $\mathbf{d}(\cdot)$ according to (1).

III. Fault Tolerant Control and DSM

According to the definition of DSM in [1] and [2], DSM should be $\delta(\cdot) \ge 0$; otherwise there is a fault or large disturbance. Hence, a FTC design based on DSM satisfies the desired response and maintains the system state within the safe region. Moreover, it should have the ability to bring the system states to the safe region as fast as possible when it, for some reason, reached an abnormal situation. FTC based on DSM can be passive or active [3],[4]. If the passive FTC with DSM has not the ability to recover the system then active FTC with DSM is preferred. The involvement of DSM in active FTC systems is important because the information of fault detection and isolation (FDI) system, in most cases, is not accurate. Hence, DSM with active FTC can improve the FTC system. In this work, DSM only in passive FTC systems is introduced, where there is no information about the fault.

IV. Predictive Controller with DSM

In general, most of the control algorithms used to recover the system performance usually stem from linear quadratic, adaptive or robust control. Thus, this section explains how DSM can be involved into predictive controllers to achieve safety requirements in addition to system performance.

The control law of predictive controller, for a system defined by the state-space model, is determined from the minimization of a 2-norm measure of predicted performance [5]

$$\min_{\underline{\mathbf{u}}} J = \min_{\underline{\mathbf{u}}} \left(\mathbf{e}^{\mathrm{T}}(N+k)\mathbf{S}_1\mathbf{e}(N+k) + \sum_{i=N_1}^{N-1} \mathbf{e}^{\mathrm{T}}(i+k)Q_1\mathbf{e}(i+k) + \sum_{i=0}^{N_u-1} \mathbf{u}^{\mathrm{T}}(i+k)R\mathbf{u}(i+k) \right) \quad (7)$$

subject to $\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\mathbf{u}(k)$

$$\mathbf{y}(k) = \mathbf{C}\,\mathbf{x}(k) + \mathbf{D}\,\mathbf{u}(k) \quad (8)$$

where $\underline{\mathbf{u}} = [\mathbf{u}(k)\,\mathbf{u}(k+1)\cdots\mathbf{u}(k+N_u-1)]^T \in \Re^{r.N_u}$;
$\mathbf{e}(k) = \mathbf{y}_d(k) - \mathbf{y}(k)$; $\mathbf{e}\in\Re^m$ is the error between the desired and measured response. $\mathbf{x}\in\Re^n$ is the system state vector, $\mathbf{y}_d\in\Re^m$ is the reference output vector, $\mathbf{y}\in\Re^m$ is the measured output vector, $\mathbf{u}\in\Re^r$ is the input vector, **A**, **B**, **C** and **D** system parameter matrix of adequate dimensions, $\mathbf{S}_1$, $\mathbf{Q}_1$ are the error weighting matrix, **R** is the input weighting matrix, N, N_1 and N_u are the maximum, minimum and control horizons, respectively.

The involvement of DSM into the predictive controller can be handled in two different forms, where both present advantages and disadvantages. Both methods consider that the system model is described by a state-space model and that the safe region is defined by linear boundaries.

A. Method 1

To control DSM (DSM positive) all elements of the distance vector ($\mathbf{d}\in\Re^q$) in (6) must be positive. Thus, the vector **d** can be introduced in the predictive control problem as additional Linear Matrix inequality (LMI) constraints. Hence, the control signal is obtained by minimizing equation (7) subjected to

$$\begin{aligned} &\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\mathbf{u}(k) \\ &\mathbf{y}(k) = \mathbf{C}\,\mathbf{x}(k) \\ &\mathbf{d}(k+i) \geq 0 \text{ or } \mathbf{x}(k+i) \in \Phi,\, i = N_1,\ldots, N \end{aligned} \quad (9)$$

