Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Group Synchronization in Coordination Tasks via Network Control Methods

Sidney N. Givigi * **Kleber M. Cabral** ** **Peter T. Jardine** **

* *School of Computing, Queen's University, Kingston, ON K7L 3N6 Canada (e-mail: sidney.givigi@queensu.ca)*
** *Department of Electrical and Computer Engineering, Royal Military College National, Kingston, ON K7K 7B4 Canada (e-mail: {cabral,peter.jardine}@rmc.ca).*

Abstract: This paper deals with the coordination problem among robots and between robots and humans using network control methods. In several applications, robots need to collaborate with humans in order to perform tasks, such as in collaborative transportation of objects, cooperative assembly of structures, or production line activities. In all these cases, the robot needs to observe the environment and take actions according to one of more other agents; human or robotic. When only one more agent is involved (*dyadic* interactions), the problem is relatively well studied. However, this paper focuses on when more than one other agent is involved in the collaboration. All agents are represented as part of a graph that determines their communication architecture. Network controllers are proposed for this environment and simulations show that synchronization of all agents is achieved. Furthermore, experiments show that a virtual agent is able to efficiently interact with humans and synchronize itself to their motion. To our knowledge this is the first time network control was explicitly applied to this problem.

Keywords: Networked control, pinning control, robots, human-machine interaction, synchronization.

1. INTRODUCTION

In the last decade, more emphasis has been given to smart manufacturing processes (Kang et al., 2016) for increased productivity. The use of robots (usually called collaborative robots, or cobots) in partnership with humans is one of the technologies that have been investigated in this context (Sadrfaridpour and Wang, 2018; Mitrea and Tamas, 2018). Several of the tasks that require Human-Robot Interaction (HRI) depend on the automatic synchronization of all involved players, humans and robots. Some of these tasks are the hybrid manufacturing cells for assembly (Sadrfaridpour and Wang, 2018), collaborative lifting of objects (DelPreto and Rus, 2019), folding cloths (Saxena et al., 2017), or interacting with multiple users (Faria et al., 2017).

The types of interactions that are usually considered in the literature only involve a single human and a single robot. These types of interactions are called *dyadic*. Other cases involve more than one human per robot (Saxena et al., 2017) or more than one robot per human (Lombardi et al., 2019).

Some tasks need that the robots be synchronized with other agents. For example, humans can hand objects to a robot or vice-versa (Edsinger and Kemp, 2007), or robots must cooperate with humans in sawing a log (Peternel et al., 2014). These tasks can be modeled as oscillatory motion and both humans and robots need to move at approximately the same frequency.

For tasks that are oscillatory, from a control perspective (from the point of view of the robot), one important issue is how to model the interaction between humans and robots. Synchronization, in this sense, is better modeled as a nonlinear dynamical process and involves some prediction of the behaviour of the other agents so synchronism can be achieved. One agent predicts the movement of the other agent and adjusts its motion accordingly (Vesper et al., 2010). This is true of humans and possibly also of robots.

This paper deals with the problem of synchronization between humans and robots in oscillatory tasks. The particular task in which we are interested is known as the *mirror game* (Słowiński et al., 2016). In this game, all players try to synchronize their motion. For example, the agents oscillate their fingers in order to approximate the motion of their neighbours (or of the agents that she/he sees). The problem is modeled in a network control formulation and the desired behaviour is that all connected agents synchronize their motion. In other words, the controllers seek the ability of guiding the motion of individual agents towards desired states (Liu et al., 2011; Nozari et al., 2019). In the specific case, the position of the finger of all the neighbours (more realistically the average position of the fingers of the neighbours). The problem is modeled as a graph in which the nodes are the agents and the egdes represent the communication among the agents.

