



LUND UNIVERSITY

A Scalable Method for Continuous-Time Distributed Control Synthesis

Mårtensson, Karl; Rantzer, Anders

Published in:
American Control Conference (ACC), 2012

DOI:
[10.1109/ACC.2012.6314762](https://doi.org/10.1109/ACC.2012.6314762)

2012

[Link to publication](#)

Citation for published version (APA):
Mårtensson, K., & Rantzer, A. (2012). A Scalable Method for Continuous-Time Distributed Control Synthesis. In *American Control Conference (ACC), 2012* (pp. 6308-6313). IEEE - Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/ACC.2012.6314762>

Total number of authors:
2

General rights

Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

A Scalable Method for Continuous-Time Distributed Control Synthesis

Karl Mårtensson and Anders Rantzer

Abstract—In this paper a synthesis method for distributed controllers for continuous time distributed systems, is discussed. The systems considered consists of subsystems interconnected in a graph structure. This graph represents a communication structure of the system and hence governs the structure of the admissible controller, meaning that distributed controllers are considered. The objective of the synthesis is to obtain such admissible controllers that optimize a given performance. The method is scalable with respect to the size of the system and is therefore suitable for large-scale systems.

Distributed controllers are suboptimal with respect to centralized ones and it is desirable to measure the amount of performance degradation. Using the variables of the synthesis scheme, it is shown how to determine such a measure of suboptimality.

I. INTRODUCTION

Decision making when the decision makers have access to different information concerning underlying uncertainties has been studied since the late 1950s [9], [11]. The subject is sometimes called team theory, sometimes decentralized or distributed control. The theory was originally static, but work on dynamic aspects was initiated by Witsenhausen [17], who also pointed out a fundamental difficulty in such problems. Some special types of team problems were solved in the 1970's [15], [6], but the problem area has recently gain renewed interest. Spatial invariance was exploited in [1], [2], conditions for closed loop convexity were derived in [14], [13] and methods using linear matrix inequalities were given in [7], [12], [5].

In this paper we will focus on finding a solution to the linear quadratic regulator (LQR) problem for systems in continuous time. The method for finding the centralized solution for this problem has been known for a long time. However, when considering large-scale systems conventional methods for finding this solution are no longer tractable. The reason lies in that the computational time and memory requirements scales as $O(n^3)$ and $O(n^2)$, respectively, for these methods, see for example [4]. Methods for large-scale systems needs to exploit some structure in the problem. How to take advantage of the structure of a system with sparse dynamics matrices is presented in [3]. Here it is also assumed that the number of input signals is a lot less than the number of states. The resulting controllers will approximately solve the centralized control problem. When considering distributed systems, they usually have a communication constraint, meaning that a subsystem does not have access to the full state vector. With the use of

the augmented Lagrangian method, iterative schemes to find structured state feedback matrices minimizing a quadratic cost, are treated in [16], [8]. The methods are initialized with the centralized solution for the problem, and recursively the solution approaches a structured feedback matrix. The methods hinge on the solutions of Lyapunov equation and the solution of the centralized problem, thus not applicable to large-scale systems. In [10] an iterative distributed control synthesis scheme for discrete-time systems is considered. We will follow a similar approach when developing the theory in this paper. The synthesis method operates by iteratively updating the controller in a descent direction of the LQR performance. This direction is determined by simulating the system and the corresponding adjoint system. When the system matrices are sparse it is realized that this produces a scalable method, hence suitable for large-scale systems. Since distributed controllers are suboptimal compared to centralized solutions, it is desirable to have a measure of suboptimality. We show how to use the variables of the synthesis algorithm to determine a bound of suboptimality of the current controller.

Section II contains a description of the distributed systems considered and the notations used in the paper are defined. In section III the method for updating the control laws using descent directions to the cost function is presented. In section IV the theory for finding the suboptimality bound to the previously mentioned method, is formulated. An example is given in section V, showing the described methodology.

