Sub-Optimality Bound on a Gradient Method for Iterative Distributed Control Synthesis

Karl Mårtensson and Anders Rantzer

***Abstract*— A previous paper introduced an online gradient method to iteratively update local controllers for improved performance. In this paper we modify that method to get an offline method for distributed control synthesis. The complexity of the method is linear in the number of neighbors to each agent.**

Since the controllers are constructed to be distributed and the method is an iterative scheme, the controllers will always be sub-optimal compared to a centralized controller. We describe a method to calculate bounds of the sub-optimality of the controllers, using the same variables that take part in the update scheme.

I. INTRODUCTION

Decision making when the decision makers have access to different information concerning underlying uncertainties has been studied since the late 1950s [8], [9]. 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 [14], who also pointed out a fundamental difficulty in such problems. Some special types of team problems were solved in the 1970's [13], [5], but the problem area has recently gain renewed interest. Spatial invariance was exploited in [1], [2], conditions for closed loop convexity were derived in [12], [11] and methods using linear matrix inequalities were given in [7], [10], [3].

In [6] an online iterative control synthesis scheme is considered. This paper uses a similar approach but instead focuses on an offline distributed control synthesis scheme. For a centralized control problem, the method would be a special case of iterative feedback tuning [4]. In this distributed setup, agents only have access to information of the local dynamic model and local states. The agent uses the local knowledge to locally change the control law in order to improve a global performance. The controllers will always be sub-optimal, both since the controllers are constructed to be distributed and since the method presented is an iterative scheme. Using the same variables determined in the controller update step, sub-optimality bounds can be found.

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

II. PROBLEM FORMULATION

A. Distributed Systems

Consider linear time-discrete systems

$$x(t+1) = \Phi x(t) + \Gamma u(t), \quad x(0) = x_0, \tag{1}$$

where $x_0 \in \mathcal{N}(0, \sigma)$. The systems are restricted to have a distributed structure, described by an associated graph. The nodes (called agents) v_i, $i = 1, \ldots, n$, of the graph represent subsystems of the complete system, that is the agents are a partition of the states of the system. The edges of the graph are represented by ordered pairs, (i, j). Let E be the collection of edges (by convention $(i, i) \in E,\ \forall i$). We call agents v_i and v_j neighbors if at least one of (i, j) and (j, i) are among the edges. An edge (i, j) means that agent v_i directly influence agent v_j through the dynamics of the system, that is

$$\Phi_{ji} = 0 \text{ if } (i, j) \notin E,$$

(throughout the paper, subscripts i, j will refer to blocks associated with agents i and j, respectively). Hence the dynamics matrix has a sparsity structure which resembles the graph structure of the distributed system. In the paper we assume that each agent has one set of distinct control signals, i.e. each control signal affects only one agent directly. This is represented by the Γ matrix being block-diagonal, that is,

$$\Gamma = \operatorname{diag}(\Gamma_1, \ldots, \Gamma_n),$$

where Γ_i is associated with agent v_i. The case that an agent does not have an input signal could be modeled as letting the corresponding block in Γ be zero. This is not necessary, and the columns corresponding to such zero entries in Γ will be removed. One example of the complete setup is found in Figure 1.

The system (1) is controlled using state feedback $u(t) = -Lx(t)$. When we consider a distributed setup, each agent v_i is restricted to use only the states of its neighboring agents to calculate its control $u_i(t)$. This imposes the restriction on the feedback matrix L,

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

With this restriction, the closed loop dynamics matrix $\Phi - \Gamma L$ satisfies the property that $[\Phi - \Gamma L]_{ij} = 0$ unless agent i and j are neighbors.