Several different solutions for this problem have been proposed in the literature. For example, an optimal controller was developed to control a virtual player interacting with a human (Zhai et al., 2016). If machine learning is used, Q-learning (Lombardi et al., 2018) or deep learning (Lombardi et al., 2019) can be successfully used. Optimal control can be very time consuming and depends on a series of parameters to be properly tuned. Q-learning and deep learning techniques depend on training data that may be difficult to come by. In this paper, the control of the network of agents follows two strategies, either a *pinning control* strategy (Wang and Su, 2014) when synchronization among different virtual agents is done, or a network control strategy when the virtual agent interacts with human players.

After formulating the solution of the mirror game as a network control strategy, we first show that our solution provides good results when only simulated agents are used. The simulations use agents in different communication configurations and show that synchronization is achieved very quickly. Then, we use the same approach in a live experiment in which a virtual agent (or robot) is executed in a computer in a network and two human volunteers interact with it through a track-pad. Results show that synchronization is achieved as if only human operators were interacting.

The rest of the paper is organized as follows. In Section 2, the mirror game problem is defined, including the metrics for evaluating the players' performance. Section 3 describes how networked control and more specifically the pinning control strategy is used in the simulation for synchronization. The experimental setup is also described. In Section 4, the main results of the paper are presented. Finally, section 5 concludes the paper.

2. PROBLEM FORMULATION

In this section we discuss the formulation of the problem. We start by describing the interaction between agents as a graph in Section 2.1. The dynamic model of the virtual agents is presented in Section 2.2. The performance metrics are introduced in Section 2.3.

2.1 Graph Formulation

The game is defined as a graph in which each agent (human or robot) is a node and the communication between them is an edge.

Formally, the network of agents can be represented as a graph, $G = \{V, E, \mathbf{A}\}$. The node set is defined as $V = \{v_i\}$, and the edge set is $E = \{e_{ij}\}$, $i, j = 1, 2, ..., N$. Two agents are connected, i.e., they interact, if there is an edge between them. In practice, this means that one player can see the other and measure the position of its "finger" (an actual finger in the case of human players or a moving dot in the case of a virtual agent), i.e., if $e_{ij} \neq 0$, agent i can sense (or measure) the state of agent j . Note that the set is not necessary symmetric, i.e., if player i can see player j, it does not necessarily follow that player j can see player i. Therefore, it is possible that the graph is undirected, thus $e_{ij} \neq e_{ji}$. If there are N players in the game, the adjacency matrix $\mathbf{A}$ is an $N \times N$ positive semidefinite matrix that describes the network topology and a_{ij} are the elements of $\mathbf{A}$. We consider that $a_{ij} = 1$ if $\exists\, e_{ij}$, and $a_{ij} = 0$ if $\nexists\, e_{ij}$.

2.2 Virtual Agent Model

The virtual agent is defined as a Haken-Kelso-Bunz (HKB) oscillator (Haken et al., 1985):

$$\ddot{x}_i + (\alpha_i x_i^2 + \beta_i \dot{x}_i^2 - \gamma_i)\dot{x}_i + \omega_i^2 x_i = u_i \tag{1}$$

where x_i is the position of the i^{th} agent; α_i, β_i, γ_i and ω_i are parameters intrinsic to the motion of the i^{th} agent.

For the simulations to be discussed in Section 4, the derivative $\dot{x}_i$ in equation (1) can be calculated with the integration of equation (1); whereas, for the experiments, the derivatives are approximated by its first order approximation

$$\dot{x}_i = \frac{x_i(t_k) - x_i(t_{k-1})}{t_k - t_{k-1}} \tag{2}$$

2.3 Performance Evaluation

The performance of the task can be measured in different ways. For example, we can track the error of each player's position $(x_i(t))$ with respect to the mean position of the group $(\bar{x}(t) = \sum_{i=1}^{N} x_i(t))$, i.e.,

$$\epsilon_i = |x_i(t) - \bar{x}(t)| \tag{3}$$

Another way is by calculating the relative phase of each agent. If we let $\theta_i(t)$ be the phase of the position signal $x_i(t)$ of the $i^t h$ player, we can define the average phase of the group as

