System Level Synthesis with State and Input Constraints

Yuxiao Chen and James Anderson

***Abstract*— This paper addresses the problem of designing distributed controllers with state and input constraints in the System Level Synthesis (SLS) framework. Using robust optimization, we show how state and actuation constraints can be incorporated into the SLS structure. Moreover, we show that the dual variable associated with the constraint has the same sparsity pattern as the SLS parametrization, and therefore the computation distributes using a simple primal-dual algorithm. We provide a stability analysis for SLS design with input saturation under the Internal Model Control (IMC) framework. We show that the closed-loop system with saturation is stable if the controller has a gain less than one. In addition, a saturation compensation scheme that incorporates the saturation information is proposed which improves the naive SLS design under saturation.**

I. Introduction

System Level Synthesis (SLS) is a recently developed framework for formulating optimal distributed control problems. Traditional methods seek to minimize the closed-loop map from exogenous disturbance to error, formally they try to solve

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & \|f(\mathbf{P},\mathbf{K})\| \\ \text{subject to} \quad & \mathbf{K} \text{ stabilizes } \mathbf{P}, \quad \mathbf{K}\in\mathcal{C}, \end{aligned}$$

where $f(\mathbf{P},\mathbf{K}) = \mathbf{P}_{11}+\mathbf{P}_{12}\mathbf{K}(I-\mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}$. The distributed nature of the problem is encapsulated in the information constraint $\mathbf{K}\in\mathcal{C}$. The sub-space $\mathcal{C}$ encodes for sparsity of the controller structure as well as various delays incurred through communication, actuation, and sensing. The seminal work of Rotkowitz and and Lall [12] characterizes when this problem has a convex solution. In later work the conditions for convexity in [8] were shown to be both necessary and sufficient.

In comparison, the SLS framework [1] does not consider the map $\bar{\mathbf{z}} = f(\mathbf{P},\mathbf{K})\mathbf{w}$. Instead, it takes into account the system state and optimizes the closed-loop map from disturbance to state and disturbance to control action:

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix}\mathbf{w}, \tag{1}$$

where

$$\mathbf{\Phi}_x = (zI-A-B\mathbf{K})^{-1}, \mathbf{\Phi}_u = \mathbf{K}(zI-A-B\mathbf{K})^{-1}.$$

One of the central results of SLS is the ability to design the *system response* pair $\{\mathbf{\Phi}_x,\mathbf{\Phi}_u\}$ using convex programming (see Section I-A.2 for details). In this work we show how constraints on the state and control action can be incorporated into the SLS formulation. In related work [5], Dean et al consider the problem of *safely learning* LQR dynamics which incorporate system constraints in an SLS setting.

Yuxiao Chen is with the Department of Mechanical and Civil Engineering, Caltech, Pasadena, CA, 91106, USA. Emails: `chenyx@caltech.edu`

James Anderson is with the Computing and Mathematical Sciences Department, Caltech, Pasadena, CA, 91106, USA. Emails: `james@caltech.edu`

A. Preliminaries

In this section we provide the necessary background material on SLS distributed control.

1) Nomenclature: $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. Transfer matrices and signals are denoted in bold, constant matrices and vectors are not. Matlab notation is used to denote the vertical concatenation of two objects of appropriate dimension, e.g., $[A;B]=[A^\intercal\ B^\intercal]^\intercal$.

$\mathcal{P}(P,q)=\{x \mid Px\le q\}$ denotes the polyhedron defined with P and q. $\mathbf{BlkDiag}(A_1,A_2,\ldots,A_k)$ represents the block diagonal matrix with diagonal blocks $A_1,A_2,\ldots,A_k$.

2) System Level Synthesis: Here we provide a very brief overview of the system level synthesis framework for distributed control. This paper is intended to be self contained, but the reader is referred to [1], [13] for a more complete picture of both theory and computation. We consider a linear time-invariant plant model of the form $x(k+1) = Ax(k)+Bu(k)+w(k)$, where $x(k),w(k)\in\mathbb{R}^n$ are the state and noise vectors at time k, and $u(k)\in\mathbb{R}^m$ is the control action at time k. The control synthesis problem is to design a dynamic state-feedback policy $\mathbf{u}=\mathbf{K}\mathbf{x}$. The system level synthesis approach to control synthesis is based on the notion of designing the closed-loop system responses $\{\mathbf{\Phi}_x,\mathbf{\Phi}_u\}$ (by choice of $\mathbf{K}$). Any stable and strictly-proper transfer matrices $\{\mathbf{\Phi}_x,\mathbf{\Phi}_u\}$ that satisfy the affine expression

$$\begin{bmatrix} zI-A & -B\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix} = I \tag{2}$$

can be used to construct an internally stabilzing controller $\mathbf{K}=\mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$. In this work we find it more convenient to work in the time-domain. We thus work with a convolutional representation of the system response given by

$$x(k)=\sum_{t=1}^{k+1}\Phi_x[t]w(k-t), \quad u(k)=\sum_{t=1}^{k+1}\Phi_u[t]w(k-t).$$

