Contents

1 Distributed estimation and control 1
S. Hirche, B. Lohmann, P. Philipp, M. Ulbrich, F. Wirth

References 15

1

Distributed estimation and control

S. Hirche, B. Lohmann, P. Philipp, M. Ulbrich, and F. Wirth

!!!Synopsis of the chapter!!!

Chapter contents

1.1 Towards a two-layer architecture for distributed systems . . 2
1.1.1 Distributed optimization 2
1.1.2 Control of distributed systems 5
1.1.3 Dual decomposition of DC optimal power flow problem 8
1.1.4 Distributed controller design with local model information 10

1.1 Towards a two-layer architecture for distributed systems

New application fields like smart power grids, water distribution systems, traffic systems, or large arrays of micro-electromechanical systems (MEMS) require novel optimization and control algorithms that operate in a distributed, event-triggered fashion and are able to react to dynamic changes within the overall system on different time scales. The research presented here is targeted at a two-layer architecture for the optimal control of distributed systems. It consists of two interacting, distributed layers: an upper optimization layer and a lower feedback control layer. The general idea is that the feedback controller adapts itself to the changing physical structure and is able to deal with fast dynamics, while the distributed optimization layer can be adjusted to the communication topology and addresses global aspects on a slower time scale. The overall control scheme uses an event-triggered, local information exchange.

In this chapter, we present a selection of important techniques for distributed optimization and control. In the first section, methods from the field of distributed optimization are shown, afterwards we give an overview over important results in distributed control. Lastly, we present two particular methods, one from optimization, the other from controller design. The two methods have in common that they are both based on dual approaches.

1.1.1 Distributed optimization

Distributed optimization algorithms are gaining significant importance due to, in particular, the ubiquitous parallel processing of new computer generations and the growing need for distributed approaches to optimized decision making and control in large scale networked systems. In this section we review several central ideas that are at the core of distributed optimization algorithms.

In general, optimization problems in a multi-agent system have the following structure: Every agent $i \in \{1, \dots, m\}$ has control over a sub-block $x_i \in \mathbb{R}^{n_i}$ of the decision variables $x = (x_1, \dots, x_m) \in \mathbb{R}^n$ and the aim is to solve the following constrained optimization problem:

$$\min_{x \in X \subseteq \mathbb{R}^n} f(x), \tag{1.1}$$

where $X \subseteq \mathbb{R}^n$ is the feasible set and $f : U \to \mathbb{R}$ is the cost function, defined on an open set $U \supset X$. The structure of f and X determines to what extent the agents have to interact in order to jointly solve (1.1). Favorable properties for decomposing the problem are separability of the cost function, i.e., $f(x) = \sum_{i=1}^{m} f_i(x_i)$ or separability of the constraint set, i.e., $X = X_1 \times \cdots \times X_m$. If both are separable then problem (1.1) decomposes into m subproblems $\min_{x_i \in X_i} f_i(x_i)$ that can be solved independently and in parallel by each agent i. Of course, this full separability is usually not given

and depending on the type of non-separability different approaches can be followed to obtain distributed solution methods.

For the case of a separable feasible set $X = X_1 \times \cdots \times X_m$ and a non-separable objective function $f(x)$, variants of block coordinate descent methods are appropriate. *Parallel Variable Distribution* (PVD) [8, 24, 27, 28] is a general class of methods that follow this paradigm. In each iteration, the PVD algorithm performs a parallelization step and a synchronization step to compute the new iterate x^{k+1} from the current iterate x^k. In the parallelization step, the main responsibility of agent i is to find an improvement for the ith block of x by solving a suitable subproblem. This is done in parallel by all agents. In addition, the search space of agent i can be extended by one dimension for each of the other $m-1$ subblocks x_j, $j \neq i$. This "forget-me-not approach" improves the robustness and accelerates the convergence compared to earlier methods such as block Jacobi [1] and coordinate descent [30]. The subproblem of the ith agent looks as follows:

$$\begin{aligned} \min_{y_i \in \mathbb{R}^{n_i}, (\lambda_{ij})_{j \neq i} \in \mathbb{R}^{m-1}} & f_i\left(y_i, x_{\bar{i}}^k + \sum_{j \neq i} \lambda_{ij} d_j^k\right) \\ \text{s.t. } & y_i \in X_i, \quad x_j^k + \lambda_{ij} d_j^k \in X_j \quad (j \neq i). \end{aligned}$$

