Technical University of Denmark

Swarm Planner

Jeppe Heini Mikkelsen

18th July 2022

Contents

1 Introduction

This document presents a method for planning for the motion of a swarm while ensuring communication among the robots. Finding planar paths for N robots while ensuring communication amounts to searching through the 2^N dimensional joint configuration space of all the robots, with additional constraints. As N increases this becomes infeasible. A way to handle this is to constrain the set of possible positions that the robots can take, and thereby reducing the search space. In this document, it is suggested that this set of positions can be expressed as the following set of transformations of a basis swarm configuration, see fig. 1.

1. Scaling
2. Rotation
3. Translation

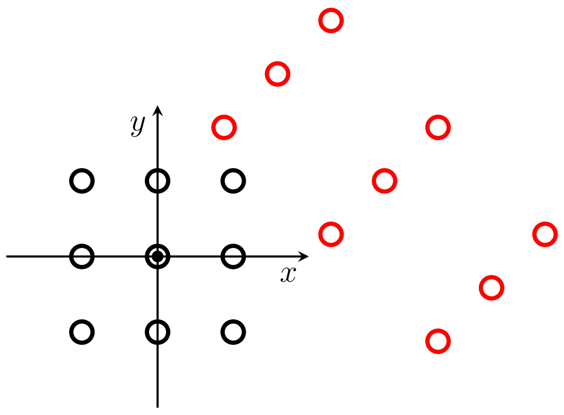

Figure 1: *Example of a transformation of a swarm configuration, with $\varphi = \pi/4$, $s_x = 1$, $s_y = 2$, $t_x = 3$ and $t_y = 1$.*

Rotations and translations are Euclidean transformations, meaning they preserve distances between the robots and, thereby, communication. Scaling does not necessarily preserve communication, but by including constraints on the scaling parameters, communication can be preserved. The transformation of the swarm can be calculated as

$$\boldsymbol{p_i} = \mathbf{T}\boldsymbol{c_i} + \boldsymbol{t}, \quad \forall i \in \{1, 2, \ldots, N\}, \tag{1}$$

where

$$\mathbf{T} = \mathbf{R}\mathbf{S}, \quad \mathbf{R} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}, \quad \boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \tag{2}$$

$\boldsymbol{p_i}$ is the position of robot i in the swarm after the transformation, $\mathbf{T}$ is a transformation matrix consisting of a rotation $\mathbf{R}$ and a scaling $\mathbf{S}$, $\boldsymbol{t}$ is a translation vector and $\boldsymbol{c_i}$ is the position of robot i in the swarm configuration. The parameters of the transformation are denoted as

$$\boldsymbol{\eta} = (\varphi, s_x, s_y, t_x, t_y). \tag{3}$$

2 Method

Having found a set of transformations for expressing the position of the swarm, the transformation that will bring each robot to some desired position, while ensuring communication, has to be found. This is done in three steps.

2.1 Step 1: Error correction

Using the chain-rule, the derivative of the position of robot i in the swarm can be expressed as

$$\frac{d}{dt}\boldsymbol{p_i} = \mathbf{J}_{\boldsymbol{\eta}_\mathbf{i}} \frac{d}{dt}\boldsymbol{\eta_i} \tag{4}$$

where $\mathbf{J}_{\boldsymbol{\eta}_\mathbf{i}}$ is the Jacobian of the transformation in eq. (1) with regard to the parameters $\boldsymbol{\eta_i}$

$$\mathbf{J}_{\boldsymbol{\eta_i}} = \begin{bmatrix} -\sin\varphi_i s_{x,i} c_{x,i} - \cos\varphi_i s_{y,i} c_{y,i} & \cos\varphi_i c_{x,i} & -\sin\varphi_i c_{y,i} & 1 & 0 \\ \cos\varphi_i s_{x,i} c_{x,i} - \sin\varphi_i s_{y,i} c_{y,i} & \sin\varphi_i c_{x,i} & \cos\varphi_i c_{y,i} & 0 & 1 \end{bmatrix}. \tag{5}$$

Using this, the derivative of the parameters for robot i can be found as

$$\frac{d}{dt}\boldsymbol{\eta_i} = \mathbf{J}^+_{\boldsymbol{\eta_i}} \frac{d}{dt}\boldsymbol{p_i} \tag{6}$$

where $(\cdot)^+$ denotes the Moore-Penrose pseudo-inverse. To find the parameters that bring the position of robot i to some desired position, the derivative of the position can be expressed as $\frac{d}{dt}\boldsymbol{p_i} = \boldsymbol{p_{des,i}} - \boldsymbol{p_i}$, where $\boldsymbol{p_{des,i}}$ is the desired position of robot i. This results in the update law becoming

$$\frac{d}{dt}\boldsymbol{\eta_i} = \mathbf{J}^+_{\boldsymbol{\eta_i}}(\boldsymbol{p_{des,i}} - \boldsymbol{p_i}). \tag{7}$$

However, since $\mathbf{J}_{\boldsymbol{\eta}_\mathbf{i}}$ is indefinite, applying the update in eq. (7) does not result in a unique solution and therefore there will be discrepancies among the robots as to the parameters. This also makes intuitive sense since there is an infinite amount of ways a point can be scaled, rotated and translated to move it to another point. For swarms of more than three robots there is only one unique transformation that will bring it to some desired formation though.