The relationship between $\mathbf{\Phi}_x,\mathbf{\Phi}_u$ and $\Phi_x[1],\ldots$,and $\Phi_u[1],\ldots$ is given through the spectral decomposition of a transfer matrix: $\mathbf{\Phi}_x = \sum_{k=0}^{\infty}\Phi_x[k]z^{-k}$, $\mathbf{\Phi}_u = \sum_{k=0}^{\infty}\Phi_u[k]z^{-k}$. Note that (2) imposes the constraint $\Phi_x[1]=I$.

II. SLS WITH STATE AND INPUT CONSTRAINT

In this section, we consider the SLS synthesis problem with state and input constraint.

The system dynamics are modeled by a discrete time linear model:

$$x(k+1) = Ax(k) + Bu(k) + w(k), \tag{3}$$

as described in Section I-A.2. We assume that a bound on exogenous disturbance w_k is known and is provided as part of the problem specification. The bounds we use will be time invariant, therefore we drop time dependencies from our notation when describing these constraints. Let the disturbance constraint be modeled as $Gw \leq g$, where $G \in \mathbb{R}^{q\times n}$, $g \in \mathbb{R}^q$ specify a polytopic bound on w. Under this assumption, our goal is to design a distributed controller such that the state x and control input u are bounded for all time. To write this compactly, we use $H[x;u] \leq h$ to denote the state and input bound, where $H \in \mathbb{R}^{p\times(n+m)}$, $h \in \mathbb{R}^p$. Although, the constraint is assumed to be time invariant, the proposed method can be easily extended to the time varying formulation. The state bound may come from the specification for the design, whereast the input bound usually comes from actuator saturation. Furthermore, locality (in space and time) constraints can be easily incorporated into the synthesis procedure. Informally, spatial locality is encoded via sparsity constraints on the matrices $\Phi_x[i], \Phi_u[i]$ for all i and a finite impulse response (FIR) constraint sets $\Phi_x[t] = 0, \Phi_u[t] = 0$ for all $t > T$. The following lemma links the different components and constraints together.

Lemma 1. *For a closed-loop stable FIR system with horizon T, if the closed-loop response is described by* (1)*, then the following statements are equivalent:*

1) *For all allowable w, for all $t \geq 0$, $H[x(t); u(t)] \leq h$*
2) *For all allowable $w(0:T-1)$,*
$$H\begin{bmatrix}\sum_{i=0}^{T-1}\Phi_x[T-i]w(i)\\ \sum_{i=0}^{T-1}\Phi_u[T-i]w(i)\end{bmatrix} \leq h,$$
where $w(0:T-1)$ denotes $w(0), \ldots, w(T-1)$.

Proof. It is trivial to see that 2) is a necessary condition to 1) by taking $t = T$; For sufficiency, notice that the constraint on w, x and u are time invariant, and the system response is FIR with horizon T. ■

Using the lemma above, we can now state the state- and input-constrained distributed SLS problem:

$$\begin{aligned}
\min_{\mathbf{\Phi}_x,\mathbf{\Phi}_u}\ & J(\mathbf{\Phi}_x,\mathbf{\Phi}_u)\\
\text{s.t.}\ & \begin{bmatrix}Iz-A & -B\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix} = I,\\
& \mathbf{\Phi}_x,\mathbf{\Phi}_u \in \frac{1}{z}\mathcal{RH}_\infty \cap \mathcal{S}\\
& \forall w \in \mathcal{P}(\widehat{G},\hat{g}),\ H\left[\sum_{i=0}^{T-1}\Phi[T-i]w(i)\right] \leq h
\end{aligned} \tag{4}$$

where $\widehat{G} = I_T \otimes G$, $\hat{g} = \mathbf{1}_T \otimes g$ with $\otimes$ denoting the Kronecker product. The set $\mathcal{S}$ encodes the FIR constraint and spatial locality constraint, see [13] for details. We refer to (4) as the robust SLS problem.