Consider (5) and (8) then a general form of error prediction and distance vector **d** are

$$\underset{\rightarrow}{\mathbf{e}} = (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k) + \mathbf{C}_B\underset{\rightarrow}{\mathbf{u}}) \quad (10)$$

$$\underset{\rightarrow}{\mathbf{d}} = \mathbf{d}_t + \mathbf{D}_A\mathbf{x}(k) - \mathbf{D}_b\underset{\rightarrow}{\mathbf{u}} \geq 0 \quad (11)$$

where

$$\underset{\rightarrow}{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_d(k+N_1) \\ \mathbf{y}_d(k+N_1+1) \\ \vdots \\ \mathbf{y}_d(k+N) \end{bmatrix} \in \Re^{m.(N-N_1+1)};\quad \underset{\rightarrow}{\mathbf{e}} = \begin{bmatrix} \mathbf{e}(k+N_1) \\ \mathbf{e}(k+N+1) \\ \vdots \\ \mathbf{e}(k+N) \end{bmatrix} \in \Re^{m.(N-N_1+1)}$$

$$\underset{\rightarrow}{\mathbf{d}} = \begin{bmatrix} \mathbf{d}(k+1) \\ \mathbf{d}(k+2) \\ \vdots \\ \mathbf{d}(k+N) \end{bmatrix} \in \Re^{q(N-N_1+1)} \quad \mathbf{D}_A = \begin{bmatrix} D_a A^{N_1} \\ D_a A^{N_1+1} \\ \vdots \\ D_a A^{N} \end{bmatrix} \in \Re^{q(N-N_1+1)\times n};$$

$$\mathbf{d}_t = \begin{bmatrix} \mathbf{d}_c \\ \mathbf{d}_c \\ \vdots \\ \mathbf{d}_c \end{bmatrix} \in \Re^{q(N-N_1+1)};\quad \mathbf{C}_a = \begin{bmatrix} CA^{N1} \\ CA^{N_1+1} \\ \vdots \\ CA^{N} \end{bmatrix} \in \Re^{m.(N-N_1+1)\times n}$$

and $\mathbf{C}_B = \mathbf{C}_b + \mathbf{D}_u$;

$$\mathbf{C}_b = \begin{bmatrix} \mathbf{CA}^{N_1-1}\mathbf{B} & \cdots & \mathbf{CB} & 0 & 0 \\ \mathbf{CA}^{N_1}\mathbf{B} & \cdots & & \ddots & 0 \\ \vdots & & \ddots & & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \cdots & \cdots & \cdots & \mathbf{CA}^{N-N_u}\mathbf{B} \end{bmatrix} \in \Re^{m(N-N_1+1)\times r.N_u};$$

$$\mathbf{D}_b = \begin{bmatrix} \mathbf{D}_a\mathbf{A}^{N_1-1}\mathbf{B}\cdots\mathbf{D}_a\mathbf{B} & 0 & \cdots\cdots 0 \\ \mathbf{D}_a\mathbf{A}^{N_1}\mathbf{B} & \cdots\cdots\mathbf{D}_a\mathbf{B} & \cdots\cdots & 0 \\ \vdots & & & \\ \mathbf{D}_a\mathbf{A}^{N-1}\mathbf{B} & \cdots\cdots\cdots\cdots & \mathbf{D}_a\mathbf{A}^{N-N_u}\mathbf{B} & \end{bmatrix} \in \Re^{q(N-N_1+1)\times r.N_u};$$

$$\mathbf{D}_u = \begin{bmatrix} \overbrace{0 \cdots 0}^{r(N_1)} & \overbrace{\mathbf{D} \cdots 0}^{r(N_u-N_1)} \\ \vdots \ddots \vdots & \vdots \ddots \vdots \\ 0 \cdots 0 & 0 \cdots \mathbf{D} \\ 0 \cdots 0 & 0 \cdots 0 \\ \vdots \ddots \vdots & \vdots \ddots \vdots \\ 0 \cdots 0 & 0 \cdots 0 \end{bmatrix} \begin{matrix} \} m(N_u - N_1) \\ \\ \} m(N - N_u + 1) \end{matrix}$$

$\mathbf{d}_c$ and $\mathbf{D}_a$ are constant matrices calculated from (6).

