Abstract — An independent stability assurance layer for the multivariable controller in the hierarchical control architecture is introduced in this paper. This stability layer provides an integrated safety and protection measure to process control systems from closed-loop instabilities, which may cause process upsets and shutdowns. It prevents disruptive shutdowns by providing corrective actions to the controller outputs. The Supervisory Stability Layer is established based on the renowned dissipative systems theory. The implementation of this approach is illustrated by an example of stability guarantee of model predictive controllers.

I. INTRODUCTION

The IEC-61508/61511 safety standard for process systems and guideline to engineer safety instrumented systems (SIS) for the process industry sector recommends the segregation between the process control system (such as the distributed control system – DCS) and the SIS (or safety shutdown/protection systems). The SIS will automatically initiate an immediate safety shutdown action to put the system into a safe state when one or more process variables have been steered into an unsafe region or out of the allowable limits. The physical segregation with the DCS ensures that the SIS achieves the assigned quantitative reliability and safety integrity level.

The DCS design and implementation should have been such that to minimize the escalation of unsafe process upsets into a plant-wide shutdown that may disrupt the operation. Inexperienced operators, human errors, disturbances, and controllers whose performance degrades over time without regular adjustments, may, however, cause unexpected system instability and/or constraint violations. An automatic tool that prevents these problems to occur will improve safety, efficiency and business continuity.

An independent stability assurance tool that continuously monitors the stability of control systems, as well as limits the trajectory within the stabilizing and safe boundaries will be ideal for this purpose. To the authors’ knowledge, such a tool has not been reported in the literature of plant-wide controllers within the hierarchical architecture of plant-wide control, as shown in Figure 2.1. It provides stabilized outputs for a dedicated multivariable controller to assure the closed-loop system stability in a fully segregated manner. Here, a single multivariable controller is discussed.

Based on the dissipative systems theory, a novel real-time stability condition for closed-loop systems is firstly introduced. Here, the stability condition for input affine systems, either linear or nonlinear, is derived on-the-fly from the dissipativity index which is defined as the accumulative supply rate of the controller. The bounds on immediate-future values of the manipulated variables are iteratively computed from the above stability condition. It is then used as a measure to monitor in real-time the system stability, and at the same token, enforces the stabilized limits on the manipulated inputs, whenever the stability threshold is violated.

This paper is organized as follows. The concept of dissipativity and quadratic supply function are introduced in brief in Section 3. The definition of the trajectory-based dissipativity index and the stability condition are given in Section 4. The implementation of this stability condition as a supervisory stability layer is given in Section 5 followed by the simulations results of a case study presented in Section 6.

II. MOTIVATION

Motivated by the theory of dissipativity for stability analysis, that is applicable for both linear and nonlinear systems [1], we propose a stability condition that can be verified online for real-time control of state space plants. Within this setup, the past behavior of system inputs, outputs and the correlations between them are used for calculating the stabilizing bounds, to guarantee the closed-loop stability. It is designated as Supervisory Stability Condition (SSC) due to the real-time aspect of the condition.

Figure 2.1 – A New Stability Assurance Layer for the Multilayer Plant-wide Control Architecture.
The approach of having an additional stability condition for the existing multivariable control setups is, in fact, practical because the hierarchical architecture of supervisory controls is currently the most widely used implementation structure [2]. Such multilayer configurations are also more fault-tolerant [3].

III. BRIEF ON DISSIPATIVITY

A. Dissipativity and Supply Rate

Consider a state space nonlinear system $\Sigma$ of

$$\begin{align*}
\dot{x}(t) &= f(x,u), \quad x \in \mathcal{X}, u \in \mathcal{U} \\
y(t) &= h(x,u), \quad y \in \mathcal{Y}
\end{align*}$$

(3.1)

where $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is a smooth mapping of its arguments, with $\mathcal{X}$ an open set on $\mathbb{R}^n$, $\mathcal{U}$ an open set on $\mathbb{R}^m$; and function $h : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$, with $\mathcal{Y}$ an open set on $\mathbb{R}^p$.