The cost functional J is chosen to be convex in $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$, a typical choice would be the $\mathcal{H}_2$-norm. This is a robust convex optimization problem in the sense that the last constraint is required to hold over all possible disturbances w. We use the dualization method that converts this robust optimization into a tractable convex program.

Lemma 2. *Consider the following robust optimization problem:*

$$\begin{aligned}
\min_{\alpha}\ & J(\alpha)\\
\text{s.t.}\ & \forall\beta \in \{\beta \mid G\beta \leq g\},\\
& H_1\beta + \alpha^\intercal H_2\beta + H_3\alpha \leq h,
\end{aligned} \tag{5}$$

where α is the decision vector, $J(\cdot)$ is a convex cost function, β is the uncertain variable, and $G\beta \leq g$ is the bound for uncertainty. $H_{1,2,3}$ are constant matrices of appropriate dimensions. The robust optimization problem (5) *is equivalent to the following convex program:*

$$\begin{aligned}
\min_{\alpha,\lambda}\ & J(\alpha)\\
\text{s.t.}\ & H_3\alpha + \lambda g \leq h\\
& H_1 + \alpha^\intercal H_2 = \lambda G\\
& \lambda \geq 0,
\end{aligned} \tag{6}$$

where λ is the dual variable.

Proof. This is a specific instantiation of the general theory developed in [2], see the proof therein. ■

For notational convenience let $\mathbf{\Phi} = [\mathbf{\Phi}_x; \mathbf{\Phi}_u]$. Applying Lemma 2 to the robust SLS synthesis problem (4), the robust optimization is transformed into the following convex optimization:

$$\begin{aligned}
\min_{\mathbf{\Phi}_x,\mathbf{\Phi}_u,\Lambda\geq 0}\ & J(\mathbf{\Phi}_x,\mathbf{\Phi}_u)\\
\text{s.t.}\ & \begin{bmatrix}Iz-A & -B\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_x\\ \mathbf{\Phi}_u\end{bmatrix} = I & (a)\\
& \mathbf{\Phi}_x,\mathbf{\Phi}_u \in \frac{1}{z}\mathcal{RH}_\infty \cap \mathcal{S} & (b)\\
& H\Phi[k] = \Lambda[k]\widehat{G}, \forall k = 1, ..., T & (c)\\
& \Lambda\hat{g} \leq h & (d)
\end{aligned} \tag{7}$$

In this setting, the variable λ from (6) is now T matrices $\Lambda[1:T], \Lambda[i] \in \mathbb{R}^{p\times q}$ and the inequality is applied in an element-wise manner. One issue with this approach is that it requires an additional variable Λ of dimension $\mathbb{R}^{p\times q\times T}$, which can be large. We will show that when the state space is decomposed into patches, and the state and input constraints are uncoupled, Λ has the same sparsity pattern as the SLS parametrization Φ_x and Φ_u.

Suppose the state is decomposed into k patches: $x = [x_1; x_2; \ldots x_k]$, where $x_i \in \mathbb{R}^{n_i}$ are of different dimensions and $\sum_{i=1}^k n_i = n$. Accordingly, w is decomposed in the same pattern $w = [w_1; w_2; \ldots w_k]$.

Definition 1. *The closed loop performance constraint $\mathcal{P}(H, h)$ is decoupled if H is block diagonal under decomposition $x = [x_1; x_2; \dots x_k]$, $u = [u_1; u_2; \dots u_k]$. The bound $\mathcal{P}(G, g)$ on exogenous disturbance w is decoupled if G is block diagonal under decomposition $w = [w_1; w_2; \dots w_k]$.*

Theorem 1. *If the performance constraint $\mathcal{P}(H, h)$ and the disturbance constraint $\mathcal{P}(G, g)$ are both decoupled, Λ has the same sparsity pattern as Φ_x and Φ_u.*

Proof. By assumption, we can write

$$H = \mathbf{BlkDiag}(H_1, H_2, \dots H_k),$$
$$G = \mathbf{BlkDiag}(G_1, G_2, \dots G_k).$$

Take $t = 1$ as an example. The equality constraint for $\Phi[1]$ and $\Lambda[1]$ can be written in blocks:

$$\begin{bmatrix} H_1\Phi^{1,1}[1] & \cdots & H_1\Phi^{1,k}[1] \\ \vdots & \ddots & \vdots \\ H_k\Phi^{k,1}[1] & \cdots & H_k\Phi^{k,k}[1] \end{bmatrix} = \begin{bmatrix} \lambda^{1,1}[1]G_1 & \cdots & \lambda^{1,k}[1]G_k \\ \vdots & \ddots & \vdots \\ \lambda^{k,1}[1]G_1 & \cdots & \lambda^{k,k}[1]G_k \end{bmatrix},$$

which can be decomposed into blocks as

$$\forall i, j \in \{1, \dots, k\}, \forall t \in \{1, \dots, T\}, H_i \Phi^{i,j}[t] = \lambda^{i,j}[t] G_j.$$