The objective function according to (10) is

$$\min_{\underline{\mathbf{u}}} J = \min_{\underline{\mathbf{u}}} (\underline{\mathbf{u}}\,\mathbf{M}\underline{\mathbf{u}} + 2\mathbf{H}\,\underline{\mathbf{u}} + \mathbf{c}_r) \quad (12)$$

subject to (11), where

$$\mathbf{M} = \mathbf{C}_B{}^T\mathbf{Q}_t\mathbf{C}_B + \mathbf{R}_t;\mathbf{H} = (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k))^T\mathbf{Q}_t\mathbf{C}_B;$$

$$\mathbf{c}_r = (\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k))^T\mathbf{Q}_t(\underset{\rightarrow}{\mathbf{y}} - \mathbf{C}_a\mathbf{x}(k));$$

$$\mathbf{R}_t = \begin{bmatrix} \mathbf{R}_1 0\cdots 0 \\ 0\,\mathbf{R}_1\cdots \\ \vdots \quad \ddots \\ 0\,0\cdots\mathbf{R}_1 \end{bmatrix} \in \Re^{r.N_u\times r.N_u};\ \mathbf{Q}_t = \begin{bmatrix} \mathbf{Q}_1 0\cdots 0 \\ 0\,\mathbf{Q}_1\cdots \\ \vdots \quad \ddots \\ 0\,0\cdots\mathbf{S}_1 \end{bmatrix} \in \Re^{m.(N-N_1+1)\times m.(N-N_1+1)}$$

Equation (12) is known as a quadratic programming (QP) problem, for which standard solvers exist [8]-[12]. One line optimization of equation (12) gives the desired control sequence, which achieves the output performance and the safety performance. Note, **Q**, **S**, **R**, N, N_1 and N_u are free design parameters. DSM constraints are considered here as hard constraints.

The feasibility of the above MPC with DSM constrains can be analysed as in [7] and it has not been included in this work. Moreover, the safety region Φ can be considered as an invariant set [13] and can be analysed according to that. This method can give a good results but the computation burden is so high that it can only be applied, where the process time constants are slow.

B. Method 2

In order to present the second method, the following lemma is necessary:

Lemma 1: if the safe operating region, Φ, is convex, then the condition to minimize any-norm of **d**(·) subject to system

model is to move the states to be inside the safety region Φ.

The proof of lemma 1 is simple; it can be easily proofed according to the convex set properties

Hence, the 2-norm of $\mathbf{d}(\cdot)$ can be introduced as additional term in the main objective function (7). The objective function of the predictive controller in this case can be rewritten in the following form

$$\begin{aligned}\min_{\underline{\mathbf{u}}} J = \min_{\underline{\mathbf{u}}} \Big[& \mathbf{e}_d{}^{\mathrm{T}}(N+k)\bar{\mathbf{S}}\mathbf{e}_d(N+k) \\ & + \sum_{i=N_1}^{N} \mathbf{e}_d{}^{\mathrm{T}}(i+k)^{\mathrm{T}}\,\bar{\mathbf{Q}}\,\mathbf{e}_d(i+k) \\ & + \sum_{i=0}^{N_u-1} \mathbf{u}(i+k)^{\mathrm{T}}\,\mathbf{R}\mathbf{u}(i+k)\Big]\end{aligned} \tag{13}$$

subject to (7) and (4), where

$$\mathbf{e}_d(.)=\begin{bmatrix}\mathbf{e}(.)\\ \mathbf{d}(.)\end{bmatrix}\in\Re^{m+q},\ \bar{\mathbf{Q}}_1=\begin{bmatrix}\mathbf{Q}_1 & 0\\ 0 & \mathbf{P}_1\end{bmatrix},\ \bar{\mathbf{S}}_1=\begin{bmatrix}\mathbf{S}_1 & 0\\ 0 & \mathbf{P}_1\end{bmatrix}\ \text{and}\ \mathbf{P}_1=\mathbf{P}_o\,e^{-\delta(k)}.$$

This is an unconstrained quadratic problem, where $\mathbf{P}_1$ is the weighting matrix for $\mathbf{d}$ and it depends on the value of DSM (δ) i.e. if δ is negative then the weighting matrix increased and vice-versa. $\mathbf{P}$ is a constant weighting matrix**.** The number of free design parameters in this case increased by $\mathbf{P}_o$. DSM constraints, here, is considered as a soft constraints. This optimization problem can be solved in two different ways

