Model reduction and control design of a multi-agent line formation of mobile robots*

Adrian-Josue Guel-Cortez, Eun-jin Kim
Centre for Fluid and Complex Systems, Coventry University;
Priory St, Coventry CV1 5FB, UK
guelcortea@uni.coventry.ac.uk

December 20, 2021

Abstract

In this work, a model reduction of a line robotic formation driven by simple PD-controllers is presented. The proposed mathematical model describes the control interactions between the agents which permits us to easily design a decentralised control strategy. To select the PD-controller gains for each agent, we employ a population-based algorithm that takes into consideration the formation stability analysis. Finally, we discuss the future work and the manner the proposed methodology can be used in more complex robotic scenarios. Multi-agent systems Line formation Evolutionary algorithms Control design.

1 Introduction

Control of multi-agent systems (MAS) is currently a trend topic with plenty of different approaches [1, 2]. The study of MAS includes not only physical systems but cyberphysical systems [3], biological neuronal networks [4], big data [5] and social networks [6]. This makes MAS a very multidisciplinary and complicated area of research. One of the main issues, where significant efforts have been made is the design of control techniques. This is because MAS control implies a variety of challenges including: modelling, scalability and communication [7].

[*] To be published in https://www.springer.com/gb/book/9783030820633

MAS control design can be simplified if we obtain simpler and reliable mathematical models. For instance, to find a linear representation of the interactions between agents that also includes the control actions. If every agent is controlled by simple low-order controllers, we may use well known linear system control design techniques (see [8, 9]) or new low-order control approaches (see [10, 11, 12]). On the other hand, there are other approaches that could be explored, for example the use of information metrics for guided self-organisation in the MAS formation [13] or the use of a fokker-planck control approach [14, 15].

Considering the given context and inspired by our previous experience with a robotic formation experimental setup (see [16]), in this work, we propose a model reduction of a n-agent line formation driven by simple PD-controllers. Our model considers that the agents need to keep a desired distance between each other while being limited to see only the agent in front of them. Besides, the nature of the proposed model allows us to study its stability by means of simple linear systems techniques. To solve the control design problem, we use an evolutionary algorithm to find the set of controller gains that improve the system dynamics. In this regard, a discussion on some simulation results is included. Finally, a set of conclusions which include the future work is given.

2 Mathematical model

Consider a system of two robots moving in the x-axis interacting within each other through a control action that keeps them apart by a desired distance d. The controller is described as an operator L over the relative error ϵ as shown in the left of Fig. 1.

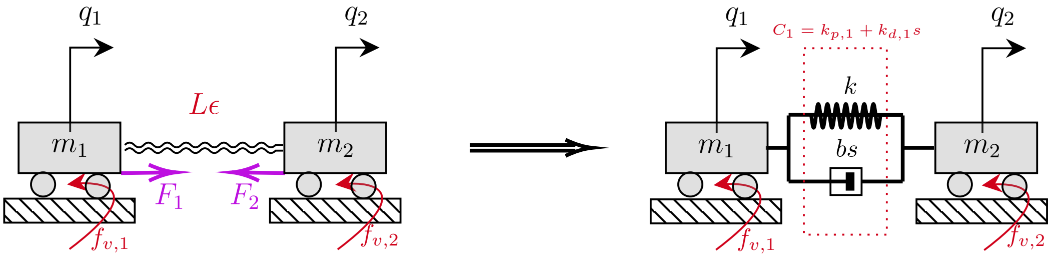


Figure 1: Interaction of 2 Robots in 1D.

In our case, the operator L is given by a classical PD controller, $L = k_p + k_d \frac{d}{dt}$. Here ϵ corresponds to the relative error between the position q_1 and q_2 of the mobile robots, and F_i represents the control force applied to the i-th robot. Note that we can use a spring and a damper in parallel to describe our PD's control action. Specifically, for the two-robot system, we proceed as follows:

The time-evolution of m_1 and m_2 is governed by

$$\begin{aligned} m_1\ddot{q}_1 + f_{v,1}\dot{q}_1 &= F_1, & (1)\\ m_2\ddot{q}_2 + f_{v,2}\dot{q}_2 &= F_2. & (2)\end{aligned}$$

Since $F_2 = L\epsilon = L(d-(q_2-q_1))$ and $F_2 = -F_1$ we have

$$\begin{aligned} m_1\ddot{q}_1 + f_{v,1}\dot{q}_1 &= k_p(-d-(q_1-q_2)) - k_d(\dot{q}_1-\dot{q}_2), & (3)\\ m_2\ddot{q}_2 + f_{v,2}\dot{q}_2 &= k_p(d-(q_2-q_1)) - k_d(\dot{q}_2-\dot{q}_1). & (4)\end{aligned}$$

Eqs. (3)-(4) give the dynamics of the system shown on the right-hand side of Fig. 1.