II. PROBLEM FORMULATION

The systems treated in this paper are continuous time, linear time invariant systems

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (1)$$

where $x(t) \in \mathbb{R}^m$, $u(t) \in \mathbb{R}^p$ and $x_0 \in \mathcal{N}(0, \sigma)$. We will assume that the system is distributed, a property that will be explained by a graph associated to the system. The vertices v_1, v_2, \dots, v_n of the graph represent subsystems or *agents* of the complete system. Hence, the vertices can be thought as a partition of all of the states. If the states are rearranged such that the states of each subsystem are next to each other, we write $x = [x_1^T \ x_2^T \ \dots \ x_n^T]^T$ where x_i are the states of subsystem i . The sparsity structure of the system is now defined by the edges of the graph. The collection of all edges is denoted by \mathcal{E} , where $(i, j) \in \mathcal{E}$ if there is an edge from v_i to v_j . By convention we assume that $(i, i) \in \mathcal{E}$ for all i . We call two distinct subsystems neighbors if there is an edge between the corresponding vertices. The edges describe the

The authors are with Automatic Control LTH, Lund University, Box 118, SE-221 00 Lund, Sweden, {karl, rantzer}@control.lth.se

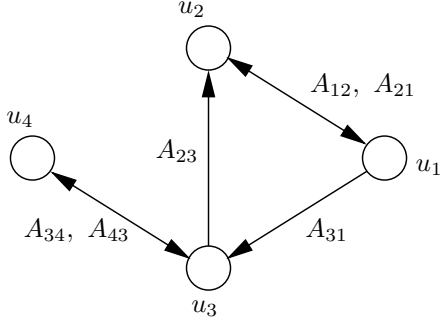


Fig. 1. Graphical representation of a distributed system. The arrows shows how each agent directly affects the others. The set $\mathcal{E} = \{(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 1), (1, 3), (3, 2), (3, 4), (4, 3)\}$

sparsity of the dynamics matrix and its structure is restricted by

$$A_{ij} = 0 \quad \text{if} \quad (j, i) \notin \mathcal{E},$$

(throughout the paper the subscript will denote blocks of the intended matrices corresponding to the subsystems). Hence, the dynamics of a subsystem is only directly affected by the states of the subsystem and its neighbors. The subsystems will also be assumed to have a distinct set of control signals, i.e. each control signal affects only one subsystem directly. This assumption is translated to assuming that the matrix B is block-diagonal, i.e. $B = \text{diag}(B_1, B_2, \dots, B_n)$.

With these initial definitions, an example of the setup is given in Figure 1.

The system (1) is controlled using state feedback $u(t) = -Kx(t)$. The graph also imposes a communication constraint on the system and the admissible controllers are only those where each subsystem uses only the states of itself and its neighbors to determine its control input. This restriction translates to a structure in the feedback matrix according to

$$K_{ij} = 0 \quad \text{if} \quad (j, i) \notin \mathcal{E}.$$

The subspace of admissible controllers is denoted by

$$\mathcal{K} = \{K \mid \forall K \text{ such that } K_{ij} = 0 \text{ if } (j, i) \notin \mathcal{E}\}.$$

The set of admissible *stabilizing* controllers is denoted by

$$\mathcal{K}_{\text{stab}} = \{K \mid \forall K \in \mathcal{K} \text{ and } K \text{ is stabilizing}\}.$$

With the restrictions of the matrices A , B and K , the closed loop dynamics matrix $A - BK$ has the same structure as A , and hence satisfies the structural constraint that the graph gives. That the closed loop dynamics matrix still is sparse will prove valuable later on when looking into the complexity of the coming method.