Likewise, the inequality constraint in block form is

$$\forall i = 1, \dots, k, \sum_{t=1}^{T} \sum_{j=1}^{k} \lambda^{i,j}[t] g_j \le h_i. \tag{8}$$

Due to the locality constraint $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\} \in \mathcal{S}$, many blocks of $\mathbf{\Phi}$ will be zero. For those zero blocks $\Phi^{i,j}[t]$, the equality constraint becomes $\lambda^{i,j}[t]G_j = 0$. Note that since $\lambda \ge 0$,

$$\lambda^{i,j}[t] g_j \ge \lambda^{i,j}[t] G_j w = 0 \cdot w = 0$$

This means that for any feasible solution of the original optimization, we can change $\lambda^{i,j}[t]$ to zero and the solution would still be feasible. Besides, λ does not appear in the cost function, therefore, we can simply set $\lambda^{i,j}[t] = 0$ for all $\Phi^{i,j}[t] = 0$. ∎

In [9], the authors showed that the SLS constraints ((a),(b) in (7)) are column-wise separable, therefore, when the objective function is separable, for example the H_2-norm, the whole synthesis can be decomposed into patches. Theorem 1 showed that the equality constraint induced by the state and input constraint ((c) in (7)) can be column-wise separated, and even inherits the sparsity pattern of Φ. Unfortunately, the last inequality constraint ((d) in (7)) is not column-wise separable. However, at this point, the patches are only coupled by several linear constraints and the coupling only exists between neighbors determined by the locality constraint. Therefore, one can still decentralize the synthesis in (7) and resolve the coupling with a simple primal-dual algorithm. Let the dual variable associated with the inequality constraint be $\sigma \in \mathbb{R}^{1 \times p}$, then the primal-dual update is as follows:

Primal update:

$$\begin{aligned} \min_{\mathbf{\Phi}_x, \mathbf{\Phi}_u, \Lambda \ge 0} \; & J(\mathbf{\Phi}_x, \mathbf{\Phi}_u) + Tr[(\hat{g} \cdot \sigma)\Lambda] \\ \text{s.t.} \; & \begin{bmatrix} Iz - A & -B \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I \\ & \mathbf{\Phi}_x, \mathbf{\Phi}_u \in \frac{1}{z} \mathcal{RH}_\infty \cap \mathcal{S} \\ & H \begin{bmatrix} \Phi[T] & \dots & \Phi[1] \end{bmatrix} = \Lambda \widehat{G} \end{aligned} \tag{9}$$

Dual update:

$$\sigma = \max(0, \sigma + \alpha(\Lambda \hat{g} - h)),$$

where α is the step size, and the stopping criteria is

$$(\Lambda \hat{g} \le h) \textbf{ and } (|\sigma(\Lambda \hat{g} - h)| \le \epsilon), \tag{10}$$

where ϵ is the tolerance. The first condition is on primal feasibility and the second condition is on complementary slackness. Note that the primal update is column-wise separable and the dual update only requires local communication between neighbors.

Remark 1. *Compared to the invariant set method [3], [4], [7], [11], that also guarantees closed-loop state bounds under a bounded disturbance signal, the SLS method proposed in this paper does not use the invariance condition of a set, but rather directly parameterize the closed-loop response. Therefore, the bound obtained is tight, as is shown in Section IV. In addition, the problem setup can be naturally extended to situations with time varying disturbance bounds. Such a setup is useful in situations when there is a nominal bound on disturbance but unexpected large disturbances may occur, but not frequently.*

III. Input Saturation

Sometimes even with design that respects the input bound, the actuator saturates under unexpected large exogenous disturbance. In this section, we discuss the SLS design under input saturation.

A. Stability of saturated SLS

We will show that the SLS controller resembles the internal model control (IMC) scheme discussed in [6], [10], and via the existing result in IMC, we prove stability of the system by saturating the internal model. Consider the dynamical system (3). Suppose a stabilizing controller parameterized by $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$ is obtained via SLS synthesis. The proposed control structure is depicted in Fig. 1.