In a two robot system, if m_2 corresponds to the robot formation leader and it does not care about the position of m_1, the force $F_2 = 0$.

2.1 1D model for n robots

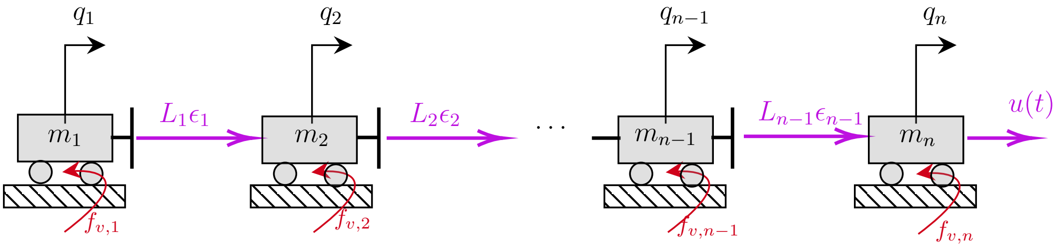


Figure 2: Interaction of n Robots in 1D.

We can generalise the previous procedure to the case of n mobile robots. If the robots can only spot the robot in front, they form a line moving in one dimension and are controlled by a classical PD controller as shown in Fig. 2. Then, considering the change of variable $q_{i,1} = q_i$ and $q_{i,2} = \dot{q}_i$, the operator $L_i = k_{p,i} + k_{d,i}\frac{d}{dt}$ and $\epsilon_i = -d_i - (q_i - q_{i+1})$, the mathematical model describing the interaction between the n robots is given by

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{q} + \mathbf{B}\mathbf{u}, \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{H} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{O} \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k_{p,1}}{m_1} & -\frac{k_{d,1}+f_{v,1}}{m_1} & \frac{k_{p,1}}{m_1} & \frac{k_{d,1}}{m_1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{k_{p,2}}{m_2} & -\frac{k_{d,2}+f_{v,2}}{m_2} & \frac{k_{p,2}}{m_2} & \frac{k_{d,2}}{m_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\frac{k_{p,3}}{m_3} & -\frac{k_{d,3}+f_{v,3}}{m_3} \end{bmatrix},$$

$$\mathbf{O} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_{p,n-1}}{m_n-1} & -\frac{k_{d,n-1}+f_{v,n-1}}{m_n-1} & \frac{k_{p,n-1}}{m_n-1} & \frac{k_{d,n-1}}{m_n-1} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{f_{v,n}}{m_n} \end{bmatrix},$$

$$\mathbf{q} = \begin{bmatrix} q_{1,1} & q_{1,2} & q_{2,1} & q_{2,2} & q_{3,1} & q_{3,2} & \cdots & q_{n-1,1} & q_{n-1,2} & q_{n,1} & q_{n,2} \end{bmatrix}^T,$$

$$\mathbf{Bu} = \begin{bmatrix} 0 & -\frac{k_{p,1}d_1}{m_1} & 0 & -\frac{k_{p,2}d_2}{m_2} & 0 & -\frac{k_{p,3}d_3}{m_3} & \cdots & 0 & -\frac{k_{p,n-1}d_{n-1}}{m_{n-1}} & 0 & u(t) \end{bmatrix}^T. \quad (6)$$

Or in the form of the recurrence equations

$$\begin{aligned} \dot{q}_{i,1} &= q_{i,2} \\ \dot{q}_{i,2} &= \frac{1}{m_i}\left(-k_{p,i}q_{i,1} - (f_{v,i} + k_{d,i})q_{i,2} + k_{p,i}q_{i+1,1} + k_{d,i}q_{i+1,2} - d_i k_{p,i}\right), \\ \dot{q}_{n,1} &= q_{n,2} \\ \dot{q}_{n,2} &= \frac{1}{m_n}\left(-f_{v,n}q_{n,2} + u(t)\right). \end{aligned} \quad (7)$$

where $i = 1, 2, \dots, n-1$.

