Supervised-distributed control with joint performance and communication optimization

Romain Guicherd[a], Paul A. Trodden[a], Andrew R. Mills[a] and Visakan Kadirkamanathan[a]

[a]Rolls-Royce University Technology Centre,
Department of Automatic Control & Systems Engineering,
University of Sheffield,
Mappin Street, Sheffield, S1 3JD, UK

ARTICLE HISTORY
Compiled October 12, 2020

ABSTRACT
This paper is concerned with the control of systems composed of multiple coupled subsystems. In such architectures, communication between different local controllers is desired in order to achieve a better overall control performance. Any resultant improvement in control performance needs, however, to be significant enough to warrant the additional design complexity and higher energy consumption and costs associated with introducing communication channels between controllers. A practical distributed control design aims, therefore, to achieve an acceptable balance between minimizing the use of communication between controllers and maximizing the system-wide performance. In this article, a new approach to the problem of synthesizing stabilizing distributed control laws for discrete-time linear systems that balances performance and communication is presented. The approach employs a supervisory agent that, periodically albeit not necessarily at every sampling instant, solves an optimization problem in order to synthesize a stabilizing state feedback control law for the system. The online optimization problem, which maximizes sparsity of the control law while minimizing an infinite-horizon performance cost, is formulated as a bilinear matrix inequality (BMI) problem; subsequently, it is then relaxed to a linear matrix inequality (LMI) problem, and (i) convergence to a solution as well as (ii) that early termination guarantees a feasible (but suboptimal) control law are proved. Stability of the closed-loop system under what is a switched control law is guaranteed by the inclusion of dwell-time constraints in the LMI problem. Finally, the efficacy of the approach is demonstrated through numerical simulation examples.

KEYWORDS
Distributed Control; Optimal Control; Optimization; Semi-definite Programming

1. Introduction

Decentralized control has attracted attention since the 1960s (Lunze, 1992). In non-centralized architectures each subsystem is equipped with a local controller that may or may not share information with the other local controllers. Distributing the control actions brings different advantages, for instance it allows to decrease

Corresponding author Romain Guicherd. Email: r.guicherd@sheffield.ac.uk

the weight of a system and also provides an increase in its modularity. Addressing this control challenge has been done with multiple techniques , most of them assume that the control system architecture is known a priori. Therefore, the first problem to tackle is often to find the control system architecture (Scattolini, 2009). This task is a complex problem in itself and can be achieved by minimizing the dynamical couplings between subsystems (Bristol, 1966; Guicherd, Trodden, Mills, & Kadirkamanathan, 2017; Kariwala, Forbes, & Meadows, 2003). Another common technique consists of finding the input and output pairs with the highest sensitivity (Manousiouthakis, Savage, & Arkun, 1986). In the case of distributed control architectures, a communication scheme has to be selected and implemented (Gross & Stursberg, 2016; Maestre, Muñoz de la Peña, & Camacho, 2009). Simply broadcasting the states between all the local controllers at every time step will increase the overall performance of the control system but this will be achieved at a high communication cost (Ye, Heidemann, & Estrin, 2002). Consequently, efficient communication strategies can prove vital to preserve the battery life of wireless sensors and actuators. This problem has been addressed with a coalitional communication scheme based on game theory, for systems coupled through their inputs, after the system partitioning is performed (Maestre, Muñoz de la Peña, Jiménez Losada, Algaba, & Camacho, 2014). In this framework, control laws with distinct topologies are computed offline, based on structural constraints added to the Riccati equation solutions. Online, the control algorithm selects the most appropriate control mode, providing the best combined control and communication cost. Even though the number of controller topologies is finite, the computation of the control gains adapted to the system dynamics and performance can only be conducted as a function of the current system states. Hence, the combined topology and control design has to be the result of an online synthesis. Another shortcoming of the offline computation approaches comes from the large amount of control gains, growing combinatorially with the system size, that have to be computed, stored and then compared. One of the major challenges for distributed control systems is to solve online the joint optimization problem of balancing the communication effort between local subsystem controllers along with system-wide performance. This combined optimization of control gains and communication topologies requires the use of structured control synthesis methods as well as sparsity inducing techniques.

Structured control synthesis methods have emerged to tackle the co-design of control architecture and control law. Some structurally constrained control synthesis approaches have focused on the implementation of fully decentralized controllers (Bakule, 2008; Šiljak, 1991). The synthesis of decentralized control laws has become an important research topic, especially for large-scale systems (Lunze, 1992). Decentralized control systems answer the needs for decreasing the shared information between the subsystems, and therefore the number of communication links between local controllers. This guarantees that the subsystems do not broadcast their states, but it does not account for potential dynamic couplings between them. Following the work on decentralized control design, more research works have addressed the problem of structurally constrained control law synthesis. Modifying the sparsity structure of control laws directly affects the use of subsystem to subsystem communication, actuator or sensor usage.

It is well known that in most cases, the computation of structurally constrained controllers is a difficult problem, only some particular systems complying with a quadratic invariance condition are tractable and can be formulated as convex optimization problems (Rotkowitz & Lall, 2006). Some approaches have performed the optimal design of partially decentralized controller in a framework such that the feedback gain obtained is as sparse as possible, while keeping acceptable $\mathcal{H}_\infty$ performance (Schuler, Münz, & Allgöwer, 2014), or $\mathcal{H}_2$ performance (Babazadeh & Nobakhti, 2017), when compared with centralized controllers. These works have studied offline control synthesis, without considering the direct impact on system performance based on current state variables. Most of them are based on compressed sensing techniques developed to deal with sparsity (Candès & Tao, 2005). For instance, common convex relaxations of the l_0-norm, such as the l_1-norm (Candès, 2008), or reweighed l_1-norm are frequently implemented as sparsity inducing costs (Candès, Wakin, & Boyd, 2008). Other approaches have focused on row or column sparsity, in order to minimize respectively the number of sensors and actuators in use (Polyak, Khlebnikov, & Shcherbakov, 2013), whilst others have managed to induce a desired block structure on the feedback gain using sufficient constraint conditions (Crusius & Trofino, 1999). Finally, other works have considered a path-following technique in order to solve the bilinear matrix inequality (BMI) problems resulting from the synthesis of structured or low-authority control laws (Hassibi, How, & Boyd, 1999a, 1999b), once again adopting an offline design strategy. The joint optimization of control law and communication topology has only been implemented based on control laws or communication topologies designed offline. To the best of the authors' knowledge this work presents the first approach to the joint online control law and communication topology optimization.

This paper provides a solution to the joint online control problem of communication and performance optimization. A control technique where a supervisory agent recomputes online a state feedback control law in order to minimize a cost combining the infinite-horizon control cost with a subsystem to subsystem communication metric is proposed. The control law is updated online based on the values of the state variables, and then broadcast to the local controllers for implementation. The novelty of this approach resides in optimizing online both the distributed control law gains and the communication topology, in a single tractable problem. While this is achieved using standard techniques, it does result in a bilinear matrix inequality problem that is challenging to solve and a time and structure varying control law that poses difficulties for establishing stability. Such an online control strategy offers a trade-off between the system predicted performance and the use of subsystem to subsystem communication. Thus, the main application of this type of control method would be to provide steady-state disturbance rejection relying on communication channels only to tackle efficiently the disturbances.

The work presented in this paper shows how to compute a new linear state feedback gain optimally with regards to the predicted control performance cost as well as a communication metric. The optimization problem is formulated as a semi-definite programming (SDP) problem including a bilinear matrix equality. This optimization problem is then relaxed using an alternate convex search method, also convergence of such a technique for this type of problem is proved. A feasible stable solution can always be computed, even after early termination of the algorithm. In addition, the stability of the closed-loop system is guaranteed based on a dwell time requirements for switching between the different control modes generated. Finally, recursive feasibility of the supervised control algorithm is demonstrated.

The paper is organized as follows, section 2 states the problem and section 3 introduces the required notations for the constraints and objective function. Section 4 demonstrates the stability and recursive feasibility of the distributed control technique. In section 5, the control algorithm is presented along with a proof of its convergence with the possibility of generating a feasible solution even after early termination, and sufficient conditions to include system constraints. In order to illustrate the efficacy of the control algorithm section 6 includes numerical examples. Finally, section 7 concludes the paper.