1) One-shot Optimization

Solving problem in the form of (13) using direct optimization can be found in [5],[14]. Substituting (10) and (11) in (12)

$$\min_{\underline{\mathbf{u}}} J = \min_{\underline{\mathbf{u}}} \left(\begin{bmatrix}\underline{\mathbf{e}}\\ \underline{\mathbf{d}}\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{Q}_t & 0\\ 0 & \mathbf{P}_t\end{bmatrix} \begin{bmatrix}\underline{\mathbf{e}}\\ \underline{\mathbf{d}}\end{bmatrix} \right) + \underline{\mathbf{u}}^{\mathrm{T}}\mathbf{R}_t\,\underline{\mathbf{u}}$$

is obtained, where

$$\mathbf{P}_t \in \Re^{q.N\times q.N} = \begin{bmatrix}\mathbf{P}_1 & 0 & \cdots & 0\\ 0 & \ddots & & \vdots\\ \vdots & & \ddots & \vdots\\ 0 & \cdots & \cdots & \mathbf{P}_1\end{bmatrix}.$$

The control law is given by

$$\underline{\mathbf{u}} = \left[\mathbf{K}_y\,\underline{\mathbf{y}} + \mathbf{K}_d\,\mathbf{d}_c - \mathbf{K}_x\,\mathbf{x}(k) \right] \tag{14}$$

where

$$\mathbf{K}_y = \left[\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{D}_b \right]^{-1}\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t$$

$$\mathbf{K}_d = \left[\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{D}_b \right]^{-1}\mathbf{D}_b{}^{\mathrm{T}}\mathbf{P}_t$$

$$\mathbf{K}_x = \left[\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_B + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{D}_b \right]^{-1}\left[\mathbf{C}_B{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{C}_a + \mathbf{D}_b{}^{\mathrm{T}}\mathbf{Q}_t\mathbf{D}_A \right]$$

The first component in $\underline{\mathbf{u}}$, namely $\mathbf{u}(k)$, is the control vector applied to the system. This control vector can be obtained from (14) as

$$\mathbf{u}(k) = \left[\mathbf{I}_r : 0 : \cdots : 0 \right] \left[\mathbf{K}_y\,\underline{\mathbf{y}} + \mathbf{K}_d\,\mathbf{d}_c - \mathbf{K}_x\,\mathbf{x}(k) \right] \tag{15}$$

Despite simplicity of the direct optimization algorithm, it needs much memory space because the matrices usually have large dimensions. Moreover, the problem could be numerical unstable when the horizons are very large.

2) Dynamic programming

The solution of the control problem by applying dynamic programming is given by the affine control law [6] but without integral action

$$\mathbf{u}(k) = \mathbf{K}\,(N-1)\ \left[\mathbf{u}_w(k) - \left[\mathbf{M} + \mathbf{B}^{\mathrm{T}}\mathbf{P}(N-1)\mathbf{A} \right]\mathbf{x}(k \right] \tag{16}$$

where $\mathbf{u}_w$ represents the control vector due to the reference ($\mathbf{y}_r$) and $\mathbf{u}_x$ the control action based on the state feedback

$$\mathbf{K}\,(N-1) = \left[\bar{\mathbf{R}} + \mathbf{B}^{\mathrm{T}}\ \mathbf{P}\,(N-1)\ \mathbf{B} \right]^{-1},$$

where $\bar{\mathbf{R}} = \mathbf{R}_j + \mathbf{D}^{\mathrm{T}}\bar{\mathbf{Q}}_1\mathbf{D}$. $\mathbf{P}(N\text{-}1)$ is calculated by solving the Riccati Difference Equation (RDE)

$$\mathbf{P}(j+1) = \mathbf{Q} + \mathbf{A}^{\mathrm{T}}\mathbf{P}(j)\mathbf{A} - \left[\mathbf{M} + \mathbf{B}^{\mathrm{T}}\mathbf{P}(j)\mathbf{A} \right]^{\mathrm{T}}\ \mathbf{K}_1(j)$$

for j = 1,…N-2, and a specific $\mathbf{P}(0) = \bar{\mathbf{C}}^{\mathrm{T}}\bar{\mathbf{S}}_1\bar{\mathbf{C}}$. $\mathbf{K}_1$ is calculated from

$$\mathbf{K}_1(j) = \mathbf{K}(j)\ \ [\mathbf{M} + \mathbf{B}^{\mathrm{T}}\mathbf{P}(j)\mathbf{A}]\,,$$

