Multi-Robot Control by Utilizing Distributed Model Predictive Controllers

KTH Royal Institute of Technology
School of Electrical Engineering
Department of Automatic Control

Student: Alexandros Filotheou (alefil@kth.se)
Supervisor: Alexandros Nikou (anikou@kth.se)
Examiner: Dimos Dimarogonas (dimos@kth.se)

January 30, 2017

Abstract

This paper addresses the problem of position- and orientation-based formation control of a class of second-order nonlinear multi-agent systems in 3D space, under static and undirected communication topologies. More specifically, we design a decentralized control protocol for each agent in the sense that each agent uses only local information from its neighbors to calculate its own control signal. Additionally, by introducing certain inter-agent distance constraints, we guarantee collision avoidance both between among the agents and between the agents and possible obstacles of the workspace. Connectivity maintenance between agents that are initially connected is also achieved by the proposed controller scheme. Finally, simulation results verify the performance of the proposed controllers.

Contents

1 Introduction 2

2 Notation and Preliminaries 2
2.1 Notation 2
2.2 Graph Theory 3

3 Problem Formulation 3
3.1 System Model 3
3.2 Problem Statement 6

4 Problem Solution 7
4.1 Distributed MPC 7
4.2 Feasibility Analysis 7
4.3 Stability Analysis 7

5 **Simulation Results** **7**

6 **Conclusions and Future Work** **7**

1 Introduction

formation of multi-agent systems, mpc intro etc.

motivation why we need mpc controllers...

In many control problems it is desired to design a stabilizing feedback such that a performance criterion is minimized while satisfying constraints on the controls and the states. Ideally one would look for a closed solution for the feedback law satisfying the constraints while optimizing the performance. However, typically the optimal feedback law cannot be found analytically, even in the unconstrained case, since it involves the solution of the corresponding Hamilton-Jacobi-Bellman partial differential equations. One approach to circumvent this problem is the repeated solution of an open-loop optimal control problem for a given state. The first part of the resulting open-loop input signal is implemented and the whole process is repeated. Control approaches using this strategy are referred to as Model Predictive Control (MPC).

2 Notation and Preliminaries

2.1 Notation

The set of positive integers is denoted as $\mathbb{N}$. The real n-coordinate space, with $n \in \mathbb{N}$, is denoted as $\mathbb{R}^n$; $\mathbb{R}^n_{\geq 0}$ and $\mathbb{R}^n_{>0}$ are the sets of real n-vectors with all elements nonnegative and positive, respectively. Given a set S, we denote as $|S|$ its cardinality. The notation $\|x\|$ is used for the Euclidean norm of a vector $x \in \mathbb{R}^n$. Given a symmetric matrix $A, \lambda_{\min}(A) = \min\{|\lambda| : \lambda \in \sigma(A)\}$ denotes the minimum eigenvalue of A, respectively, where $\sigma(A)$ is the set of all the eigenvalues of A and $\text{rank}(A)$ is its rank; $A \otimes B$ denotes the Kronecker product of matrices $A, B \in \mathbb{R}^{m \times n}$, as was introduced in [**?**]. Define by $1_n \in \mathbb{R}^n, I_n \in \mathbb{R}^{n \times n}, 0_{m \times n} \in \mathbb{R}^{m \times n}$ the column vector with all entries 1, the unit matrix and the $m \times n$ matrix with all entries zeros, respectively. A matrix $A \in \mathbb{R}^{n \times n}$ is called skew-symmetric if and only if $A^\top = -A$. $\mathcal{B}(c, r) = \{x \in \mathbb{R}^3 : \|x - c\| \leq r\}$ is the 3D sphere of radius $r \in \mathbb{R}_{\geq 0}$ and center $c \in \mathbb{R}^3$. The vector connecting the origins of coordinate frames $\{A\}$ and $\{B\}$ expressed in frame $\{C\}$ coordinates in 3D space is denoted as $p^C_{B/A} \in \mathbb{R}^3$. Given $a \in \mathbb{R}^3$, $S(a)$ is the skew-symmetric matrix defined according to $S(a)b = a \times b$. We further denote as $q_{B/A} \in \mathbb{T}^3$ the Euler angles representing the orientation of frame $\{B\}$ with respect to frame $\{A\}$, where $\mathbb{T}^3$ is the 3D torus. The angular velocity of frame $\{B\}$ with respect to $\{A\}$, expressed in frame $\{C\}$ coordinates, is denoted as $\omega^C_{B/A} \in \mathbb{R}^3$. We also use the notation $\mathbb{M} = \mathbb{R}^3 \times \mathbb{T}^3$. For notational brevity, when a coordinate frame corresponds to an inertial frame of reference $\{0\}$, we will omit its explicit notation (e.g., $p_B = p^0_{B/0}, \omega_B = \omega^0_{B/0}$ etc.). All vector and matrix differentiations are derived with respect to an inertial frame $\{0\}$, unless otherwise stated.