Definition 2.1 – (Willems, 1972) [4] A $C^1$ function called supply function or supply rate, denoted by $w(u,y)$ with $w : \mathcal{Y} \times \mathcal{U} \to \mathbb{R}$, satisfying

$$\int_0^T w(u(t),y(t))dt < \infty, \quad \forall T > 0.$$  

(3.2)

System $\Sigma$ (3.1) with a properly chosen output is said to be dissipative with respect to the supply rate $w(u,y)$ if there exists a positive definite, $C^1$ function, addressed as the storage function $V : \mathcal{X} \to \mathbb{R}$, $V(0) = 0$, for any $T > 0$, the following inequality is satisfied irrespectively of the initial value of the state $x(0)$:

$$V(x(T)) - V(x(0)) \leq \int_0^T w(u(t),y(t))dt, \quad \forall x \in \mathcal{X} \times \mathcal{U}.  
$$

(3.3)

If it is a strict inequality, the system is called strictly dissipative. The dissipation inequality is equivalent to the infinitesimal form with $C^1$ function,

$$\dot{V}(x) \leq w(u,y), \quad \forall x \in \mathcal{X} \times \mathcal{U}.  
$$

(3.4)

And in discrete-time domain,

$$V(x(k+1)) - V(x(k)) \leq w(u(k),y(k)), 
\forall x, u \in \mathcal{X} \times \mathcal{U}, \forall k, k \text{ is integer time index}.  
$$

(3.5)

With respect to the behavioral framework, the dissipative systems are defined in association with all trajectories of a supply rate inside a behavior of the universe, together with its coupled storage treated as a latent variable.

Definition 2.2 – (Willems, 2007) [5] Let $\Xi = (T, \mathcal{W}, \mathcal{B})$ be a dynamical system, with $T \subseteq \mathbb{R}$ the time set, $\mathcal{W} \subseteq \mathbb{R}$ the signal space, and $\mathcal{B} \subseteq \mathcal{W}$ the behavior. A trajectory $w : \mathcal{W} \to \mathbb{R}, w \in \mathcal{B}$, is the rate of supply absorbed by the system $\Xi$ is said to be dissipative if and only if $\forall w \in \mathcal{B}$ and $\forall t_0 \in T$, $\exists K \in \mathbb{R}$ such that

$$-\int_{t_0}^T w(t)dt \leq K \text{ for } T \geq t_0.  
$$

(3.6)

The storage is viewed as a latent variable $\lambda$ that is coupled to the supply rate $w$. This leads to a latent variable system $\Xi_L = (T, \mathcal{W}, \mathcal{B}_{\text{full}})$ such that $(w, \lambda)$ belongs to the full behavior $\mathcal{B}_{\text{full}}$ if the pair $\lambda : \mathbb{L} \to \mathbb{R}$ and $w : \mathcal{W} \to \mathbb{R}$ is a possible history of the way the supply flows in and out of and is stored in the system.

B. Dissipative System with Quadratic Supply Rate

Albeit the general definition of a supply rate only requires a $C^1$ function, the quadratic supply rates are often employed in the stability analysis and control designs as they are related to both gain and phase information of systems.

Definition 2.3 – (Hill and Moylan, 1976) [6] Dissipative system $\Sigma$ (3.1) having the supply rate of the following quadratic form:

$$w(u,y) = y^T Q y + 2 y^T S u + u^T R u.  
$$

(3.7)

where $Q, S, R$ are appropriately dimensioned matrices, $Q$ and $R$ are symmetric, is called $(Q, S, R)$-dissipative system.

IV. DISSIPATIVITY INDEX AND TRAJECTORY BASED STABILITY CONDITION

From the nonlinear control literature [7], the stability of two interconnected dissipative systems in a feedback configuration is established if their supply rates satisfy certain conditions. These conditions, however, cannot be applied directly to real-time controllers whose closed-form models for all possible mappings between plant outputs and control inputs do not exist.