where $\mathbf{M} = \bar{\mathbf{D}}^{\mathrm{T}}\bar{\mathbf{Q}}_{1j}\bar{\mathbf{C}}$ and $\mathbf{Q} = \bar{\mathbf{C}}^{\mathrm{T}}\bar{\mathbf{Q}}_{1j}\bar{\mathbf{C}}$. The matrices $\mathbf{C}$ and $\mathbf{D}$ are obtained from

$$\bar{\mathbf{C}} = \begin{bmatrix}\mathbf{C}\\ \mathbf{D}_a\end{bmatrix} \in \Re^{(m+q)\times n}\ \text{and}\ \bar{\mathbf{D}} = \begin{bmatrix}\mathbf{D}\\ 0\end{bmatrix} \in \Re^{m+q}.$$

The matrices $\bar{\mathbf{Q}}_{1j}$ and $\mathbf{R}_j$ are defined as

$$\bar{\mathbf{Q}}_{1j} = \begin{cases}\bar{\mathbf{Q}}_1\ \forall 1\le j\le N-N1\\ 0\ \ \forall N-N_1\le j\le N-1\end{cases}\quad \mathbf{R}_j = \begin{cases}\infty\mathbf{I}\ \forall 1\le j\le N-N_n\\ \mathbf{R}\ \ \forall N-N_u+1\le j\le N-1\end{cases}$$

The control vector $\mathbf{u}_w(k)$ is given by

$$\mathbf{u}_w(k) = \mathbf{D}_d{}^{\mathrm{T}}\bar{\mathbf{Q}}_{1j}\mathbf{w}(k) + \mathbf{B}^{\mathrm{T}}\mathbf{p}(N-1)$$

where $w(k) = \begin{bmatrix}\mathbf{y}_r(k)\\ \mathbf{d}_c\end{bmatrix} \in \Re^{m+q}$; $\mathbf{p}(N\text{-}1)$ is obtained from

$$\begin{aligned}\boldsymbol{p}(j+1) = & \left[\bar{\boldsymbol{C}} - \bar{\boldsymbol{D}}\,\boldsymbol{K}_1(j) \right]^{\mathrm{T}}\bar{\boldsymbol{Q}}_{1j}\left[\boldsymbol{w}(k+N-(j+1)) \right] \\ & + \left[\boldsymbol{A} - \boldsymbol{B}\,\boldsymbol{K}_1(j) \right]^{\mathrm{T}}\ \boldsymbol{p}(j)\end{aligned}$$

and $\mathbf{p}(0) = \bar{\mathbf{C}}^{\mathrm{T}}\bar{\mathbf{S}_1}\left[\mathbf{w}(k+N) \right]$.

The advantage of using dynamic programming optimization instead of direct optimization is that the matrices dimensions are smaller. However, the number of calculation steps is increased.

Notice that both methods to implement DSM in predictive controller are general for MIMO systems. The following example shows the advantages of each method.

Example

State-space model parameters of a separately exited dc motor, described in [1], are

A	B	C	D
$\begin{bmatrix}-1 & 1\\ -1 & -1\end{bmatrix}$	$\begin{bmatrix}0\\ 1\end{bmatrix}$	$\begin{bmatrix}1 & 0\end{bmatrix}$	[0]

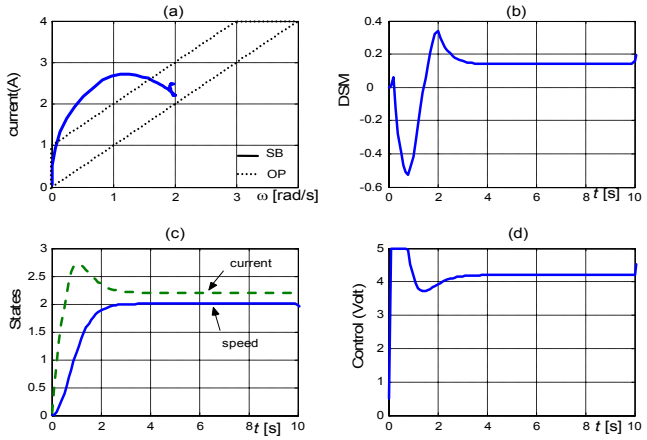

Fig. 3. DC motor response with PID
(SB: Boundary and OP: Trajectory)

where $\mathbf{x} = [\omega \quad i]^{\mathrm{T}}$ and $u = v_i$.