III. ITERATIVE DISTRIBUTED CONTROL SYNTHESIS

The objective of the control synthesis is to determine a feedback matrix that minimizes some performance. The

performance that is considered is the commonly known LQR cost for continuous time systems

$$J(K, x_0) = \int_0^\infty x(t)^T Q_x x(t) + u(t)^T Q_u u(t) dt, \quad (2)$$

where $x(t)$ satisfies the dynamics equation (1) and $u(t) = -Kx(t)$. The weights Q_x and Q_u are assumed to be block-diagonal meaning that it is possible to separate the cost into costs for each subsystem. For all stabilizing K , the cost (2) can be determined by solving certain Lyapunov equation. Specifically, we have that $J(K, x_0) = \text{tr}((Q_x + K^T Q_u K) X_0) = \text{tr}(P x_0 x_0^T)$ where X_0 and P are the solutions to the following Lyapunov equations, respectively.

$$(A - BK)X_0 + X_0(A - BK)^T + x_0 x_0^T = 0, \quad (3)$$

$$(A - BK)^T P + P(A - BK) + Q_x + K^T Q_u K = 0. \quad (4)$$

With these solutions an expression for the gradient of J with respect to K can also be determined.

Proposition 1: Given the system (1) and a stabilizing K , the gradient of the cost function (2) with respect to K is

$$\nabla_K J = 2(Q_u K - B^T P) X_0. \quad (5)$$

Proof. For simpler expression we define the matrices

$$A_K = A - BK, \\ M = Q_u K - B^T P.$$

By differentiating (4) with respect to K we get the following Lyapunov equation

$$A_K^T dP + dP A_K + dK^T M + M^T dK = 0.$$

The integral solution to this equation is

$$dP = \int_0^\infty e^{tA_K^T} (dK^T M + M^T dK) e^{tA_K} dt.$$

Now, since $dJ = \text{tr}(dP x_0 x_0^T)$, we get

$$\begin{aligned} dJ &= \text{tr} \left(\int_0^\infty e^{tA_K^T} (dK^T M + M^T dK) e^{tA_K} x_0 x_0^T dt \right) \\ &= 2 \cdot \text{tr} \left(dK^T M \int_0^\infty e^{tA_K} x_0 x_0^T e^{tA_K^T} dt \right) \\ &= 2 \cdot \text{tr} (dK^T M X_0). \end{aligned}$$

By using the relation about differentials

$$dZ = \text{tr}(dX^T \cdot Y) \implies \nabla_X Z = Y,$$

relation (5) is verified. \square

In order to calculate the gradient of J with respect to K using the result of Proposition 1, the Lyapunov equations (3) and (4) needs to be solved. If large-scale systems are considered it is not possible to solve these equations in a reasonable time. Hence, for a scalable method these solutions we need to find an expression that does not rely on these solutions. In the next proposition adjoint variables are introduced and it is shown how to use them to get rid of X_0 and P from (5).

Theorem 1: Given the system (1) and a stabilizing K , let the adjoint states λ be defined by

$$-\dot{\lambda}(t) = (A - BK)^T \lambda(t) - (Q_x + K^T Q_u K) x(t), \quad (6)$$

where $x(t)$ are the states of (1) and $\lim_{t \rightarrow \infty} \lambda(t) = 0$. Then

$$\nabla_K J = 2 \int_0^\infty (-Q_u u(t) + B^T \lambda(t)) x(t)^T dt. \quad (7)$$

Proof. By denoting $Q_K = Q_x + K^T Q_u K$, the adjoint states can be determined

$$\lambda(t) = \int_t^\infty -e^{(s-t)A_K^T} Q_K x(s) ds.$$

With this expression we can rewrite to following

$$\begin{aligned} \int_0^\infty \lambda(t) x(t)^T dt &= \int_0^\infty \int_t^\infty -e^{(s-t)A_K^T} Q_K x(s) ds \cdot x(t)^T dt \\ &= \int_0^\infty \int_t^\infty -e^{(s-t)A_K^T} Q_K e^{(s-t)A_K} x(t) ds \cdot x(t)^T dt \\ &= -P X_0. \end{aligned}$$

This relation and that $Q_u K X_0 = -Q_u \int_0^\infty u(t) x(t)^T dt$ fitted into (5) gives the desired result. \square

Remark 1: The dynamical system for the adjoint variables is stable when considering time going from the future backwards, i.e. from $t = \infty$ to $t = 0$. Hence, it is simulated in the backwards time direction.