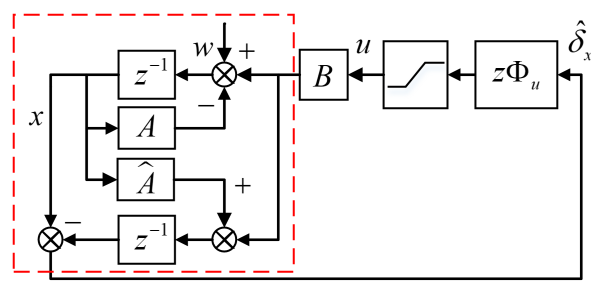

Fig. 1: Block diagram with saturation.

Theorem 2. *Assume that the dynamic system in* (3) *is open-loop stable, then (i) when $\widehat{A} = A$ is a perfect model for the dynamics, the closed loop system depicted in Fig. 1 is stable; (ii) when $\widehat{A} = A$ and the actuator is not saturated, the block diagram is a realization of the SLS controller; (iii) when $\widehat{A} \neq A$, but*

$$\left\|(A - \hat{A})(Iz - A)^{-1} z\Phi_u\right\| < 1, \tag{11}$$

the closed loop system is stable.

Proof. We prove stability using a small gain theorem argument. The transfer function from Bu to $\hat{\delta}_x$ (the red block) in Fig. 1 is $(A - \widehat{A})(Iz - A)^{-1}$. Clearly, when $\widehat{A} = A$ and the system is open-loop stable, the gain is zero. By constraint (b) of (7), $z\mathbf{\Phi}_u \in \mathcal{RH}_\infty$ which implies stability, therefore the overall loop gain is less than 1, and the system is stable. (iii) can be proved directly with small gain theorem.

For the second claim, first notice that without saturation, the closed-loop system becomes Fig. 2 after some block diagram manipulation.

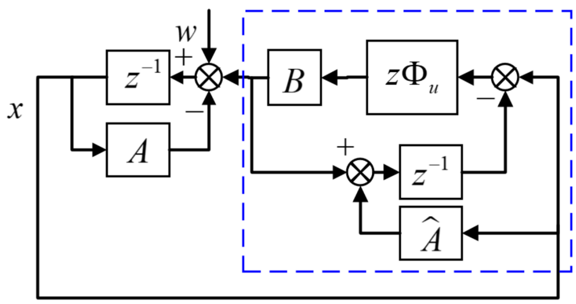

Fig. 2: Block diagram without saturation.

It is sufficient to show that the blocks within the blue square are equivalent to $B\mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$. To see this, note that $\hat{A} = A$, and from the block diagram we have

$$\begin{aligned} z\mathbf{\Phi}_u(x - z^{-1}(\widehat{A}x + Bu)) &= u \\ \Rightarrow u + \mathbf{\Phi}_u Bu &= \mathbf{\Phi}_u(Iz - A)x. \end{aligned} \tag{12}$$

Taking the SLS constraint (a) in (7) and pre-multiplying by $(Iz - A)^{-1}$ and post-multiplying by $\mathbf{\Phi}_x^{-1}$ we obtain

$$\begin{aligned} B\mathbf{\Phi}_u &= (Iz - A)\mathbf{\Phi}_x - I && (a) \\ (Iz - A)^{-1}B\mathbf{\Phi}_u &= \mathbf{\Phi}_x - (Iz - A)^{-1} && (b) \\ Iz - A - \mathbf{\Phi}_x^{-1} &= B\mathbf{\Phi}_u\mathbf{\Phi}_x^{-1} && (c) \end{aligned} \tag{13}$$

respectively. Utilizing these expressions, and starting by multiplying (12) on both sides by B, the derivation follows as

$$Bu + B\mathbf{\Phi}_u Bu = B\mathbf{\Phi}_u(Iz - A)x,$$

apply (a) to the LHS and rearrange to get

$$\mathbf{\Phi}_x Bu = (Iz - A)^{-1}B\mathbf{\Phi}_u(Iz - A)x.$$

Applying (b) to the RHS gives

$$\begin{aligned} \mathbf{\Phi}_x Bu &= (\mathbf{\Phi}_x - (Iz - A)^{-1})(Iz - A)x \\ \Rightarrow Bu &= (Iz - A - \mathbf{\Phi}_x^{-1})x, \end{aligned}$$

and finally, applying (c) to the RHS gives

$$Bu = B\mathbf{\Phi}_u\mathbf{\Phi}_x^{-1},$$

which proves that the transfer function from x to Bu is $B\mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$. ■