Proposition 4.1 below provides a new stability condition called dissipative-trajectory stability condition. It is conceptually different to the dissipation condition determined from the mapping of the entire input output spaces of the controller. Consider a system $\Sigma$ (3.1) in discrete time domain. Firstly, the notion of dissipative trajectory is given in the following:

Definition 4.1 – Consider a real-time controller with input and output trajectories denoted as $y(k)$ and $u(k)$ respectively ($k$ is integer time index). The dissipation index $W_c$ is defined as follows:

$$W_c(k) = \sum_{i=0}^{k} w_c(i),  
$$

(4.1)

$$w_c(i) = y^T(i) R_c y(i) + 2 y^T(i) S_c u(i) + u^T(i) Q_c u(i).  
$$

This dissimilarity index can be calculated recursively on-line based on the past output, input and its own past values, as per formulæ below:

$$W_c(k) = W_c(k-1) + w_c(k).  
$$

(4.2)

Definition 4.2 – A real-time controller is said to have a $(Q_c, S_c, R_c)$-dissipative trajectory, within a finite time domain, if the corresponding dissimilarity index is positive in the domain, i.e.
Proposition 4.1 [8] – Suppose that a discrete-time system \( \Sigma \) is \((Q,S,R)\)-dissipative. The closed-loop system will be stable, if the controller draws a \((Q_c,S_c,R_c)\)-dissipative trajectory initiating from time zero, and the following matrix is negative definite:

\[
D_M = \begin{bmatrix} Q + R_c & S + S_c \\ \mathbf{R}^T + S_c^T & R + Q_c \end{bmatrix} < 0,
\]

with \((Q_c,S_c,R_c)\) and \((Q,S,R)\) are the matrices defined in (4.1) and (3.7) respectively. This condition is defined as dissipative-trajectory stability condition for the negative feedback system with real-time controllers.

The strict \((Q_c,S_c,R_c)\)-dissipative trajectory (i.e. the controller dissipativity index is always positive) used with the stability condition (4.4) is, however, difficult to be maintained online. It will become a dead lock for the algorithm if it is arbitrarily negative at a time instant, which is an inevitably foreseeable circumstance. The dissipative trajectory can, fortunately, be relaxed by employing its real-time version illustrated below.

Proposition 4.2 – Consider a \((Q,S,R)\)-dissipative discrete-time system \( \Sigma \) with a real-time controller. Assume \( Q_c,S_c,R_c \) satisfy the condition given in (4.4) but the controller does not have a \((Q_c,S_c,R_c)\)-dissipative trajectory as \( W_c(k) < 0 \) at certain step \( k \). Once \( W_c(k) \) is negative, the closed-loop system is stable if the following inequalities for the instant dissipativity index of the controller are true at all time:

\[
0 > w_c(k+1) + w_c(k) > 0 \quad \forall \ k.
\]

With \( w_c(\cdot) \) is defined in (4.1).

Proof: The \((Q,S,R)\)-dissipativity of \( \Sigma \) implies

\[
V(k+2) - V(k+1) < w(k+1).
\]

Due to \( D_M < 0 \) in (4.4), we have \( w(k) + w_c(k) < 0 \) \( \forall y(k), u(k) \)

(i) If \( w_c(k) > 0 \) \( \forall k \) while \( W_c(k) < 0 \) we have:

\[
(A.1) \implies V(k+2) - V(k+1) < 0 \quad \forall k.
\]

(ii) If \( w_c(k+1) > w_c(k) \), \( w_c(k) > w_c(k+1) > W_c(k) - W_c(k-1) \) while \( W_c(k) < 0 \), we have:

\[
\begin{align*}
& V(k+2) - V(k+1) + [W_c(k) - W_c(k-1)] < 0 \\
& \implies \left[ V(k+2) + [W_c(k-1)] - [V(k+1) + [W_c(k)]] \right] < 0. \quad (A.3)
\end{align*}
\]

(A.3) implies that the sum of two positive functions monotonously decreases over time. From the results of (i) and (ii) in (A.4) and (A.2) we have \( V(k) \to 0 \) as \( k \to \infty \), hence if \( V(k) \) is chosen as \( x^T(k)Px(k) \), with \( P \) is positive definite, then \( x(k) \to 0 \) as \( k \to \infty \).