Notation: For $(a, b) \in \mathbb{N}^2$ such that $a < b$, the set $[\![a, b]\!]$ defines the set containing the integers from a to b included. The operator $|\cdot|$ is used to denote the magnitude of a complex number. The operator $\|\cdot\|_p$ with $p \in \{0, 1, 2\}$ defines the l_p-norm for real vectors and matrices. For a set $\mathbb{N}$, the notation $\mathbb{N}^*$ defines $\mathbb{N} \setminus \{0\}$. $\mathbb{R}$ and $\mathbb{R}_+$ respectively denotes the set of real and non-negative real numbers. The superscript $^\top$ represents the transpose of a vector or matrix. A matrix $B \in \mathbb{R}^{n\times m}$ will be noted $(b_{ij})_{(i,j)\in[\![1,n]\!]\times[\![1,m]\!]}$ and $(B_{kl})_{(k,l)\in[\![1,N]\!]\times[\![1,M]\!]}$, respectively for the element and block notations, with N row blocks and M column blocks. The operator $\mathrm{vec}(\cdot)$ denotes the vectorization for matrices, and the notation $\mathrm{diag}(\cdot)$ denotes the block diagonal matrix formed from the arguments. The sets of symmetric positive semi-definite and symmetric positive definite matrices of size $n \in \mathbb{N}^*$ will be noted $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ respectively. For conciseness, the zero matrix of appropriate size will simply be noted 0, and the symbol $*$ will denote the symmetric matrix block when used in a symmetric matrix. For $n \in \mathbb{N}^*$, the matrix I_n represents the identity matrix of dimension n. The generalized order on the positive semi-definite cone will be denoted by $\succ$ and $\succeq$, respectively for the strict and non-strict inequalities. For all $(A, B) \in (\mathbb{R}^{n\times m})^2$, the notation $A \circ B$ defines the Hadamard product of A and B.

2. Problem Statement

2.1. Distributed System Dynamics

Consider a discrete time, linear, time invariant system (1a), it is assumed throughout this paper that a decomposition of this system into subsystems exists. Such a decomposition could be achieved using relative gain array techniques (Bristol, 1966), other interactions based system partitioning approaches (Guicherd et al., 2017; Kariwala et al., 2003) or is readily available when the system is built as a concatenation of stabilizable subsystems. Therefore, there exists an integer $N \in \mathbb{N}^*$ representing the number of subsystems composing the system (1a), such that, for all $p \in [\![1, N]\!]$ the subsystem indexed by p is modelled as per equation (1b).

$$x^+ = Ax + Bu \tag{1a}$$

$$x_p^+ = A_{pp}x_p + B_{pp}u_p + \sum_{\substack{j=1\\j\neq p}}^{N} \left\{A_{pj}x_j + B_{pj}u_j\right\} \tag{1b}$$

with, for all $p \in [\![1, N]\!]$, $x_p \in \mathbb{R}^{n_p}$ and $u_p \in \mathbb{R}^{m_p}$. The vectors x^+ and x_p^+ respectively denote the successor system and subsystem state variables. A_{pj} and B_{pj} as well as A_{pp} and B_{pp} are matrices of appropriate dimensions, such that

$$\sum_{p=1}^{N} n_p \geq n \tag{2a}$$

$$\sum_{p=1}^{N} m_p = m \tag{2b}$$

where, $n \in \mathbb{N}^*$ and $m \in \mathbb{N}^*$. The variables $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are respectively the state and input variables of the system (1a) with A and B matrices of appropriate dimensions. Dynamic couplings are permitted to exist between the different subsytems and are represented by the right-hand side sum in equation (1b). The equation (2a) defines a possible overlapping condition for the subsystem state variables, whereas the equation (2b) denotes a non-overlapping condition for the subsystem input variables. In other words, a state variable and an input can respectively be shared by multiple subsystems or belongs to a unique subsystem. The following assumption is made with regards to the system (1a) and the subsystems defined as per equation (1b),

Assumption 1. *The system (A, B) as well as, for all $p \in [\![1, N]\!]$ the subsystems (A_{pp}, B_{pp}) are stabilizable.*

Assumption 1 implies the existence of a linear stabilizing state feedback controller $F \in \mathbb{R}^{m\times n}$. Also, this assumption guarantees that there exists a controller F_{pp} stabilizing for the subsystem (A_{pp}, B_{pp}), for any value of subsystem index p. However, this does not imply that the block diagonal controller $F = \mathrm{diag}(F_{11}, \ldots, F_{NN})$ stabilizes the system (A, B) for every stabilizing choice $(F_{pp})_{p\in[\![1,N]\!]}$.

2.2. Control Problem

The aim of this paper is to synthesize a state feedback control law $u(k) = f(x(k))$, minimizing simultaneously the classical linear quadratic regulator (LQR) cost for the system as well as a subsystem to subsystem communication metric, defined within the next section. Without loss of generality, the system model (1a) can be decomposed into subsystem models using the following block decomposition, $A = (A_{ij})_{(i,j)\in[\![1,N]\!]^2}$ and $B = (B_{ij})_{(i,j)\in[\![1,N]\!]^2}$ where the blocks correspond to the subsystem models given in equation (1b). This decomposition is always achievable throughout the reorganization of the system state and input variables. When the control law is linear defined by $u(k) = Fx(k)$, then the block matrices composing the matrix F represent linear feedback gains between a subsystem state variable and a subsystem input variable. More specifically, diagonal blocks represent feedback gains within a subsystem, whereas off-diagonal blocks represent gains between two distinct subsystems. Hence, the sparsity structure of F defines the system communication topology. Similarly, F can be decomposed into N^2 block matrices as follows,

$$\forall (i, j) \in [\![1, N]\!]^2,\ F_{ij} \in \mathbb{R}^{m_i\times n_j},\ F = (F_{ij})_{(i,j)\in[\![1,N]\!]^2}. \tag{3}$$

The simplest distributed system architecture is a system composed of two subsystems. This simple case is presented in Figure 1, it illustrates that the control law F

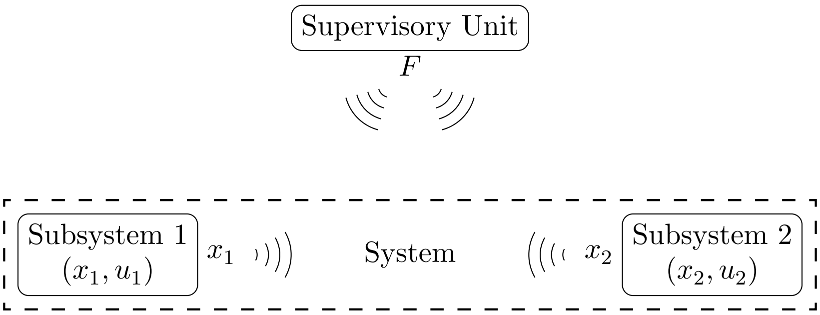

Figure 1. Representation of the information exchanged between the subsystems of a system composed of two subsystems and a supervisory unit.

not only defines the performance of the system, but also the communication structure between the subsystems. Indeed, decomposing F into four block matrices as per equation (3) yields the following relations,

$$\begin{cases} u_1 = F_{11}x_1 + F_{12}x_2 \\ u_2 = F_{21}x_1 + F_{22}x_2, \end{cases} \tag{4}$$

where the system state and input variables are decomposed into two simple subsystems, respectively with $x^\top = \begin{bmatrix} x_1^\top & x_2^\top \end{bmatrix}$ and $u^\top = \begin{bmatrix} u_1^\top & u_2^\top \end{bmatrix}$. Subsequently, the block matrices F_{21} and F_{12} respectively represents the need for communication from the subsystem 1 to the subsystem 2 and from the subsystem 2 to the subsystem 1. More generally, if the block matrix F_{ij} is different from the zero matrix, it implies communication from the subsystem j to the subsystem i. The next section formulates the communication metric as well as the approach taken.

3. Proposed Approach

The cost function formulated in this section includes a part linked to the predicted control performance, representing the effort to steer the states to a reference using a linear control law F, as well as a penalty on the states discrepancy. The second part of the cost function is a metric associated with the number of subsystem to subsystem communication links in use within the control law F.

3.1. Control Cost

The infinite-horizon control quadratic cost is given by an infinite sum on the state and control input variables (5). In the linear time invariant case, when $u(k) = Fx(k)$, the control cost only depends on the initial state, the state space model and the control law (Zhou, Doyle, & Glover, 1996).