Here, $x_{\bar{i}} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$ denotes x after removal of the ith block x_i and $f_i(x_i, x_{\bar{i}}) := f(x_1, \ldots, x_m)$ is just the function f with conveniently reordered variables x_j. This problem is solved to obtain (approximate) solutions y_i^k and λ_{ij}^k, $j \neq i$. Convergence analyses exist for different flavors of accuracy requirements to which these subproblems are solved: exact, ϵ-solutions, or sufficient decrease conditions [24, 27, 28]. The synchronization step then collects the agents' new candidate iterates $x^{k,i}$ with subblocks $x_i^{k,i} = y_i^k$ and $x_j^{k,i} = x_j^k + \lambda_{ij}^k d_j^k$, $j \neq i$. Here, the vectors d_j^k are subblocks of a suitable direction $d^k \in \mathbb{R}^n$ (e.g. steepest descent). The new iterate $x^{k+1} \in X$ is determined such that $f(x^{k+1}) \leq \min_{1 \leq i \leq m} f(x^{k,i})$. This condition can be relaxed, too.

Another class of distributed optimization methods, *bilevel decomposition algorithms* (BDA) [2, 14, 15, 29], is suitable for partially separable optimization problems. These are separable with respect to m local variables $x_i \in \mathbb{R}^{n_i}$, $i = 1, \ldots, m$, and coupled via global variables $y \in \mathbb{R}^p$:

$$\min_{y, x_1, \ldots, x_m} \sum_{i=1}^{m} f_i(y, x_i) \quad \text{s.t.} \quad g_j(y, x_j) \leq 0 \quad (1 \leq j \leq m),$$

with $g_i : \mathbb{R}^p \times \mathbb{R}^{n_i} \to \mathbb{R}^{l_i}$. The general approach of BDA to solve this problem is to break it up into a master problem and a set of m subproblems each depending only on the local variable block x_i. A more formal description of the algorithm is the following:
Solve the master problem

$$\min_y \sum_{i=1}^{m} F_i^*(y),$$

where $F_i^*(y)$ is the optimal value function of the ith subproblem:

$$F_i^*(y) = \min \{ f_i(y, x_i) \mid g_i(y, x_i) \leq 0,\ x_i \in \mathbb{R}^{n_i} \}$$

for $i = 1, \ldots, m$. The latter problems can be solved in parallel.

Both, PVD methods and BDA require to some extent a global data exchange. PVD distributes the current iterate to all agents before the synchronization step and collects the results in the parallelization step. In the BDA, each evaluation of the master problem's cost function requires to distribute the global variable y to the agents and to collect the function values $F_i^*(y)$ afterwards. If the optimization procedure for the master problem requires additional information, e.g., (sub-) gradients of F_i^*, these have to be collected as well. Alternatives to BDA were proposed that tackle the whole coupled problem by interior point methods and perform a BDA-like decomposition on the linear algebra level by a Schur complement approach [15].

Many optimization problems arising in the context of large-scale networks such as utility maximization problems (NUM) [19], distributed estimation [22], and the DC optimal power flow problem [20] have the following common structure: The cost function is convex and separable and the coupling occurs only via linear constraints:

$$\min_{x \in X} \sum_{i=1}^{m} f_i(x_i) \quad \text{s.t.} \quad Ax \leq c\ , \quad Bx = d,$$

with $X = X_1 \times \cdots \times X_m$, $X_i \subset \mathbb{R}^{n_i}$ convex and closed, $A = (A_1, \ldots, A_m) \in \mathbb{R}^{l \times n}$, $A_i \in \mathbb{R}^{l \times n_i}$, $c \in \mathbb{R}^l$, $B = (B_1, \ldots, B_m) \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times n_i}$, and $d \in \mathbb{R}^n$. Due to this structure *dual decomposition algorithms* [3, 4, 17, 32] find broad application, where the constraints are decoupled using the Lagrangian $\mathcal{L}(x, \lambda, \mu) = \sum_{i=1}^{m} [f_i(x_i) + \lambda^T A_i x_i + \mu^T B_i x_i] + \lambda^T c + \mu^T d$. Then the dual problem is considered:

$$\max_{\lambda \in \mathbb{R}^l_+,\ \mu \in \mathbb{R}^n} d(\lambda, \mu),$$

where $d(\lambda, \mu) = \inf_{x \in X} \mathcal{L}(x, \lambda, \mu)$.
Obviously, the evaluation of the dual function can be done in parallel and a classical approach to solve the dual problem is the application of the projected gradient method [1] (or the projected subgradient method [1], if the dual function is not differentiable) which is easily parallelizable too. distribution could be used as well. regarding the dual multipliers λ the classical approach Often, the structure of the network optimization problems mentioned above in combination with dual decomposition requires only local communication between the agents, i.e., only neighboring agents in the network need to communicate with each other (see section 1.1.3). By incorporating event-triggered communication [13] into the dual decomposition approach, additional flexibility with

respect to physical and communicational network topology changes can be achieved. This is sketched in the following section.