B. Compensation controller for actuator saturation

Under this IMC design, when input saturation occurs, it is possible that a large transient would occur if the system is close to instability. This is because the state deviation caused by saturation is not treated with the IMC controller since it is not an exogenous disturbance and therefore gets canceled by the internal model. One approach to fixing this is to treat the saturation as an additional source of disturbance, i.e., $\bar{w} = B\Delta u$, where Δu is the part of input that gets cut off due to saturation. Since the local controller can detect the actuator saturation, the saturation status can be broadcast to the neighbors alongside state x. With this additional signal we can design a compensation controller that deals with the "disturbance" caused by saturation. Let $\bar{\mathbf{\Phi}}_\mathbf{x}$ and $\bar{\mathbf{\Phi}}_\mathbf{u}$ denote the closed-loop state and input response to $\bar{w}$, then we have

$$\begin{aligned} x &= \mathbf{\Phi}_x w + \bar{\mathbf{\Phi}}_\mathbf{x}\bar{w} \\ u &= \mathbf{\Phi}_u w + \bar{\mathbf{\Phi}}_\mathbf{u}\bar{w}, \end{aligned} \tag{14}$$

where $\bar{w}$ is directly known to the controller. When designing $\bar{\mathbf{\Phi}}_\mathbf{x}$ and $\bar{\mathbf{\Phi}}_\mathbf{u}$, an additional constraint should be added. For the saturated node i, the control capacity is zero, i.e., $\bar{\mathbf{\Phi}}_\mathbf{u}(i, :) = 0$. Since the control policy is localized, the i-th node would have to compute N_i compensation controllers that correspond to N_i different saturation scenarios, where N_i is the number of neighbors for node i (including itself).

Remark 2. *We only consider the case where one node in a local patch is saturated since the compensation controller with one saturated node is guaranteed to have the same locality structure as the original SLS design. However, the compensation controller with more than one saturated node may not be feasible due to communication limitations. When multiple neighbors are saturated, we simply superimpose a compensation controller for each saturated node on top of each other.*

Remark 3. *With the additional input constraint caused by actuation saturation, the synthesis of the compensation controller $\bar{\mathbf{\Phi}}_\mathbf{u}$ may be infeasible, in which case we use the robust version of SLS introduced in [9] to obtain a controller.*

With $\bar{\mathbf{\Phi}}_\mathbf{x}$ and $\bar{\mathbf{\Phi}}_\mathbf{u}$ designed, the implementation of the compensation controller is shown in Fig. 3.

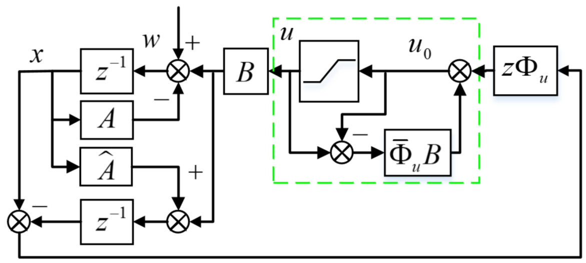

Fig. 3: Closed-loop system with the compensation controller

For notational convenience, define the saturation operator as

$$\mathbf{Sat}_{\mathcal{U}}(u) = \underset{\bar{u} \in \mathcal{U}}{\arg\min} \|u - \bar{u}\|,$$

and the cut-off operator is defined as

$$\mathbf{Co}_{\mathcal{U}}(u) = u - \mathbf{Sat}_{\mathcal{U}}(u).$$

Theorem 3. *When $\|z\bar{\mathbf{\Phi}}_{\mathbf{u}}\| \leq 1$ and the feasible input set $\mathcal{U}$ contains the origin, the block diagram in Fig. 3 is stable. Furthermore, when $\bar{\mathbf{\Phi}}_{\mathbf{u}} = \mathbf{\Phi}_u$, the closed-loop system is equivalent to the original SLS controller with saturation.*

Proof. The stability follows from small gain theorem. Based on the proof of Theorem 2, all we need to show is the stability of the blocks in the green square, which is a feedback loop consisting of the cut-off operator and $z\bar{\mathbf{\Phi}}_{\mathbf{u}}$. When $\mathcal{U}$ contains the origin, it is obvious that $\|\mathbf{Co}_{\mathcal{U}}(\cdot)\| < 1$. Therefore, when $\|\bar{\mathbf{\Phi}}_{\mathbf{u}}\| \leq 1$, the compensation block is stable.

The original SLS with saturation is shown in Fig. 4.