$$J_{ctrl} = \sum_{k=0}^{+\infty} \left\{ x_k^\top Q x_k + u_k^\top R u_k \right\} = x_0^\top P x_0 \tag{5}$$

where $(Q, R) \in \mathbb{S}_{+}^{n} \times \mathbb{S}_{++}^{m}$, such that the pair $(Q^{\frac{1}{2}}, A)$ is detectable. The matrices Q and R represent the weighting matrices used in order to penalize adequately, respectively the state and input discrepancies. The vectors x_k and u_k represent the system state and input variables at time step k, and the matrix $P \in \mathbb{S}_{++}^{n}$ is the unique positive definite solution of the discrete time Lyapunov equation

$$(A + BF)^{\top} P(A + BF) - P + Q + F^{\top} RF = 0. \tag{6}$$

In particular, when the control law is designed as the optimal LQR gain F that minimizes the cost (5), F is then state invariant and fully centralized in general. The system (1a) in closed-loop control is represented as follows,

$$x^{+} = (A + BF)x. \tag{7}$$

3.2. Communication Cost

The cost linked to the communication is associated to the sparseness of the off-diagonal blocks of the state feedback gain F. The corresponding entries of the feedback gain matrix rely on subsystem communication. However, this communication metric does not evaluate the communication exchange information rate and only represents the need for a communication link. Previously, a study of the effect of the bit rate allocation has been performed (Xiao, Johansson, Hindi, Boyd, & Goldsmith, 2003). The communication cost promoting sparsity employs the l_0-norm of the vectorized form of the feedback gain such that

$$J_{comm} = \left\| \operatorname{vec}(W \circ F) \right\|_0 \tag{8}$$

where $W \in \mathbb{R}^{m \times n}$ is a binary matrix used to select the entries of F to penalize. Sparseness of a vector or a matrix is penalized using the l_0-norm which associates a binary element to each of the entries based on their value. The l_0-norm corresponds to the number of non-zero entries in a matrix or vector. Finding sparse solutions represents a combinatorial problem and is in general NP-hard, usually requiring exhaustive search with exponential complexity (Blondel & Tsitsiklis, 1997). Since the l_0-norm does not comply with all the norm properties (Hurley & Rickard, 2009), and is therefore not a true norm, a common convex relaxation is to use the l_1-norm. Indeed, the l_1-norm is a well known sparsity promoting penalty. This heuristic has been used in the past in different fields such as optimal control, compressed sensing and signal reconstruction (Candès & Tao, 2005). Necessary and sufficient conditions have been established for this convex relaxation to be tight (Candès, 2008). Therefore, as it has been done in previous research works, sparsity is promoted using the following l_1-norm as a convex relaxation.

$$J_{comm} = \left\| \operatorname{vec}(W \circ F) \right\|_1 \tag{9}$$

The next subsection presents the optimization problem to be solved recursively in order to determine what communication links are needed based on the system predicted performance.

3.3. Minimization of the Control and Communication Costs

In this subsection, the optimization problem dealing with the infinite-horizon control cost and the communication cost is formulated using linear matrix inequality (LMI) constraints as well as a bilinear matrix equality. The objective is to minimize the convex performance index taking into account both the control and communication costs given in (10).

$$\begin{aligned} J &= \lambda J_{ctrl} + (1-\lambda) J_{comm} && \text{(10a)} \\ \Leftrightarrow J &= \lambda x_0^\top P x_0 + (1-\lambda) \big\| \operatorname{vec}(W \circ F) \big\|_1 && \text{(10b)} \end{aligned}$$

where $\lambda \in (0,1]$ is a design tuning parameter implemented in order to balance the control (5) and communication (9) objectives. The constraints formulated for the optimization problem are LMI constraints as per the optimization problem (11). These constraints are equivalent to the discrete time algebraic Riccati equation and are used to minimize an upper bound on the infinite-horizon control cost (Kothare, Balakrishnan, & Morari, 1996). A bilinear matrix equality is added to the LMI constraint set so that the sparsity promoting objective can be expressed within the cost function. The optimization problem (11) presented below articulates the trade-off between the control and communication costs.

$$\begin{aligned} \underset{\rho,F,X,K}{\text{minimize}} \quad & \lambda\rho + (1-\lambda)\big\| \operatorname{vec}(W \circ F) \big\|_1 \\ \text{subject to} \quad & \begin{bmatrix} X & * & * & * \\ AX+BK & X & * & * \\ Q^{\frac{1}{2}}X & 0 & \rho I_n & * \\ R^{\frac{1}{2}}K & 0 & 0 & \rho I_m \end{bmatrix} \succeq 0 \\ & \begin{bmatrix} 1 & x_0^\top \\ x_0 & X \end{bmatrix} \succeq 0 \\ & FX = K \end{aligned} \tag{11}$$

Remark 1. Strictly speaking the optimization variables should be indexed with the control mode index to emphasize the fact that these modes are recomputed online and periodically. However, for the sake of conciseness, these indices are used only for the asymptotic stability and recursive feasibility proofs.

Note that the optimization problem presented in (11) is biconvex. Fixing respectively the variable F or X renders a convex optimization problem. With the given parametrization, the Lyapunov solution P and the feedback gain F are expressed as follows:

$$P = \rho X^{-1} \tag{12a}$$

$$F = KX^{-1}. \tag{12b}$$

3.4. Time-varying Control Modes

When the parameter λ is set to 1, the standard LQR feedback gain is returned by the optimization as the global optimum. Consequently, the optimization problem (11) without the sparsity promoting objective is state invariant. The minimum is reached for the solution of the discrete algebraic Riccati equation. In the case of (11), different degrees of sparsity for F will affect the infinite-horizon control cost throughout the subsystem dynamic couplings. The time-varying nature of the control technique is demonstrated with numerical examples later within this paper. In the literature different techniques have been used in order to solve these types of optimization problems. For instance, one approach implemented a sequential method where a penalty on the bilinear equality gap is added to the objective (Doelman & Verhaegen, 2016). A convergence guarantee has been proven, however there is no assurance that the optimum reached will be the global optimum. Another technique called the alternate convex search uses iterative convex relaxations of the biconvex problem (Gorski, Pfeuffer, & Klamroth, 2007). The alternate convex search also known as alternate SDPs method (Fukuda & Kojima, 2001), works by alternately fixing one of the variables in the bilinear matrix constraint. The two convex optimization problems are solved alternately until a stopping criteria is met. Other techniques, relying on a branch-and-cut strategy have been developed in order to reach the global optimum value for a bilinear matrix inequality optimization problem (Fukuda & Kojima, 2001; Goh, Safonov, & Papavassilopoulos, 1995). However, the technique mentioned above requires a lot of time and computational power. The next section introduces a new set of convex constraints to ensure global stability of the switched closed-loop control system. Also, the recursive feasibility of the distributed controller is demonstrated.

4. Stability and Feasibility of the Distributed Controller

4.1. Controller Stability

Since, the distributed controller minimizes not only the infinite-horizon control cost but also the communication effort, and that no constraints are added to ensure the stability of the switched system, instability can become an issue. It is a well known fact that switching between two stabilizable controllers can trigger plant instability. For example, the system presented in (13) is stable and can be stabilized by any of the two feedback control laws given in (14). Also, the two closed-loop systems obtained by implementing the controller F_0 and F_1 comply with the constraints of the optimization problem (11), as stabilizing control laws. Nonetheless, instability can be triggered by a rapid switching between the two control laws F_0 and F_1.

$$A = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{13}$$

$$F_0 = \begin{bmatrix} 2 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}, \qquad F_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & -2 \end{bmatrix} \tag{14}$$

The instability question can be tackled by a dwell time constraint. Every time a new feedback gain is broadcast to the plant subsystems, the plant becomes a switched system (Liberzon, 2003). Consequently, it is important to ensure that switching between two consecutive feedback gains is done so that one can guarantee global asymptotic stability of the closed-loop system (Geromel & Colaneri, 2006). In order to prevent any unstable behaviour from happening, an extra LMI constraint is added to the optimization problem (11) enforcing switched asymptotic system stability by design. This stability constraint is formulated using the previous Lyapunov solution as well as the previous control law combined with a dwell time design parameter Δ, thus leading to the formulation given in (17). The control mode dwell time Δ, with $\Delta \in \mathbb{N}^*$ guarantees that instability will not be triggered during control mode switching, if the switching is performed at least after Δ time steps. Also, the dwell time constraint allows enough time to perform the online optimization. The Lemma 4.1 proves that the extra constraint added in (17) enforces system stability and the recursive feasibility of the control algorithm is proved in Lemma 4.3.

Lemma 4.1. *Consecutive feasible solutions of the distributed control optimization problem* (17) *are globally asymptotically stabilizing for the switched closed-loop system.*

Proof. The constraint added to the problem (17) corresponds to a stable switch from the previous mode l to the next mode $l+1$, after Δ time steps. This is shown in (17) by applying the Schur complement to the extra constraint, with the previous parametrization (12a) of the optimization problem (i.e. $P_{l+1} = \rho X^{-1}$).

$$\begin{bmatrix} \rho P_l & \rho(A+BF_l)^{\Delta\top} & \varepsilon\rho I_n \\ \rho(A+BF_l)^{\Delta} & X & 0 \\ \varepsilon\rho I_n & 0 & \varepsilon\rho I_n \end{bmatrix} \succeq 0 \tag{15a}$$

$$\Leftrightarrow \begin{cases} \varepsilon\rho \geq 0 \\ X \succeq 0 \\ \rho P_l - (A+BF_l)^{\Delta\top}\rho^2 X^{-1}(A+BF_l)^{\Delta} \succeq \varepsilon\rho I_n \end{cases} \tag{15b}$$

$$\Leftrightarrow \begin{cases} \varepsilon\rho \geq 0 \\ X \succeq 0 \\ (A+BF_l)^{\Delta\top}P_{l+1}(A+BF_l)^{\Delta} - P_l \preceq -\varepsilon I_n \end{cases} \tag{15c}$$