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

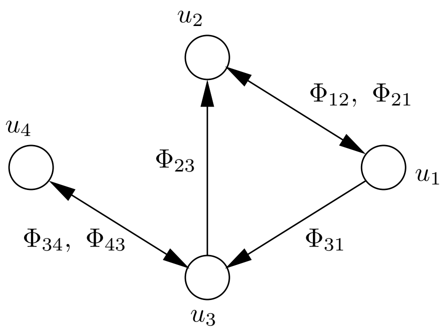

Fig. 1. Graphical representation of a distributed system. The arrows shows how each agent affects the others. The set $E = \{(1,1),(2,2),(3,3),(4,4),(1,2),(2,1),(1,3),(3,2),(3,4),(4,3)\}$

B. *Iterative Distributed Control Synthesis*

The performance we consider is the LQR cost

$$J(L, x_0) = \sum_{t=0}^{\infty} \left(|x(t)|_Q^2 + |u(t)|_R^2 \right), \qquad (2)$$

where $x(t)$ and $u(t) = -Lx(t)$ satisfies (1). This cost function is only defined for stabilizing feedback matrices. We assume that the cost can be separated to decentralized costs for each agent, meaning that Q and R are block-diagonal and

$$J(L, x_0) = \sum_i J^{(i)}(L, x_0) = \sum_i \sum_{t=0}^{\infty} \left(|x_i(t)|_{Q_i}^2 + |u_i(t)|_{R_i}^2 \right)$$

Using a similar approach as in [6] the following is stated.

Proposition 1: Given the system (1) and a stabilizing L, the gradient of the cost function (2) is

$$\nabla_L J = 2 \left[RL - \Gamma^T P(\Phi - \Gamma L) \right] X_0, \qquad (3)$$

where

$$X_0 = (\Phi - \Gamma L) X_0 (\Phi - \Gamma L)^T + x_0 x_0^T \qquad (4)$$
$$P = (\Phi - \Gamma L)^T P (\Phi - \Gamma L) + Q + L^T R L \qquad (5)$$

Proof. The cost function can be written as $J(L, x_0) = \text{tr}(X_0 Q) + \text{tr}(L X_0 L^T R) = \text{tr}(P x_0 x_0^T)$. To calculate the differential of P, let

$$\Phi_L = \Phi - \Gamma L$$
$$M = [L^T R - \Phi_L^T P \Gamma] dL$$

Differentiating (5) shows that dP satisfies the Lyapunov equation

$$dP = \Phi_L^T dP \Phi_L + M + M^T$$

Hence

$$dP = \sum_{k=0}^{\infty} (\Phi_L^T)^k (M + M^T) \Phi_L^k$$
$$\text{tr}(dP x_0 x_0^T) = \text{tr} \left(2M^T \sum_{k=0}^{\infty} \Phi_L^k x_0 x_0^T (\Phi_L^T)^k \right)$$
$$= \text{tr} \left(2 dL^T (RL - \Gamma^T P \Phi_L) X_0 \right)$$

This concludes the proof. □

Introducing adjoint states, we can rewrite this expression, which is summarized in the following Proposition.

Proposition 2: Given the system (1) and a stabilizing L, let the adjoint states λ be defined by the backwards iteration

$$\lambda(t-1) = (\Phi - \Gamma L)^T \lambda(t) - (Q + L^T R L) x(t), \qquad (6)$$

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

$$\nabla_L J = 2 \left(R L X_0 + \Gamma^T \sum_{t=0}^{\infty} \lambda(t) x(t)^T \right)$$

Proof. For simplicity, let $Q_L = Q + L^T R L$. For any j,

$$\lambda(j) = - \sum_{k=j+1}^{\infty} ((\Phi - \Gamma L)^T)^{k-j-1} Q_L x(k)$$
$$= - \sum_{k=0}^{\infty} ((\Phi - \Gamma L)^T)^k Q_L (\Phi - \Gamma L)^{k+1} x(j)$$

Hence (letting $\Phi_L = \Phi - \Gamma L$)

$$\sum_{t=0}^{\infty} \lambda(t) x(t)^T = - \sum_{t=0}^{\infty} \sum_{k=0}^{\infty} (\Phi_L^T)^k Q_L (\Phi_L)^{k+1} x(t) x(t)^T$$
$$= -P(\Phi - \Gamma L) X_0$$

Fitting this into (3) gives the desired result. □

Projecting the gradient to the set of admissible feedback matrices, we obtain a descent direction of $J(L)$. To determine the local update direction for an agent, each agent first simulates its states for a time interval $[0, N]$ by only sharing states and control signals with neighboring agents. Thereafter each agent simulates its adjoint states, $\lambda_i(t)$, backwards in time (ending with $\lambda(N) = 0$), by only sharing adjoint states. That this can be done using only local states and adjoint states can be found in [6]. With the adjoint states, each agent approximately determines the blocks in the gradient of the cost corresponding to its neighbors. The method is summarized in the following update scheme