2.2 Graph Theory

An *undirected graph* $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of nodes, representing a team of agents, and $\mathcal{E} \subseteq \{\{i,j\} : i,j \in \mathcal{V}, i \neq j\}$, with $M = |\mathcal{E}|$, is the set of edges that model the communication capability between neighboring agents. For each agent, its neighbors' set $\mathcal{N}_i$ is defined as $\mathcal{N}_i = \{i_1, \dots, i_{N_i}\} = \{j \in \mathcal{V} : \{i,j\} \in \mathcal{E}\}$, where $i_1, \dots, i_{N_i}$ is an enumeration of the neighbors of agent i and $N_i = |\mathcal{N}_i|$.

If there is an edge $\{i,j\} \in \mathcal{E}$, then i, j are called *adjacent.* A *path* of length r from vertex i to vertex j is a sequence of $r+1$ distinct vertices, starting with i and ending with j, such that consecutive vertices are adjacent. For $i = j$, the path is called a cycle. If there is a path between any two vertices of the graph $\mathcal{G}$, then $\mathcal{G}$ is called *connected.* A connected graph is called a tree if it contains no cycles.

3 Problem Formulation

3.1 System Model

Consider a set of N rigid bodies, with $\mathcal{V} = \{1, 2, \dots, N\}$, $N \geq 2$, operating in a workspace $W \subseteq \mathbb{R}^3$, with coordinate frames $\{i\}, i \in \mathcal{V}$, attached to their centers of mass. The workspace is assumed to be modeled as a bounded sphere $\mathcal{B}(0, r_w)$. We consider that each agent occupies a sphere $\mathcal{B}(p_i(t), r_i)$, where $p_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^3$ is the position of the agent's center of mass and r_i is the agent's radius. We also denote as $q_i : \mathbb{R}_{\geq 0} \to \mathbb{T}^3, i \in \mathcal{V}$, the Euler angles representing the agents' orientation with respect to an inertial frame $\{0\}$, with $q_i = [\phi_i, \theta_i, \psi_i]^\tau$. By defining $x_i : \mathbb{R}_{\geq 0} \to \mathbb{M}, v_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^6$, with $x_i = [p_i^\tau, q_i^\tau]^\tau, v_i = [\dot{p}_i^\tau, \omega_i^\tau]^\tau$, we model each agent's motion with the 2nd order dynamics:

$$\dot{x}_i(t) = J_i(x_i) v_i(t), \tag{1a}$$

$$M_i(x_i)\dot{v}_i(t) + C_i(x_i, \dot{x}_i) v_i(t) + g_i(x_i) = u_i, \tag{1b}$$

where $J_i : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is a Jacobian matrix that maps the Euler angle rates to v_i, given by

$$J_i(x_i) = \begin{bmatrix} I_3 & 0_{3\times 3} \\ 0_{3\times 3} & J_q(x_i) \end{bmatrix},$$

$$J_q(x_i) = \begin{bmatrix} 1 & \sin(\phi_i)\tan(\theta_i) & \cos(\phi_i)\tan(\theta_i) \\ 0 & \cos(\phi_i) & -\sin(\phi_i) \\ 0 & \dfrac{\sin(\phi_i)}{\cos(\theta_i)} & \dfrac{\cos(\phi_i)}{\cos(\theta_i)} \end{bmatrix},$$

for which we make the following assumption:

Assumption 3.1. The angle θ_i satisfies the inequality $-\frac{\pi}{2} < \theta_i(t) < \frac{\pi}{2}, \forall i \in \mathcal{V}, t \in \mathbb{R}_{\geq 0}$.

