

Iterative optimal feedback control design under relaxed rigidity constraints for multi-robot cooperative manipulation

Dominik Sieber*, Frederik Deroo* and Sandra Hirche

Abstract—Cooperative manipulation of multiple robots presents an interesting control application scenario of coupled dynamical systems with a common goal. Here, we treat the problem of moving a formation of physically interconnected robots to a desired goal while maintaining the formation. This control problem is for example relevant in cooperative transport of an object from an initial to a final configuration by mobile robotic manipulators. To achieve the control goal we formulate an LQR-like optimal control problem that, in addition to goal regulation and minimization of input energy, includes the formation rigidity constraint in a relaxed form expressed as a biquadratic penalty term. The control problem is solved by two different iterative algorithms, a gradient descent using adjoint states and a quasi-Newton method, that determine a static linear state-feedback matrix. The proposed control design and the iterative algorithms are validated and compared in numerical simulations showing the efficacy of both approaches.

I. INTRODUCTION

Technological advances in communications and robotics have inspired researchers to investigate the interaction of several mobile robots to achieve a common goal. This includes for example the situation where the group of robots collaboratively transport an object from an initial point to a specified end point. The benefits of cooperative object manipulation are higher manipulation performance in terms of higher payload, geometric extension of the object, and potential synergies in the sensing and actuation capacities compared to a single robot. An important issue of mobile cooperative manipulation is the desired rigidity of the robot formation in order to ensure that the object is not deformed. Our goal here is to design an optimal control law that both maintains rigidity and regulates the desired goal configuration. Both design goals are essential when multiple mobile robot manipulators cooperatively drive an object from an initial to a final configuration without squeezing the object.

Many recent research activities on multi-robot problems focus on distributing motion commands of interconnected robots based on the desired movement of a rigid body using advanced robot dynamics [1]–[3]. Since contact forces are always present in the cooperation of robots, their inclusion into the control scheme facilitates a desired compliant interaction, called impedance control. These impedance-based control schemes are distributed among several cooperating manipulators in [4] including a stability analysis. A second research direction treats formation control with agents with

integrator dynamics. In [5], a biquadratic, positive definite function is used for the leader-follower approach in mobile robots with unicycle dynamics. Using the rigidity constraint in the control design in combination with a leader-follower network is presented in [6]. A general framework for maintaining the formation of agents with integrator dynamics is employed for directed triangle formations in [7], [8].

The intention of this work is to bring these two research directions closer by considering a formation control problem with robotic manipulators with advanced dynamics. The contribution of this work includes the derivation of advanced system dynamics as they appear in multi-robot cooperative manipulation. We then formulate an optimal control problem which takes both control goals of formation rigidity and goal regulation into account. By relaxing the formation rigidity constraint of a cooperative multi-robot transportation we obtain a biquadratic cost functional, closely resembling the LQR cost functional. However, because of the specific form of the cost functional, standard LQR methods are not applicable. Hence, we present two iterative algorithms to determine the feedback matrix, inspired by results in [9], [10]. These iterative algorithms are based on the simulation of trajectories of the system state and an adjoint state. Numerical simulations demonstrate the effectiveness of the approach. A full-scale experimental evaluation of the proposed algorithm with multiple mobile manipulators is presented in [11].

The remainder of this paper is structured as follows. Section II describes the system dynamics of a cooperative multi-robot team. Furthermore a control problem is presented which relaxes the rigid formation constraint into a biquadratic cost term. Two algorithms to compute the feedback law are presented in Section III. An evaluation of the approach through simulations is shown in Section IV.

Notation: Bold symbols denote vectors. The pose of the i -th end-effector frame Σ_i is expressed in a world coordinate system Σ_w . An object-centered frame Σ_o is aligned to the principal axes of the object. The matrix R_o^i describes the rotation of Σ_i relative to Σ_o . The vectorization of a matrix is denoted by $\text{vec}(\cdot)$, the inverse operation of reshaping a vector into a matrix of appropriate dimensions is denoted by $\text{mat}(\cdot)$. The identity matrix of dimension n is signified by I_n . The zero matrix of dimension $m \times n$ is denoted by $0_{m \times n}$.

II. PROBLEM FORMULATION

A system model for cooperative multi-robot cooperation is presented in this section and forms the system dynamics. Then, based on the dynamics, the control goal considered in this paper is formulated.

* Dominik Sieber and Frederik Deroo contributed equally to this work.