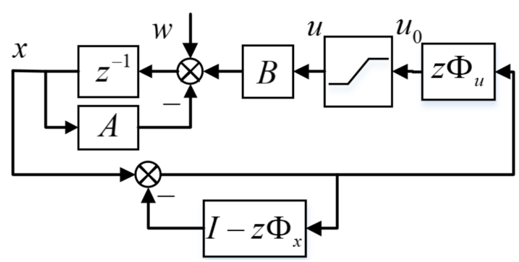

Fig. 4: Closed-loop system under original SLS controller with saturation

The second claim can be proved by observing that when $\bar{\mathbf{\Phi}}_{\mathbf{u}} = \mathbf{\Phi}_u$, the two block diagrams in Fig. 3 and Fig. 4 can be both simplified to

$$B\mathbf{\Phi}_u B u + B\mathbf{\Phi}_u w = (Iz - A)\mathbf{\Phi}_x B u_0, \tag{15}$$

where $u = \mathbf{Sat}_{\mathcal{U}}(u_0)$. ■

We note that the sparsity constraint $\bar{\mathbf{\Phi}}_u$ may seem restrictive, however, as we show in our numerical examples this doesn't appear to be the case. Moreover, should an infeasibility occur due to this constraint, we may apply results from robust SLS as described in [9] that relax the affine constraint (a) in 7.

With $\bar{\mathbf{\Phi}}_{\mathbf{u}}$ taking the control incapability caused by actuation saturation into account, we show in Section IV that it outperforms the original SLS controller.

IV. Numerical Example

We first show the simulation result of SLS design with state and input constraint. The model is the classic bidirectional chain system considered in [9]. The dynamic equation is

$$x_i^+ = \rho\left(1 - |\mathcal{N}_i|\alpha\right) x_i + \rho\alpha \sum\nolimits_{j \in \mathcal{N}_i} x_j + u_i, \tag{16}$$

where x_i and u_i are the state and control of node i, $\mathcal{N}_i$ denotes the set of neighbors of x_i and $|\mathcal{N}_i|$ denotes the cardinality of the neighbor set. α and ρ are parameters, we pick $\alpha = 0.4$ and $\rho = 1$, which makes the system neutrally stable.

We assume an infinity norm bound on w, and the constraint on x and u is also in the infinity norm:

$$\mathcal{P}(G, g) := \{w \mid w_{\max} \geq w \geq -w_{\max}\}$$
$$\mathcal{P}(H, h) := \left\{ x, u \;\middle|\; \begin{array}{l} x_{\max} \geq x \geq -x_{\max} \\ u_{\max} \geq u \geq -u_{\max} \end{array} \right\},$$

where $x_{\max} \in \mathbb{R}^n$, $u_{\max} \in \mathbb{R}^m$ and $w_{\max} \in \mathbb{R}^n$ are the bound on state, input and disturbance.

Remark 4. *The proposed framework naturally extends to the case with time varying bounds, for simplicity, we assume a constant bound for w for all $t \geq 0$.*

For reasonable locality constraints and FIR horizon (horizon length $T = 4$, locality radius $d = 3$), the SLS synthesis is feasible. In addition, the decentralized primal-dual algorithm (9) converges to the centralized synthesis solution. To validate the result, we first solve for the worst-case disturbance, i.e., $w_{0:T-1} \in \mathcal{P}(\hat{G}, \hat{g})$ that is the worst case for one of the constraints, which is formulated as the following convex optimization problem:

$$\begin{aligned} \max_{w(0:T-1)} \; & H_i \sum\nolimits_{t=1}^{T} \Phi_x[T-t] w(t) \\ s.t.\; & \forall t = 0, \ldots, T-1, Gw(t) \leq g, \end{aligned} \tag{17}$$

where H_i is any row of H. Since (17) searches for the worst-case disturbance for one row of the constraint matrix, which can be either a state or input constraint, the result will depend on which constraint it tries to violate. As an example, we take the state constraint on one of the nodes in the chain system, and the worst-case disturbance is shown in Fig. 5. Because of the locality constraint, the worst-case disturbance is also local, i.e., only nonzero among neighbors of node i. The closed loop response of x_i is shown in Fig. 6, and it stays within the bound specified as part of the SLS specification. Note that in general one cannot find a worst-case disturbance sequence for multiple state and input constraints, therefore we simply pick one of the constraint and solve for the worst-case disturbance sequence.

Next, we simulate the compensation controller. As discussed in Section III-B, the saturated IMC design may have large transient, and the IMC design with the original SLS

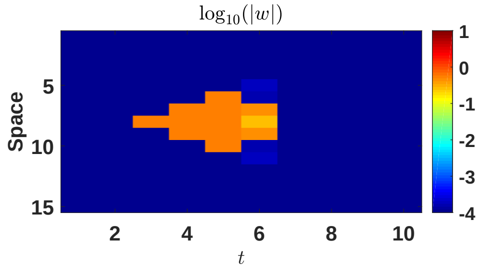

Fig. 5: Worst-case disturbance.