The constraint (15) added to the optimization (17) yields the following relation when pre- and post-multiplied by respectively $x_k^\top$ and x_k

$$
\begin{aligned}
& x_k^\top \left[(A+BF_l)^{\Delta\top} P_{l+1}(A+BF_l)^\Delta - P_l\right] x_k < 0 && \text{(16a)}\\
\Leftrightarrow\ & \left[(A+BF_l)^\Delta x_k\right]^\top P_{l+1}(A+BF_l)^\Delta x_k < x_k^\top P_l x_k && \text{(16b)}\\
\Leftrightarrow\ & x_{k+\Delta}^\top P_{l+1} x_{k+\Delta} < x_k^\top P_l x_k && \text{(16c)}
\end{aligned}
$$

where x_k is the system full state vector at time step k, distinct from the zero vector. Therefore, this ensures a strict decrease of the Lyapunov function for the closed-loop switched system, implying global asymptotic stability. □

$$
\begin{aligned}
\underset{\rho,F,X,K}{\text{minimize}} \quad & \lambda\rho + (1-\lambda)\big\|\operatorname{vec}(W \circ F)\big\|_1 \\
\text{subject to} \quad & \begin{bmatrix} X & * & * & * \\ AX+BK & X & * & * \\ Q^{\frac{1}{2}}X & 0 & \rho I_n & * \\ R^{\frac{1}{2}}K & 0 & 0 & \rho I_m \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} \rho P_l & * & * \\ \rho(A+BF_l)^\Delta & X & * \\ \varepsilon\rho I_n & 0 & \varepsilon\rho I_n \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} 1 & x_0^\top \\ x_0 & X \end{bmatrix} \succeq 0 \\
& FX = K
\end{aligned} \tag{17}
$$

Remark 2. In order to guarantee a strict decrease of the switched system Lyapunov functions after Δ time steps, the equations (15) must be strict inequalities. However, for a practical numerical implementation, the strict inequality is replaced by an inequality with a given ε precision, where $\varepsilon \in \mathbb{R}_+^*$ is a small but strictly positive constant.

Proposition 4.2. *For a given stable control mode $l \in \mathbb{N}$, the set*

$$
\Pi_l^\Delta = \left\{(\rho, X) \,\middle|\, \begin{bmatrix} \rho P_l & \rho(A+BF_l)^{\Delta\top} \\ \rho(A+BF_l)^\Delta & X \end{bmatrix} \succ 0 \right\}
$$

is non-decreasing with $\Delta \in \mathbb{N}^*$, in the set inclusion sense, i.e. $\Pi_l^\Delta \subseteq \Pi_l^{\Delta+1}$.

Proof. For a given stable control mode $l \in \mathbb{N}$ defined by a feedback gain F_l as well as a Lyapunov function P_l, the following set inclusion arises

$$
\begin{aligned}
& \forall \Delta \in \mathbb{N}^*, \\
& (\rho, X) \in \Pi_l^\Delta \Leftrightarrow (A+BF_l)^{\Delta\top} P_{l+1}(A+BF_l)^\Delta \prec P_l \\
& (\rho, X) \in \Pi_l^\Delta \Rightarrow \left[(A+BF_l)^{\Delta+1}\right]^\top P_{l+1}(A+BF_l)^{\Delta+1} \preceq (A+BF_l)^\top P_l (A+BF_l) \\
& (\rho, X) \in \Pi_l^\Delta \Rightarrow \left[(A+BF_l)^{\Delta+1}\right]^\top P_{l+1}(A+BF_l)^{\Delta+1} \prec P_l \\
& (\rho, X) \in \Pi_l^\Delta \Rightarrow (\rho, X) \in \Pi_l^{\Delta+1}.
\end{aligned}
$$

Therefore, for all $\Delta \in \mathbb{N}^*$ and for all stable control mode $l \in \mathbb{N}$, the set of stable control laws under switching is non-decreasing with Δ, $\Pi_l^{\Delta} \subseteq \Pi_l^{\Delta+1}$. $\square$

The dwell time Δ is a designer tuning parameter. It will condition the time allocated to the optimization algorithm as well as the maximum control feedback gain refreshing rate. Therefore, if a solution exists it will make the closed-loop system globally asymptotically stable. Subsequently, the next logical step is to prove that the optimization problem stays recursively feasible. The recursive feasibility property of the control algorithm is presented in the next subsection.

4.2. Controller Feasibility

The global asymptomatic stability of the controller is given by the previous LMI constraint in (11). Therefore, it is important to make sure that this new set of constraints will not trigger infeasibility of the optimization problem (17).

Lemma 4.3. *If there exists an initial asymptotically stabilizing control law for the system* (1) *then the optimization problem* (17) *is recursively feasible.*

Proof. Consider the system given by the state space model (1). Due to system stabilizability, an initial stabilizing state feedback controller F_0 can be designed. This initial control mode can be computed by solving the discrete algebraic Riccati equation. In this case, F_0 is obtained by minimizing the infinite-horizon quadratic cost. Consequently, there exists at least one feasible asymptotically stabilizing mode F_l, with $l \in \mathbb{N}$, for the system (1). Therefore, the control mode l must be feasible for the optimization time step $l+1$ with the lowest possible dwell time of 1. The control law F_l along with the Lyapunov solution P_l are feasible candidate solutions for the next optimization time step. Thus, even if the control mode F_l is not optimal for the optimization step $l+1$, it still constitutes a feasible solution. Hence, by induction, the optimization problem (17) remains recursively feasible. $\square$

The Theorem 4.4 summarizes the previous results about the optimization problem (17) when applied to system (1).

Theorem 4.4. *If there exists an initial asymptotically stabilizing control law for the system* (1)*, then the optimization problem* (17) *is recursively feasible and its consecutive solutions are globally asymptotically stabilizing for the switched closed-loop system.*

Proof. The proof is straight forward, and follows from Lemma 4.1 and Lemma 4.3. $\square$

Following the discussions on the stability and recursive feasibility of the distributed controller, the control system algorithm will be presented. The next section will explain how the algorithm proceeds and presents a sufficient way to include physical system constraints. Also, a proof of convergence of the control algorithm is given.

5. Supervised-distributed Control Algorithm

5.1. Unconstrained Supervised-distributed Controller

The optimization problem (17) yielding the distributed control laws is non-convex. Therefore, it is difficult to find a global optimum or even to verify that a certain value is a global optimum. However, since the problem is biconvex, including a bilinear matrix equality, known techniques exist to reach a partial local optimum (Wendell & Hurter, 1976). For instance, the alternate convex search technique is used to relax the problem into two well defined convex optimization sub-problems (Boyd & Vandenberghe, 2010). This technique is applied to the optimization problem (17) in Algorithm 1. According to Theorem 5.1, the biconvex structure of this optimization problem, makes it possible to ensure convergence of the alternate convex search.

Theorem 5.1. *The sequence of solutions generated by Algorithm 1 converges monotonically to a stationary point that is a partial local minimum of problem* (17).

Proof. According to Lemma 4.3, the optimization problem (17) is recursively feasible, thus, feasible in the sets of decision variables (ρ, F) and (ρ, X). The optimization problems obtained with these sets of decision variables are convex and the objective function is positive as a convex combination of two positive objectives. Finally, the sequence of solutions obtained by solving the optimization problem (17) whilst fixing alternately the variables X and F is monotonically non-increasing. Subsequently, the sequence of solutions converges monotonically to a stationary point that constitutes a partial local optimum of the optimization problem (17) (Gorski et al., 2007; Wendell & Hurter, 1976). □

The optimization problem (17) is denoted by $\mathcal{P}_{x_0,X}(\rho, F)$ and $\mathcal{P}_{x_0,F}(\rho, X)$ respectively when (i) (ρ, F) are decision variables and (x_0, X) are fixed parameters, (ii) (ρ, X) are decision variables and (x_0, F) are fixed parameters. The supervised-distributed controller (SDC) in Algorithm 1, relies on the predicted state variables after Δ time steps under the previous control mode, in order to generate the next control mode by alternate convex search.

One can notice that the order of the alternate convex search can be changed, which means that X_q can be fixed first, before performing the optimization with F_{q+1}. Indeed, the control mode l has been shown to be a suboptimal but feasible solution of the succeeding optimization problem, at time step $l + 1$. Therefore, any of the previous decision variables can be used to start the alternate convex search.

As presented in Figure 2, at time step k the control law F_l is received by the subsystems and used for the next Δ time steps. In exchange, the subsystems broadcast their state variables so the optimization can be performed in order to compute the next control mode F_{l+1}. The optimization problem is solved based on the current control mode as well as a prediction of the system state $\overline{x}_{k+\Delta|k}$ after Δ time steps. The next control law is obtained when one of the convergence stopping criteria is met. Although, the optimization can be terminated if the alternate convex search had not converged within the allocated time, yielding a feasible but suboptimal solution. The next step is to broadcast the solution to the subsystems so that the control law F_{l+1} can be implemented for the following Δ time steps. Then, this process is repeated in a receding horizon fashion.