All authors are with the Institute for Information-Oriented Control, Department of Electrical Engineering and Information Technology, Technische Universität München, D-80290 Munich, Germany. {dominik.sieber, fred.deroo, hirche}@tum.de

A. Multi-robot cooperation - a state space model

We consider a team of cooperating robots manipulating a single object. The dynamics of each individual robot $i = 1, \dots, N$ evolves according to the feedback linearized system dynamics incorporating impedance control given by [12]

$$M_i \ddot{\xi}_i + D_i (\dot{\xi}_i - \dot{\xi}_{i,d}) + K_i (\xi_i - \xi_{i,d}) = \mathbf{f}_d - \mathbf{f}_i, \quad (1)$$

where $\xi_i \in \mathbb{R}^n, i = 1, \dots, N$ is the Cartesian position of the individual robot, and its time derivatives $\dot{\xi}$ and $\ddot{\xi}$ are velocity and acceleration, respectively. The positive definite impedance parameters inertia M_i , damping D_i , and stiffness $K_i \in \mathbb{R}^{n \times n}$ are chosen constant. The movement of each robot is driven by the desired robot velocity $\dot{\xi}_d$ and the desired force of each manipulator \mathbf{f}_d , representing the control inputs. The coupling between the subsystems in (1) is expressed through the force $\mathbf{f}_i \in \mathbb{R}^n$ at the i th manipulator which results not only from environment contact but also from the coupling of the robots through the object.

The force $\mathbf{f}_i = \frac{1}{N}(\mathbf{f}_{\text{motion}} + \mathbf{f}_{\text{ext}}) + \mathbf{f}_{i,\text{int}}$ is composed of the interaction forces of all n robots $\mathbf{f}_{i,\text{int}}$, called internal force, of the rigid body dynamics of the object $\mathbf{f}_{\text{motion}}$, and of environmental contact force \mathbf{f}_{ext} , called external force. With the factor $\frac{1}{N}$, we assume that the force is distributed equally among the cooperating robots which is valid for equal impedance parameters M_i, D_i , and K_i . Since we assume the object dynamics to be accurately known, $\mathbf{f}_{\text{motion}}$ is neglected and no external force \mathbf{f}_{ext} acting on the object is considered, i.e. $\mathbf{f}_i = \mathbf{f}_{i,\text{int}}$. In order to express $\mathbf{f}_{i,\text{int}}$ in terms of ξ and ξ_d , we make the following assumption.

Assumption 1: The force $\mathbf{f}_{i,\text{int}}$ can be approximated as

$$\begin{aligned} \mathbf{f}_{i,\text{int}} = & K_i (\xi_{i,d} - \xi_i) \\ & - R_o^o G_i^\dagger \sum_{j=1}^n G_j (R_o^w K_j (\xi_{j,d} - \xi_j)), \end{aligned} \quad (2)$$

where $G = [G_1, \dots, G_N] \in \mathbb{R}^{n \times nN}$ is the grasp matrix describing the relation between the forces of each robot frame and the object frame, G^\dagger is its Pseudo-inverse, and R_o^w is the orthonormal transformation matrix from Σ_o to Σ_w . This is valid since movements in multi-robot manipulation are generally not high-speed, yielding vanishing terms w.r.t D_i and M_i .

Let $\mathbf{x}_i = \left[\left(\int_{t_0}^t \dot{\xi}_{i,d} dt \right)^T, \xi_i^T, \dot{\xi}_i^T \right]^T$ be the system state and $\mathbf{u}_i = \left[\dot{\xi}_{i,d}^T, \mathbf{f}_{i,d}^T \right]^T$ be the control input. Then by combining (1) and (2) a state space model for the single manipulator in cooperation is given by

$$\dot{\mathbf{x}}_i = A_{ii} \mathbf{x}_i + B_{ii} \mathbf{u}_i + \sum_{j \in \{1 \dots N\} \setminus \{i\}} A_{ij} \mathbf{x}_j, \quad (3)$$

with

$$A_{ii} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ \frac{1}{N} M_i^{-1} K_i & -\frac{1}{N} M_i^{-1} K_i & -\frac{1}{N} M_i^{-1} D_i \end{pmatrix}, \quad (4)$$

$$A_{ij} = M_i^{-1} R_o^o G_i^\dagger \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ G_{j0} R_o^w K_j & -G_j R_o^w K_j & 0 \end{pmatrix}, \quad (5)$$

and

$$B_{ii} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ M_i^{-1} D_i & -M_i^{-1} \end{pmatrix}, \quad (6)$$

where A_{ii} is the system matrix of a single decoupled robotic system, B_{ii} is the input matrix, and A_{ij} is the interaction matrix of system j to system i . By stacking the individual robot states into $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_N^T]^T \in \mathbb{R}^{3nN}$ and the individual control inputs into $\mathbf{u} = [\mathbf{u}_1^T, \dots, \mathbf{u}_N^T]^T \in \mathbb{R}^{2nN}$ we obtain the system dynamics for the multi-robot cooperative system in a standard LTI system form

$$\dot{\mathbf{x}} = A \mathbf{x} + B \mathbf{u}. \quad (7)$$

In the following, we consider the complete system matrix A which consists of the blocks from (4) and (5) and is thus written as $A = [A_{ij}]$. Equivalently, the input matrix results in $B = \text{diag}(B_{11}, \dots, B_{NN})$.

B. Control goal

We want to design a linear state feedback control law $\mathbf{u} = -K \mathbf{x}$ which optimally drives a formation of interconnected robots described by the system dynamics (7) from an initial configuration \mathbf{x}_0 to a desired end point \mathbf{x}_e while maintaining the initial formation. This control goal is formulated in an LQR-like setting with rigidity constraint.