We close this section by mentioning that many aspects of (distributed) optimization can be interpreted from a control perspective [33]. The iteration performed in optimization methods can often be interpreted as a time discretization of a dynamical system (e.g., gradient flow). The aim, however, is not to accurately trace the trajectory, but rather to efficiently approach an equilibrium point.

1.1.2 Control of distributed systems

Today, our everyday life is influenced directly by many important large-scale dynamical systems. The electrical power grid and our water distribution systems, for example, are fundamental to our society. Traffic management has become more and more important, motivated by the growing population. Also, the recent technological revolution represented by the internet can be regarded as a dynamical system, spread out all over the globe. While this system class has existed for a long time, new challenges have emerged. First, and foremost, in control it is always the goal to achieve a good performance, and today, more than ever, the desire for energy efficiency calls for new technologies. Furthermore, specific challenges call for new control methods. For instance, the idea to couple national and even continental power grids more closely, while simultaneously including more and more renewable energy sources, creates even larger dynamical systems which become harder to control. Distributed control aims to become one of the enabling technologies to deal with these new challenges.

The size of large-scale systems is the reason why classical, centralized control methods cannot be applied. Traditionally, completely decentralized control methods are used which means that the individual subsystems do not exchange information and only use their own measurements for the control task. However, the wide-spread use of modern communication networks has sparked ideas and interest to employ these new technologies in the context of control to find an intermediate between centralized and decentralized control. The basic idea is to allow communication between some – but not all – subsystems, in order to improve performance when compared to decentralized control, while maintaining the low complexity compared to centralized (full information) control. The structure of the allowed communication connections – sometimes referred to as the *communication topology* – imposes structural constraints on the controller.

In this section, we will restrict ourselves to structured linear-time invariant systems with N subsystems. We assume that the subsystems can influence each other via the state, but that the input of each agent is restricted to the one respective agent. In the time-domain, this can be formulated as follows.

The dynamics of each subsystem i is given by

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{A}_{ii}\boldsymbol{x}_i(t) + \boldsymbol{B}_i\boldsymbol{u}_i + \sum_{\substack{j=1 \\ j\neq i}}^{N} \boldsymbol{A}_{ij}\boldsymbol{x}_j,\ i = 1, ..., N,$$

where $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$, $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$, $\boldsymbol{A}_{ii} \in \mathbb{R}^{n_i \times n_i}$, $\boldsymbol{A}_{ij} \in \mathbb{R}^{n_i \times n_j}$ and $\boldsymbol{B}_i \in \mathbb{R}^{n_i \times m_i}$.

The coupling structure of the dynamical system is often described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the collection of nodes (i.e. subsystems, agents) and $\mathcal{E}$ is the collection of edges (i.e. interconnections). The edge set describes which subsystems influence each other, or more specifically, if an edge $(j, i) \in \mathcal{E}$, subsystem i is influenced directly by the states of agent j. Hence, the matrix $\boldsymbol{A}_{ij} \neq 0$ only if $(j, i) \in \mathcal{E}$.

$$\mathcal{N}_i = \{j | (i, j) \in \mathcal{E} \text{ or } (j, i) \in \mathcal{E}\}.$$

By concatenation of the subsystems' states, the overall interconnected system can be written compactly as the continuous-time time-invariant linear system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1.2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state, $\boldsymbol{u} \in \mathbb{R}^m$ is the input and $\boldsymbol{x}_0 \in \mathbb{R}^n$ is the initial condition. The matrix $\boldsymbol{A}$ consists of the blocks $\boldsymbol{A}_{ij}$, and $\boldsymbol{B}$ has the form $\mathrm{diag}(\boldsymbol{B}_i)$.

Similarly, some approaches in the literature consider the system in frequency domain. This means that the plant transfer function matrix $\boldsymbol{G}(s)$ has a sparsity structure (zero entries) which can also be described by a graph.

The goal of distributed optimal control is to design a controller which also has a specific sparsity structure, and which is optimal in some sense. The literature offers a variety of performance specifications which will be explored in more detail below.

Regarding the structure, we want to give an example. Suppose we want to design a linear static state feedback control where communication is allowed among neighbors (neighborhood as defined above). Mathematically, this can be formulated as

$$\boldsymbol{K}_{ij} \neq 0 \text{ only if } j \in \mathcal{N}_i. \tag{1.3}$$

Obviously, in the frequency domain, similar structural constraints can be imposed on the feedback transfer function matrix $\boldsymbol{K}(s)$.

In the following, we want to present several approaches to control large-scale systems.