$$q(t) = \frac{1}{N} \sum_{i=1}^{N} e^{j\theta_i(t)}.$$

Then, the relative phase of each agent is

$$\phi_i(t) = \theta_i(t) - q(t) \tag{4}$$

Then, we can track the group synchronization index (Alderisio et al., 2017). If we further define $\bar{\phi}_i$ as the mean phase of the i^{th} agent, the coordination of the group (or the group synchronization index (Alderisio et al., 2017)) is

$$\rho_g(t) = \frac{1}{N} \left| \sum_{i=1}^{N} e^{j(\phi_i(t) - \bar{\phi}_i)} \right| \tag{5}$$

For the results presented in Section 4, we will calculate the metrics (3), (4) and (5) for the agents.

3. CONTROL

In this section we describe the techniques used for the control of the virtual agent. Section 3.1 introduces the general networked control problem. Section 3.2 describes how synchronization is defined in terms of the networked control problem. Section 3.3 describes the pinning control strategy used in simulation. Finally, the mirror game is discussed in Section 3.4.

3.1 Networked Control

In multi-agent networked systems, information flows from agent to agent and decisions are made be each individual controller. Using the notation discussed in Section 2.1, each node in the set of vertices V is identified as an agent and the edges in the set E define the communication between the agents. By using the information received, agents change their behaviour. In our problem, information is received from agents via measurement of their positions. Then the problem is to find the control signal such that the network reaches the desired behaviour. The dynamics of the agent with respect to their neighbours is described by

$$\bar{u}_i(t) = \sum_{j=1, j\neq i}^{N} c_{ij} a_{ij}$$
$$\dot{\bar{x}}_i = f(\bar{x}_i(t), \bar{u}_i(t), t) \tag{6}$$

where:

- $\bar{u}_i$ is a control signal;
- c_{ij} is the coupling strength between the nodes v_i and v_j. These parameters are chosen in order to drive the system to its desired behaviour;
- a_{ij} are the values from the *adjacency matrix* $\mathbf{A}$;
- $\bar{x}_i = [x_i^1, \dot{x}_i^n]^\top \in \mathbb{R}^n$ is the state vector of the i^{th} node;
- $f : \mathbb{R}^n \times \mathbb{R}^m \times [0, +\infty) \to \mathbb{R}^n$ is a continuous map function implemented directly from (1).

Notice that it is possible to generalize the formulation in which the control signal $\bar{u}_i(t)$ is not given for all agents, but only a subset $K \subseteq V$. This will be further explored in Section 3.3. Now we will define the objective of the control, i.e., the synchronization of the agents.

3.2 Synchronization of the Agents

Network synchronization means that the states of all nodes converge to a common value. A complete synchronization occurs when:

$$\lim_{t\to\infty} \|\bar{x}_i(t) - \bar{x}_j(t)\| = 0 \tag{7}$$

for all $i, j = 1, ..., N$.

Synchronization is an interesting problem for the evaluation of a control law, since all the nodes in the network must reach the same value. Note that the system behaviour achieved through network synchronization depends on what the nodal states are representing. For example, if the states are modeling the position of the agents, all players would try to synchronize the amplitude and phase of the motion (as in the case studied in this paper). However, if the nodal states are describing the frequency of the motion, "synchronized" agents would only repeat the same motion without regard for the amplitude of the motion.

3.3 Pinning Control

As discussed before, in the context of networked control, the term *pinning control* defines the placement of local feedback controllers on a small fraction of the network nodes. Such nodes are called pins or pinned nodes. The nodes that are not directly actuated will be influenced only through their connections with the pins (Zhi-Hong Guan et al., 2010; Adaldo et al., 2015).

The problem discussed in this paper is similar to synchronization of networks with time-delay (Liu et al., 2010). Each agent observes the output of the other agents and uses that as its own input as if that was a common input reference trajectory.