In order to derive the rigidity constraint consider the set of edges E such that the structure is rigid during the movement phase. Rigidity of the formation is described by an edge function $f(\mathbf{x}) = (\dots, \|\xi_i - \xi_j\|_2^2, \dots) \in \mathbb{R}^{|E|}$ which is required to satisfy $f(\mathbf{x}) = \mathbf{p}$. The state variable \mathbf{x} is concatenated by \mathbf{x}_i and thus involves the manipulator positions ξ_i . Here, $\mathbf{p} = (\dots, p_{ij}, \dots)$ is the desired rigid distance between the manipulators, and therefore considered to be constant in the following. Differentiating f w.r.t time leads to

$$(\xi_i - \xi_j)^T (\dot{\xi}_i - \dot{\xi}_j) = 0 \quad \forall (i, j) \in E. \quad (8)$$

The geometrical interpretation of (8) is that the difference in position between two linked robots is orthogonal to the difference in velocity. In order to include the rigidity condition (8) into our LQR-like setting, we transform it into a quadratic term of the states. Thus (8) is written as $\bar{\mathbf{x}}_{i,j}^T Q_{ij} \bar{\mathbf{x}}_{i,j}$ with $\bar{\mathbf{x}}_{i,j} = [x_i^T, x_j^T]^T$ by defining the blocks

$$\begin{aligned} [q_{ii}] = & \begin{pmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & \frac{1}{2} I_n \\ 0_{n \times n} & \frac{1}{2} I_n & 0_{n \times n} \end{pmatrix} \quad \forall (i, j) \in E \quad \text{and} \\ [q_{ij}] = [q_{ji}] = & \begin{pmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & -\frac{1}{2} I_n \\ 0_{n \times n} & -\frac{1}{2} I_n & 0_{n \times n} \end{pmatrix} \quad \forall (i, j) \in E. \end{aligned} \quad (9)$$

The resultant matrix $Q_{ij} = \begin{bmatrix} q_{ii} & q_{ij} \\ q_{ji} & q_{jj} \end{bmatrix}$ is symmetric but indefinite and thus it cannot be employed in a standard LQR

problem directly. Since the equality constraint described in (8) can be violated in both directions, the indefiniteness of $\bar{x}_{i,j}^T Q_{ij} \bar{x}_{i,j}$ is obvious, and its apparent global minimum is $-\infty$. The biquadratic term $(\bar{x}_{i,j}^T Q_{ij} \bar{x}_{i,j})^2$ on the other hand has a minimum of 0, and is thus suitable to be included in an optimization as a penalty term. Minimizing $(\bar{x}_{i,j}^T Q_{ij} \bar{x}_{i,j})^2$ for all $(i,j) \in E$ relaxes the rigidity equality constraint (8) into a minimization problem. Proper partitioning allows writing $(\bar{x}_{i,j}^T Q_{ij} \bar{x}_{i,j})^2$ as

$$(\mathbf{x}^T Q_k \mathbf{x})^2 \forall k \in \{1, \dots, ||E||\}. \quad (10)$$

This gives the first part of the desired cost functional. In order to achieve goal regulation to the desired end point \mathbf{x}_e , we employ the standard transformation of \mathbf{x} into

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_e, \quad (11)$$

and including the term $\tilde{\mathbf{x}}^T \tilde{Q} \tilde{\mathbf{x}}$ into the cost functional. It is important to note however that the rigidity condition (10) still needs to be satisfied in the original coordinate system \mathbf{x} . In order to combine both coordinate systems in the same cost functional we introduce an extended state vector

$$\bar{\mathbf{x}} = (\tilde{\mathbf{x}}^T, 1)^T. \quad (12)$$

Thus, we reformulate the rigidity penalty term (10) into

$$\begin{aligned} & (\mathbf{x}^T Q_k \mathbf{x})^2 \forall k \in \{1, \dots, ||E||\} \\ &= \left(\begin{pmatrix} \tilde{\mathbf{x}} \\ 1 \end{pmatrix}^T \begin{pmatrix} Q_k & Q_k \mathbf{x}_e \\ \mathbf{x}_e^T Q_k & \mathbf{x}_e^T Q_k \mathbf{x}_e \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}} \\ 1 \end{pmatrix} \right)^2 \\ &= (\bar{\mathbf{x}}^T \bar{Q}_k \bar{\mathbf{x}})^2 \forall k \in \{1, \dots, ||E||\}. \end{aligned} \quad (13)$$

We can now combine all of the terms into one cost functional and restate our control goal.