As already mentioned, the earliest approaches dealing with large-scale systems are decentralized control [26] where each subsystem uses only its own information for the control task. While stability may be guaranteed, the performance is usually degraded when compared to centralized (full information) control. So distributed control can be regarded as a compromise between classical centralized control and decentralized control. A big distinction in the

approaches can be made which is especially interesting when large-scale systems are considered. This divides all approaches into those which assume that complete, centralized model information is available while others consider the control problem with only limited or locally available model information. We begin our overview with the complete-model approaches.

An early result [34] in distributed control already illustrates the difficulty of the problem by showing that the optimal controller to a two-agent linear quadratic cost problem is nonlinear when a structural constraint is imposed. A more recent important result is given in [21] where conditions on the communication topology are given based on the plant transfer function such that the controller design problem can be formulated as a convex optimization problem. The paper uses the Youla parametrization to parameterize all stabilizing controllers and if the set of admissible controllers satisfies the so-called property of quadratic invariance, the optimal controller can be found via a convex problem. Also, a computational test for quadratic invariance based on the structure of the controller matrix $\boldsymbol{K}$ and on the structure of the plant transfer function $\boldsymbol{G}$ is given. However, the resulting optimization problems are in general infinite-dimensional and thus, though convex, still difficult to solve.

In some cases, it may be of interest to have a minimal communication topology while maintaining good performance levels in comparison to the centralized case and this is achieved in [23]. This is first formulated as a minimization of the l_0-norm, i.e. the direct minimization of the number of entries in the feedback matrix. Since this problem is hard to solve, a convex relaxation is introduced in form of a weighted l_1-minimization. The minimization is subject to a given performance degradation in an $\mathcal{H}_\infty$-sense. This means that the performance of the decentralized controller is compared to the performance of the optimal centralized controller and the discrepancy cannot become too large.

Interesting approaches for distributed controller synthesis based on state-space formulations are for example [11, 31] which are based on LMIs and where the controller topology is identical to the plant topology, i.e. if subsystems are coupled dynamically, they are allowed to exchange information. This assumption on the structure is fairly common and is based on the idea that such a communication topology would be relatively easy to implement.

In [25], a state-space method for distributed $\mathcal{H}_2$ optimal controller design is presented which is applicable when the subsystems are partially ordered, e.g. chains, trees or nested structures. It is shown that the problem can be decomposed into N subproblems, where N is the number of subsystems in the whole system. Each subproblem then consists of the solution of an uncoupled Riccati equation for which efficient techniques are known.

If one is interested in the largest eigenvalue as a performance measure, the approach in [10] is applicable. The minimization of the real part of the largest eigenvalue aims at maximizing the convergence rate. In order to do so, a decentralized controller is designed in a first step. This controller is simply used to stabilize the system and no performance is optimized. The advantage of this is that the system remains stable even if communication links fail. In a second step, a limited number of communication links between the subsystems are added to get optimal performance. This constitutes a mixed integer problem which is NP-hard. For a special case of scalar subsystems and several conditions on the dynamics matrix, a more efficient solution is also derived.

All the methods presented above have in common that the controller is implemented distributedly but it is designed in a centralized way, i.e. with the assumption that a complete model is available. So in the following, we want to present some of the very few approaches that try to satisfy their control task with only limited model information. One of these papers is [7]. The authors show for discrete-time LTI systems that when each agent knows only its own system model, deadbeat control cannot be outperformed by any other control strategy when an LQR cost functional is considered. For a specific structure of the dynamics matrix $\boldsymbol{A}$ (tridiagonal), which is present in vehicle following problems, the authors in [6] formulate a distributed optimization problem using dual decomposition to find the optimal linear quadratic distributed controller. In order to facilitate convergence of the problem, they use a relaxed augmented Lagrangian technique. Another approach based on distributed optimization is presented in [9]. The authors use dual decomposition in a distributed model predictive method which involves repeated minimization of a cost function to determine an optimal control action profile.

In [12] it is assumed that subsystems know their own dynamic model and the model of their direct neighbors (neighborhood is defined through direct physical coupling). The optimal distributed controller for an LQR cost functional is then computed iteratively by introducing an adjoint (dual) state and using simulated trajectories of the system and adjoint states to compute a gradient descent direction for the state feedback matrix.

1.1.3 Dual decomposition of DC optimal power flow problem

We now consider the static DC optimal power flow problem, where a convex and separable energy production cost function in an electrical power network has to be minimized subject to coupling constraints that enforce the power balance within the network as well as bounds on power generation and power line flow. The power system is modeled by a directed graph $G = \{V, E\}$ with $V = \{1, \ldots, n\}$ representing the set of buses, where all buses contain a load and for simplicity of notation are directly connected to a generator. The generalization of only $p < n$ buses directly connected to a generator is