The aforementioned assumption guarantees that J_i is always well-defined and invertible, since $\det(J_i) = \frac{1}{\cos\theta_i}$. Furthermore, $M_i : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is the positive definite inertia matrix, $C_i : \mathbb{M} \times \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ is the Coriolis matrix and $g_i : \mathbb{M} \to \mathbb{R}^6$ is the gravity vector. We consider that the aforementioned vector fields are unknown and continuous. Finally, $u_i \in \mathbb{R}^6$ is the control input vector representing the 6D generalized force acting on the agent. Let as also define the vectors $x = [x_1^\tau, \dots, x_N^\tau]^\tau : \mathbb{R}_{\geq 0} \to \mathbb{M}^N, v = [v_1^\tau, \dots$ and $v_N^\tau]^\tau : \mathbb{R}_{\geq 0} \to \mathbb{R}^{6N}$.

Remark 3.1. According to [**?**], the matrices $\dot{M}_i - C_i, i \in \mathcal{V}$ are skew-symmetric. From [**?**], we have that a quadratic form of a skew-symmetric matrix is always equal to 0. Hence, for the matrices $\dot{M}_i - C_i$ it holds that:

$$y^\top \left[\dot{M}_i - 2C_i\right] y = 0, \forall y \in \mathbb{R}^n, i \in \mathcal{V}. \tag{2}$$

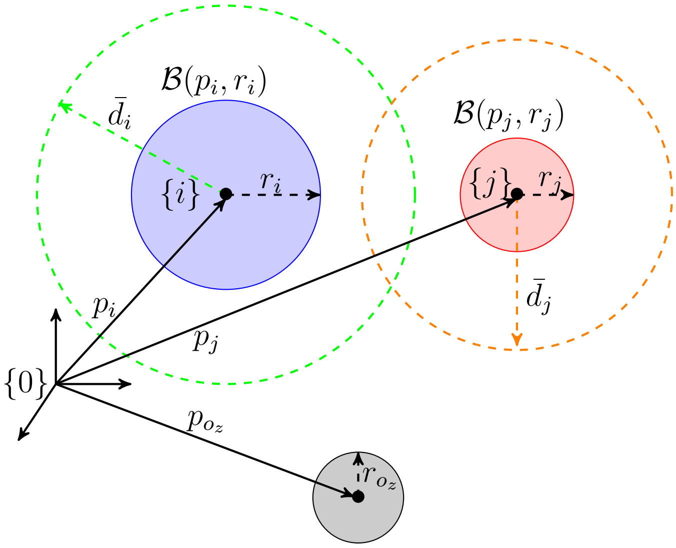

Figure 1: Illustration of two agents $i, j \in \mathcal{V}$ and an static obstacle o_z in the workspace; $\{0\}$ is the inertial frame, $\{i\}, \{j\}$ are the frames attached to the agents' center of mass, $p_i, p_j, p_{o_z} \in \mathbb{R}^3$ are the positions of the center of mass of the agents i, j and the obstacle o_z, respectively, with respect to $\{0\}$. r_i, r_j, r_{o_z} are the radii of the agents i, j and the obstacle o_z respectively. $\bar{d}_i, \bar{d}_j$ with $\bar{d}_i > \bar{d}_j$ are the agents' sensing ranges.

It is also further assumed that each agent i can measure its own $p_i, q_i, \dot{p}_i, v_i, i \in \mathcal{V}$, and has a limited sensing range of $\bar{d}_i > \max\{r_i + r_j : i, j \in \mathcal{V}\}$. Therefore, by defining the neighboring set $\mathcal{N}_i(t) = \{j \in \mathcal{V} : p_j(t) \in \mathcal{B}(p_i(t), s_i)\}$, agent i also knows at each time instant t all $p_{j/i}^i(t), q_{j/i}(t)$ and, since it knows its own $p_i(t), q_i(t)$, it can compute all $p_j(t), q_j(t), \forall j \in \mathcal{N}_i(t), t \in \mathbb{R}_{\geq 0}$.