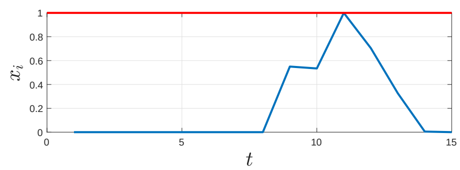

Fig. 6: State response under the worst-case disturbance.

controller $\mathbf{\Phi}_u$ as the compensation controller is equivalent to the naive SLS control with saturation. We show the comparison in Fig. 7 between the naive SLS controller with saturation (top three plots) and SLS with saturation compensation, i.e., a compensation controller that assumes zero control capability for the saturated node (bottom three plots).

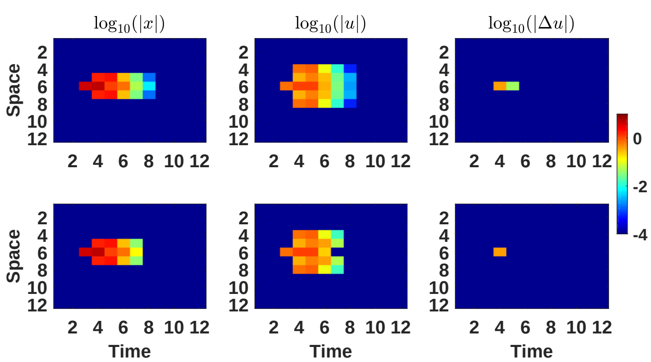

Fig. 7: Simulation of naive SLS with saturation (top) and SLS with saturation compensation (bottom).

The results show that with a compensation controller that incorporates the saturation information, the state and input converges to zero faster than the original SLS with saturation.

V. Conclusion

We have described a method for incorporating state and input constraints into distributed control problems formulated using the system level synthesis framework. It was shown that we can use robust optimization to incorporate the constraints, and that the coupling constraints distribute when using a simple primal-dual algorithm. It was further shown that the closed-loop transient dynamics can be improved upon by a slight modification to the controller (provided a simple gain condition is satisfied).

In future work we will consider the case of linear time-varying systems. The theory for SLS in this setting exists and we are currently mapping it to the constrained control setting. Such an approach will form the basis of a distributed model-predictive control scheme.

VI. Acknowledgements

We gratefully acknowledge John C. Doyle for suggesting we look at internal model control theory in connection with this problem.

References

[1] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni. System level synthesis. *Annual Reviews in Control*, 2019.

[2] A. Ben-Tal, D. Den Hertog, and J.-P. Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical programming*, 149(1-2):265–299, 2015.

[3] Y. Chen, J. Anderson, K. Kalsi, S. H. Low, and A. D. Ames. Compositional set invariance in network systems with assume-guarantee contracts. *arXiv preprint arXiv:1810.10636*, 2018.

[4] Y. Chen, H. Peng, J. Grizzle, and N. Ozay. Data-driven computation of minimal robust control invariant set. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4052–4058. IEEE, 2018.

[5] S. Dean, S. Tu, N. Matni, and B. Recht. Safely learning to control the constrained linear quadratic regulator. *arXiv preprint arXiv:1809.10121*, 2018.

[6] J. C. Doyle, R. S. Smith, and D. F. Enns. Control of plants with input saturation nonlinearities. In *1987 American Control Conference*, pages 1034–1039. IEEE, 1987.

[7] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical problems in engineering*, 4(4):317–367, 1998.

[8] L. Lessard and S. Lall. Quadratic invariance is necessary and sufficient for convexity. In *Proceedings of the 2011 American Control Conference*, pages 5360–5362. IEEE, 2011.

[9] N. Matni, Y.-S. Wang, and J. Anderson. Scalable system level synthesis for virtually localizable systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3473–3480. IEEE, 2017.

[10] M. Morari and E. Zafiriou. *Robust process control*. Morari, 1989.

[11] S. V. Rakovic and M. Baric. Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks. *IEEE Transactions on Automatic Control*, 55(7):1599–1614, 2010.

[12] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 50(12):1984–1996, 2005.

[13] Y.-S. Wang, N. Matni, and J. C. Doyle. Separable and localized system-level synthesis for large-scale systems. *IEEE Transactions on Automatic Control*, 63(12):4234–4249, 2018.