straightforward. The edge $e_{ij} \in E \subseteq V \times V$ with $|E| = m$ represents the transmission line from bus i to bus j. Let I be the $m \times n$ incidence matrix of the graph G and define a diagonal matrix $D \in \mathbb{R}^{m \times m}$ with $d_{ll} = 1/x_l$ where x_l is the reactance of the lth transmission line.
Denoting by P_i^g the generated power, by P_i^d the load and by θ_i the voltage phase angle at bus i, the DC optimal power flow problem can be stated as follows:

$$\min_{P_i^g \in P_i,\ \theta_i \in \Theta_i} \sum_{i=1}^{n} C_i(P_i^g)$$

$$\text{s.t. } B\theta = P^g - P^d \tag{1.4}$$

$$F^{\min} \leq DI\theta \leq F^{\max}, \tag{1.5}$$

where P_i, Θ_i are compact and convex sets, $C_i(x) = a_{i2}x^2 + a_{i1}x + a_{i0}$ is the quadratic cost of power production at bus i with nonnegative coefficients a_{i2}, a_{i1}, a_{i1} for $i = 1, \ldots, n$. Matrix $B \in \mathbb{R}^{n \times n}$ is defined as:

$$B = I^T DI\ , \qquad B_{ij} = \begin{cases} \sum_{k \in N(i)} \frac{1}{x_{ik}}, & \text{if } i = j, \\ -\frac{1}{x_{ij}}, & \text{if } j \in N(i), \\ 0, & \text{else,} \end{cases}$$

where $N(i) = \{j \mid (i,j) \in E \vee (j,i) \in E\}$ denotes the set of indices of bus i's neighbors and x_{ij} denotes the reactance of the transmission line connecting bus i and bus j. The constraint (1.5) limits the power flow on each transmission line l to $[F_l^{\min}, F_l^{\max}]$, whereas constraint (1.4) expresses the power flow !¿!L equations.
We used the partially separable structure of this problem in combination with a dual decomposition approach [16] to derive a dual problem that allows for a parallel evaluation of the dual cost function:

$$f(\lambda, \mu) = \min_{P_i^g \in P_i,\ \theta_i \in \Theta_i} \sum_{i=1}^{n} [C_i(P_i^g) + \mu_i P_i^g] + \sum_{i=1}^{n} \Bigg[\sum_{l \in L(i)} (\lambda_l - \lambda_{l+m}) A_{li}$$

$$- \sum_{j \in N(i) \cup i} \mu_j B_{ji} \Bigg] \theta_i + \sum_{l=1}^{m} [\lambda_{l+m} F_l^{\min} - \lambda_l F_l^{\max}] - \sum_{i=1}^{n} \mu_i P_i^d,$$

where $L(i) = \{l \mid v_i \in e_l \wedge e_l \in E\}$ denotes the set of indices of lines connected to bus i. Nesterov's smoothing method [18] is utilized by adding strongly convex functions $d_i(x) = \frac{\sigma_i}{2}\|x\|_2^2$, $\sigma_i > 0$ for $i = 1, \ldots, 2n$ scaled with a smoothing parameter $c > 0$ to obtain a Lipschitz continuously differentiable smoothed dual cost function that inherits the favorable structure of the original dual function:

$$
\begin{aligned}
f_c(\lambda,\mu) = & \min_{P_i^g \in P_i,\ \theta_i \in \Theta_i} \sum_{i=1}^{n} [C_i(P_i^g) + \mu_i P_i^g + c\, d_i(P_i^g)] \\
& + \sum_{i=1}^{n} \left\{ \left[\sum_{l \in L(i)} (\lambda_l - \lambda_{l+m}) A_{li} - \sum_{j \in N(i) \cup i} \mu_j B_{ji} \right] \theta_i + c\, d_{i+n}(\theta_i) \right\} \\
& + \sum_{l=1}^{m} [\lambda_{l+m} F_l^{\min} - \lambda_l F_l^{\max}] - \sum_{i=1}^{n} \mu_i P_i^d .
\end{aligned}
$$

The considered augmented dual problem has the form:

$$
\max_{\lambda \in \mathbb{R}_+^{2m},\ \mu \in \mathbb{R}^n} f_c(\lambda,\mu).
$$