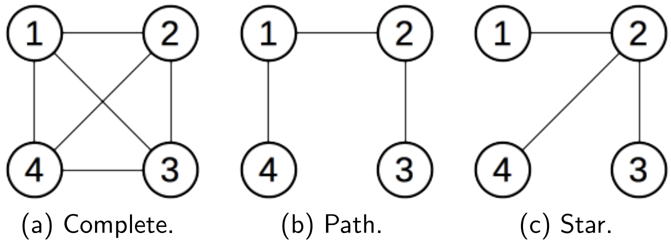


Fig. 1. Communication configurations for the mirror game.

In pinning control, the number of pins could be as high as the total number of nodes in the network, i.e., if K is the set of pins (the nodes that are actuated as defined in Section 2.1), $K \subseteq V$. However, Chen et al. (2007) proved that under certain conditions, as few as only one controller (actuating on a single node) is capable of synchronizing the whole network to a target trajectory.

Let us consider again equation (6). Let us assume that:

- the controlled nodes (input set) are the first M nodes of the network, thus $i = 1, ..., M, ..., N$;
- the controllers to be placed in a node $v_i \in K$ are $\bar{u}_i(t) = \kappa(\bar{x}_i(t) - \bar{s}(t))$, where κ is the *control gain.* This control law moves the states of the node $(\bar{x}_i(t))$ to a reference $(\bar{s}(t))$. Note that all nodes in K have the same controller gain (κ) and will track the same reference $(\bar{s})$.

The objective of the controller is that all nodes be able to track $\bar{s}$ at the same time, even the nodes that are not being directly controlled (Sorrentino et al., 2007) (i.e., they are all synchronized). The pinning controllability depends on the values of c_{ij} and κ (coupling and controller gains) over which the synchronization of the network with the reference is possible.

Given the assumptions above, we can rewrite equation (6) as (Sorrentino et al., 2007)

$$\bar{u}_i = c_{ij} \sum_{j=1, j\neq i}^{N} a_{ij}(\bar{x}_j(t) - \bar{x}_i(t))$$
$$+ c_{ij} \sum_{k=1}^{M} \delta(i-k)\kappa(\bar{s}(t) - \bar{x}_i(t))$$
$$\dot{\bar{x}}_i = f(\bar{x}_i(t), \bar{u}_i(t), t) \tag{8}$$

where δ is the Dirac delta function, defining if either a control input will be applied at the node v_i or not.

The control signal $\bar{u}_i(t)$ is formed by two terms. The first one describes the relationship between the node and its neighbours. The second term only applies the feedback control law if v_i is a controlled node (node in K), weighted by c_{ij}. The two values combined are applied to the internal dynamic of the node v_i.

3.4 Mirror game

For the application of networked control systems, we can represent the mirror game as a graph using the notation of Section 2.1. The game was implemented first on simulation and then in an experiment. For the simulation, the scenarios that we implemented are shown in Figure 1.

Table 1. Parameters for simulated agents.

Parameter	Player 1	Player 2	Player 3	Player 4
α	1	1	3	1
β	2	1	7	1
γ	1	1	0.2	1
ω	1	1.17	1.75	3.05

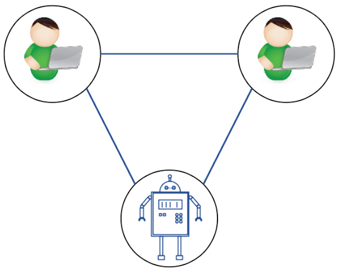


Fig. 2. Configurations for the experimental mirror game.

The three configurations were implemented to assess the robustness of the technique. In all cases, the pin node was always node 2, i.e., this node received the reference, and all the others tried only to synchronize themselves using equation (8). For the simulations, the parameters for the parameters in equation (1) are listed in Table 1. The values for α, β, γ, and ω are usually defined empirically (Lombardi et al., 2019) and all values were taken from the literature (Lombardi et al., 2019; Zhai et al., 2016; Haken et al., 1985; Zhai et al., 2015; Alderisio et al., 2016).