The gradient gives a direction in which the feedback matrix K can be updated in to decrease the cost $J(K, x_0)$. Though, since we impose a structure on K , the gradient $\nabla_K J$ needs to be projected to the subspace \mathcal{K} defining this structure. This projected gradient will also produce a descent direction of $J(K, a)$. To understand this, consider the restriction of J on \mathcal{K} . The gradient of this function is exactly the projection of $\nabla_K J$ on \mathcal{K} .

In order to get a tractable algorithm, the time for simulating the states in (1) and (6) must be truncated to some finite time t_{final} . The truncation implies that an approximation of the gradient in Theorem 1 will be determined. The algorithm of iteratively updating the feedback matrix is given below.

Algorithm 1: Consider the system (1) with control $u(t) = -Kx(t)$ where $K \in \mathcal{K}_{\text{stab}}$. To find a local minimizer to (2), start with $K^{(0)} \in \mathcal{K}_{\text{stab}}$ and for each $\tau \geq 0$

- 1) Simulate the states $x(t)$ of (1) with control $u(t) = -K^{(\tau)} x(t)$ for times $t \in [0, t_{\text{final}}]$.
- 2) Simulate the adjoint states $\lambda(t)$ of (6) with for times $t \in [0, t_{\text{final}}]$ in the backwards time direction with $\lambda(t_{\text{final}}) = 0$.
- 3) For all agents i and all j such that $(j, i) \in \mathcal{E}$

I) Calculate

$$G_{ij} = 2 \int_0^{t_{\text{final}}} (-[Q_u]_i u_i(t) + B_i^T \lambda_i(t)) x_j(t)^T dt.$$

II) Update the feedback matrix

$$K_{ij}^{(\tau+1)} = K_{ij}^{(\tau)} - \gamma G_{ij},$$

for some step length γ .

- 4) Increase τ with 1 and goto 1).

Remark 2: As previously mentioned the closed loop matrix $A - BK$ follow the sparsity pattern described by the graph associated with the distributed system. Examining the matrix $Q_x + K^T Q_u K$ we find that it also has a distributed structure related to the graph. Hence, if the distributed system is sparse we understand that, using a sparse ODE solver, Algorithm 1 benefits from this structure. In fact, the scheme is linear in complexity when considering systems consisting of subsystems with the same average state space size and number of neighbors. This can be compared to solving Lyapunov equations which in standard implementation requires $\mathcal{O}(n^3)$ flops. This means that the alternate version of Algorithm 1 where the gradient instead is determined through (5) would not be tractable for large-scale systems.

Remark 3: In order to approximate the gradient in Algorithm 1 a final time t_{final} needs to be determined to ensure that the approximation is still a descent direction. For any descent direction D , $\text{tr}(\nabla_K J^T \cdot D) < 0$ must hold. Letting G be the truncated gradient and $H = \nabla_K J - G$. Then G is a descent direction if $\text{tr}((G + H)^T G) < 0$, that is $\text{tr}(G^T G) < \text{tr}(H^T G)$. Since $\text{tr}(G^T G)$ can be determined, a valid final time would be one for which it is possible to determine a bound on H in order for the inequality to hold. A strategy could be to analyse the decrease in the state trajectory to find such bound. This is an issue that needs further attention.