In [13] we adjusted Nesterov's first order method [16, 18] to solve this problem in parallel and achieved ϵ-optimality with optimal iteration complexity $O(\sqrt{L_c/\epsilon})$, where L_c is the Lipschitz constant of the gradient of the smoothed dual function f_c. The error introduced by the smoothing is of the order $O(c)$, where c is the smoothing parameter. Since there holds $L_c = O(1/c)$, the choice $c = O(\epsilon)$ achieves an ϵ-optimal solution in $O(1/\epsilon)$ iterations. We further showed in [13] that this method can be applied in such a way that each agent needs to acquire only data that are maintained by direct neighbors, i.e., by those agents to which it is linked by a direct edge (line) of the power grid graph. In order to further reduce communication traffic, we introduced event-triggered communication in the developed method. Here, an agent transmits data only when a certain threshold is exceeded. Convergence was proved in [13] while maintaining the favorable features of the original scheme and numerical tests show that communication between the agents could be reduced by up to 60%. Moreover, we observed that an incautious choice of the convexity parameters $\sigma_i > 0$ of the d_i can lead to a quite large Lipschitz constant L_c due to the dependence

$$
L_c(\sigma) = \frac{1}{c} \sum_i \frac{u_i}{\sigma_i},
$$

where the vectors u_i are problem-depending constants. We showed that the resulting optimization problem to minimize the Lipschitz constant subject to nonnegative convexity parameters $\sigma_i > 0$ can be solved analytically and in a distributed, parallel way. Thus, it is well suited also for the case of dynamically changing network topologies. Finally, we developed a scaling technique to further reduce the number of iterations necessary to achieve ϵ-optimality.

1.1.4 Distributed controller design with local model information

While the majority of the research on distributed control considers a centralized design and a distributed implementation, it is the authors' opinion that

at least for large-scale distributed systems, distributed design methods are necessary. The reasons for this are manifold. A first issue is privacy, i.e. not every subsystem might be willing to share model information with every other subsystem they are cooperating with. Second, a dynamic model might not be readily available for a large system. Furthermore, a system model might be too large to be handled by a centralized design method. A last point is that large-scale systems are likely to change at some point (nodes are added or removed), and a centralized approach would require a lot of effort to react to these changes, while a distributed approach could react locally in the affected nodes only.

Motivated by this reasoning, and inspired by the results in [12], an accelerated iterative method to determine a distributed controller for continuous-time LTI systems given in Eq. (1.2) is presented in [5]. As a performance criterion, the finite horizon LQR cost functional

$$J(\boldsymbol{x}, \boldsymbol{u}) = \int_0^T \boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\mathrm{d}t, \tag{1.6}$$

is considered, and it is minimized in the following way. The feedback matrix $\boldsymbol{K}$ is restricted to the structure described by the constraint given in (1.3). The gradient of the cost functional (1.6) with respect to the (i,j)th entry of the feedback matrix is given by

$$(\nabla_{\boldsymbol{K}} J)_{ij} = -2\boldsymbol{R}_i \int_0^T \boldsymbol{u}_i(t)\boldsymbol{x}_j(t)^T \mathrm{d}t - \boldsymbol{B}_i^T \int_0^T \boldsymbol{\lambda}_i(t)\boldsymbol{x}_j(t)^T \mathrm{d}t,$$

where

$$\dot{\boldsymbol{\lambda}}(t) = -(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^T\boldsymbol{\lambda}(t) - 2(\boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K})\boldsymbol{x}(t),$$

with $\boldsymbol{\lambda}(T) = 0$. Thus, the gradient is computed using simulated trajectories of the system. The (locally) optimal feedback matrix can then be determined iteratively using a gradient descent method. Thus, in order to compute their respective entries of the feedback matrix, the agents only need information from their direct neighbors.

The main results of the paper deal with the challenge of determining a good step size that leads to relatively fast convergence. Step size selection is a difficult problem for distributed optimization because no centralized or collected knowledge is available. In the paper, the step size is determined according to a Barzilai-Borwein scheme. Applied to the presented problem, the Barzilai-Borwein method gives the step size with

$$\gamma_k = \frac{\langle \Delta\mathrm{vec}(\boldsymbol{K}), \Delta\mathrm{vec}(\boldsymbol{K})\rangle}{\langle \Delta\mathrm{vec}(\boldsymbol{K}), \Delta\mathrm{vec}(\nabla_{\boldsymbol{K}} J)\rangle} \tag{1.7}$$

of where Δ denotes the difference between iteration k and $(k-1)$, $\langle\cdot,\cdot\rangle$ is the scalar product, superscript (k) denotes the kth iteration and vec represents the column-wise vectorization of a matrix. While this formula cannot be computed

distributedly directly, the computation can be completely distributed using a consensus algorithm in the following way.