Corollary 5.2. *A feasible asymptotically stabilizable solution of problem* (17) *can be*

Algorithm 1: SDC algorithm

Inputs : x_k, F_l, P_l
Outputs : F_{l+1}, P_{l+1}
Parameters : Δ, ε_F, ε_X
Initialization: $q = 0$, $F_0 = F_l$
Compute state prediction: $\overline{x}_{k+\Delta|k} = (A + BF_l)^{\Delta} x_k$
while $\|F_{q+1} - F_q\|_2 > \varepsilon_F$ or $\|X_{q+1} - X_q\|_2 > \varepsilon_X$
do
  Solve optimization problem: $\mathcal{P}_{\overline{x}_{k+\Delta|k},F_q}(\rho, X)$,
  $(\rho, X_{q+1}) = \text{argmin}(\mathcal{P}_{\overline{x}_{k+\Delta|k},F_q}(\rho, X))$.
  Solve optimization problem: $\mathcal{P}_{\overline{x}_{k+\Delta|k},X_{q+1}}(\rho, F)$,
  $(\rho, F_{q+1}) = \text{argmin}(\mathcal{P}_{\overline{x}_{k+\Delta|k},X_{q+1}}(\rho, F))$.
  $q = q + 1$
end
return
$F_{l+1} = F_q$
$P_{l+1} = \rho X_q^{-1}$

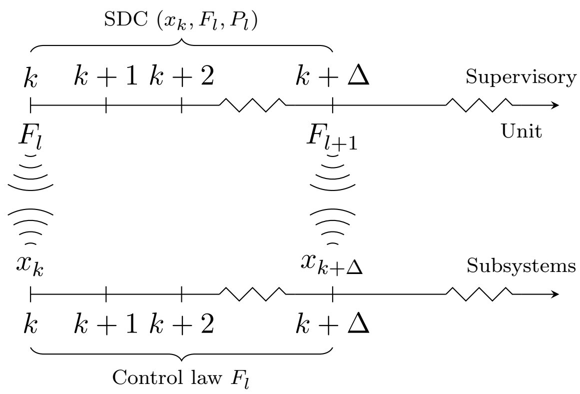

Figure 2. Supervised-distributed controller time line.

obtained by terminating Algorithm 1 after any iteration.

Proof. According to Theorem 4.4, the optimization problem (17) stays recursively feasible and a suboptimal solution $(F_0,\, X_0)$ is known at every time step from the previous optimization solution. Subsequently, for all $q \in \mathbb{N}$, Algorithm 1 uses a feasible solution $(F_q,\, X_q)$ in order to solve alternately the two following convex SDP problems, $\mathcal{P}_{x_0,F_q}(\rho, X)$ which then generates $(F_q,\, X_{q+1})$ and $\mathcal{P}_{x_0,X_{q+1}}(\rho, F)$ which generates $(F_{q+1},\, X_{q+1})$. Hence, by induction the Algorithm 1 produces a new feasible solution after every iteration, which concludes the proof. $\square$

The next subsection presents a way to introduce sufficient conditions in order for the control laws to comply with given physical system constraints.

5.2. Constrained Supervised-distributed Controller

Without changing the results presented previously, it is possible to introduce constraints on the states, the outputs as well as the inputs within the optimization problem (17). The input vector can be constrained in two ways, using sufficient LMI constraints. First, the Euclidean norm of the input vector can be bounded (19) and secondly the maximum magnitude of each input variable can be constrained (21). This technique has been studied previously (Boyd, El Ghaoui, Feron, & Balakrishnan, 1994; Kothare et al., 1996).

$$\begin{bmatrix} u_{max}^2 I_m & K \\ K^\top & X \end{bmatrix} \succeq 0 \tag{18}$$

Adding the constraint (18) to the optimization problem (17) is a sufficient condition to provide the following peak constraint,

$$\forall k \in \mathbb{N}^*,\ \|u_k\|_2 \leq u_{max}. \tag{19}$$

A peak value constraint can be implemented on each of the input vector components with the inequality (20).

$$\begin{bmatrix} U^{max} & K \\ K^\top & X \end{bmatrix} \succeq 0 \tag{20}$$

where for all $j \in [\![1, m]\!]$, $U_{jj}^{max} \leq u_{j,max}^2$. Therefore providing the following peak component constraints,

$$\forall k \in \mathbb{N}^*,\ \forall j \in [\![1, m]\!],\ \left|u_{j,k}\right| \leq u_{j,max}. \tag{21}$$

In a similar fashion, sufficient LMI conditions can be used to bound the Euclidean norm of the state or output vectors as presented in (22).

$$\begin{bmatrix} X & (AX+BK)^\top C^\top \\ C(AX+BK) & x_{max}^2 I_n \end{bmatrix} \succeq 0 \tag{22}$$

where C represents either the identity matrix or the output matrix for respectively a bound on the Euclidean norm of the states or the outputs. Similarly, by using only the appropriate row of C, sufficient conditions can be established to bound the magnitude of a single state or output variable. Hence, the following constraints can be implemented on the vector peak magnitude,

$$\forall k \in \mathbb{N}^*,\ \|x_k\|_2 \leq x_{max} \tag{23a}$$

$$\forall k \in \mathbb{N}^*,\ \|y_k\|_2 \leq y_{max}. \tag{23b}$$

The next section will provide numerical examples to demonstrate the efficacy of Algorithm 1.

6. Numerical Examples

Two examples have been used to demonstrate the trade-off offered by the optimal distributed state feedback control. These numerical examples have been solved using YALMIP (Löfberg, 2004) along with the SeDuMi solver (Sturm, 1999).

6.1. First Example

This example is a second-order plant (24), where the subsystem interactions come from both the state and input matrices. All the simulations start from the initial state $x_0^\top = \begin{bmatrix} 1 & -1 \end{bmatrix}$ and include multiple additive disturbances added at time steps 24, 26, 32 and 34 only to the state variable x_2.

$$x^+ = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} x + \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix} u \tag{24}$$

This first example has been used as a proof of concept to demonstrate two main aspects of the control technique developed within this paper. First of all, three simulations are conducted to highlight the trade-off provided between the fully connected or centralized architecture and the decentralized architecture. Secondly, changing the trade-off parameter λ shows that a Pareto front is obtained between the control and communication objectives considered, providing a tuning capability to the control engineer.

6.1.1. Control Methods Comparison

The weighting matrices Q and R used to design the control laws as well as to evaluate the control systems performance are taken equal to the identity matrices of appropriate dimensions. The system is partitioned into two subsystems each including the state and input having the same index. In order to compare the performance of the distributed controller, two benchmark controllers are designed F_C and F_D respectively the fully connected or centralized controller and the decentralized controller.

$$F_C = \begin{bmatrix} -0.8708 & -0.6210 \\ 0.6069 & -0.5330 \end{bmatrix} \tag{25a}$$

$$F_D = \begin{bmatrix} -0.7551 & 0 \\ 0 & -0.9078 \end{bmatrix} \tag{25b}$$

The spectral radii of the autonomous plants with these two controllers are respectively equal to 0.4012 and 0.8803.

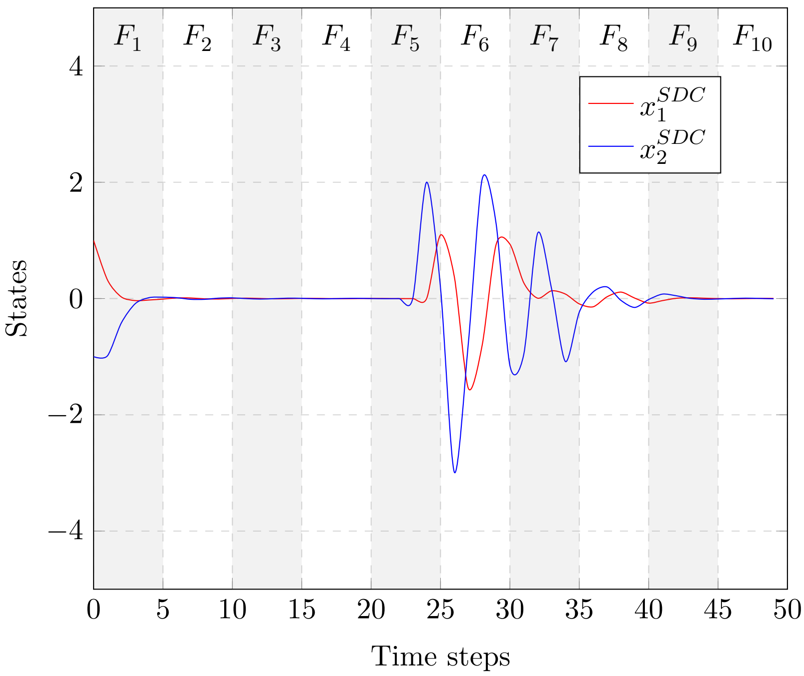

Figure 3. Supervised-distributed controller applied to the small-scale system (24).