Algorithm 1: At iterate k, let the feedback matrix be $L^{(k)}$ and let the time interval for the simulation be $[0, N]$. To update the feedback matrix in agent i

1) Let $x_0 \in \mathcal{N}(0, \sigma)$ and simulate the states $x_i(t)$ of the system (1) for times $t = 0, \dots, N$ by communicating states and control signals from and to neighboring agents.
2) Simulate the adjoint states $\lambda_i(t)$ of the system (6) for times $t = 0, \dots, N$ in the backwards direction (with $\lambda(N) = 0$), by communicating adjoint states from and to neighboring agents

$$\lambda_i(t-1) = \sum_{j \in E_i} \left(\Phi - \Gamma L^{(k)} \right)_{ji}^T \lambda_j(t)$$
$$- \left(Q_i x_i(t) - \sum_{j \in E_i} (L_{ji}^{(k)})^T R_j u_j(t) \right)$$

3) For every neighboring agent j, calculate the approximation of the projected gradient by

$$G_{ij} = -2\Bigg(R_i \sum_{t=0}^{N} u_i(t)x_j(t)^T + \Gamma_i^T \sum_{t=0}^{N} \lambda_i(t)x_j(t)^T\Bigg)$$

4) For each neighboring agent j, update

$$L_{ij}^{(k+1)} = L_{ij}^{(k)} - \gamma_k G_{ij},$$

for some step length γ_k.

5) Increase k and go to 1) and repeat. □

An important property of the posed scheme is that the complexity is linear in the number of neighbors to each agent. Introducing more agents to the system only changes the calculations for the agents which are to be neighbors to the new ones. Hence it does not involve much effort to add more agents to existing system.

III. SUB-OPTIMALITY BOUND

A. Centralized Sub-Optimality Bound Calculations

Solving the ordinary LQR control problem is a well-studied problem and has a tractable solution. But finding the minimizing feedback matrix, when imposing a structure, is not even guaranteed to be convex. The underlying method in Alorithm 1 is a descent method, and hence we can not guarantee that the optimal structured feedback matrix is ever reached. A measure of the sub-optimality in each iteration step of the update alforithm, is $\alpha \geq 1$ such that

$$J(L, x_0) \leq \alpha J(L_{\text{opt}}, x_0), \tag{7}$$

where $L_{\text{opt}} = \operatorname{argmin}_K J(K, x_0)$. That is, $J(L, x_0)$ is within a factor of α of the actual optimal value. This means that if we can verify that an α close to 1 must satisfy (7), then even though L might not be the optimal feedback matrix, we will not find one that reduces the cost greatly. For the remaining part of the paper

$$J(L, x_0) = \sum_{t=0}^{N} \left(|x(t)|_Q^2 + |u(t)|_R^2\right)$$

Theorem 1: If $\alpha \geq 1$ is such that for a given sequence of dual (or adjoint) variables $\lambda(t)$, with $\lambda(N) = 0$

$$J(L, x_0) \leq \alpha \sum_i \min_{x_i, u_i} \sum_{t=0}^{N} \Bigg[|x_i(t)|_{Q_i}^2 + |u_i(t)|_{R_i}^2 + 2\lambda_i(t)^T \left(x_i(t+1) - \Phi_i x - \Gamma_i u_i(t)\right) \Bigg], \tag{8}$$

then

$$J(L, x_0) \leq \alpha J(L_{\text{opt}}, x_0), \tag{9}$$

where

$$L_{\text{opt}} = \operatorname*{argmin}_K \min_x J(K, x_0)$$

Proof. Assume that α is such that for a given sequence of $\lambda(t)$, (8) holds. We have that