First, the individual agents use their own entries of both feedback and gradient matrix to determine an estimate of the BB-step size γ_k. Afterwards, the value of (1.7) is given by a distributed consensus algorithm based on the previous estimates. Specifically, as a first step, each node $i, i = 1, ...n$ initializes the two scalar values

$$\rho_i(k(0)) = \left\langle \Delta\mathrm{vec}(\boldsymbol{K}_i^T), \Delta\mathrm{vec}(\boldsymbol{K}_i^T) \right\rangle,$$

and

$$\psi_i(k(0)) = \left\langle \Delta\mathrm{vec}(\boldsymbol{K}_i^T), \Delta\mathrm{vec}((\nabla_{\boldsymbol{K}} J)_i^T) \right\rangle.$$

In other words, in every iteration k, the agents will calculate the completely local values $\rho_i(k(0))$ and $\psi_i(k(0))$ using only their own respective row(s) of the feedback and gradient matrix. After this, consensus iterations start which require information exchange:

$$\rho_i(k(t+1)) = \boldsymbol{W}_{ii}\rho_i(k(t)) + \sum_{j\in\mathcal{N}_i} \boldsymbol{W}_{ij}\rho_j(k(t)),$$
$$\psi_i(k(t+1)) = \boldsymbol{W}_{ii}\psi_i(k(t)) + \sum_{j\in\mathcal{N}_i} \boldsymbol{W}_{ij}\psi_j(k(t)),$$

where $\boldsymbol{W}$ is a symmetric, non-negative with strictly positive diagonal entries, doubly stochastic matrix, compatible with graph $\mathcal{G}$, and it requires only information exchange between neighbors. It is then shown that the consensus algorithm converges to γ_k of the BB-formula (1.7). Additionally, using the presented step size method, convergence of the gradient method to a stationary point is shown using the so-called Armijo rule.

As a numerical demonstration of the effectiveness of the new step size method, the approach is applied to 500 randomly created stable systems of size $n = 10$ with $m = 10$ inputs. An average of 44 entries of the state matrices are non-zero. The time horizon was is to 10 s, all threshold values of the algorithm are 10^{-3}. A comparison of the BB step size is done with a constant step size of $\gamma = 1$. The result is that on average, the BB step size needs only 18% of the iterations that are needed with the constant step size.

is

Furthermore, the problem setup allows reference tracking of the state. This offers the possibility to link this control level with an upper optimization level which computes optimal setpoints through distributed optimization.

A visualization of the distributed controller computation is given in Fig. 1.1 for an example system with four subsystems.

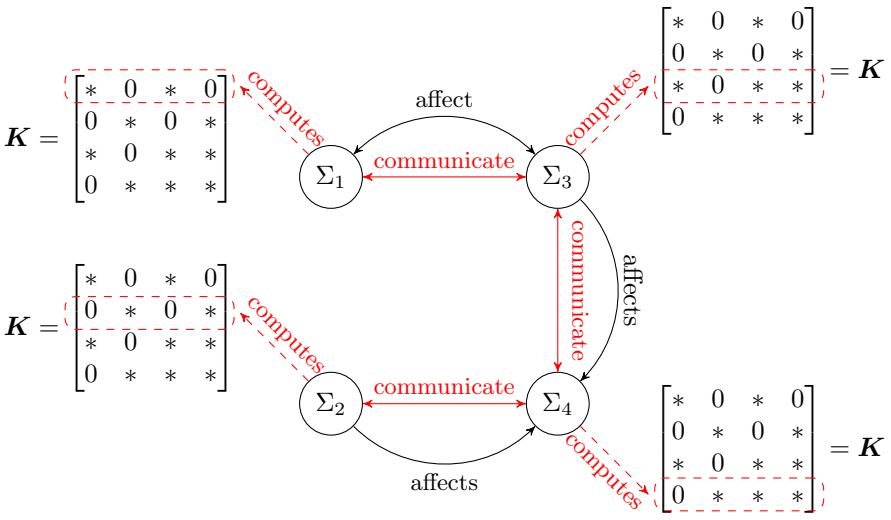

Fig. 1.1. Representation of the general approach for an example of 4 subsystems: Σ_i denotes subsystem i, black arrows represent physical coupling, red arrows represent communication, red dashed arrows computation

References

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1989.

[2] R. D. Braun and I. M. Kroo. Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In M. Y. Hussaini N. M. Alexandrov, editor, *Multidisciplinary Design Optimization: State of the Art*, pages 98–116. SIAM, Philadelphia, 1997.

[3] G. Chert and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathematical Programming*, 64:81–101, 1994.

[4] R. De Leone, R. R. Meyer, and S. Kontorgiorgis. Alternating Direction Splittings For Block-Angular Parallel Optimization. *Journal of Optimization Theory and Applications*, 90(1):2–9, 1996.

[5] F Deroo, M. Ulbrich, B. D. O. Anderson, and S. Hirche. Accelerated iterative Distributed Controller Synthesis with a Barzilai-Borwein Step Size. In *Proc. 51st IEEE Conf. on Decision and Control (CDC)*, 2012 (accepted).

[6] M. Fardad, F. Lin, and M. R. Jovanovic. On the dual decomposition of linear quadratic optimal control problems for vehicular formations. In *Proc. 49th IEEE Conference on Decision and Control (CDC)*, pages 6287–6292. IEEE, 2010.