IV. SUBOPTIMALITY BOUND

Solving the ordinary LQR control problem is a well-studied problem and has a tractable solution when the system is of moderate size. But when we introduce restrictions in the structure of the feedback matrix, there is no general formula for finding the optimal one. The minimization problem is not even guaranteed to be convex. The underlying method in Algorithm 1 is a descent method, thus we can not guarantee that the globally optimal structured feedback matrix is ever attained. We only know that a locally optimal solution is reached. A measure of the suboptimality in each iteration step of the Algorithm 1, is $\alpha \geq 1$ such that

$$J(K, x_0) \leq \alpha J(K_{\text{opt}}, x_0), \quad (8)$$

where $K_{\text{opt}} = \arg \min_K J(K, x_0)$. The value of α tells us that the cost of the feedback matrix, $J(K, x_0)$, is within a factor α of the optimal value. If there is a way to verify that an α close to 1 must satisfy (8), then even though K might not be the optimal feedback matrix, we will not find one that reduces the cost greatly compared to this one. Hence, the suboptimality bound can be used as a stop criterion for Algorithm 1. That is when the bound goes below a given value we consider that the current feedback matrix is satisfactory and return it from the algorithm.

To determine the suboptimality bounds, we start by define

the truncated cost

$$\hat{J}(K, x_0, t_{\text{final}}) = \int_0^{t_{\text{final}}} x(t)^T Q_x x(t) + u(t)^T Q_u u(t) dt, \quad (9)$$

where $x(t)$ satisfies (1) and $u(t) = -Kx(t)$. The following theorem gives us a suboptimality bound telling us that in the time interval $[0, t_{\text{final}}]$ we are within a factor of α of the optimal solution on this interval.

Theorem 2: If $\alpha \geq 1$ is such that for a given trajectory of adjoint (or dual) variables $\lambda(t)$, with $\lambda(t_{\text{final}}) = 0$

$$\hat{J}(K, x_0, t_{\text{final}}) \leq \alpha \min_{\substack{x, u \\ x(0)=x_0}} \int_0^{t_{\text{final}}} \left[x(t)^T Q_x x(t) + u(t)^T Q_u u(t) + 2\lambda(t)^T (\dot{x}(t) - Ax - Bu(t)) \right] dt, \quad (10)$$

then

$$\hat{J}(K, x_0, t_{\text{final}}) \leq \alpha \hat{J}(K_{\text{opt}}, x_0, t_{\text{final}}), \quad (11)$$

where

$$K_{\text{opt}} = \arg \min_K \hat{J}(K, x_0, t_{\text{final}}).$$

Proof. Assume that α is such that for a given trajectory of $\lambda(t)$, (10) holds. We have that

$$\begin{aligned} \hat{J}(K_{\text{opt}}, x_0, t_{\text{final}}) &= \\ &= \begin{cases} \min_{K, x} \int_0^{t_{\text{final}}} x(t)^T (Q_x + K^T Q_u K) x(t) dt \\ \text{subject to: } \dot{x}(t) = (A - BK)x(t) \\ x(0) = x_0 \end{cases} \\ &\geq \begin{cases} \min_{x, u} \int_0^{t_{\text{final}}} x(t)^T Q_x x(t) + u(t)^T Q_u u(t) dt \\ \text{subject to: } \dot{x}(t) = Ax(t) - Bu(t) \\ x(0) = x_0 \end{cases} \\ &\geq \min_{\substack{x, u \\ x(0)=x_0}} \int_0^{t_{\text{final}}} \left(x(t)^T Q_x x(t) + u(t)^T Q_u u(t) + 2\lambda(t)^T (\dot{x}(t) - Ax(t) - Bu(t)) \right) dt, \end{aligned}$$

where the second inequality comes from introducing dual variables. Hence, if (10) holds, so must (11). \square