The experiment was executed with three instead of four nodes. In that case, the ring and complete graphs are exactly the same as shown in Figure 2. The two human volunteers had a track-pad and moved their fingers while trying to synchronize their motion. They also were shown an image in a monitor that simulated the motion done by the virtual agent, who also tried to synchronize its motion with the humans. The virtual agent implemented equation (1) with $\alpha = 1$, $\beta = 2$, $\gamma = 1$, and $\omega = 1$.

4. RESULTS

In this section we show the results for the simulation in Section 4.1 and the experiment in Section 4.2.

4.1 Simulation

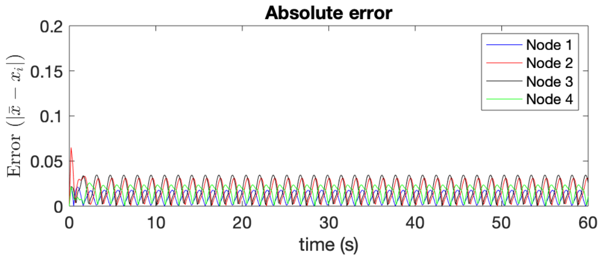
The simulation was run for several different initial conditions (initial position and velocity for each virtual player). The parameters of equation (8) were set so that all $c_{ij} = 2$ and $\kappa = 2$. Even though all players have different oscillation frequencies (the different values of ω in Table 1), the desired frequency of oscillation was set to 2 rad/s (i.e., the reference signal was set to be a sinusoidal with frequency 2 rad/s).

First we ran 100 simulations for the complete graph case (Figure 1a) and the path graph case (Figure 1b) . The results are shown in Table 2. As the table shows, the relative phase and root mean square (RMS) error for both cases are very small. This means that the players can efficiently follow the other players.

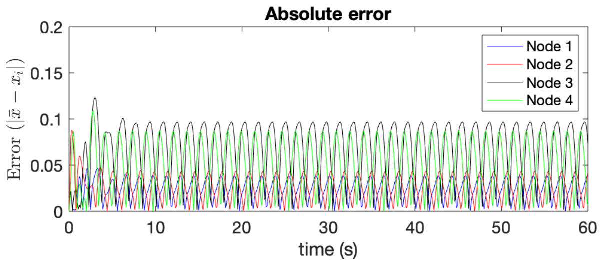
Figure 3 shows the results for a single run of the two graphs. In both cases, the complete graph (Figure 3a) and the path graph (Figure 3b), the errors are very small and all agents indeed converge to the behaviour of the others.

Table 2. Metrics for 100 trials in the complete graph and path graph cases.

Metric	Complete graph	Path graph
Relative phase	-0.0045 ± 0.0025	-0.0376 ± 0.0043
RMS	0.0228 ± 0.0038	0.0524 ± 0.0029



(a) Complete graph simulation.



(b) Path graph simulation.

Fig. 3. Absolute error for two graph configurations.

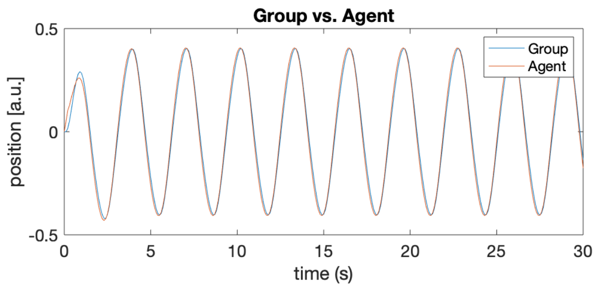


Fig. 4. Trajectories of the mean of the neighbours and an agent.