ω is the motor speed (angular velocity), i armature current, v_i input voltage $\in[0,5]$. The safe operation region Φ is defined by

$$i-(w+0.5) < 0;\ i-w > 0;\ i<4 \text{ and } w<4. \qquad (17)$$

Fig. 3 shows the motor state trajectory (Fig. 3a), DSM variation (Fig. 3b) and speed response (Fig. 3c) to step input 2 rad/sec using PID controller with $K_p = 4$, $K_i = 1$ and $K_d = 1$. The response of the controller is accepted w.r.t. the error and rise-time but the state trajectory lies outside the safe boundaries at transient (Fig. 3a) i.e. DSM is negative. To improve DSM at transient time, the controller should be redesigned according to DSM. Some method of solving this problem is stated in [1] and [3] using switched PID, adaptive PID and optimal state-feedback controllers based on DSM.

Fig. 4 shows the motor response using a predictive controller in the form of (9) subject to (11) (method 1) with the following parameters

$$\mathbf{A}_c = \begin{bmatrix} -1 & 1 & 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 1 & 0 & -1 \end{bmatrix}^{\mathrm{T}};\ \mathbf{C}_c = [0.2 \quad 0 \quad 4 \quad 4 \quad 0 \quad 0]^{\mathrm{T}};$$

$$\mathbf{D}_{ia} = diag(0.702, 0.702, 1, 1, 1, 1, 1);$$

$$N = 3;\ N_1 = 1;\ N_u = 3;\ \mathbf{Q}_1 = \mathbf{S}_1 = [10];\ \mathbf{R} = [0.001]$$

Fig. 5 shows the best motor response can be obtained, w.r.t safety and transient performance, using a predictive controller in the form of (15) (method 2) with the following parameters

$$P_o = \mathrm{Diag}(10,10,0,0,0,0);\ \mathbf{Q}_1 = \mathbf{S}_1 = [30];\ \mathbf{R} = [0.001];$$

$$N=10;\ N_1=1;\ N_u=5.$$

Note that the rise time in Fig. 4 is similar to Fig. 3 but the state trajectory (Fig. 4a) is forced to be inside the safe region during transient period. Response in Fig. 5a satisfies the safety bounds but the rise-time increased (Fig. 5c).

The performance of *method* 1 is more accepted but the computation algorithm is difficult. On the other hand, algorithms of *method* 2 are quite easy and provide smoother control signals but the overall performance of the system is for particular example worse and the free design parameters

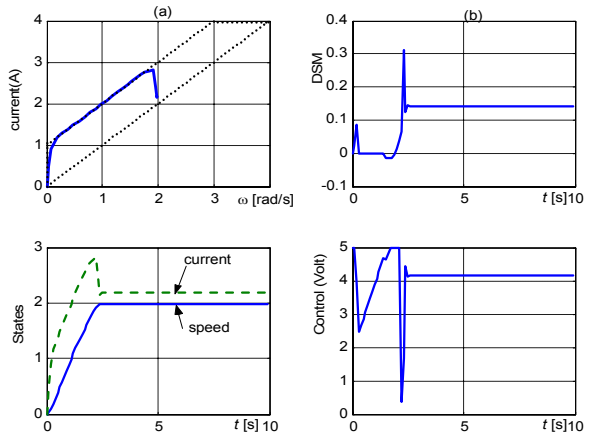

Fig. 4. DC Motor response with predictive controller *method* 1

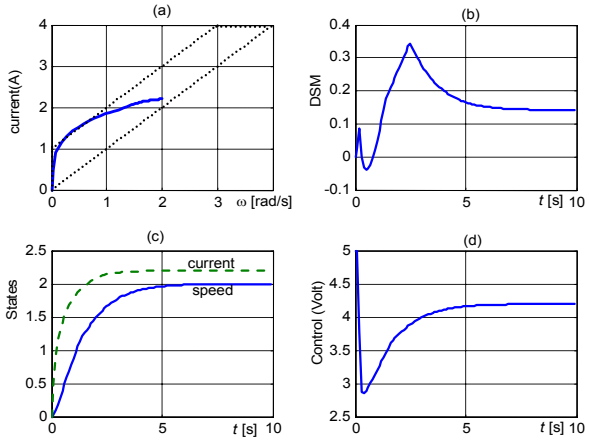

Fig. 5. DC Motor response with predictive controller *method* 2

increased.