[7] F. Farokhi, C. Langbort, and K. H. Johansson. Control design with limited model information. In *Proc. American Control Conf. (ACC)*, 2011.

[8] M. C. Ferris and O. L. Mangasarian. Parallel Variable Distribution. *SIAM Journal on Optimization*, 4(4):815–832, 1994.

[9] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proc. 49th IEEE Conf. Decision and Control (CDC)*, number 0, pages 7272–7277. IEEE, 2010.

[10] A. Gusrialdi and S. Hirche. Performance-oriented communication topology design for large-scale interconnected systems. In *Proc. 49th IEEE Conf. Decision and Control (CDC)*, pages 5707–5713, 2010.

[11] C. Langbort, R.S. Chandra, and R. D'Andrea. Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control*, 49(9):1502–1519, 2004. ISSN 0018-9286.

[12] K. Martensson and A. Rantzer. A scalable method for continuous-time distributed control synthesis. In *Proc. American Control Conf. (ACC)*, pages 6308–6313, 2012.

[13] M. Meinel and M. Ulbrich. A Class of Distributed Optimization Methods with Event-Triggered Communication. Preprint, 2012.

[14] V. De Miguel and W. Murray. A Local Convergence Analysis of Bilevel Decomposition Algorithms. *Optimization and Engineering*, 7(2):99–133, 2006.

[15] V. De Miguel and F. J. Nogales. On Decomposition Methods for a Class of Partially Separable Nonlinear Programs. *Mathematics of Operations Research*, 33(1):119–139, 2008.

[16] I. Necoara and J. Suykens. Application of a smoothing technique to decomposition in convex optimization. *IEEE Transactions on Automatic Control*, 53(11):2674 –2679, 2008.

[17] I. Necoara and J. A. K. Suykens. Interior-Point Lagrangian Decomposition Method for Separable Convex Optimization. *Journal of Optimization Theory and Applications*, 143(3):567–588, 2009.

[18] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programmming*, 103(1):127–152, 2005.

[19] D. P. Palomar and M. Chiang. A Tutorial on Decomposition Methods for Network Utility Maximization. *IEEE Journal on Selected Areas in Communication*, 24(8):1439–1451, 2006.

[20] K. Purchala, L. Meeus, D. Van Dommelen, and R. Belmans. Usefulness of dc power flow for active power flow analysis. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 454 – 459 Vol. 1, 2005.

[21] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized Control. *IEEE Transactions on Automatic Control*, 51(2): 274–286, 2006.

[22] S. Samar, S. Boyd, and D. Gorinevsky. Distributed Estimation via Dual Decomposition. In *Proceedings of the European Control Conference (ECC)*, pages 1511–1516, 2007.

[23] S. Schuler, W. Zhou, U. Münz, and F. Allgöwer. Controller structure design for decentralized control of coupled higher order subsystems. In *Proc. 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, Annecy, France*, pages 269–274, 2010.

[24] C. A. Sgastizabal and M. V. Solodov. Parallel Variable Distribution for Constrained Optimization. *Computational Optimization and Applications*, 22(1):111–131, 2002.

[25] P. Shah and P. A. Parrilo. H2-optimal decentralized control over posets: A state space solution for state-feedback. In *Proc. 49th IEEE Conf. Decision and Control (CDC)*, 2010.

[26] D. D. Siljak. *Large-Scale Dynamic Systems: Stability and Structure*. North-Holland, 1978.

[27] M. V. Solodov. New Inexact Parallel Variable Distribution Algorithms. *Computational Optimization and Applications*, 7(2):165–182, 1997.

[28] M. V. Solodov. On The Convergence of Constrained Parallel Variable Distribution Algorithms. *SIAM Journal on Optimization*, 8(1):187–196, 1998.

[29] K. Tammer. The application of parametric optimization and imbedding for the foundation and realization of a generalized primal decomposition approach. In J. Guddat, H. Jongen, B. Kummer, and F. Nozicka, editors, *Parametric Optimization and Related Topics, Mathematical Research*, volume 35, pages 376–386. Akademie-Verlag, Berlin, 1989.

[30] P. Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Mathematical Programming*, 59(1-3):231–248, 1993.

[31] A. S. M. Vamsi and N. Elia. Design of distributed controllers realizable over arbitrary directed networks. In *Proc. 49th IEEE Conf. Decision and Control (CDC)*, pages 4795–4800, 2010.

[32] P. Wan and M. D. Lemmon. Distributed Network Utility Maximization using Event-triggered augmented Lagrangian methods. In *American Control Conference, 2009. ACC '09.*, volume 1, pages 3298–3303, 2009.

[33] J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *CDC-ECE'11*, pages 3800–3805, 2011.

[34] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):131, 1968. doi: 10.1137/0306011.