This is further shown in Figure 4. An agent (in this case agent 4) can perfectly track the mean of the behaviour of the other players. Also notice that the frequency of the group is in fact the value set in the reference, 2 rad/s. Note that we clipped the data to 30 seconds for better visualization. All the results shown are similar to the ones found in the literature using other control and/or machine learning techniques.

We then simulated the scenario shown in Figure 1c in order to set up the parameters of the controller for the experiment. In this case, there is no pin node and the equation used is (6). The value of the control parameters were set such as $c_{ij} = 2$. These are the same parameters used in the simulations shown so far. The mean position of the group and that of agent 2 is shown in Figure 5. Notice that the agent is indeed able to track the behaviour of the neighbors (we clipped again the data to 30 seconds

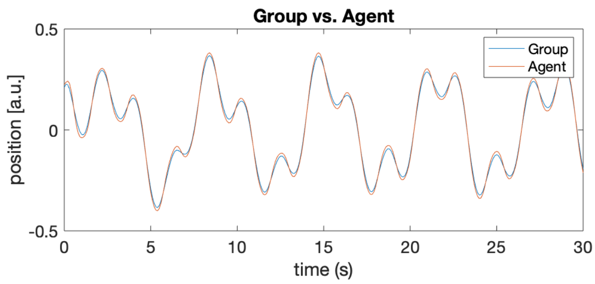


Fig. 5. Trajectories of the mean of the neighbours and agent 2 for the star graph.

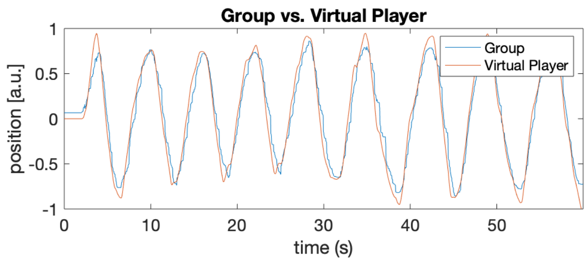


Fig. 6. Trajectories of the mean of two human volunteers and a virtual agent.

for better visualization). With this result, we proceeded to run the experiment described in Section 3.4.

4.2 Experiment

The experiment consisted of two human volunteers and a virtual agent. Each one of the volunteers was given a track-pad in which they could move their fingers and they were asked to try to synchronize their motion. After three trials, they were presented to a monitor representing a third track-pad. The motion in the monitor was that of a virtual agent running a networked control strategy described in equation (6). The setup just described is exactly the one shown in Figure 2.

Again, the values of the control parameters were set to $c_{ij} = 2$. Ten different tests were executed with varying parameters for the virtual agent. Different virtual agent parameters were used in the experiments. The parameters are the same described in Table 1 and three experiments were performed for the parameters shown in column "Player 1", three for the parameters shown in column "Player 2", two for the parameters shown in column "Player 3", and two for the parameters shown in column "Player 4". The paths of the group and the virtual agent for one experiment (with parameters of "Player 1") are shown in Figure 6. As it can be seen in the figure, the virtual agent can mimic the behaviour of the human players.

Another way to access the behaviour of the virtual agent is by comparing its error with respect to the mean of the neighbors to the errors of the human players. This is shown in Figure 7. We shown only the error for the interval $10\ s - 20\ s$ for clarity as the remaining of the interval has the same characteristics. Notice that the errors of the virtual agent and the humans are indistinguishable.

The same metrics calculated for the simulations were computed for the experiments. The results are shown in

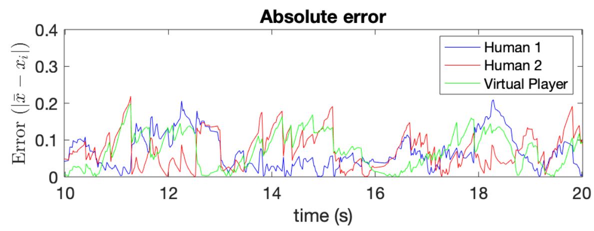


Fig. 7. Errors of human players and the virtual agent.