The goal of our optimal control problem is to find a control law $\mathbf{u} = -\bar{K} \bar{\mathbf{x}}$ with structure $\bar{K} = [\bar{K}, 0_{2nN \times 1}]$ in order to minimize the following cost functional

$$\begin{aligned} J = & \bar{\mathbf{x}}(T)^T S \bar{\mathbf{x}}(T) + \int_0^T \sum_{k=1}^{||E||} (\bar{\mathbf{x}}^T(t) \bar{q}_k \bar{Q}_k \bar{\mathbf{x}}(t))^2 \\ & + \mathbf{u}^T(t) R \mathbf{u}(t) + \bar{\mathbf{x}}^T \bar{Q} \bar{\mathbf{x}} dt, \end{aligned} \quad (14)$$

where S and \bar{Q} have the structure $S = \text{diag}(\tilde{S}, 0)$ and $\bar{Q} = \text{diag}(\tilde{Q}, 0)$ in order not to penalize the additional 1-state, \bar{Q}_k is given in (13) and $\bar{q}_k \geq 0$ is a scalar weighting factor. The zero column in \bar{K} is used to discard the augmented 1-state. The term $\bar{\mathbf{x}}(T)^T S \bar{\mathbf{x}}(T)$ represents the final penalty term resulting from the distance between \mathbf{x} and \mathbf{x}_e for the final time T . An indirect influence on the control input constraints is realized by $\mathbf{u}^T(t) R \mathbf{u}(t)$. This cost functional represents our combined control goals of maintained formation, through the term $\sum_{k=1}^{||E||} (\bar{\mathbf{x}}^T(t) \bar{Q}_k \bar{\mathbf{x}}(t))^2$, and goal regulation, through the term $\bar{\mathbf{x}}^T \bar{Q} \bar{\mathbf{x}}$.

III. MAIN RESULTS

In this section, we present two algorithms to achieve our control goal described by the cost functional (14). Furthermore, an idea is presented to alleviate the local character of the resulting control law.

A. Optimal control design via gradient descent method using adjoint states

In this subsection we describe a solution algorithm to determine a suboptimal feedback to minimize the biquadratic cost functional (14), inspired by the results in [10]. While there is a linear relationship in the standard LQR problem between the primal states \mathbf{x} and the adjoint states $\boldsymbol{\lambda}$ given by $\boldsymbol{\lambda}(t) = P \mathbf{x}(t)$, allowing for the solution to use a Riccati equation for the matrix P , this is not the case here. Because of the biquadratic term in the cost functional we are forced to use an alternative method based on simulated trajectories which is explained in the following.

Given the biquadratic cost functional (14) we want to iteratively determine the optimal state-feedback law. The corresponding Lagrangian function of the problem is

$$\begin{aligned} L = & \bar{\mathbf{x}}(T)^T S \bar{\mathbf{x}}(T) + \int_0^T \sum_{k=1}^{||E||} (\bar{\mathbf{x}}^T(t) \bar{q}_k \bar{Q}_k \bar{\mathbf{x}}(t))^2 + \bar{\mathbf{x}}^T \bar{Q} \bar{\mathbf{x}} \\ & + \bar{\boldsymbol{\lambda}}^T(t) (\dot{\bar{\mathbf{x}}}(t) - (\bar{A} - \bar{B} \bar{K}) \bar{\mathbf{x}}(t)) \\ & + \bar{\mathbf{x}}^T(t) \bar{K}^T R \bar{K} \bar{\mathbf{x}}(t) dt + \bar{\boldsymbol{\mu}}(\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}_0), \end{aligned} \quad (15)$$

where $\bar{A} = \text{diag}(A, 0)$ and $\bar{B} = [B^T, 0_{1 \times 2nN}^T]^T$.

Partial integration of (15) gives

$$\begin{aligned} L = & \bar{\mathbf{x}}(T)^T S \bar{\mathbf{x}}(T) + \int_0^T \sum_{k=1}^{||E||} (\bar{\mathbf{x}}^T(t) \bar{q}_k \bar{Q}_k \bar{\mathbf{x}}(t))^2 \\ & + \bar{\mathbf{x}}^T(t) \bar{K}^T R \bar{K} \bar{\mathbf{x}}(t) + \bar{\mathbf{x}}^T \bar{Q} \bar{\mathbf{x}} - \bar{\mathbf{x}}^T(t) \dot{\bar{\boldsymbol{\lambda}}}(t) \\ & - \bar{\mathbf{x}}^T(t) (\bar{A} - \bar{B} \bar{K})^T \bar{\boldsymbol{\lambda}}(t) dt \\ & + [\bar{\boldsymbol{\lambda}}(t)^T \bar{\mathbf{x}}(t)]_0^T + \bar{\boldsymbol{\mu}}(\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}_0). \end{aligned} \quad (16)$$

We can derive equations for the adjoint state through the optimality condition $\frac{\partial L}{\partial \bar{\boldsymbol{\lambda}}} = 0$. This gives

$$\begin{aligned} \dot{\bar{\boldsymbol{\lambda}}}(t) = & (\bar{A} - \bar{B} \bar{K})^T \bar{\boldsymbol{\lambda}}(t) - 2 \bar{K}^T R \bar{K} \bar{\mathbf{x}}(t) - 2 \bar{Q} \bar{\mathbf{x}}(t) \\ & - 4 \sum_{k=1}^{||E||} (\bar{\mathbf{x}}^T(t) \bar{q}_k \bar{Q}_k \bar{\mathbf{x}}(t)) \bar{q}_k \bar{Q}_k \bar{\mathbf{x}}(t) \end{aligned} \quad (17a)$$

$$\bar{\boldsymbol{\lambda}}(T) = -2S \bar{\mathbf{x}}(T), \quad \bar{\boldsymbol{\mu}} = -\bar{\boldsymbol{\lambda}}(0). \quad (17b)$$

The last equation gives justification that $\bar{\boldsymbol{\lambda}}(0)$ is free.