In the workspace there are $|Z|$ static obstacles, modeled as spheres with centers and radii $p_{o_z}, r_{o_z} \in \mathbb{R}^3, z \in \mathcal{Z} = \{1, \dots, |Z|\}$, respectively. Thus, the obstacles are modeled by the spheres $\mathcal{B}(p_{o_z}, r_{o_z}), z \in \{1, \dots, |Z|\}$. The geometry in the workspace W of agents i and j as well as an obstacle z is depicted in Fig. 1.

Let us also define the distances:

$$d_{ij,a} = \|p_i - p_j\|, i, j \in \mathcal{V}, i \neq j, \tag{3a}$$
$$\underline{d}_{ij,a} = r_i + r_j, i, j \in \mathcal{V}, i \neq j, \tag{3b}$$
$$d_{iz,o} = \|p_i - p_{o_z}\|, i \in \mathcal{V}, z \in \mathcal{Z}, \tag{3c}$$
$$\underline{d}_{iz,o} = r_i + r_{o_z}, i \in \mathcal{V}, z \in \mathcal{Z}, \tag{3d}$$

where (3a) stands for the distance between agents i, j, (3b) stands for the minimum distance that two agents do not collide, (3c) stands for the distance between agent i and obstacle z and (3d) stands for the minimum distance that agent i and obstacle z do not collide.

The topology of the multi-agent network is modeled through the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \dots, N\}$ and $\mathcal{E} = \{\{i, j\} \in \mathcal{V} \times \mathcal{V} : j \in \mathcal{N}_i(0) \text{ and } i \in \mathcal{N}_j(0)\}$. The latter implies that at $t = 0$ the graph is undirected, i.e.,

$$\|p_i(0) - p_j(0)\| < \bar{d}_i, \forall i \in \mathcal{V}, j \in \mathcal{N}_i(0). \tag{4}$$

We also consider that $\mathcal{G}$ is static in the sense that no edges are added to the graph. We do not exclude, however, edge removal through connectivity loss between initially neighboring agents, which we guarantee to avoid, as presented in the sequel. It is also assumed that at $t = 0$ the neighboring agents are at a collision-free configuration, i.e.,

$$\underline{d}_{ij,a} < \|p_i(0) - p_j(0)\|, \forall i, j \in \mathcal{V}, i \neq j. \tag{5}$$

Let us define the distance:

$$D = \min\{d_o, d_{o,w}\},$$

where:

$$\begin{aligned} d_o &= \min\{\|p_{o_z} - p_{o_{z'}}\| : z, z' \in \mathcal{Z}\}, \\ d_{o,w} &= \min\{r_w - (r_{o_z} + \|p_{o_z}\|) : z \in \mathcal{Z}\}, \end{aligned}$$

are the distance between the two most closest obstacles and the distance between the closest obstacle to the boundary with the boundary of the workspace, respectively. We define the *diameter of formation*, as the maximum distance between two agents, when the formation is achieved, i.e.:

$$\begin{aligned} \Delta = \max\{& d_{ij,a} + r_i + r_j : i, j \in \mathcal{V}, i \neq j, \\ & p_k - p_\ell = p_{k\ell,\text{des}}, q_k - q_\ell = q_{k\ell,\text{des}}, k \in \mathcal{V}, \ell \in \mathcal{N}_k(0)\}, \end{aligned}$$

Assumption 3.2. In order for the problem to be feasible, we make the following natural geometric assumptions:

- All the agents should be able to pass between any two the obstacles and between an obstacle and the boundary of the workspace, simultaneously, and without colliding to each other or with the obstacles as well as the boundary of the workspace. Thus, it is required $D > \sum_{i \in \mathcal{V}} 2r_i$.
- When the multi-agent system reach the desired formation, it should be able to pass between two of the obstacles and between an obstacle and the boundary of the workspace. Thus, it is required $D > \Delta$.