V. EXPERIMENTAL RESULTS

The above algorithms are tested in real-time operation of an experimental laboratory process described in [15]. The process, shown in Fig. 6, consists of two-tank system. Each tank has a control valve at the output line to control the level in the tank. In the current experiment, the interconnecting valve was fully opened, the leakage valve (control valve of 2nd tank) was adjusted to simulate a constant leakage and the control valve, of the first tank, was used to adjust the level in both tanks. The two-tank system was fed at constant flow 1 l/s in the first tank. The discrete linear model of the system at sampling rate equal to 10 Hz is given in Table 1.

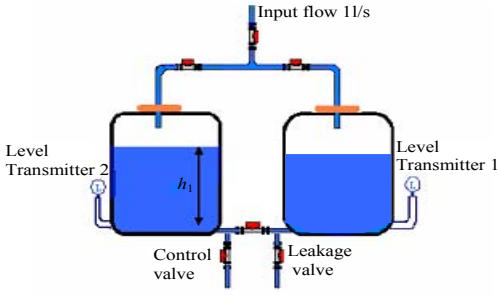

Fig. 6. Schematic diagram of a two-tank system

Table 1: Linear state-space model of the two-tank-system

A

$$\begin{bmatrix} 0.9748 & 0.0019 & -0.0146 \\ -0.1616 & -0.2104 & 0.5555 \\ -2.4323 & -1.1408 & 0.2307 \end{bmatrix}$$

B

$$\begin{bmatrix} -0.0004 \\ -0.0105 \\ -0.0173 \end{bmatrix}$$

C

$$[1 \quad 0 \quad 0]$$

D

$$[0]$$

The output h is the level in the tank (m) and the input is the valve opening.

Consider that the variables, which are relevant to system safety, are the tank level rate (dh/dt) and v_i, the valve limb movement (m) which simulates the valve opening. The safe operation region (Φ) is given by:

$$dh/dt + 0.8\, v_i - 0.08 < 0;\ dh/dt + 0.75\, v_i + 0.14 > 0;$$
$$-0.4 < dh/dt < 0.4;\ -0.5 < v_i < 0.5. \qquad (18)$$

where the valve opening is normalized within [-0.5, 0.5] i.e. 0.5 means fully opened and -0.5 completely closed. The level rate (dh/dt) changes *in (mm/sec)*.

A predictive controllers with and without DSM are used, according to Section 3, to regulate the level of the left tank at a set point of 0.3 *m* in case of leakage fault. Fig. 7 shows real-time results without considering DSM in predictive controller for the actual two-tank system when the leakage valve is opened 10% after 500 sec, 30 after 650 sec, and 50% after 800 sec (fault scenario). Fig. 8 shows the real-time results of the above algorithm with considering DSM in predictive controller for the same faults. It is clear from Fig. 8 that in case of fault the controller has the ability to operate the system within the safety limit until the fault be repaired or isolated.

VI. CONCLUSIONS

Controller design based on DSM improves safety-assessment of safety-critical systems particularly by using MPC. Results of a simulation example as well as of a real-time implementation on a two-tank process demonstrate the advantage of this approach mainly in FTC. However, the feasibility of MPC with DSM constraint was not treated in this work. There are some open areas in applying this