V. DISSIPATIVE-TRAJECTORY STABILITY CONDITION AS AN INDEPENDENT STABILITY ASSURANCE LAYER

The real-time stability conditions presented in Proposition 4.1 and 4.2 form the foundation for the proposed stability assurance scheme. The main idea is not to integrate this condition into the control design problem offline, but rather use it in a supervisory manner. As such the stability assurance layer can, in principle, be applied to different multivariable control schemes.

A. Supervisory Stability Layer

Corollary 5.1 – Consider a \((Q,S,R)\)-dissipative discrete-time system with a real-time controller. Assume \( Q_c,S_c,R_c \) satisfy the condition given in (4.4) but the controller does not have a \((Q_c,S_c,R_c)\)-dissipative trajectory as \( W_c(k) < 0 \) at certain step \( k \) and \( w_c(k) \) is also negative. Define an intermediary supply rate \( \bar{w}_c(\cdot) \) as follows:

\[
\bar{w}_c(i) = y^T(k)S_cy(k) + 2y^T(k)S_c\bar{u}(k) + \bar{u}^T(k)Q_c\bar{u}(k).
\]

The output vector \( y(k) \) in (5.1) is known. The closed-loop system will then be stable, if there exists a signal vector \( \bar{u}(k) \) so that the actual manipulated input vector \( u(k) \) belongs to the following set:

\[
\Omega(k) = \{ u(k): 0 > w_c(k+1) \geq \bar{w}_c(k) > w_c(k), \quad (5.2)
\]

or \( w_c(k+1) \geq \bar{w}_c(k) > 0 \).

Proof: The above condition is an immediate result of Proposition 4.2.

This set will establish a new boundary for the control input at the iterative step \( k \). At every iteration, a signal for \( \bar{u}(k) \) is computed based on the inequalities (5.2) above. The boundary for the control input is obtained from the resultant \( \bar{u}(k) \). This is the core of the Supervisory Stability Layer (SSL). It produces the stability boundary as an independent second value. The controller output will be limited within this boundary in case the real-time control algorithm produces out-of-limit values.

A switch is installed between such the controller output and the SSL output to provide a stabilized manipulated input to the plant. The implementation for this SSL will, nevertheless, face some difficulties as described following.

Due to the fact that, the dissipative-trajectory stability condition is a sufficient condition only, the SSL may be conservative. A less conservative result can be obtained if the \((Q_c,S_c,R_c)\) and \((Q,S,R)\) matrices are re-computed and revised at every time step as per Proposition 5.1 below. It is recommended that the SSL shall not override the manipulated inputs unless the LMIS are infeasible.

Proposition 5.1 – Consider a discrete time LTI system (5.1) without direct feed-through (\( D = 0 \)). The closed-loop system is stable, if the matrices \( Q_c(k),S_c(k),R_c(k) \) and \( Q(k),S(k),R(k) \) and \( P \) at the iterative time step \( k \) satisfy the following three LMIs for \( k = 0 \ldots \infty \) (the bold typefaces are decision variables):

\[
y_k^TQ_ky_k + 2y_k^TS_ku_k + u_k^TR_ku_k \geq 0,
\]

4411
Using the measured data of input and output in parallel with Proposition 5.1, which is currently under investigation, especially when the number of time steps required in the stability boundary. Alternatively, an additional condition on performance or sensitivity can be used in parallel with Proposition 5.1, which is currently under investigation.