Figure 3 shows the behaviour of the plant when the supervised-distributed controller is implemented. The supervisory unit computes a trade-off between the communication and the system performance infinite-horizon cost. In this example, a dwell time parameter of five has been implemented, the parameter lambda has been set to 0.99 and the maximum number of iterations has been limited to ten. Therefore, every five time steps, a new control law is computed and sent to the local controllers in order to be implemented for the next five time steps. The supervisory unit is initialized with the centralized controller and as soon as the predicted system performance change, the supervisory unit switches off communication channels accordingly. This translates directly by the broadcast of an updated state feedback controller, composed of new gain entries and topology. However, when some disturbance pushes the system out of its steady state value, the cost balance changes and the next control laws computed become centralized again. As it can be seen in Table 1, the structure of the control law is optimized in real time and allows to rely on communication only to improve the disturbance rejection performance. However, because of the precision of the SDP solver, the values of the off-diagonal elements are not strictly zero, therefore some threshold has to be applied before a communication channel can be completely switched off.

Table 1. State feedback gains computed by the supervised-distributed controller in Figure 3.

State feedback gains				
$\mathrm{vec}(F_1)^\top$	$[-0.8708$	$+0.6069$	-0.6210	$-0.5330]$
$\mathrm{vec}(F_2)^\top$	$[-0.8083$	$+0.0374$	$+0.0000$	$-0.9484]$
$\mathrm{vec}(F_3)^\top$	$[-0.7551$	$+0.0000$	$+0.0000$	$-0.9078]$
$\mathrm{vec}(F_4)^\top$	$[-0.7551$	$+0.0000$	$+0.0000$	$-0.9078]$
$\mathrm{vec}(F_5)^\top$	$[-0.7551$	$+0.0000$	$+0.0000$	$-0.9078]$
$\mathrm{vec}(F_6)^\top$	$[-0.7551$	$+0.0000$	$+0.0000$	$-0.9078]$
$\mathrm{vec}(F_7)^\top$	$[-0.8687$	$+0.6015$	-0.6170	$-0.5348]$
$\mathrm{vec}(F_8)^\top$	$[-0.8108$	$+0.0397$	-0.0008	$-0.9497]$
$\mathrm{vec}(F_9)^\top$	$[-0.8028$	$+0.3953$	$+0.4510$	$-0.6194]$
$\mathrm{vec}(F_{10})^\top$	$[-0.7551$	$+0.0000$	$+0.0000$	$-0.9078]$

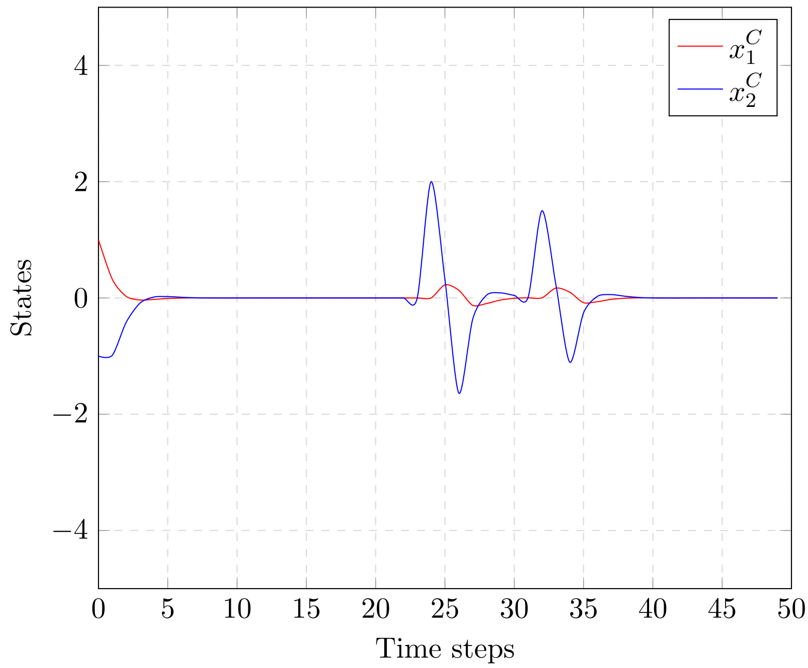

Figure 4. Centralized or fully connected controller applied to the small-scale system (24).

Figure 4 presents the response to the second-order system controlled with the centralized state feedback controller F_C. This control law relies on all the available system communication, however it offers the best system performance and disturbance rejection characteristics as it can be seen in Table 2.

Finally, Figure 5 shows the response of the system when the decentralized control law F_D is implemented. This automatic control system does not rely on communication but offers the worst performance and disturbance rejection capabilities when compared

Table 2. Comparison of the cumulative control and communication costs for the different control methods applied to the system (24).

Costs	C-LQR	SDC	D-LQR
Control	23.1	58.8	110.9
Communication	61.4	16.8	0
Total	84.5	75.6	110.9

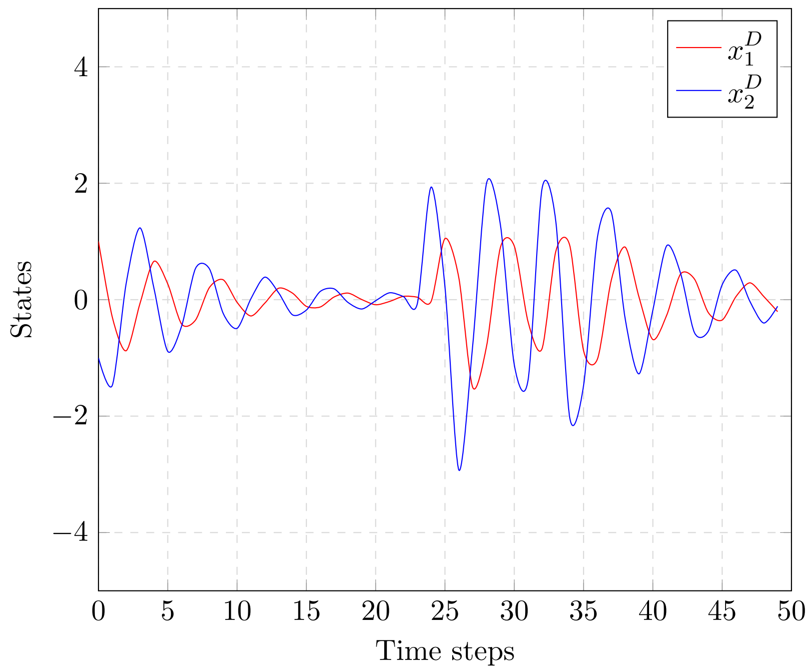

Figure 5. Decentralized controller applied to the small-scale system (24).

to the two other system behaviours. The cumulative costs regarding the control and communication metrics for the different control techniques are gathered in Table 2.

6.1.2. Control and Communication Trade-off

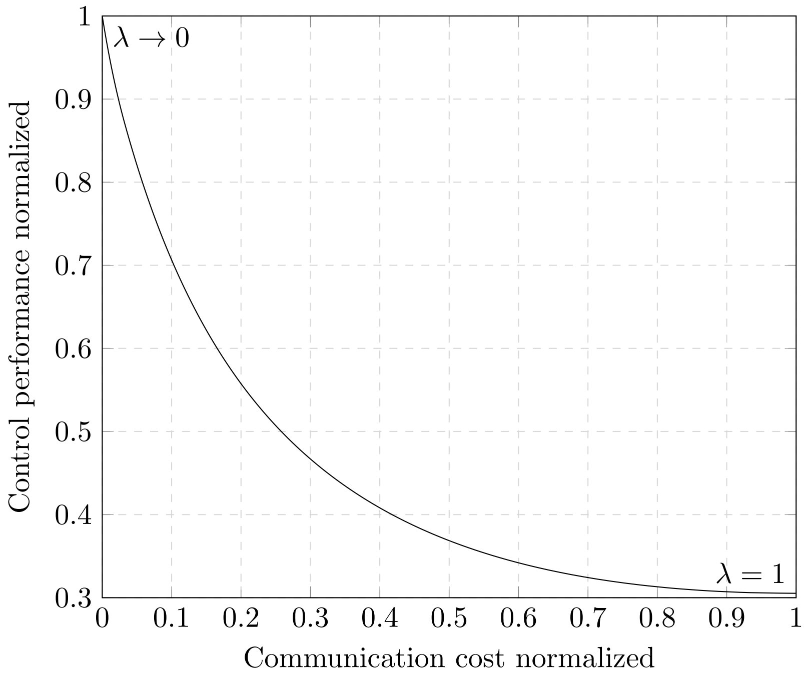

Figure 6. Trade-off between control performance and communication cost for the small-scale system (24).

The supervised-distributed controller offers a trade-off between the system control performance and the communication costs and performs better with regards to these combined objectives. The trade-off between the communication and control objectives is achieved via the parameter λ. When λ is set to one, the fully centralized control is yielded as a solution, when λ is close to zero, more decentralized solutions are obtained. Figure 6 presents the trade-off curve when the parameter λ is changed from 0.85 to 1, using the same parameters as previously, without any disturbance. For this example, the control laws will be fully decentralized on the left of the graph, with partial communication in the middle of the graph and fully centralized on the right.