Table 3. Metrics for 10 experimental trials.

Metric	Experimental results
Relative phase	-0.0008 ± 0.0126
RMS	0.0913 ± 0.0235

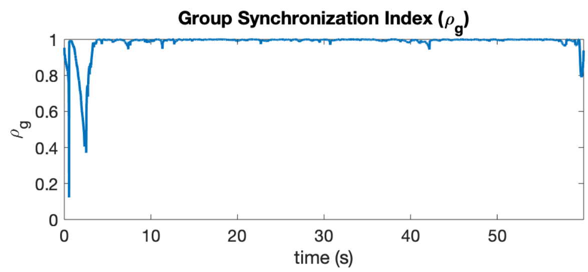


Fig. 8. Group synchronization index for an experiment between two human agents.

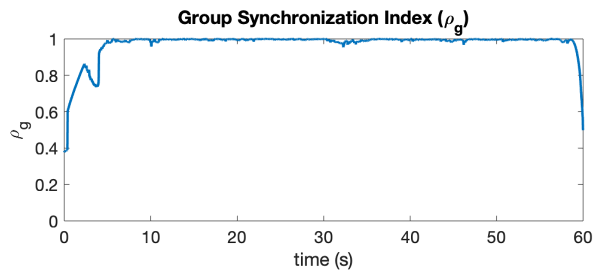


Fig. 9. Group synchronization index for an experiment involving two human and one virtual agents.

Table 3. The values for the metrics in the simulations (Table 2) and in the experiments (Table 3) have the same order of magnitude, meaning that the virtual agent has the same qualitative behaviour in simulation and experiments.

Finally, the group synchronization index was calculated for all experiments. First, the value was calculated for interaction between only the two human volunteers. The result for one trial is shown in Figure 8, where one can see that the players are able to synchronize perfectly. This result is consistent over all the trials. For the case of three agents interacting (two humans and a virtual agent), the behaviour is the same. A sample of the time evolution of the group synchronization index including the virtual player is shown in Figure 9. The pattern is always the same. The players take some time to reach a synchronization steady-state and then keep synchronized until the end of the trial.

5. CONCLUSION

In this paper we implemented a networked control strategy for the synchronization of a virtual agent (robot) and human players. Simulations demonstrated that the quality of the results using this technique can be compared to solutions using other control and/or machine learning techniques. However, the implementation of the networked control approach is more straightforward when compared to optimal control, as it does not need the correct derivation of a model for predictions or the implementation of and optimization approach such as gradient descent, or deep learning approaches, as it does not require a large amount of data for training.

The solution is robust to changes in the configuration of the network and to different types of agents. In the future, we intend to implement it to a robotic arm to analyze if the implementation advantages of the technique can be realized for more complex tasks such as cooperative assembly of structures.

REFERENCES

Adaldo, A., Alderisio, F., Liuzza, D., Shi, G., Dimarogonas, D.V., di Bernardo, M., and Johansson, K.H. (2015). Event-triggered pinning control of switching networks. *IEEE Transactions on Control of Network Systems*, 2(2), 204–213.

Alderisio, F., Antonacci, D., Zhai, C., and di Bernardo, M. (2016). Comparing different control approaches to implement a human-like virtual player in the mirror game. In *2016 European Control Conference (ECC)*, 216–221.

Alderisio, F., Fiore, G., Salesse, R.N., Bardy, B.G., and Bernardo, M.d. (2017). Interaction patterns and individual dynamics shape the way we move in synchrony. *Scientific Reports*, 7(1), 6846.

Chen, T., Liu, X., and Lu, W. (2007). Pinning complex networks by a single controller. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(6), 1317–1326.

DelPreto, J. and Rus, D. (2019). Sharing the load: Human-robot team lifting using muscle activity. In *2019 International Conference on Robotics and Automation (ICRA)*, 7906–7912.