When the LMIs (5.4)-(5.6) are infeasible, the least conservative matrices of \( Q_c, S_c, R_c \) can be found by solving the following optimization problem:

\[
\begin{bmatrix}
Q(k) + R_c(k) & S^T(k) + S_c(k) \\
S(k) + S_c(k) & R(k) + Q_c(k)
\end{bmatrix} < 0, \tag{5.5}
\]

\[
\begin{bmatrix}
A^T P(k) A - P(k) - C^T Q(k) C & A^T P(k) B - C^T S(k) \\
B^T P(k) A - S(k) C & B^T P(k) B - R(k)
\end{bmatrix} < 0. \tag{5.6}
\]

Where, \( w_k = [u(0)^T \cdots u(k)^T]^T \); \( y_k = [y(0)^T \cdots y(k)^T]^T \); \( Q_k = \text{diag}(Q(k) - Q(k)_{k+1}) \); \( S_c(k) = \text{diag}(S(k) - S(k)_{k+1}) \); \( R_k(k) = \text{diag}(R(k) \cdots R(k)_{k+1}) \).

With \( u_k, y_k \) and matrices \( A, B, C \) are known at time step \( k \).

**Proof:** Inequality (5.6) implies system (5.1) is \( (Q, S) \)-dissipative [8]. Inequality (5.4) is a vector form of the controller \( (Q_c, S_c, R_c) \)-dissipative trajectory, as defined in (4.1):\n
\[
W_k^x(k) = \sum_{i=0}^{k} w_i^x(i) > 0;
\]

\[
w_k^y(j) = y^T(j)R_c(k)y(j) + 2y^T(j)S_c(k)u(j) + u^T(j)Q_c(k)u(j)
\]

Therefore, \( W_k^x(k) - W_k^y(k) = w_k^x(k) \).

The \((Q, S, R)\)-dissipativity of (5.1) implies\n
\[
V(k + 1) - V(k) < w_k^x(k) \Rightarrow [V(k + 1) - V(k)] + [W_k^x(k) - W_k^y(k) \mid k - 1) \]

\[
< y^T(k)\{Q + R \mid y(k) + 2y^T[k + S_c^T(k)]u(k) + u^T(k)\}R + Q_c]u(k)
\]

The right hand side is negative due to (5.5), thus\n
\[
[V(k + 1) + W_k^x(k)] - [V(k) + W_k^y(k)] < 0, \forall k. \tag{5.7}
\]

The sum of these two positive functions monotonically decreases over time, consequently, each of the positive function, specifically, \( V(k) \rightarrow 0 \) as \( k \rightarrow \infty \), hence if \( V(k) \) is chosen as \( x^T(k)P \mid k \), with \( P \) is positive definite, then \( x(k) \rightarrow 0 \) as \( k \rightarrow \infty \).

To ensure the closed-loop stability of the control system, the stability assurance layer proposed in this paper checks the sufficient condition given in Proposition 5.1. If the conditions in Proposition 5.1 do not hold (i.e. there is no solution for (5.4)-(5.6)), an alternative control action will be provided to guarantee stability based on Corollary 5.1.

The computational cost may be high due to its complexity, especially when the number of time steps required in (5.4) is exceeding high. To remove this obstacle, it is suggested to redefine the limit on the time step \( k \) used for finding the matrices \( Q_c(k), S_c(k), R_c(k) \) in (5.4), based on the computer capacity. The received matrices will be the least conservative ones the computer can afford. If the problem has a solution the closed-loop system deemed staying inside the stability boundary. Alternatively, an additional condition on performance or sensitivity can be used in parallel with Proposition 5.1, which is currently under investigation.

Minimize \( -\gamma \), subject to \( \gamma \geq 0 \), and

\[
w_c(k) = y^T(k)R_c y(k) + 2y^T(k)S_c u(k) + u^T(k)Q_c u(k) > \gamma.
\]

Where signal vectors \( y(k), u(k) \) are known.