Proposition 1: The gradient of the cost functional with respect to the feedback matrix \bar{K} is

$$\nabla_{\bar{K}} J = \int_0^T 2R \bar{K} \bar{\mathbf{x}}(t) \bar{\mathbf{x}}^T(t) + B^T \bar{\boldsymbol{\lambda}}(t) \bar{\mathbf{x}}^T(t) dt. \quad (18)$$

Proof: The gradient is determined from the Lagrange function. We get

$$\begin{aligned} \nabla_{\bar{K}} J \bullet H = & \int_0^T 2\bar{\mathbf{x}}^T(t) H^T R \bar{K} \bar{\mathbf{x}}(t) + \bar{\mathbf{x}}^T(t) H^T B^T \bar{\boldsymbol{\lambda}}(t) dt \\ = & \int_0^T 2R \bar{K} \bar{\mathbf{x}}(t) \bar{\mathbf{x}}^T(t) + B^T \bar{\boldsymbol{\lambda}}(t) \bar{\mathbf{x}}^T(t) dt \bullet H, \end{aligned}$$

where H is a variation in \bar{K} , and where \bullet denotes the Frobenius inner product. \blacksquare

The feedback matrix is then iteratively determined using the following algorithm.

Algorithm 1:

- 1) Simulate the states $\bar{\mathbf{x}}(t)$ for the finite horizon T .
- 2) Simulate the adjoint states $\bar{\boldsymbol{\lambda}}(t)$ for the same horizon according to (17).
- 3) Compute the gradient according to (18).
- 4) Update the feedback matrix

$$\bar{K}^{(k+1)} = \bar{K}^{(k)} - \gamma_k \nabla_{\bar{K}} J^{(k)},$$

where γ_k is a scalar step length.

- 5) If $\left| \frac{J^{(k)} - J^{(k-1)}}{J^{(k-1)}} \right| < \epsilon$, stop. Otherwise, increase k and go back to step 1.

The choice of the step size γ_k is important for the speed of convergence. A popular method is to choose a step size satisfying the Wolfe conditions [13] given by

$$J(\bar{K} + \gamma_k \text{mat}(\mathbf{s}_k)) - J(\bar{K}) \leq \gamma_k c_1 (\text{vec}(\nabla_{\bar{K}} J))^T \mathbf{s}_k \quad (19a)$$

$$(\text{vec}(\nabla_{\bar{K} + \gamma_k \mathbf{s}_k} J))^T \mathbf{s}_k \geq c_2 (\text{vec}(\nabla_{\bar{K}} J))^T \mathbf{s}_k, \quad (19b)$$

where $c_1 \in (0, 1)$ and $c_2 \in (c_1, 1)$. For Algorithm 1, the search direction \mathbf{s}_k is given by the vectorization of the negative gradient, i.e. $\mathbf{s}_k = -\text{vec}(\nabla_{\bar{K}} J)$. An alternative step size for Algorithm 1 is the Barzilai-Borwein step size [14] which uses information of the previous step and is given as

$$\gamma_k = \frac{(\Delta \text{vec}(\bar{K}))^T (\Delta \text{vec}(\bar{K}))}{(\Delta \text{vec}(\bar{K}))^T (\Delta \text{vec}(\nabla_{\bar{K}} J))^T}, \quad (20)$$

where $\Delta \text{vec}(\bar{K}) = \text{vec}(\bar{K}^{(k)}) - \text{vec}(\bar{K}^{(k-1)})$ and $\Delta \text{vec}(\nabla_{\bar{K}} J) = \text{vec}((\nabla_{\bar{K}} J)^{(k)}) - \text{vec}((\nabla_{\bar{K}} J)^{(k-1)})$.

B. Minimization via BFGS method

The downside of Algorithm 1 presented in the previous section is that gradient methods generally converge slowly. However, the availability of the gradient according to (18) allows us to use more advanced optimization methods, like the Quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [15]. That means that instead of the negative gradient we use the following search direction

$$\mathbf{s}_k = -D_k \text{vec}(\nabla_{\bar{K}^{(k)}} J),$$

where D_k approximates the inverse of the Hessian matrix.

We obtain the following algorithm.

Algorithm 2:

- 1) Choose $c_1, c_2 \in \mathbb{R}$, $\bar{K}_0 \in \mathbb{R}^{m \times n}$. Pick a positive definite matrix $D_0 \in \mathbb{R}^{mn \times mn}$, e.g. $D_0 = I_{mn}$.
- 2) Compute the search direction \mathbf{s}_k as

$$\mathbf{s}_k = -D_k \text{vec}(\nabla_{\bar{K}} J),$$

where the gradient $\nabla_{\bar{K}} J$ is given by to (18).