The Theorem 2 gives a method to evaluate the expected performance an updated feedback matrix will give to the system. We only have to choose the adjoint or dual variables. The name suggest that we choose the adjoint variables defined by (6). To motivate this choice, we could refer to Pontryagin's maximum principle. The motivation comes from examining the Hamiltonian

$$\max_{\lambda} \min_{x, u} \int_0^{t_{\text{final}}} \underbrace{\left(x(t)^T Q_x x(t) + u(t)^T Q_u u(t) + 2\lambda(t)^T (\dot{x}(t) - Ax(t) - Bu(t)) \right)}_{\mathcal{H}(x, u, \lambda)} dt \quad (12)$$

from Theorem 2. We let the objective function be denoted by $\mathcal{H}(x, u, \lambda)$. To find a saddle point for \mathcal{H} then

$$\begin{aligned} 0 &= \nabla_{x(t)} \mathcal{H} = 2(Q_x x(t) - \dot{\lambda}(t) - A^T \lambda(t)), \\ 0 &= \nabla_{u(t)} \mathcal{H} = 2(Q_u u(t) - B^T \lambda(t)). \end{aligned}$$

We get (6) by $\nabla_{x(t)} \mathcal{H} + K^T \nabla_{u(t)} \mathcal{H} = 0$ and replacing $u(t) = -Kx(t)$.

To show that it is actually possible to solve the minimization program in (10) we give the following proposition.

Proposition 2: The value of the minimization program in (10) can be determined by

$$- \int_0^{t_{\text{final}}} \left[\left(\dot{\lambda}(t) + A^T \lambda(t) \right)^T Q_x^{-1} \left(\dot{\lambda}(t) + A^T \lambda(t) \right) + \lambda(t)^T B Q_u^{-1} B^T \lambda(t) \right] dt - 2\lambda(0)^T x_0.$$

Proof. Introduce $f = Q_x x(t) - \dot{\lambda}(t) - A^T \lambda(t)$ and $g = Q_u u(t) - B^T \lambda(t)$. The integral in the objective in (10) is manipulated:

$$\begin{aligned} &\int_0^{t_{\text{final}}} \left[x(t)^T Q_x x(t) + u(t)^T Q_u u(t) + 2\lambda(t)^T (\dot{x}(t) - Ax - Bu(t)) \right] dt \\ &= \int_0^{t_{\text{final}}} \left[x(t)^T Q_x x(t) + u(t)^T Q_u u(t) - 2\dot{\lambda}(t)^T x(t) - \lambda(t)^T (Ax + Bu(t)) \right] dt \\ &\quad + 2 \left[\lambda(t)^T x(t) \right]_{t=0}^{t_{\text{final}}} \\ &= \int_0^{t_{\text{final}}} \left[f^T Q_x^{-1} f + g^T Q_u^{-1} g - \lambda(t)^T B Q_u^{-1} B^T \lambda(t) - \left(\dot{\lambda}(t) + A^T \lambda(t) \right)^T Q_x^{-1} \left(\dot{\lambda}(t) + A^T \lambda(t) \right) \right] dt \\ &\quad - 2\lambda(0)^T x_0, \end{aligned}$$

where the first equality comes from partial integration and the second equality from completing the squares. When minimizing the last expression we understand that choosing $x(t)$ and $u(t)$ to make $F = G = 0$ minimizes the integral (the only point on the trajectory of $x(t)$ that can not be chosen is $x(0)$ but this point does not change the value of the integral). \square

V. EXAMPLE

In this example we consider a small-scale system. The reason for not using a large-scale system is to be able to verify the actual cost of the feedback matrices by determining the corresponding Lyapunov solution. The system consists of 10 subsystems, each with one state, connected in a linear fashion as shown in Figure 2. This structure of the graph results in a tri-diagonal dynamics matrix. In this example its

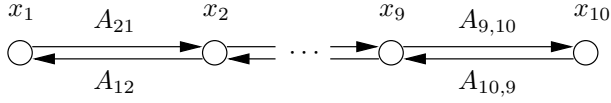


Fig. 2. Graphical representation of the system in the example. The arrows shows how each agent affects the others.