In summary, the computation for the stabilizing boundary for the Supervisory Stability Layer (SSL) will be as per Procedure 5.1 below.

**Procedure 5.1** – At every time step of the online controller,

**Step 1** - Using the measured data of input and output in the past, and their current value from the controller, solve the system of three LMIs (5.4), (5.5), (5.6) for the matrices \( Q_c(k), S_c(k), R_c(k) \) and \( Q(k), S(k), R(k) \). If the solution is infeasible, go to Step 2, otherwise stay in this Step.

**Step 2** - Solve the system of three LMIs (5.8), (5.5), (5.6) for the matrices \( Q_c(k), S_c(k), R_c(k) \) and \( Q(k), S(k), R(k) \).

**Step 3** - Find \( \bar{u}(k) \) by solving the following problem:

\[
\begin{align*}
\text{Minimize} & \quad ||\bar{u}(k) - u(k)||_2^2, \text{ subject to} \\
& \quad 0 > \bar{w}_c(k) > w_c(k), \text{ or } \bar{w}_c(k) > 0.
\end{align*}
\]

With \( Q_c(k), S_c(k), R_c(k) \) matrices are found from Step 2, and the signal vector \( y(k) \) is known.

Then obtain the controller output so that \( w_c(k + 1) > \bar{w}_c(k) \). Repeat this Step in the next following iterations, if Step 1 is infeasible.

The above problem (5.9) can be converted into an optimization problem with constraints in LMIs, using the Schur complement if \( Q_c < 0 \). This is actually a condition that should be enforced to guarantee the stability of the controller (itself). It is worth noting here that this optimization problem is always feasible for stable (Hurwitz) systems.

**B. Constrained Problems**

If the constraints on the manipulated variables \( u(k) \) are considered, the following optimization problem is to be solved instead of the one in Proposition 5.1 or (5.8).

**Lemma 5.2** – For a control problem with \( q \) polytopic input constraints \( \mathcal{P} = \{u \mid a_i u \leq 1\} \), for \( i = 1 \ldots q \), the dissipativity parameter matrices \( Q_c(k), S_c(k), R_c(k), Q(k), S(k), R(k) \) and \( P(k) \) at the step \( k \) are the solutions to the following optimization problem:

\[
\begin{bmatrix}
\xi \\
\Phi u(k - 1) \\
\Phi(k) a_i - S_c^T y(k) \\
1
\end{bmatrix} > 0, \quad i = 1 \ldots q,
\]

\[
\begin{bmatrix}
-\gamma \\
\Phi(k) a_i - S_c^T y(k) \\
1
\end{bmatrix} > 0.
\]

(5.5) and (5.6).
With, \( \xi = y^T(k)R_yy(k) + W_c(k - 1) - 2u^T(k - 1)S_yy(k) \),
\( \Phi(k) = -Q_c(k) \).

**Proof:** It is to find the maximum ellipsoid volume containing \( u(k - 1) \) inside the constraint region that satisfies the conditions in Proposition 5.1. From (5.4), we have:
\[
y^T(k)R_yy(k) + 2y^T(k)S_yu(k) + u^T(k)Q_yu(k) \geq -W_c(k - 1).
\]
This inequality is represented by an ellipsoid \( E \), defined by \( \mathcal{E} = \{ u(k) | M_c(u(k)) \leq 0 \} \), with
\[
M_c(u(k)) = u^T(k)P_yu(k) + 2u^T(k)b + c.
\]
(5.11)

With, \( b = -S_cT\overline{y}(k), c = -W_c(k - 1) - y^T(k)R_cy(k), \)
\( \Phi = \Phi^T = -Q_c > 0 \).