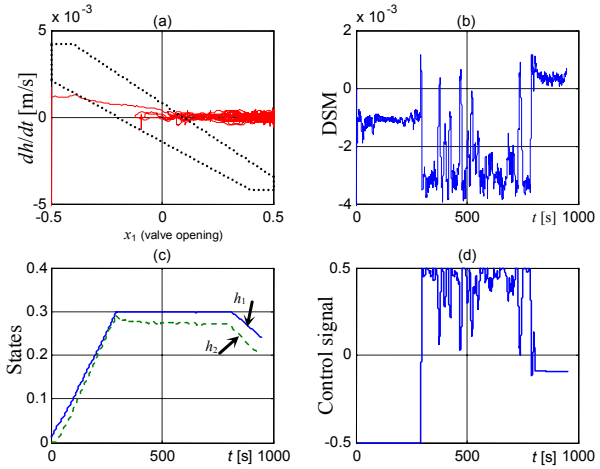

Fig. 7: Level response using predictive controller without DSM

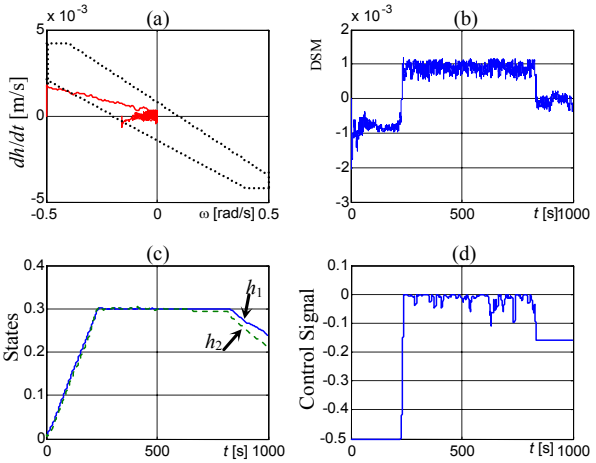

Fig. 8: Level response using predictive controller with DSM

approach, which should be covered in the future: For example, DSM measuring for large-scale system and the problem of determining safety boundaries. Hence, all these topics will be undertaken in the future work as well as the problem of applying DSM to fault prognosis.

REFERENCES

[1] E. Badreddin and M. Abdel-Geliel. "Dynamic Safety Margin Principle and Application in Control of Safety Critical System," *IEEE International conference of control application (CCA 2004) conference,* September 2-4, 2004, Taiwan. 689-695.

[2] M. Abdel-Geliel and E. Badreddin, "Dynamic Safety Margin in Fault Diagnosis and Isolation," *European Safety and Reliability* (*ESREL*) *conf.,* Tri city Poland, June 27-30, 2005.

[3] M. Blanke, M. Staroswieki and N. Eva Wu, "Concept and Methods in Fault-Tolerant Control," *Tutorial at American Control Conference*, June 2000.

[4] M. Mahmoud, J. Jiang, and Y. Zhang, "*Active Fault Tolerant Control System: Lecture Notes in Control and information Sciences,*" Springer, 2003.

[5] J.A Rossiter, "*Model-Based Predictive Control: Practical Approach,*" CRC, 2003.

[6] A. Gambier and H. Unbehauen, "Multivariable Generalized State-Space Receding Horizon Control in a Real-time Environment," *Automatica*, **35**, 1787-1797, 1999.

[7] D. Q. Mayne, J. B. Rawlings, C. V. Rao and Po. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica,* **36**, 789-814, 2000.

[8] E.F. Camacho and C. Bordons, "*Model Predictive Control,"* Springer, 1999.

[9] T.F. Coleman, and Y. Li, "A Reflective Newton Method For Minimizing a Quadratic Function Subject to Bounds on Some of the Variables," *SIAM Journal on Optimization*, Vol. 6, N° 4, 1040-1058, 1996.

[10] J.M. Maciejowski, "*Predictive Control with Constrained,*" Prentice Hall, 2001

[11] F. Borrelli, *"Constrained Optimal Control of Linear and Hybrid systems: Lecture Notes in Information Science,"* Springer, 2003.

[12] Thomas C., M.A. Branch and A. Grace, "*Optimization Toolbox,*" Math Work, Inc., 1999.

[13] F. Blanchini. "Set Invariance in Control," *Automatic*, 35:1747-1767, 1999.

[14] A. Gambier. "*State-space Design of Predictive Control for MIMO Systems,*" PhD thesis, Bochum University, Germany, 1995.

[15] A. Gambier, T. Miksch and E. Badreddin, "A control Laboratory Plant to Experiment with Hybrid System," *Proc. of American Control Conference*, Denver, 2003.

[16] M. Abdel-Geliel and E. Badreddin, "Adaptive controller using dynamic safety margin for hybrid laboratory plant," *Proc. of American Control Conference 2005*, Portland, Oregon, USA, 1443-1448.