- 3) Compute the step size γ_k according to the Wolfe conditions (19).
- 4) Update the feedback matrix

$$\bar{K}^{(k+1)} = \bar{K}^{(k)} + \gamma_k \text{mat}(\mathbf{s}_k).$$

- 5) Set $\mathbf{p}_k = \text{vec}(\bar{K}^{(k+1)}) - \text{vec}(\bar{K}^{(k)})$ and $\mathbf{q}_k = (\text{vec}(\nabla_{\bar{K}^{(k+1)}} J) - \text{vec}(\nabla_{\bar{K}^{(k)}} J))$. Update D_k as

$$D_{k+1} = D_k + \frac{(\mathbf{p}_k - D_k \mathbf{q}_k) \mathbf{p}_k^T + \mathbf{p}_k (\mathbf{p}_k - D_k \mathbf{q}_k)^T}{\mathbf{p}_k^T \mathbf{q}_k} - \frac{(\mathbf{p}_k - D_k \mathbf{q}_k)^T \mathbf{q}_k}{(\mathbf{p}_k^T \mathbf{q}_k)^2} \mathbf{p}_k \mathbf{p}_k^T.$$

- 6) If $\left| \frac{J^{(k)} - J^{(k-1)}}{J^{(k-1)}} \right| < \epsilon$, stop. Otherwise, increase k and go back to step 1.

Remark 1: Even though the two algorithms should lead to the same control law with the same cost this is not always the case. We attribute this to the nonconvexity of the cost functional causing the algorithms to converge to different local minima caused by the differences in search directions and step sizes. Also note that because the optimization problem has a finite horizon, the resulting control law is not guaranteed to be stabilizing for $t \rightarrow \infty$.

C. Averaging over the initial configuration x_0

One possible problem concerning the resulting feedback matrices of Algorithms 1 and 2 is that they are optimized w.r.t. to one specific initial configuration x_0 . In practice however, the initial configuration might not be known in advance or might be slightly disturbed from the assumed one. In order to circumvent this problem, we propose to average over several initial configurations for the simulated trajectories to obtain a control law that performs well for an area. The algorithms principally remain unchanged except for the gradient which is now given by

$$\nabla_{\bar{K}} J = \frac{1}{n_{\text{samples}}} \left(\sum_{i=1}^{n_{\text{samples}}} \int_0^T 2R\bar{K} \bar{\mathbf{x}}_i(t) \bar{\mathbf{x}}_i^T(t) + B^T \bar{\boldsymbol{\lambda}}_i(t) \bar{\mathbf{x}}_i^T(t) dt \right), \quad (21)$$

where $\bar{\mathbf{x}}_i(t)$ and $\bar{\boldsymbol{\lambda}}_i(t)$ are the trajectories resulting from the i th initial configuration, and n_{samples} is the number of selected initial configurations, see [10] for more details.

Remark 2: While this extension enlarges the area of possible initial configurations, this does not lead to a globally optimal control law. In fact, our numerical investigations show that initial configurations that are not considered directly in the design may lead to undesirable performance. This is also later shown in the example in Section IV-B.

IV. NUMERICAL INVESTIGATION

This section validate the control design algorithm.

A. Comparison between the presented algorithms

In this subsection we want to compare the computational performance between Algorithm 1 using the Barzilai-Borwein step size (20) and Algorithm 2. As a system, we consider three physically interconnected robots with system dynamics described by (7), all with identical parameters $M_i = I_3$, $D_i = 2\sqrt{3}I_3$ and $K_i = 3I_3$. Since each system has nine states, we have a total system dimension of 27.

As a comparison scenario we move an initial triangle on the edges of the rectangle marked by the four crosses in Figure 1 in steps of length 0.5, resulting in 40 different starting points. The weighting matrices are chosen as $\bar{q}_k = 5$, $R = I_{18}$, $\bar{Q} = \text{diag}(I_{27}, 0)$, $S = \text{diag}(I_{27}, 0)$ with a horizon of 40. The optimization algorithms stop when the change in cost between iterations is less than 10^{-3} . The results of the comparison are summarized in Table I. We see that the number of iterations is comparable for both algorithms. The fact that the number of iterations is lower for the gradient method is counter-intuitive and might be due to the fact that different local minima are found by the two algorithms. We observed in our investigations that even though both algorithms may achieve almost comparable costs, the actual resulting control matrices might be completely different. While the gradient descent algorithm has advantages in the computation time, the BFGS algorithm always achieves lower cost. The longer computation time of the BFGS algorithm for the same number of iterations is due to the step size computation according to the Wolfe conditions (19) which takes considerably longer than the computation of the Barzilai-Borwein step size. Note that for some categories we chose the median instead of the average because the BFGS algorithm produces some outliers that are not representative for its overall performance. In conclusion, if computational time is not an issue, the BFGS algorithm should be preferred because of the lower achieved cost.