The maximum ellipsoid \( E \) containing \( u(k - 1) \) inside the constraint region, therefore, can be found by (5.10). ■

The implementation of Step 1 in Procedure 5.1, therefore, must be modified for the constraint problems, based on Lemma 5.2, as follows:

**Step 1a:** Solve the optimization problem (5.10) for the matrices \( Q_c(k), S_c(k), R_c(k) \) and \( q(k), s(k), r(k) \). If the solution is infeasible, go to Step 2, otherwise stay in this Step.

**C. Graphical Presentation**

Consider a system (5.1) with two control inputs. Figures 5.1 and 5.2 below represent the evolution of the stability bounds when time elapses. Two different cases of dissipative ellipsoid regions, one is convex the other is concave relatively to the constraint region of manipulated inputs are shown in these Figures. They show how the stabilizing bounds on the control inputs are found.

![Figure 5.1](image1)

(a). \( W_c(k) < 0, W_c(k - 1) > 0 \) (b). \( W_c(k) < 0 \) for 2nd time

**Figure 5.1 – Real-time stability bound evolution.**

When the controller dissipativity index becomes negative for the first time, i.e. \( \bar{u}(k) \) – the computed (or predicted) controller output, is just outside the dissipative trajectory region, the signal vector \( \bar{u}(k) \) will be found by using the algorithm described in Procedure 5.1 above. With a known \( y(k) \) in the dissipativity index \( W_c(k) \), which is a quadratic function of vector \( \bar{u}(k) \), the solution will reside inside or outside an ellipsoid region, depending on the matrix \( Q_c \).

The algorithm in Procedure 5.1 will produce a point \( (\bar{u}_1(k), \bar{u}_2(k)) \) inside the stability boundary as shown in Figure 5.1/5.2 (a), which will become the re-assigned values \( (\bar{u}_1(k), \bar{u}_2(k)) \). Similarly, the new dissipative region emerges in Figure 5.1/5.2 (b) as a result of Procedure 5.1 with different ellipses, coming from the respective values of \( y(k + 1) \).

![Figure 5.2](image2)

(a). \( W_c(k) < 0, W_c(k - 1) > 0 \) (b). \( W_c(k) < 0 \) for 2nd time

**Figure 5.2 – Different dissipative ellipsoid regions.**

The dissipativity index \( W_c(k + 1) \) is negative again in Figure 5.1/5.2 (b), i.e. \( \bar{u}(k + 1) \) is outside the new dissipative region. The same process of finding the new stabilizing bound incurs in this case. Dependent upon the matrices, the plant model, the dissipative trajectory and sections will vary. The resultant stabilizing bounds and the system response time will therefore be affected.

**D. Feasibility**

The optimization problem in Procedure 5.1 may become infeasible in the next following steps due to constraint violations. One of the solutions for assuring feasibility is to have a shrinking dissipative ellipsoid region \( E \) over time. This is achievable by having an additional condition on the manipulated inputs as stated in Lemma 5.3 below.

**Lemma 5.3** – The diameter of the ellipsoidal dissipative region \( E \) monotonously reduces over time when the manipulated input vector at the current time step \( k \) satisfies the following quadratic constraint:
\[
\begin{align*}
&u^T(k)R_uu(k) + 2u^T(k)S + Q < 0, \quad \text{with} \\
&R = -B^TQ_cB + Q_c, \\
&S = (-A^TQ_cB + A^TS_c)^T x(k - 1) + \\
&(-B^TQ_cB + B^TS_c)^u(k - 1), \\
&Q = u^T(k - 1)[-B^TQ_cB + B^TW_cB - Q_c]u(k - 1) + \\
&x^T(k - 1)[-A^TQ_cA + A^TS_c]x(k - 1),
\end{align*}
\]
where, \( S_cQ_c - S_c^T = Q_c \), \( CA = A, CB = B \).

**Proof:** The monotonously reducing diameter of \( E \) requires \( \Phi^2(k) < \Phi^2(k - 1) \), which is equivalent to:
\[
b^T(k)(-Q_c)^{-1}b(k) + c(k) < b^T(k - 1)(-Q_c)^{-1}b(k) - c(k - 1),
\]
where, \( b(i) = -S_c^Ty(i), c(i) = -W_c(i - 1) - y^T(i)R_cy(i) \).