Remark 1 *Eq.* (5) *models the interactions between the robots but not to fully describe the dynamics of each robot individually. We consider that Eq.* (5) *is a good approximation to the dynamics of the robotic formation when the robots move with low velocities. This is a feasible assumption in some practical scenarios.*

3 Stability Analysis

Since our process is a simple linear model, we can use any linear systems stability analysis methodology. For instance, by defining the system's output as $\mathbf{Y} = \mathbf{Cq} + \mathbf{D}$, we can use the expression $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ to find a transfer function to do the analysis in the Laplace domain.

To clarify this idea, let us consider the case where we have three mobile robots. Then, for $n = 3$ in Eq. (5) we have

$$\dot{\mathbf{q}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k_{p,1}}{m_1} & -\frac{k_{d,1}+f_{v,1}}{m_1} & \frac{k_{p,1}}{m_1} & \frac{k_{d,1}}{m_1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{k_{p,2}}{m_2} & -\frac{k_{d,2}+f_{v,2}}{m_2} & \frac{k_{p,2}}{m_2} & \frac{k_{d,2}}{m_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{f_{v,3}}{m3} \end{bmatrix} \begin{bmatrix} q_{1,1} \\ q_{1,2} \\ q_{2,1} \\ q_{2,2} \\ q_{3,1} \\ q_{3,2} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{k_{p,1}}{m_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{k_{p,2}}{m_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ d_1 \\ 0 \\ d_2 \\ 0 \\ u(t) \end{bmatrix}. \tag{8}$$

Specifically, let us assume that we can write the system's output as

$$\mathbf{Y} \ = \ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_{1,1} \\ q_{1,2} \\ q_{2,1} \\ q_{2,2} \\ q_{3,1} \\ q_{3,2} \end{bmatrix}, \tag{9}$$

i.e. by measuring the position of every agent in the system.

Then, we can obtain the MIMO transfer function

$$G(s) = \begin{bmatrix} 0 & G_{12} & 0 & G_{14} & 0 & G_{16} \\ 0 & 0 & 0 & G_{24} & 0 & G_{26} \\ 0 & 0 & 0 & 0 & 0 & G_{36} \end{bmatrix}, \tag{10}$$

where

$$
\begin{aligned}
G_{12} &= -\frac{k_{p,1}}{s(f_{v,1}+k_{d,1}+m_1 s)+k_{p,1}},\\
G_{14} &= -\frac{k_{p,2}(k_{d,1}s+k_{p,1})}{(s(f_{v,1}+k_{d,1}+m_1 s)+k_{p,1})(s(f_{v,2}+k_{d,2}+m_2 s)+k_{p,2})},\\
G_{16} &= \frac{m_3(k_{d,1}s+k_{p,1})(k_{d,2}s+k_{p,2})}{s(f_{v,3}+m_3 s)(s(f_{v,1}+k_{d,1}+m_1 s)+k_{p,1})(s(f_{v,2}+k_{d,2}+m_2 s)+k_{p,2})},\\
G_{24} &= -\frac{k_{p,2}}{s(f_{v,2}+k_{d,2}+m_2 s)+k_{p,2}},\\
G_{26} &= \frac{m_3(k_{d,2}s+k_{p,2})}{s(f_{v,3}+m_3 s)(s(f_{v,2}+k_{d,2}+m_2 s)+k_{p,2})},\\
G_{36} &= \frac{m_3}{f_{v,3}s+m_3 s^2}.
\end{aligned}
\tag{11}
$$