TABLE I
COMPARISON BETWEEN ALGORITHMS 1 AND 2

	Algorithm 1	Algorithm 2
Median number of iterations	184	191
Average achieved cost	82.09	75.92
Median computation time [s]	59.1	142.4
Average cost decrease by BFGS	-	9.2%
Median cost decrease by BFGS	-	8.1%

B. Illustrating numerical example

In this subsection we want to illustrate the result of Algorithm 2 with averaging over the initial configuration x_0 . The system parameters are identical to the previous subsection. As weighting matrices we choose $R = 5I_{18}$, $S = \text{diag}(10I_{27}, 0)$, $\bar{q}_k = 100$ and $\bar{Q} = \text{diag}(0.1I_{27}, 0)$ because we want to emphasize maintaining of the formation. The optimization horizon is 40 seconds. The optimization algorithm stops when the change in cost between iterations is less than 10^{-3} . For the design, we pick 4 different initial configurations. The coordinates of the center points of the respective triangles are given in Table II and are marked as gray crosses in Figure 1. In the simulations the goal is to move the triangle formation of the three interconnected robots from four different initial configurations to a desired end point. These four initial configurations and the end point are also given in the table. Three of the initial configurations belong to the area surrounded by the four points used in the design, with two of them used directly in the design, while the fourth point is outside the area.

Figure 1 shows the resulting movements from all four initial configurations. We can see that the control design works well for all three points inside the area because the desired end point is reached and all the intermediate steps also show the initial formation. This shows that the relaxed rigidity condition is satisfied by the control design and the control law achieves all of its goals. For the point outside the area the desired end point is still reached but it can clearly be seen that the formation is violated in the intermediate steps because the formation stretches in all directions. This illustrates that the optimized control law has a local character and is not guaranteed to work well away from the initial configurations used during the optimization.

Remark 3: A phenomenon we observed during our numerical investigations is that while the classical LQR problem is invariant to scaling in the cost functional, meaning that the control law for Q and R is identical as the control law for cQ and cR with $c > 0 \in \mathbb{R}$, this is not the case here. Finding the exact reason will be part of future work.

TABLE II
STARTING AND END POINTS FOR VISUALIZING EXAMPLE

Phase	Center Point
Design	$x_{0,1} = [2, -2]$
Design	$x_{0,2} = [2, 4]$
Design	$x_{0,3} = [-2, 4]$
Design	$x_{0,4} = [-2, -2]$
Simulation	$x_{0,1} = [2, 0]$
Simulation	$x_{0,2} = [2, 4]$
Simulation	$x_{0,3} = [-2, 4]$
Simulation	$x_{0,4} = [-4, -1]$
Design & Simulation	$x_e = (\frac{\sqrt{3}}{6}, 2.5)$

C. Comparison with open loop input

In this subsection we compare the performance of the resulting feedback matrix from Algorithm 2 with an open loop input obtained with the Matlab function `fminunc`. The test scenario is the movement of a formation of three robots with identical system parameters as in Section IV-B. The open loop case corresponds to the generation of desired trajectories which are tracked by the impedance control law. The shape is a triangle with an initial center point $(0, 0)$. The desired end formation is a triangle with center point $(\frac{\sqrt{3}}{6}, 2.5)$ which is rotated by $-\frac{\pi}{2}$. We choose the weighting matrices $Q_k = 40I_{27}$, $R = 5I_{18}$, $S = \text{diag}(10I_{27}, 0)$ and $\bar{Q} = \text{diag}(10^{-2}I_{27}, 0)$, with the horizon 20.

The presented Algorithm 2 leads to a feedback control law which achieves the cost 11.13 after 958 iterations and 1543 seconds of computation time. The open loop trajectory is able to achieve a lower cost of 7.92 which is clear by the additional degree of freedom in the input signal but the computation time is much higher (117021 seconds ≈ 1.35 days). Besides computation time the feedback clearly has advantages when disturbances or uncertainties are considered.

V. CONCLUSIONS

In this paper we introduce an advanced state-space model of physically interconnected robots. Based on the model, we