Both these geometrical assumptions can be summarized in the following inequality:

$$D > \max\left\{\Delta, \sum_{i \in \mathcal{V}} 2r_i\right\}. \tag{6}$$

3.2 Problem Statement

Due to the fact that the agents are not dimensionless and their communication capabilities are limited, the control protocol, except from achieving desired position formation $p_{ij,\text{des}}$ and desired formation angles $q_{ij,\text{des}}$ for all neighboring agents $i \in \mathcal{V}, j \in \mathcal{N}_i(0)$, it should also guarantee for all $t \in \mathbb{R}_{\geq 0}$ that (i) all the agents avoid collision with every other agent and (iii) all the initial edges are maintained, i.e., connectivity maintenance. Therefore, all the neighboring agents of agent i must remain within distance less than $\bar{d}_i$, for all $i \in \mathcal{V}$ and all the agents $i, j \in \mathcal{V}, i \neq j$ must remain within distance greater than $\underline{d}_{ij,a}$. We also make the following assumption that are required on the initial graph topology

Assumption 3.3. The communication graph $\mathcal{G}$ is connected at time $t = 0$ and the agents are in collision-free configuration, i.e., both (4) and (5) hold.

Formally, the control problem under the aforementioned constraints is formulated as follows:

Problem 3.1. Given N agents performing in workspace W modeled as bounded sphere $\mathcal{B}(0, r_w)$, with spherical obstacles $\mathcal{B}(p_{o_z}, r_{o_z}), z \in \mathcal{Z}$, governed by the dynamics (1), under the Assumptions 1-3, under the geometric feasibility constraint (6) and given the desired inter-agent distances and angles $p_{ij,\text{des}}, q_{ij,\text{des}}$, with $\underline{d}_{ij,a} < p_{ij,\text{des}} < \bar{d}_i$, $\forall i \in \mathcal{V}, j \in \mathcal{N}_i(0)$, design decentralized control laws $u_i \in \mathbb{R}^6, i \in \mathcal{V}$ such that:

- $\forall i \in \mathcal{V}, j \in \mathcal{N}_i(0)$, the following hold:
  1) $\lim\limits_{t \to \infty} [p_i(t) - p_j(t) - p_{ij,\text{des}}] = 0_{3\times 1}$,
  2) $\lim\limits_{t \to \infty} [q_i(t) - q_j(t) - q_{ij,\text{des}}] = 0_{3\times 1}$.
- $\forall i, j \in \mathcal{V}, i \neq j$ the following holds:
  3) $\|p_i(t) - p_j(t)\| > \underline{d}_{ij,a}, \forall\, t \in \mathbb{R}_{\geq 0}$.
- $\forall\, i \in \mathcal{V}, z \in \mathcal{Z}$ the following holds:
  4) $\mathcal{B}(p_i(t), r_i) \cap \mathcal{B}(p_{o_z}(t), r_{o_z}) = \emptyset, \forall\, t \in \mathbb{R}_{\geq 0}$.
- $\forall i \in \mathcal{V}, j \in \mathcal{N}_i(0)$ the following holds:
  5) $\|p_i(t) - p_j(t)\| < \bar{d}_i, \forall\, t \in \mathbb{R}_{\geq 0}$.

The aforementioned specifications imply the following:

- 1 stands for formation control;
- 2 stands for orientation alignment;
- 3 stands for inter-agent collision avoidance;
- 4 stands for collision avoidance between the agents and the obstacles;
- 5 stands for connectivity maintenance of the initial graph;

4 Problem Solution

In this section, a systematic solution to Problem 3.1 is introduced. Our overall approach builds on formulating a model predictive control optimization problem for each agent such that solution this captures all the desired control specifications. The following analysis is performed:

1. The form of the proposed optimizaton problem is described in Section 4.1.
2. The feasibility of the problem is given in 4.2.
3. The stability analysis is given in 4.3.

4.1 Distributed MPC

4.2 Feasibility Analysis

4.3 Stability Analysis

5 Simulation Results

6 Conclusions and Future Work