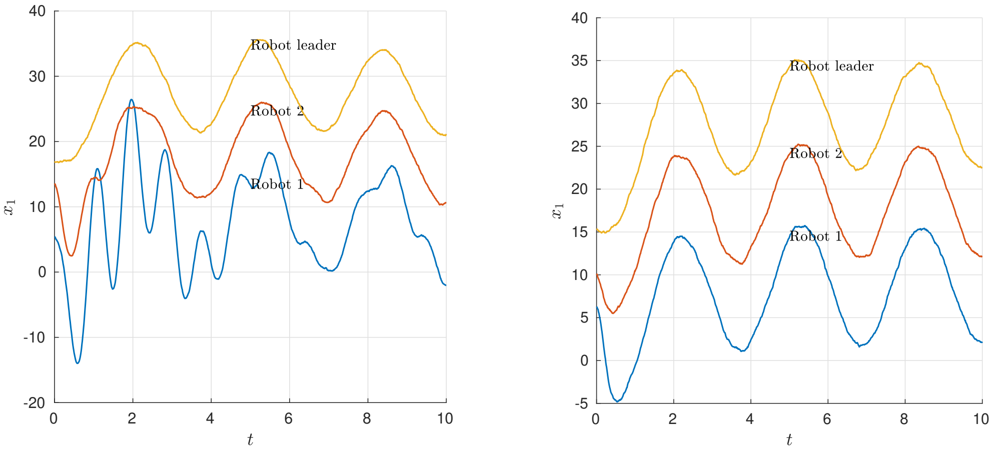
As we can see from Eq. (11), the stability of our process can be studied by the characteristic polynomial of G_{16} which is given by

$$
P(s) = s(f_{v,3}+m_3 s)(s(f_{v,1}+k_{d,1}+m_1 s)+k_{p,1})(s(f_{v,2}+k_{d,2}+m_2 s)+k_{p,2}). \tag{12}
$$

Note that the polynomial (12) contains the roots of the rest of the transfer functions' characteristic polynomials.

Since all the parameters in the system are positive, by taking $k_{p,1}, k_{d,1}, k_{p,2}, k_{d,2} > 0$ we would make the system stable (Hurwitz polynomial). We could further add time delays to the process to use more elaborated stability analysis methodologies (for instance, see [12]).

It is important to notice that a selection of gains that stabilises the process does not guarantee a proper behaviour. For instance, we consider the pair of simulations shown in figure 2. Here, even though, both simulations use stabilising gains, one of them shows a crashing between the robots. In such robotic formations, the more robots in the process the more susceptible the system is to transmit oscillations. Therefore, to design proper controller gains, we must select the transfer function from Eq. (11) that provides us with the most useful transient dynamical information. For the case of three robots, this corresponds to $G_{14}(s)$ because it relates the distance d_2 with the mobile's position q_1. In other words, $G_{14}(s)$ relates the dynamics of the mobiles next and farthest to the leader. Let us recall that every agent can only see who is in front and not who is behind of itself.



(a) Undesired behaviour. There exist crashing between the robots. The values $k_p = 8$, $k_d = 1$ are used in every robot.

(b) Desired behaviour. There is no crashing between the robots. The values $k_p = 50$, $k_d = 10$ are used in every robot.

Figure 3: System of 3 robots. $u(t) = 100\sin(t)$, all masses $m_i = 1$, $f_{v,i} \in (0, 2)$ and all $d_i = 10$.

4 Algorithm for the controller's design

Designing a good controller depends on the goal to be achieved. Thus, our problem can be described as an optimisation problem in a parameters' hyperspace (the space of the controllers gains). We can formulate the optimisation problem as

$$\min_{\mathbf{k}} \left[\mathcal{J}(t)\right] \;\; \textbf{s.t.} \;\; \mathbf{k} \in \mathbf{S} \tag{13}$$

where $\mathbf{k} \in \mathbb{R}^n$ is the set of control parameters inside the stability region $\mathbf{S}$ and $\mathcal{J}$ is our cost function. In this work we use the Root-Mean-Square Deviation (RMSD) given by:

$$\mathcal{J} \;\; = \;\; \sqrt{\frac{\sum_{t=1}^{T}(u_t - \hat{y}_t)^2}{T}}. \tag{14}$$

Here, $\hat{y}_t$ is the output of the transfer function with the characteristic polynomial $P(s)$ using a set of control parameters $\mathbf{k}$ with an input equal to the desired response u_t. T is the total number of iterations.

4.1 Solution to the optimisation problem

Due to the complexity of the optimisation problem formulated by Eq. (13), in this section, a evolutionary algorithm is proposed as an starting point. We describe it as

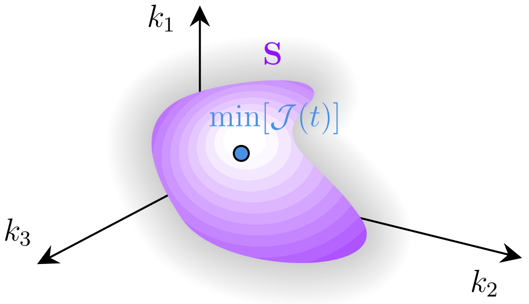


Figure 4: Diagram of the optimisation problem. Finding the vector $\mathbf{k}$ inside the stability region $\mathcal{S}$, that minimise the RMSD $\mathcal{J}$ in a 3rd order problem.

a pseudocode in Algorithm 1.