6.2. Second Example

The example used within this subsection is inspired by the benchmark system used for robust control design (Wie & Bernstein, 1992). This mechanical system is composed of three carts where each cart is linked to the next one by a spring as presented in Figure 7. The states of the system are the position and velocity of each cart and the inputs are the forces applied to them. Each subsystem consists of a cart, the subsystem state variables are the cart position and velocity and the subsystem input variable is the force applied on the cart. The system is discretized using Euler's first-order approximation for the derivatives with a sample time of $0.1s$. The masses of the carts and the spring constants K are equal to 0.1 and 1 respectively with the appropriate units, the discrete-time linear state space model is given in equation (26). The simulations conducted aims at steering the carts to the origin. The design of the control laws has been completed with the weighting matrices $Q = \mathrm{diag}(10, 1, 10, 1, 10, 1)$ and $R = I_3$. The parameters used for the supervised-distributed controller are a dwell time of five time steps, a lambda value of $1 - 10^{-4}$ and a maximum number of five iterations. This example has been used to show the impact of different disturbance magnitudes on the system as well as to study the effect of varying the system model dynamics.

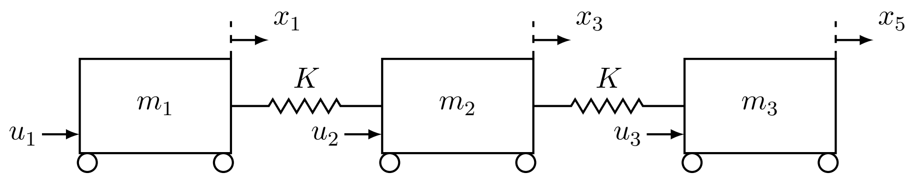

Figure 7. Three-mass-spring system.

$$x^+ = \begin{bmatrix} 1 & \frac{1}{10} & 0 & 0 & 0 & 0 \\ -\frac{K}{10m_1} & 1 & \frac{K}{10m_1} & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{10} & 0 & 0 \\ \frac{K}{10m_2} & 0 & -\frac{2K}{10m_2} & 1 & \frac{K}{10m_2} & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{10} \\ 0 & 0 & \frac{K}{10m_3} & 0 & -\frac{K}{10m_3} & 1 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{10m_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{10m_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{10m_3} \end{bmatrix} u \tag{26}$$

6.2.1. Impact of Disturbance Magnitude

Each cart can transmit its position and velocity to one of the other two carts, therefore, the total number of possible communication topologies is $2^{12} = 4096$. More generally, a similar mechanical system with N carts will have a number of $2^{2N(N-1)}$ distinct communication topologies. Therefore, any offline strategy will have to synthesize, store and compare the control laws for such a combinatorial set. The simulation is conducted from the initial state $x_0^\top = \begin{bmatrix} 10 & 1 & 10 & 1 & 10 & 1 \end{bmatrix}$ and an additive disturbance is injected at time step 50 in cart 1 velocity. The Figure 8 presents the amount of communications in use within the control modes computed by the supervised-distributed controller for different disturbance values. The main trend is that an increase in disturbance magnitude also increases the total number of subsystem communications used throughout the simulation. In general, the disturbance takes some time to propagate from cart 1 to cart 3 using the springs. Consequently, the communication channels are switched on after an extra delay of Δ time steps for small enough disturbances. Under a certain threshold, the impact of the disturbance on the system performance is too small to trigger any subsystem communication and the disturbance is directly tackled by the decentralized controller. The communication links between the different carts are indexed from 1 to 12, where the odd indices represent the position of a cart and the even indices the velocity, as displayed in Figure 9. The evolution of the communication topology for different values of the disturbance is represented in Figure 10. Figure 9 and 10 combined show the set of communication links used for a particular control mode, this highlights the importance of transmitting the velocity state variables for this mechanical system. The distributed controller then uses the cart speed differences in order to compute the control actions. In addition, it can be seen that communication between carts mechanically linked by a spring has a greater impact on the system performance, and is triggered by the supervised-distributed controller for larger disturbance values. Consequently, the supervised-distributed control algorithm could also be used to understand the most important communication links of a distributed system.

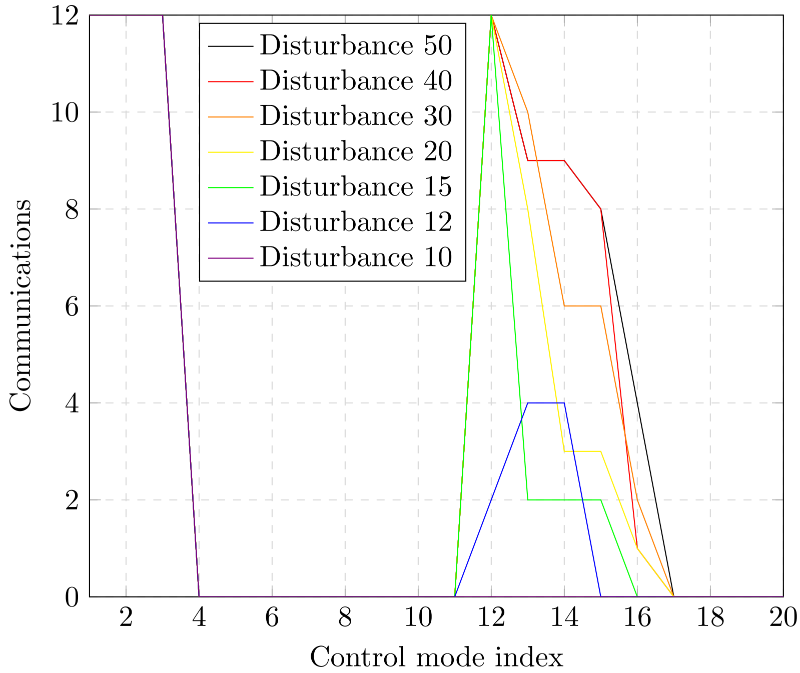

Figure 8. Use of communication in the control modes for different values of velocity disturbance applied to cart 1 in the system (26).

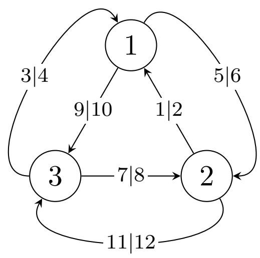

Figure 9. Communication link indices between the three carts of the system (26).

6.2.2. Impact of Model Dynamics

A change in the system parameters used for the cart masses or the spring constants modifies both the communication channels used as well as their usage frequencies. The system model used in the previous experiment has been modified to conduct two series of simulations. In the first set of three simulations, the mass of one cart is increased to 0.5, whilst the other masses are kept at 0.1, and both spring constants are set to 0.5.

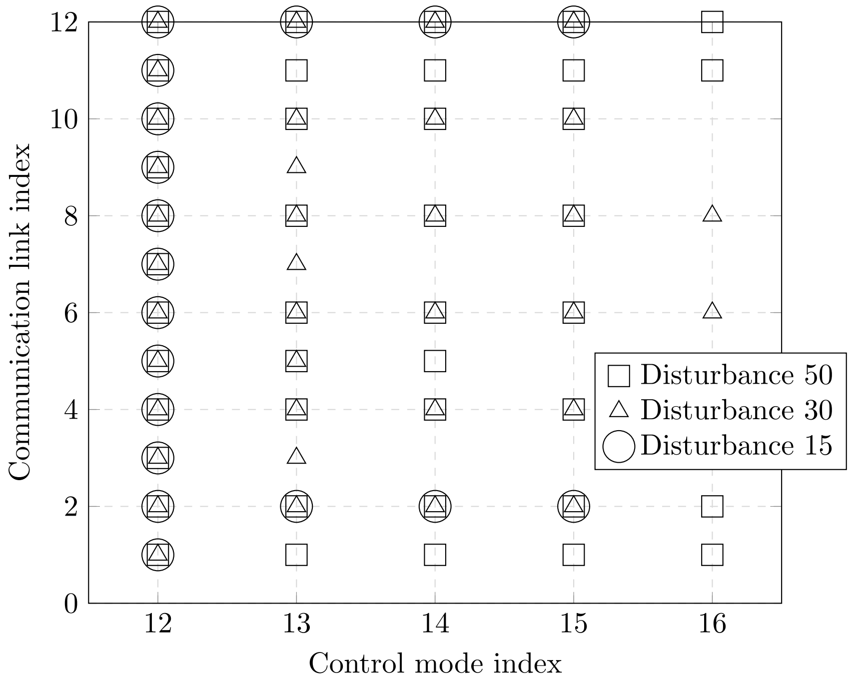

Figure 10. Evolution of the communication topology for different disturbance magnitudes applied to the system (26).

The second set varies the values of the spring constants, one spring constant is set to 0.01, while the other is set to 1 and vice versa. In both sets of simulations a disturbance of magnitude 30 is injected in cart 2 velocity at time step 50. The communication results are presented in Figure 11 for the variation of masses, and in Figure 12 for the modulation of spring constants. These figures display the total number and type of communication links used within the control modes designed online, using the same indexation as presented previously in Figure 9. The set of experiments varying the cart masses highlights that a heavy cart will communicate its position and velocity more to its neighbours to achieve good overall system performance, more specifically, a heavy cart communicates its position to direct neighbours and its speed to indirect neighbours. As it was observed in the initial simulations conducted for the study of the disturbance magnitude impact, the cart velocity has a great impact on the system performance.