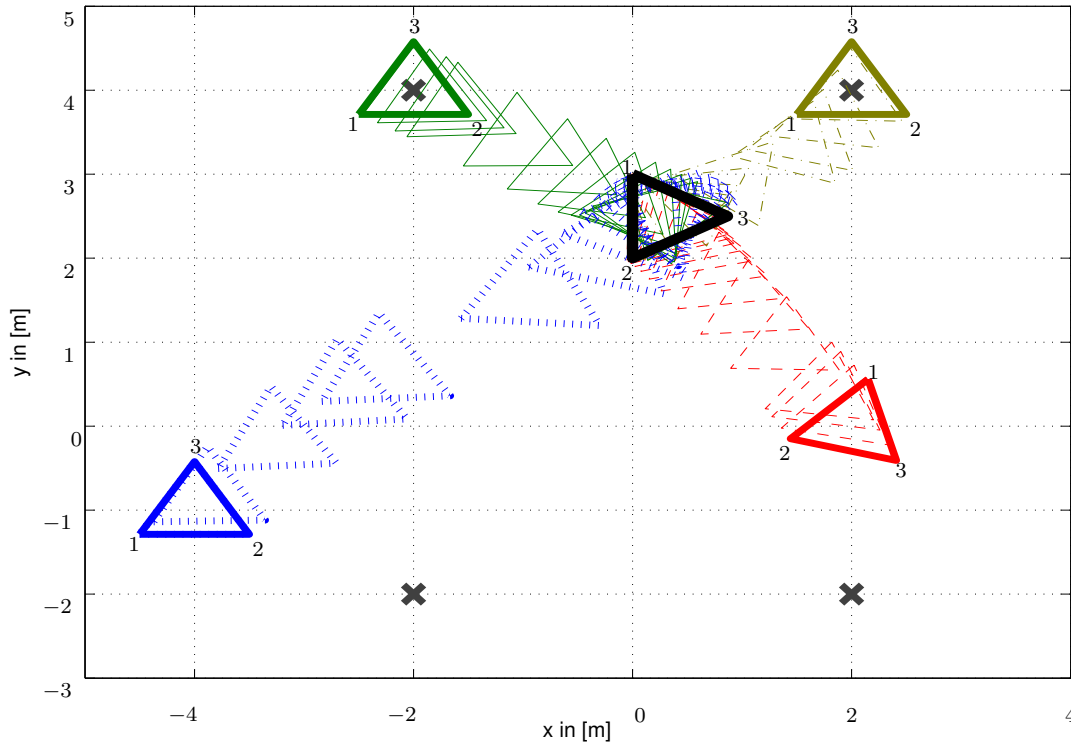


Fig. 1. Three mobile robots drive from four different initial configurations (red dashed, yellow dashed dotted, green solid, blue dashed) to a common goal while trying to maintain the formation. Bold colored triangles illustrate the initial robot configuration, the bold black triangle is the final configuration. The blue triangle clearly loses formation because the shape of the triangle stretches during the movement, while the other three triangles maintain their shape.

formulate a novel LQR-optimal control problem to design a control law which drives a formation of interconnected robots from an initial configuration to a desired end point. An important new property is the inclusion of formation rigidity as a penalty term into the cost functional, thus combining the two research directions of formation control with advanced robot dynamics. The resulting cost functional includes biquadratic terms. Two iterative algorithms, the first based on gradient descent, the second on a quasi-newton method, are presented to solve the optimal control problem and derive the feedback matrix. The presented algorithms are validated in numerical simulations. The algorithms are compared in a large-scale experiment with anthropomorphic robotic platforms in a companion paper [11].

ACKNOWLEDGMENTS

The work is partly supported by the German Research Foundation (DFG) excellence initiative research cluster “Cognition for Technical Systems CoTeSys”, the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no. 601165 of the project “WEARHAP - Wearable Haptics for Humans and Robots”, and the DFG Priority Program SPP 1305 “Control Theory of Digitally Networked Dynamical Systems”.

REFERENCES

- [1] S. Schneider and R. Cannon, “Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results,” *IEEE Trans. Robot. Automat.*, vol. 8, no. 3, pp. 383–394, 1992.
- [2] O. Khatib, K. Yokoi, K. Chang, D. Rusconi, R. Holmberg, A. Casal, A. Baader *et al.*, “Force strategies for cooperative tasks in multiple mobile manipulation systems,” in *International Symposium on Robotics Research*, vol. 7, 1996, pp. 333–342.
- [3] S. Erhart, D. Sieber, and S. Hirche, “An impedance-based control architecture for multi-robot dual-arm mobile manipulation,” in *Proc. IEEE/RSJ IROS*, 2013.
- [4] J. Szewczyk, F. Plumet, and P. Bidaud, “Planning and controlling cooperating robots through distributed impedance,” *Journal of Robotic Systems*, vol. 19, no. 6, pp. 283–297, 2002.
- [5] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 947–951, 2001.
- [6] M. Egerstedt, S. Martini, M. Cao, K. Camlibel, and A. Bicchi, “Interacting with networks: How does structure relate to controllability in single-leader, consensus networks?” *IEEE Control Systems*, vol. 32, no. 4, pp. 66–73, 2012.
- [7] T. Eren, P. N. Belhumeur, B. D. Anderson, and A. S. Morse, “A framework for maintaining formations based on rigidity,” in *Proc. of the 15th IFAC World Congress*, 2002, pp. 2752–2757.
- [8] M. Cao, C. Yu, A. S. Morse, B. Anderson, and S. Dasgupta, “Generalized controller for directed triangle formations,” in *Proc. of the 17th IFAC World Congress*, 2008, pp. 6590–6595.
- [9] K. Martensson and A. Rantzer, “Gradient methods for iterative distributed control synthesis,” in *Proc. IEEE/CSS CDC*, 2009, pp. 549–554.
- [10] F. Deroo, M. Ulbrich, B. D. O. Anderson, and S. Hirche, “Accelerated iterative distributed controller synthesis with a barzilai-borwein step size,” in *Proc. IEEE/CSS CDC*, 2012, pp. 4864–4870.
- [11] D. Sieber, F. Deroo, and S. Hirche, “Formation-based approach for cooperative manipulation based on iterative optimal controller design,” 2013, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons Hoboken, eNJ NJ, 2006.
- [13] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.
- [14] J. Barzilai and J. M. Borwein, “Two-point step size gradient methods,” *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.
- [15] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.