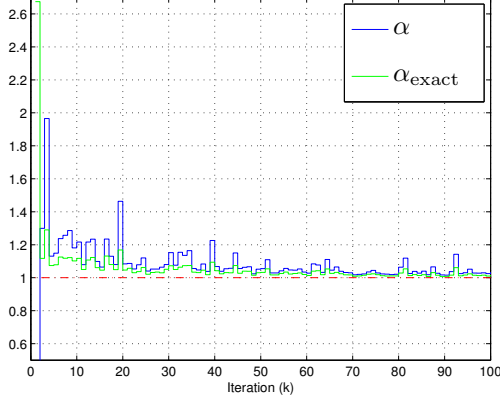


Fig. 3. Plots of the estimated sub-optimality using the described method and the exact suboptimality.

value is

$$A = \begin{bmatrix} -4 & 1 & & & & & & & & & \\ 1 & -4 & -1 & & & & & & & & \\ -2 & -4 & 1 & & & & & & & & \\ -1 & -8 & 8 & 2 & & & & & & & \\ & 1 & -2 & -9 & -1 & & & & & & \\ & & -1 & -9 & -1 & -2 & & & & & \\ & & & -2 & -5 & -2 & & & & & \\ & & & & -1 & -8 & -2 & & & & \\ & & & & & 7 & -10 & -1 & & & \\ & & & & & & -1 & -4 & & & \end{bmatrix}$$

and the remaining entries equal zero. The matrix B is diagonal and given by

$$B = \text{diag}([-1, 2, 6, -3, -6, -1, 3, -4, 7, 2]).$$

The weights $Q_x = Q_u = I$ and $x_0 \in \mathcal{N}(0, I)$.

The system is initially stable (the largest real part of the eigenvalues of A is -1.18), meaning that we can start Algorithm 1 with $K^{(0)} = 0$. The algorithm is used for 100 iterations where the systems are simulated between times $[0, 10]$ in each iteration. The method for estimating suboptimality bounds is performed in each update iteration. Also, the actual suboptimality is determined by solving the appropriate Lyapunov equation with the current feedback matrix K . The result of a simulation is given in Figures 3-4.

In Figure 3 the estimated suboptimal bound is denoted by α and shown in blue. A first remark is that in the first iteration we get a negative value of the suboptimality bound. This is due to the fact that the minimization program in (10) is not guaranteed to give a positive value. In case a negative value is obtained nothing can be said about the suboptimality with this method. Though, as we get closer to the optimal feedback matrix, the adjoint trajectory will approach the optimal (with respect to (12)) and the inequalities in the

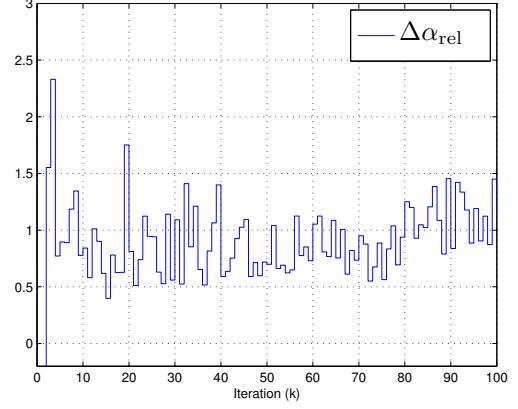


Fig. 4. Plot of the relative difference between the estimated and the exact suboptimality.

proof of Theorem 2 will almost be equal implying that we can expect a positive value from (10).

When positive, the suboptimality bound is always larger than 1 which is natural. In the same figure denoted by α_{exact} and shown in green, is the true suboptimality determined by

$$\alpha_{\text{exact}} = \frac{J(K^{(k)}, x_0)}{J(K_{\text{opt}}, x_0)}.$$

As expected the suboptimality bound (when positive) is also always larger than the true suboptimality. As the true suboptimality approaches 1, that is the cost with the feedback matrix approaches the optimal cost, the suboptimality bound also approaches 1.