Some important features of this algorithm are the way in which we compute the score of each element in the population and the way in which we create the new population. For the latter, we use a Blend crossover operator (BLX) [17]. The fitness/score is computed by using the $exp(-\text{RMSD})$ in order to normalise the RMSD while penalising big RMSDs.

5 Results

In this section, we discuss results of some simulations of system (5) for the case $n = 3$ subject to a set of controller gains $\mathbf{k}$ designed by means of Algorithm 1. For the present simulations, we have computed the RMSD of the time response of $G_{14}(s)$ using two different reference signals $u(t) = 1$ (regulation) and $u(t) = 30\sin(2t)$ (tracking) in Algorithm 1. Besides, we have used all $m_i = 1$ and $f_{v,i} = 0.5$ while keeping all agents desired distance between each other $d_i = 10$ in our simulations. Additionally, we have kept the maximum number of iterations of Algorithm 1 equal to 100. The simulation results are given in Figs. 5 to 8.

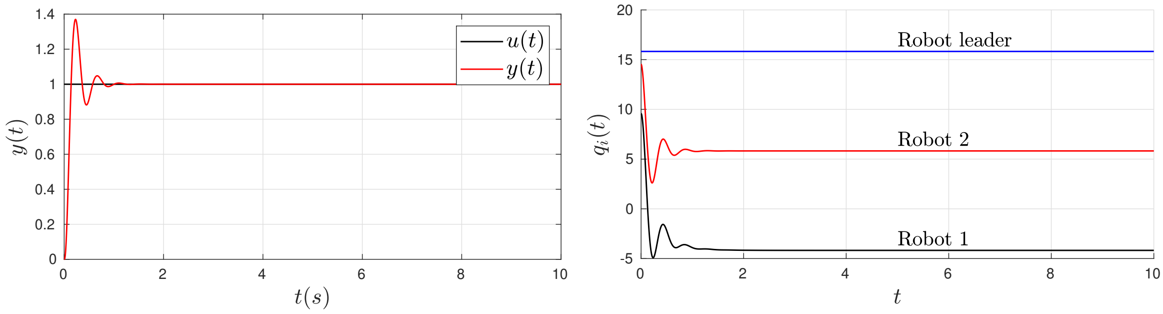
First, the results of the case $u(t) = 1$ are shown in Figs. 5 and 6. Here, Fig. 5 presents the case when the elements k_i of the controller gains vector $\mathbf{k}$ are in the range $0 < k_i < 500$, this large range of values allow us to have a faster step response of $G_{14}(s)$ as it is depicted in Fig. 5a (fast an overdamped response). Let us recall that by using the RMSD as a cost function, we force Algorithm 1 to find the set of gains $\mathbf{k}$ that minimise the difference between $u(t)$ and $y(t)$ for a given time. In our simulations, we have used a final simulation time $t_f = 1$ to get a fast response. Using a larger t_f would make our algorithm computationally more expensive. In addition, Fig. 5b shows that every agent lasts at least 2 seconds to reach to the desired separation between each other. This agrees with the step response of $G_{14}(s)$

```
Algorithm 1: Control design algorithm.
Data: Consider the initial population
      $\mathcal{P}_0 := \{\mathbf{k}_1, \ldots, \mathbf{k}_N | \mathbf{k}_i \in \mathcal{S} \forall i = 1, \ldots, N\}$ where $\mathcal{S} \subset \mathbb{R}^n$ is the stability
      region of the parameter space $\mathbf{k}$, the input $u(t)$ and the total
      simulation time $t_f$.
Result: The value of the gain vector $\mathbf{k}_{\hat{i}}$ for which $\mathcal{J}$ is minimum over the
        time $t \in (0, t_f)$ for the input $u(t)$.
/* Score the initial population $\mathcal{P}_0$ */
$\mathbf{F} := [\mathcal{F}_1, \ldots, \mathcal{F}_N]^T$  // Vector with computed fitness of each element
    in the population.
$\mathbf{E} := [E_1, \ldots, E_N]^T$ // Vector with computed RMSDs of each element in
    the population.
/* Compute the fitness value.  */
for $i \leftarrow 1$ to $N$ do
    $E_i = \mathcal{J}(\mathbf{k}_i \in \mathcal{P}_0, u, t_f)$
    $\mathcal{F}_i = e^{-E_i}$
end
$f_b = \min(|1 - \mathbf{F}|)$            // Find the best score $f_b$ in the initial
    population and compare it with a desired goal $d_g$.
/* Repeat the process until the desired goal is accomplish */
while $f_b > d_g$ do
    $\mathcal{P}_N = \mathrm{BLX}(\mathcal{P}_0)$       // Create a new population $\mathcal{P}_N$ using the BLX
        algorithm on the old population $\mathcal{P}_0$.
    for $i \leftarrow 1$ to $N$ do
        $E_i = \mathcal{J}(\mathbf{k}_i \in \mathcal{P}_N, u, t_f)$
        $\mathcal{F}_i = e^{-E_i}$
    end
    $\mathcal{P}_0 = \mathcal{P}_N$            // The new population replaces the old one.
    $f_b = \min(|1 - \mathbf{F}|)$
end
$\hat{i} = \mathrm{find}(|1 - \mathbf{F}| = f_b)$       // Find the position $\hat{i}$ of $\mathbf{k}_{\hat{i}} \in \mathcal{P}_N$ with the
    best score.
return $\mathbf{k}_{\hat{i}} \in \mathcal{P}_N$
```