$$\begin{aligned} J(L_{\text{opt}}, x_0) &= \begin{cases} \displaystyle\min_{K,x} \sum_{t=0}^{N} |x(t)|_Q^2 + |Kx(t)|_R^2 \\ \text{subject to: } x(t+1) = (\Phi - \Gamma K)x(t) \end{cases} \\ &\geq \begin{cases} \displaystyle\min_{u,x} \sum_{t=0}^{N} |x(t)|_Q^2 + |u(t)|_R^2 \\ \text{subject to: } x(t+1) = \Phi x(t) - \Gamma u(t) \end{cases} \\ &\geq \min_{u,x} \sum_{t=0}^{N} \Bigg[|x(t)|_Q^2 + |u(t)|_R^2 \\ &\qquad + 2\lambda(t)^T (x(t+1) - \Phi x(t) - \Gamma u(t)) \Bigg], \end{aligned}$$

where the second inequality comes from introducing dual variables. Hence, if (8) holds, so must (9). □

This theorem relates the finite horizon cost $J(L, x_0)$ with the optimal finite horizon cost $J(L_{\text{opt}}, x_0)$. The interpretation then becomes that over the time interval $[0, N]$ we could not reduce the cost by more than a factor $\frac{1}{\alpha}$.

B. Distributed Sub-Optimality Bound Calculations

If we consider distributed control, that is, if we restrict the feedback matrix to be such that each agent only is allowed to use its neighboring states to determine its control action, then each agent can locally solve its corresponding minimization problem in (8). Hence, each agent can obtain a sub-optimality bound locally. We state the following Corollary to Theorem 1.

Corollary 1: Let the simulation interval be the time interval $[0, N]$. Let $J^{(i)}(L_i, x_0) = \sum_{t=0}^{N} |x_i(t)|_{Q_i}^2 + |u_i(t)|_{R_i}^2$. Now, given sequence of $\lambda(t)$, if for $1 \leq i \leq n$, $\alpha_i \geq 1$ is such that

$$J^{(i)}(L_i, x_0) \leq \alpha_i \min_{x_i, u_i} \sum_{t=0}^{N} \Bigg[|x_i(t)|_{Q_i}^2 + |u_i(t)|_{R_i}^2 + 2\lambda_i(t)^T \left(x_i(t+1) - \Gamma_i u_i(t)\right) + \sum_{(i,j)\in E} \lambda_j(t)^T \Phi_{ji} x_i(t) \Bigg], \tag{10}$$

then, letting $\alpha = \max_i \alpha_i$,

$$J(L, x_0) \leq \alpha J(L_{\text{opt}}, x_0) \tag{11}$$

Proof. Assume that (10) holds. We have that

$$J(L, x_0) = \sum_i J^{(i)}(L_i, x_0)$$

Using (10) and that $\alpha = \max_i \alpha_i$ we get right hand side of (8) and by Theorem 1 we have (11). □

The Corollary 1 gives a method to evaluate the expected performance an updated feedback matrix will give to the system. We only have to choose the dual or adjoint 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. Another motivation comes from examining

$$\max_{\lambda}\min_{u,x}\underbrace{\sum_{t=0}^{N}\Big[|x(t)|_Q^2+|u(t)|_R^2 +2\lambda(t)^T(x(t+1)-\Phi x(t)-\Gamma u(t))\Big]}_{\mathcal{L}(x,u,\lambda)},$$

from Theorem 1. Let the objective function be $\mathcal{L}(x,u,\lambda)$. To find a saddle point for $\mathcal{L}$ then

$$0=\nabla_{x(t)}\mathcal{L}=2(Qx(t)+\lambda(t-1)-\Phi^T\lambda(t))$$
$$0=\nabla_{u(t)}\mathcal{L}=2(Ru(t)-\Gamma^T\lambda(t))$$

We get (6) by $\nabla_{x(t)}\mathcal{L}+L^T\nabla_{u(t)}\mathcal{L}=0$.