Edsinger, A. and Kemp, C.C. (2007). Human-robot interaction for cooperative manipulation: Handing objects to one another. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, 1167–1172.

Faria, M., Silva, R., Alves-Oliveira, P., Melo, F.S., and Paiva, A. (2017). me and you together movement impact in multi-user collaboration tasks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2793–2798.

Haken, H., Kelso, J.A.S., and Bunz, H. (1985). A theoretical model of phase transitions in human hand movements. *Biological Cybernetics*, 51(5), 347–356.

Kang, H.S., Lee, J.Y., Choi, S., Kim, H., Park, J.H., Son, J.Y., Kim, B.H., and Noh, S.D. (2016). Smart manufacturing: Past research, present findings, and future directions. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 3(1), 111–128.

Liu, T., Zhao, J., and Hill, D.J. (2010). Exponential synchronization of complex delayed dynamical networks with switching topology. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(11), 2967–2980.

Liu, Y.Y., Slotine, J.J., and Barabási, A.L. (2011). Controllability of complex networks. *Nature*, 473(7346), 167–173.

Lombardi, M., Liuzza, D., and di Bernardo, M. (2018). Using learning to control artificial avatars in human motor coordination tasks. *arXiv e-prints*, arXiv:1810.04191.

Lombardi, M., Liuzza, D., and di Bernardo, M. (2019). Deep learning control of artificial avatars in group coordination tasks. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 724–729.

Mitrea, D. and Tamas, L. (2018). Manufacturing execution system specific data analysis-use case with a cobot. *IEEE Access*, 6, 50245–50259.

Nozari, E., Pasqualetti, F., and Cortés, J. (2019). Heterogeneity of central nodes explains the benefits of time-varying control scheduling in complex dynamical networks. *Journal of Complex Networks*.

Peternel, L., Petrič, T., Oztop, E., and Babič, J. (2014). Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach. *Autonomous Robots*, 36(1), 123–136.

Sadrfaridpour, B. and Wang, Y. (2018). Collaborative assembly in hybrid manufacturing cells: An integrated framework for humanrobot interaction. *IEEE Transactions on Automation Science and Engineering*, 15(3), 1178–1192.

Saxena, K., Labuguen, R., Joshi, R.P., Koganti, N., and Shibata, T. (2017). A study on human-robot collaboration for table-setting task. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 183–188.

Słowiński, P., Zhai, C., Alderisio, F., Salesse, R., Gueugnon, M., Marin, L., Bardy, B.G., di Bernardo, M., and Tsaneva-Atanasova, K. (2016). Dynamic similarity promotes interpersonal coordination in joint action. *Journal of the Royal Society Interface*, 13.

Sorrentino, F., Di Bernardo, M., Garofalo, F., and Chen, G. (2007). Controllability of complex networks via pinning. *Physical Review E*, 75(4), 046103.

Vesper, C., Butterfill, S., Knoblich, G., and Sebanz, N. (2010). A minimal architecture for joint action. *Neural Networks*, 23(8), 998 – 1003. Social Cognition: From Babies to Robots.

Wang, X. and Su, H. (2014). Pinning control of complex networked systems: A decade after and beyond. *Annual Reviews in Control*, 38(1), 103 – 111.

Zhai, C., Alderisio, F., Tsaneva-Atanasova, K., and di Bernardo, M. (2015). A model predictive approach to control the motion of a virtual player in the mirror game. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 3175–3180.

Zhai, C., Alderisio, F., Sowiski, P., Tsaneva-Atanasova, K., and di Bernardo, M. (2016). Design of a virtual player for joint improvisation with humans in the mirror game. *PLOS ONE*, 11(4), 1–17.

Zhi-Hong Guan, Zhi-Wei Liu, Gang Feng, and Yan-Wu Wang (2010). Synchronization of complex dynamical networks with time-varying delays via impulsive distributed control. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(8), 2182–2195.