subject to the designed controller gains $\mathbf{k}$ (see Fig. 5a). On the other hand, Fig. 6 represents a similar scenario although here we have diminished the range at which ever element k_i of the set of controller gains $\mathbf{k}$ is generated. This gives us a smooth but slower step response as indicated in both the step response of $G_{14}(s)$ (see Fig.

5b) and the time response of the robotic formation (see Fig. 6b).

For the case when $u(t) = 30\sin(2t)$, we have again implemented our design algorithm by changing the range of possible values that every element k_i of our controller $\mathbf{k}$ can take. Fig. 7 shows the case when the elements k_i of the controller gains vector $\mathbf{k}$ are in the range $0 < k_i < 50$. This gives a set of gains that does not completely reduce the value of RMSD, as it is shown in the time response of $G_{14}(s)$ (see Fig. 7a). Nonetheless, for this set of values the robotic formation shows a good performance (see Fig. 7b). On the contrary, Fig. 8, shows the case when the elements k_i are in the range $0 < k_i < 500$. This allows us to obtain a smaller RMSD value, as we can see from the time response in Fig. 8a, but it also adds small oscillations at the beginning of the robotic formation experiment (see Fig. 8b).



(a) $G_{14}(s)$ step response.

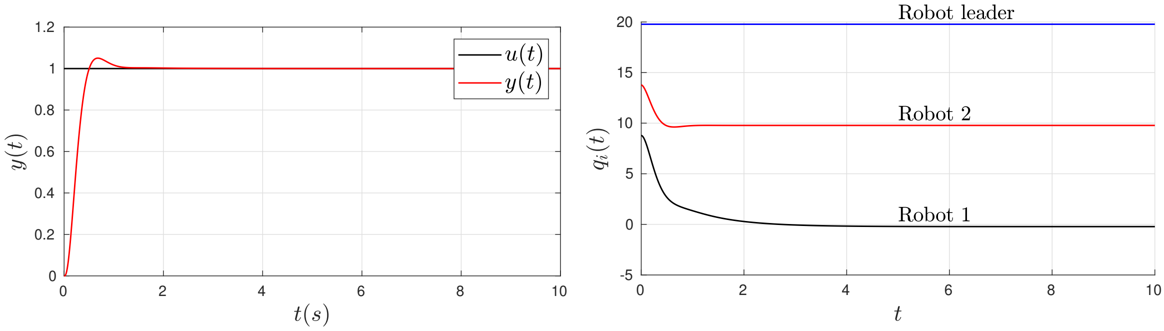
(b) Formation behaviour.