With some tedious manipulations, the inequality (5.12) is obtained ■

It is worth noting here that if the dissipativity index is monotonously decreasing, i.e. \( W_c(k) < 0 \), the inequality
(5.12) will be satisfied despite of \( u(\cdot) \) and \( x(\cdot) \), provided that
\[
\begin{bmatrix}
-R_c & S_c \\
S_c^T & -Q_c
\end{bmatrix} > 0, \text{ and } R_c < 0.
\] (5.13)

E. Bumpless transfer

To avoid the system instability caused by possible state jumps while switching between the controller outputs and the stabilizing outputs, it is required that the constraints on variable increments are imposed.

VI. ILLUSTRATIVE EXAMPLE

In this section, an MPC controller is deployed to illustrate the supervisory stability layer operation.

**Problem 6.1** – Consider a discrete-time LTI state space system (5.1). Its control inputs, input increments and plant outputs are constrained by their respective physical limits. The MPC objective function consists of these three variables associated with their weighting matrices \( W_* \). The predictive (output) horizon is \( N_y \), the control horizon is \( N_u \).

Minimize \( J(k) = \sum_{i=0}^{N_y} \bar{y}(k+i+1)^T W_y \bar{y}(k+i+1) + \sum_{i=1}^{N_u} \Delta \bar{u}(k+i)^T W_u \Delta \bar{u}(k+i) \) subject to the equality constraint of the state space model, and the inequality constraints of decision variables.

The control minimizing sequence can be computed by using an MPC numerical algorithm (such as the \textit{mpcmove} function in Matlab MPC toolbox).

**Example 6.1** - Consider the following continuous time state space model of a paper machine [9]:

\[
A = \begin{bmatrix}
-1.93 & 0.27 & 0 & 0 \\
0.94 & -0.43 & 0 & 0 \\
0 & 0 & -0.63 & 0 \\
0.82 & -0.78 & 0.41 & -0.42
\end{bmatrix}; \\
B = \begin{bmatrix}
1.27 & 1.27 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1.3 & -0.65 & 0.21 & 0.41 \\
0 & 0 & 0 & 0
\end{bmatrix}; \\
C = \begin{bmatrix}
0 & 1.0 & 0.0 \\
0 & 0 & 1.0 \\
0 & 0 & 0 & 1.0
\end{bmatrix}; \quad D = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}.
\]

Matrices \( Q,S,R \) and \( P \) (may vary in different steps) are calculated by using Proposition 5.1 with the historical data of controller outputs in 200 time steps. A very short predictive horizon \( (N_y=3, N_u=2) \) has been simulated with the \textit{mpcmove} function. The plant output constraints are omitted. Figure 6.1a depicts the output response of the MPC alone, which is unstable due to the short predictive horizon (it is stable with longer horizons). Figure 6.1b shows the implementation results of the proposed SSL where the MPC output is overridden by the SSL output (following Procedure 5.1), i.e. when the trajectory based stability condition is not satisfied.

It is clear that the SSL ensures the stability of the closed-loop system.

VII. CONCLUSION

Motivated by the idea of segregating the stability assurance problems from the multivariable control designs, a new Supervisory Stability Layer (SSL) that can be used in the multilayer plant-wide control architecture is developed in this paper. Due to its independence, the SSL is potentially applicable to all types of real-time controllers which may or may not be represented using closed-form dynamic models (e.g., in ODEs). It is employable to control systems that implement the online optimization (such as MPC) and/or artificial intelligence.

For MPC implementations, the SSL introduced herein can be used with different existing industrial MPC setups using impulse and/or step response models (such as MAC, DMC) as an external add-on mechanism, rather than having to replace/modify the controller setups.

REFERENCES