Algorithm 2: Given the feedback matrix L, the $[0,N]$-horizon sub-optimality bounds in each agent are determined by

1) Simulate the states $x_i(t)$ of the system
$$x(t+1)=(\Phi-\Gamma L)x(t),$$
for times $t=0,\dots,N$.
2) Simulate the adjoint states $\lambda_i(t)$ of the system
$$\lambda(t-1)=(\Phi-\Gamma L)^T\lambda(t)-(Qx(t)+L^TRL)x(t),$$
for times $t=0,\dots,N$ in the backwards direction (with $\lambda(N)=0$.
3) Solve the optimization problem in (10) in every agent to find α_i. □

C. Modified Distributed Sub-Optimality Bound Calculations

The distributed sub-optimality bounds found in Corollary 1 does not equal the centralized one found in Theorem 1. Some of the bounds in Corollary 1 may even be much greater. In order get distributed sub-optimality bounds that approaches the centralized one, a modification to (10) is given. In each agent, a constant d_{ij} for every neighboring agent is added

$$J^{(i)}_{[0,N]}(L_i,x_0)\le\alpha_i\Bigg(\min_{x_i,u_i}\sum_{t=0}^{N}\Big[|x_i(t)|_{Q_i}^2+|u_i(t)|_{R_i}^2 + 2\lambda_i(t)^T\left(x_i(t+1)-\Gamma_iu_i(t)\right)+\sum_{(i,j)\in E}\lambda_j(t)^T\Phi_{ji}x_i(t)\Big] + \sum_{(i,j)\in E}d_{ij}\Bigg),$$

with the restriction that $d_{ij}=-d_{ji}$. The constants d_{ij} will be used to create consensus for all α_i. Using a consensus scheme, the d_{ij} are modified in order to make all α_i equal. In essence, if $\alpha_j<\alpha_i$ for two neighboring agents, then d_{ij} should be increased while at the same time d_{ji} should be equally decreased. Since d_{ij} are constant in the minimization, the minimization does only have to be carried out once for every iteration.

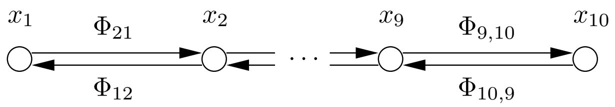

Fig. 2. Graphical representation of the system in the example. The arrows shows how each agent affects the others.

IV. EXAMPLE

The system

$$x(t+1)=\Phi x(t)+\Gamma u(t)$$

that is considered, consists of 10 agents, where the agents are connected in a linear fashion, see Figure 2. This leads to a tri-diagonal dynamics matrix, which, in this example, is

$$\Phi=\begin{bmatrix}
0.5 & 0.5 & & & & & & & & \\
-0.5 & 0.1 & -0.3 & & & & & & & \\
 & 0.4 & -0.2 & -0.5 & & & & & & \\
 & & -0.4 & -0.5 & 0.2 & & & & & \\
 & & & 0.2 & 0.3 & -0.1 & & & & \\
 & & & & -0.3 & 0.1 & 0.3 & & & \\
 & & & & & 0.2 & -0.4 & -0.4 & & \\
 & & & & & & 0.2 & -0.2 & 0.3 & \\
 & & & & & & & 0.5 & -0.5 & 0.3 \\
 & & & & & & & & -0.1 & -0.1
\end{bmatrix}$$

and with the remaining entries equal to zero. We allow each agent to have an input and set $\Gamma=I$. We wish to minimize the cost

$$J(L,x_0)=\sum_{t=0}^{N}\left(|x(t)|_Q^2+|u(t)|_R^2\right)$$

where $u=-Lx$, $Q=R=I$ and $x_0\in\mathcal{N}(0,I)$.

The magnitude of the maximal eigenvalue of A, $\rho(A)\approx 0.81$, hence we can initially let the system be uncontrolled, i.e. let $L=0$.

In Figures 3-4 the evolution of the sub-optimality when simulating the gradient method for 50 iteration. The horizon $N=10$ in the simulation. In Figure 3 the curve α is the sub-optimality bound determined by each agent when the modified suc-optimality calculations are performed. The curve denoted by α_{exact} shows the true sub-optimality the feedback matrix in that iteration gives rise to, that is

$$\alpha_{\text{exact}}=\frac{J(L^{(k)},x_0)}{J(L_{\text{opt}},x_0)}$$

In Figure 4 the relative difference between α and α_{exact} is shown, that is

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

When the feedback matrix is far from the optimal, in Theorem 1 it is not guarantee that the right hand side is positive. This happens when the feedback matix is far from the optimal. In the case of a negative right hand side, there is no positive α satisfying the inequality. In these cases, no value of α is ploted.