Figure 5: System of 3 robots. $u(t) = 0$, all $m_i = 1$, $f_{v,i} = 0.5$, $d_i = 10$ and the elements k_i of the controller gains vector $\mathbf{k}$ are in the range $0 < k_i < 500$. We iterate Algorithm 1 one hundred times. The final value of $\mathbf{k} = [k_{p,1}, k_{d,1}, k_{p,2}, k_{d,2}]^T = [291.2359, 103.4591, 229.6364, 9.0647]^T$.

6 Conclusions

This work presents a linear model reduction for a line formation of mobile robots driven by classical PD-controllers. This model permits us to easily design stabilising controllers for each agent in the formation. The proposed model can be generalised to study systems in higher dimensions by considering the particles in a plane. We have also analysed the use of an evolutionary algorithm to find the best set of control parameters that improves the network performance. The results of our algorithm are discussed together with some simulations.

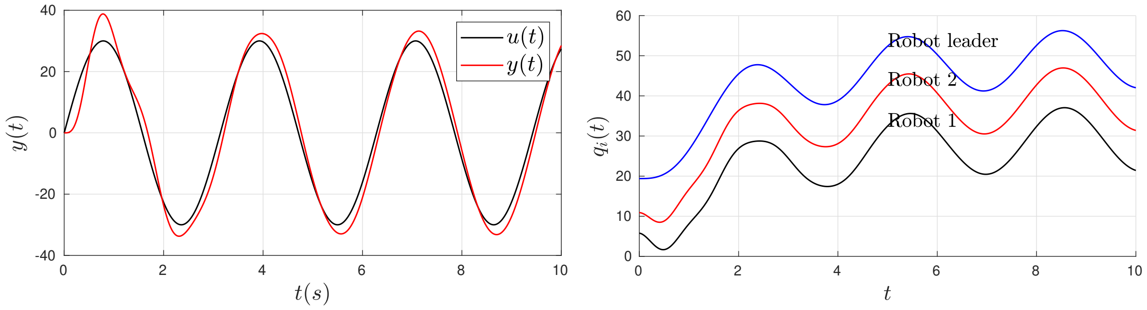
Future work includes exploring several novel and well known linear control design techniques (for instance, see [9, 10]), the generalisation of the model to a multidimen-



(a) $G_{14}(s)$ step response.

(b) Formation behaviour.

Figure 6: System of 3 robots. $u(t) = 0$, all $m_i = 1$, $f_{v,i} = 0.5$, $d_i = 10$ and the elements k_i of the controller gains vector $\mathbf{k}$ are in the range $0 < k_i < 50$. We iterate Algorithm 1 one hundred times. The final value of $\mathbf{k} = [k_{p,1}, k_{d,1}, k_{p,2}, k_{d,2}]^T = [37.8257, 33.5179, 48.2762, 9.5010]^T$.



(a) $G_{14}(s)$ time response.

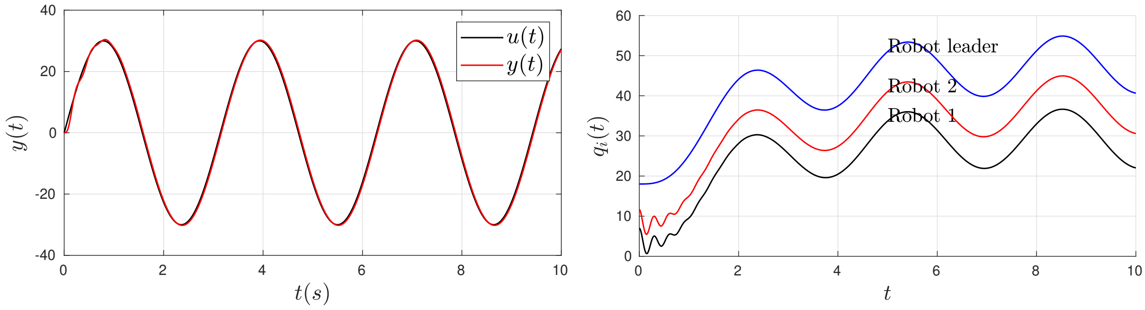
(b) Formation behaviour.

Figure 7: System of 3 robots. $u(t) = 30\sin(2t)$, all $m_i = 1$, $f_{v,i} = 0.5$, $d_i = 10$ and the elements k_i of the controller gains vector $\mathbf{k}$ are in the range $0 < k_i < 50$. We iterate Algorithm 1 one hundred times. The value of $\mathbf{k} = [k_{p,1}, k_{d,1}, k_{p,2}, k_{d,2}]^T = [31.4295, 32.2729, 45.6248, 0.7629]^T$.

sional framework and the improvement of the optimisation method for the controller's design.