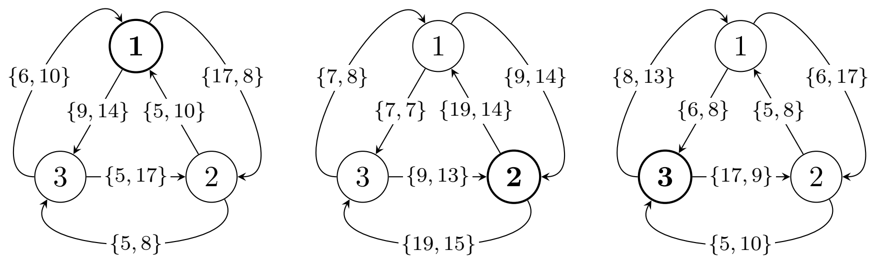

Figure 11. Comparison of the communication links usage for an increase in cart mass applied to carts 1, 2 and 3, from the left to the right graphs respectively, for the system (26).

Varying the values of the spring constants between the carts directly affects the dynamical coupling between the subsystems. Consequently, the supervised-distributed controller reacts by increasing the amount of communications between highly coupled subsystems. The two carts coupled with the spring having the highest constant communicate more, sharing more position and velocity states during the simulation. Once again, the velocity states are more important for the system-wide performance, and subsequently are communicated more often between all the subsystems.

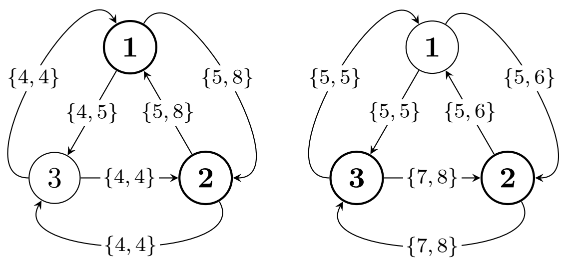

Figure 12. Comparison of the communication links usage for different spring constants applied to the system (26), the left graph has a higher K_{12} spring constant and the right graph has a higher K_{23} spring constant.

7. Conclusion

The supervised-distributed control scheme presented within this paper provides a strategy to the online optimization of distributed control laws. This optimal control technique minimizes a trade-off cost function, online and periodically, that includes the predicted infinite-horizon system performance, as well as a subsystem to subsystem communication metric. This paper has shown that this optimal control problem can be formulated as a biconvex optimization problem, thus solvable using an alternate convex search technique. The particular structure of the optimization problem employed provides a guarantee on the convergence of the control algorithm. Also, the supervised-distributed control technique is globally asymptotically stabilizing for the closed-loop system under control mode switching due to dwell time constraints. Finally, the recursive feasibility property of the supervised-distributed optimization problem was established; and it has been proven that terminating the algorithm after any number of iterations yields a feasible solution. The control strategy developed in this paper determines not only when communication between subsystems is required, but more importantly, which state variables to communicate, as well as the set of recipient subsystems and the control gains to apply. The dependency of both the control law gains and communication topology on the system state variables makes any offline control methods impractical. Numerical examples were used in order to demonstrate the efficacy of the control approach, highlighting the trade-off offered, as well as the improved system performance when communication is included as an objective. Future research directions shall explore the challenge of tuning the performance and communication trade-off parameter, using a game theoretical approach. Other investigation directions could look at conditions required to obtain a convex optimization problem, ensuring that the global optimal solution is reached.

References

Babazadeh, M., & Nobakhti, A. (2017). Sparsity promotion in state feedback controller design. *IEEE Transactions on Automatic Control*, *62*, 4066-4072.

Bakule, L. (2008). Decentralized control: An overview. *Annual Reviews in Control*, *32*, 87–98.

Blondel, V., & Tsitsiklis, J. N. (1997). NP-hardness of some linear control design problems. *SIAM Journal on Control and Optimization*, *35*, 2118–2127.

Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory.* Philadelphia: SIAM.

Boyd, S., & Vandenberghe, L. (2010). *Convex optimization.* Cambridge: Cambridge University Press.

Bristol, E. H. (1966). On a new measure of interaction for multivariable process control. *IEEE Transactions on Automatic Control*, *11*, 133–134.

Candès, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, *346*, 589–592.

Candès, E. J., & Tao, T. (2005). Decoding by linear programming. *IEEE Transaction on Information Theory*, *51*, 4203–4215.

Candès, E. J., Wakin, M. B., & Boyd, S. P. (2008). Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, *14*, 877–905.

Crusius, C. A. R., & Trofino, A. (1999). Sufficient LMI conditions for output feedback control problems. *IEEE Transactions on Automatic Control*, *44*, 1053–1057.

Doelman, R., & Verhaegen, M. (2016). Sequential convex relaxation for convex optimization with bilinear matrix equalities. *European Control Conference*, 1946–1951.

Fukuda, M., & Kojima, M. (2001). Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem. *Computational Optimization and Applications*, *19*, 79–105.

Geromel, J. C., & Colaneri, P. (2006). Stability and stabilization of discrete time switched systems. *International Journal of Control*, *79*, 719–728.

Goh, K. C., Safonov, M. G., & Papavassilopoulos, G. P. (1995). Global optimization for the biaffine matrix inequality problem. *Journal of Global Optimization*, *7*, 365–380.

Gorski, J., Pfeuffer, F., & Klamroth, K. (2007). Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research*, *66*, 373–407.

Gross, D., & Stursberg, O. (2016). A cooperative distributed MPC algorithm with event-based communication and parallel optimization. *IEEE Transactions on Control of Network Systems*, *3*, 275–285.

Guicherd, R., Trodden, P. A., Mills, A. R., & Kadirkamanathan, V. (2017). Weak interactions based system partitioning using integer linear programming. *IFAC-PapersOnLine*, *50*, 3698–3704.

Hassibi, A., How, J. P., & Boyd, S. P. (1999a). Low-Authority Controller Design by Means of Convex Optimization. *Journal of Guidance, Control, and Dynamics*, *22*, 862–872.

Hassibi, A., How, J. P., & Boyd, S. P. (1999b). A path-following method for solving BMI problems in control. *Proceedings of the American Control Conference*, 1385–1389.

Hurley, N., & Rickard, S. (2009). Comparing measures of sparsity. *IEEE Transactions on Information Theory*, *55*, 4723–4741.

Kariwala, V., Forbes, J. F., & Meadows, E. S. (2003). Block relative gain: Properties and pairing rules. *Industrial & Engineering Chemistry Research*, *42*, 4564–4574.

Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, *32*, 1361–1379.

Liberzon, D. (2003). *Switching in systems and control.* Boston: Birkhäuser.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. *IEEE International Symposium on Computer Aided Control Systems Design*, 284–289.

Lunze, J. (1992). *Feedback Control of Large-Scale Systems.* Dresden: Prentice Hall.

Maestre, J. M., Muñoz de la Peña, D., & Camacho, E. F. (2009). A distributed MPC scheme with low communication requirements. *Proceedings of the American Control Conference*, 2797–2802.

Maestre, J. M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., & Camacho, E. F. (2014). A coalitional control scheme with applications to cooperative game theory. *Optimal Control Applications and Methods*, *35*, 592–608.

Manousiouthakis, V., Savage, R., & Arkun, Y. (1986). Synthesis of decentralized process control structures using the concept of block relative gain. *AIChE Journal*, *32*, 991–1003.

Polyak, B., Khlebnikov, M., & Shcherbakov, P. (2013). An LMI approach to structured sparse feedback design in linear control systems. *European Control Conference*, 833–838.

Rotkowitz, M., & Lall, S. (2006). A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, *51*, 274–286.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, *19*, 723–731.

Schuler, S., Münz, U., & Allgöwer, F. (2014). Decentralized state feedback control for interconnected systems with application to power systems. *Journal of Process Control*, *24*, 379–388.

Šiljak, D. D. (1991). *Decentralized control of complex systems.* San Diego: Academic Press Inc.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, *11–12*, 625–653.

Wendell, R. E., & Hurter, A. P. (1976). Minimization of a Non-Separable Objective Function Subject to Disjoint Constraints. *Operations Research*, *24*, 643–657.

Wie, B., & Bernstein, D. S. (1992). A Benchmark Problem for Robust Control Design. *Journal of Guidance, Control, and Dynamics*, *15*, 1057–1059.

Xiao, L., Johansson, M., Hindi, H., Boyd, S., & Goldsmith, A. (2003). Joint optimization of communication rates and linear systems. *IEEE Transactions on Automatic Control*, *48*, 148–153.

Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of the IEEE Infocom*, 1567–1576.

Zhou, K., Doyle, J. C., & Glover, K. (1996). *Robust and optimal control.* Upper Saddle River: Prentice Hall.