In Figure 3 we see that the true sub-optimality approaches 1, meaning that the cost of the distributed feedback matrix $L^{(k)}$ approaches the cost of the optimal centralized LQR feedback matrix. After a few iterations we see that we can calculate a sub-optimality bound using the theory previously described. Figure 4 shows that this bound is within a factor 3 of the true sub-optimality.

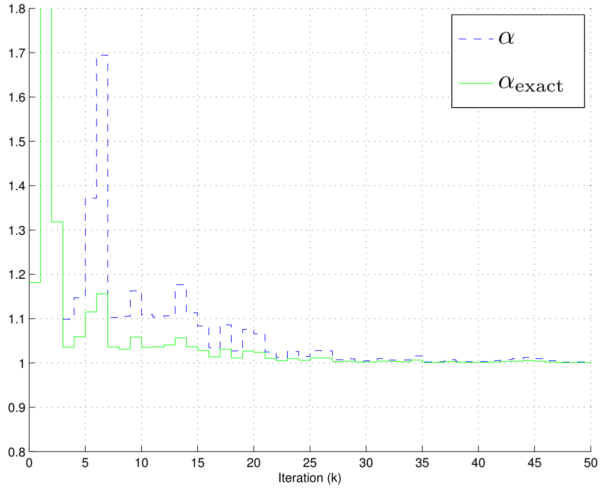

Fig. 3. Something

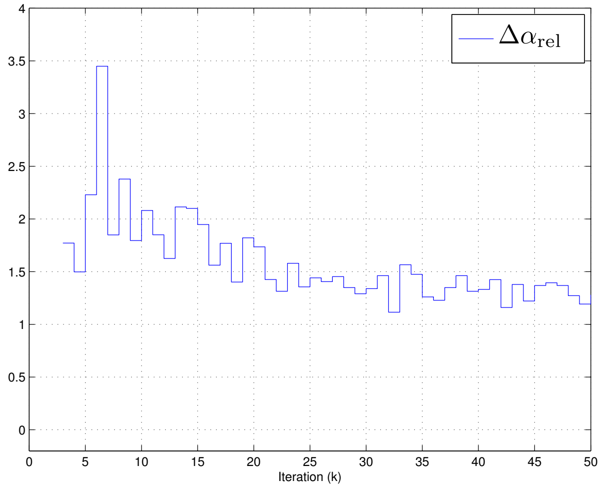

Fig. 4. Something

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

By introducing adjoint states, it is shown how to find a descent direction to a globally defined cost function, using only local information. Using this, a method to do offline distributed control synthesis is posed. The method can be used to update distributed controllers connected in a network, in order to improve a globally defined objective of the global system.

It is also shown how to use the same adjoint variables to calculate sub-optimality bounds to the current set of controllers, in a distributed fashion. The sub-optimality bounds can for example be used as a stoping criteria for the synthesis method.

B. Future Works

Using an approach similar to the one used in the sub-optimality calculations, we could have a scheme to determine an appropriate step length in the descent direction in the synthesis method. This would allow for a faster convergence.

Looking at the possiblilty to reformulate the minimization step in the calculations of the sub-optimality bounds to exclude negative sub-optimality bounds.

VI. 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 a Linnaeus Grant 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] Ather Gattami. Generalized linear quadratic control theory. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006.

[4] Håkan Hjalmarsson. Iterative feedback tuning –an overview. *International Journal on Adaptive Control and Signal Processing*, January 2002.

[5] 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.

[6] Anders Rantzer Karl Mårtensson. Gradient methods for iterative distributed control. In *Proceedings of 48th IEEE Conference on Decision and Control*, Shanghai, China, 2009.

[7] Cdric 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] J. Marshak. Elements for a theory of teams. *Management Sci.*, 1:127–137, 1955.

[9] R. Radner. Team decision problems. *Ann. Math. Statist.*, 33(3):857–881, 1962.

[10] Anders Rantzer. A separation principle for distributed control. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006.

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

[12] Michael Rotkowitz and Sanjay Lall. Decentralized control information structures preserved under feedback. In *Proceedings of the IEEE Conference on Decision and Control*, December 2002.

[13] 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.

[14] H.S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):138–147, 1968.