In Figure 4 the relative difference between the suboptimality bound and the true suboptimality is shown. The relative difference is determined by

$$\Delta\alpha_{\text{rel}} = \frac{\alpha - \alpha_{\text{exact}}}{\alpha_{\text{exact}} - 1}.$$

As can be seen in the figure, the relative difference is in most iteration below 1.5 meaning that the suboptimality bound is not more than a factor 1.5 more from the true suboptimality (when the true suboptimality is measured as the distance to 1).

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

In this paper a scalable method for doing synthesis of distributed controllers for linear, continuous time, distributed systems. The objective is to obtain a linear, structured controller which minimizes the LQR cost. The method works in an iterative fashion where, in each iteration, a descent direction (with respect to the cost) is determined by simulating the system and the corresponding adjoint system. The controller is then updated by a step in that direction. The fact that the method relies on simulation of distributed, sparse system shows that it is scalable with respect to the number of subsystems.

In each iteration the trajectories are also used to determine a bound of the suboptimality of the current controller. This bound gives qualitative information of the current controller, and can for example be used as a stopping criteria for the synthesis method.

B. Future Works

How to determine a valid final time in the simulations to guarantee that the approximated gradient still is a descent direction, needs further attention. For more discussion, see Remark 3.

We will work on connecting the state feedback synthesis with the similar observer synthesis to get a method for output feedback synthesis. We will for example look at what happens to the suboptimality bounds in this case.

VII. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement number 224428, acronym CHAT. This work was also supported by the Linnaeus Grant LCCC from the Swedish Research Council.

REFERENCES

- [1] Bassam Bamieh, Fernando Paganini, and Munther A. Dahleh. Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control*, 47(7), July 2002.
- [2] Bassam Bamieh and Petros G. Voulgaris. A convex characterization of distributed control problems in spatially invariant systems with communication constraints. *Systems & Control Letters*, 54(6):575–583, June 2005.
- [3] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15:755–777, November 2008.
- [4] D. Bini, B. Iannazzo, and B. Meini. *Numerical Solution of Algebraic Riccati Equations*. Society for Industrial and Applied Mathematics, 2011.
- [5] Ather Gattami. Generalized linear quadratic control theory. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006.
- [6] Yu-Chi Ho and Kai-Ching Chu. Team decision theory and information structures in optimal control problems—Part I. *IEEE Transactions on Automatic Control*, 17(1):15–22, Feb 1972.
- [7] Cédric Langbort, Ramu Sharat Chandra, and Raffaello D’Andrea. Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control*, 49(9):1502–1519, September 2004.
- [8] Fu Lin, M. Fardad, and M.R. Jovanovic. Augmented lagrangian approach to design of structured optimal state feedback gains. *IEEE Transactions on Automatic Control*, 56(12):2923–2929, dec. 2011.
- [9] J. Marschak. Elements for a theory of teams. *Management Sci.*, 1:127–137, 1955.
- [10] Karl Mårtensson and Anders Rantzer. Sub-optimality bound on a gradient method for iterative distributed control synthesis. In *Proc. 19th International Symposium on Mathematical Theory of Networks and Systems*, Budapest, Hungary, July 2010.
- [11] R. Radner. Team decision problems. *Ann. Math. Statist.*, 33(3):857–881, 1962.
- [12] Anders Rantzer. A separation principle for distributed control. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006.
- [13] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Trans. on Automatic Control*, 51(2):274–286, Feb 2006.
- [14] Michael Rotkowitz and Sanjay Lall. Decentralized control information structures preserved under feedback. In *Proceedings of the IEEE Conference on Decision and Control*, December 2002.
- [15] Nils R. Sandell and Michael Athans. Solution of some nonclassical LQG stochastic decision problems. *IEEE Transactions on Automatic Control*, 19(2):108–116, April 1974.
- [16] C. Wenk and C. Knapp. Parameter optimization in linear systems with arbitrarily constrained controller structure. *IEEE Transactions on Automatic Control*, 25(3):496–500, jun 1980.
- [17] H.S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):138–147, 1968.