2.2 Step 2: Consensus

To ensure that the robots ends at the same solution, a consensus step is applied to eq. (7).

$$\frac{d}{dt}\boldsymbol{\eta_i} = \mathbf{J}^+_{\boldsymbol{\eta_i}}(\boldsymbol{p_{des,i}} - \boldsymbol{p_i}) \underbrace{- \sum_{j\in\mathcal{N}_i}(\boldsymbol{\eta_i} - \boldsymbol{\eta_j}),}_{\textbf{consensus step}} \tag{8}$$

where $\mathcal{N}_i$ denotes the communication neighbours of robot i. Applying this step drives the solutions of the robots together. This with the addition of step 1 will bring the robots to the unique solution that results in them being at their desired positions.

2.3 Step 3: Constraint Satisfaction

In step 1 and 2, the parameters that resulted in the desired transformation of the swarm was found. However, this transformation allows for unbounded scaling which can result in loss of communication. To remedy this, an additional step for constraining the solution is applied.

$$\frac{d}{dt}\boldsymbol{\eta_i} = \mathbf{J}^+_{\eta_i}(\boldsymbol{p_{des,i}} - \boldsymbol{p_i}) - \sum_{j\in\mathcal{N}_i}(\boldsymbol{\eta_i} - \boldsymbol{\eta_j}) - \underbrace{\gamma_i(\boldsymbol{\eta_i} - \mathrm{proj}(\boldsymbol{\eta_i}, \mathcal{C}))}_{\textbf{constraint satisfaction step}} \tag{9}$$

where γ is a penalty multiplier and $\mathrm{proj}(\boldsymbol{\eta_i}, \mathcal{C})$ is a projection of the parameter onto the constraint satisfaction set $\mathcal{C}$. Since the rotation and translation is unconstrained, their projection is simply set to their current value

$$\mathrm{proj}(\boldsymbol{\eta_i}, \mathcal{C}) = (\varphi_{proj}, \boldsymbol{s_{proj}}, \boldsymbol{t_{proj}}), \quad \varphi_{proj} = \varphi_i, \quad \boldsymbol{t_{proj}} = \boldsymbol{t_i}. \tag{10}$$

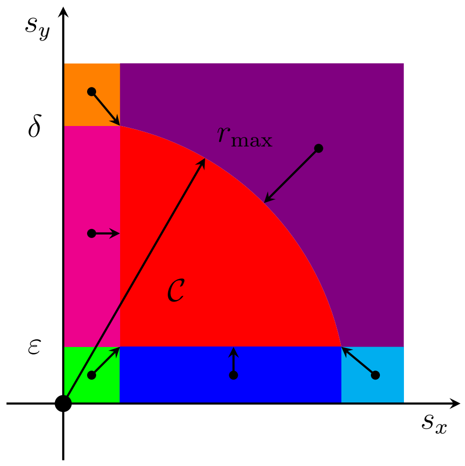

Figure 2: *Constraint satisfaction set $\mathcal{C}_S$ as an annulus with inner radius $r_{\min}$ and outer radius $r_{\max}$. The three examples A, B and C show different projection cases.*

2.4 Complete Method

Applying these three steps result in the following update law for each robot, see eq. (11).

$$\boxed{\begin{aligned} \frac{d}{dt}\boldsymbol{\eta_i} &= \mathbf{J}^+_{\eta_i}(\boldsymbol{p_{des,i}} - \boldsymbol{p_i}) - \sum_{j\in\mathcal{N}_i}(\boldsymbol{\eta_i} - \boldsymbol{\eta_j}) - \gamma_i(\boldsymbol{\eta_i} - \mathrm{proj}(\boldsymbol{\eta_i}, \mathcal{C})), \\ \frac{d}{dt}\gamma_i &= ||\boldsymbol{\eta_i} - \mathrm{proj}(\boldsymbol{\eta_i}, \mathcal{C})||_2, \quad \gamma_i(0) = 0, \quad \forall i \in \{1, 2, \dots, N\}. \end{aligned}} \tag{11}$$

2.4.1 Simulation w. Ideal Swarm

2.4.2 Simulation w. Non-Ideal Swarm

2.5 Local Translation