References

[1] Knorn, S., Chen, Z., Middleton, R.H.: Overview: Collective control of multiagent systems. IEEE Transactions on Control of Network Systems 3(4), 334–347 (2015)



(a) $G_{14}(s)$ time response.

(b) Formation behaviour.

Figure 8: System of 3 robots. $u(t) = 30\sin(2t)$, all $m_i = 1$, $f_{v,i} = 0.5$, $d_i = 10$ and the elements k_i of the controller gains vector $\mathbf{k}$ are in the range $0 < k_i < 500$. We iterate Algorithm 1 one hundred times. The value of $\mathbf{k} = [k_{p,1}, k_{d,1}, k_{p,2}, k_{d,2}]^T = [28.0426, 213.5836, 440.8379, 5.3353]^T$.

[2] Bussmann, S., Jennings, N.R., Wooldridge, M.: Multiagent systems for manufacturing control: a design methodology. Springer Science & Business Media (2013)

[3] Zhang, D., Feng, G., Shi, Y., Srinivasan, D.: Physical safety and cyber security analysis of multi-agent systems: A survey of recent advances. IEEE/CAA Journal of Automatica Sinica 8(2), 319–333 (2021)

[4] Ma, J., Tang, J.: A review for dynamics of collective behaviors of network of neurons. Science China Technological Sciences 58(12), 2038–2045 (2015)

[5] Giannakis, M., Louis, M.: A multi-agent based system with big data processing for enhanced supply chain agility. Journal of Enterprise Information Management (2016)

[6] Rodriguez, M.A., Steinbock, D.J., Watkins, J.H., Gershenson, C., Bollen, J., Grey, V., Degraf, B.: Smartocracy: Social networks for collective decision making. In: 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07). pp. 90–90. IEEE (2007)

[7] Oh, K.K., Park, M.C., Ahn, H.S.: A survey of multi-agent formation control. Automatica 53, 424–440 (2015)

[8] Åström, K.J., Hägglund, T.: The future of pid control. Control engineering practice 9(11), 1163–1175 (2001)

[9] Méndez-Barrios, C., Niculescu, S.I., Moraraescu, C.I., Gu, K.: On the fragility of pi controllers for time-delay siso systems. In: 2008 16th mediterranean conference on control and automation. pp. 529–534. IEEE (2008)

[10] Guel-Cortez, A.J., Méndez-Barrios, C.F., Kim, E.j., Sen, M.: Fractional-order controllers for irrational systems. IET Control Theory & Applications (2021)

[11] Guel-Cortez, A.J., Méndez-Barrios, C.F., González-Galván, E.J., Mejía-Rodríguez, G., Félix, L.: Geometrical design of fractional pdμ controllers for linear time-invariant fractional-order systems with time delay. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 233(7), 815–829 (2019)

[12] Hernández-Díez, J.E., Méndez-Barrios, C.F., Mondié, S., Niculescu, S.I., González-Galván, E.J.: Proportional-delayed controllers design for lti-systems: a geometric approach. International Journal of Control 91(4), 907–925 (2018)

[13] Guel-Cortez, A.J., Kim, E.j.: Information length analysis of linear autonomous stochastic processes. Entropy 22(11), 1265 (2020)

[14] Annunziato, M., Borzì, A.: A fokker–planck control framework for multidimensional stochastic processes. Journal of Computational and Applied Mathematics 237(1), 487–507 (2013)

[15] Mwaffo, V., DeLellis, P., Humbert, J.S.: Formation control of stochastic multi-vehicle systems. IEEE Transactions on Control Systems Technology (2021)

[16] Ramos-Ávila, D., Rodríguez, C., Hernández-Carrillo, J., Guel-Cortez, A.J., Sen, M., Méndez-Barrios, C.F., González-Galván, E., Goodwine, B.: Experiments with PD-controlled robots in ring formation. In: XXI Congreso Mexicano de Robótica – COMRob (2019)

[17] Abido, M.: Abido, m.a.: A novel multiobjective evolutionary algorithm for environmental / economic power dispatch. electric power systems research 65(1), 71-81. Electric Power Systems Research 2, 71–81 (04 2003)