POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Assistive controller for physical human-robot interaction based on cooperative game theory and human intention estimation

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'Automazione

Author: Davide Cassinelli

Advisor: Prof. Paolo Rocco

Co-advisor: Paolo Franceschi, Nicola Pedrocchi

Academic year: 2021-2022

1. Introduction

The application of robotics from industry to the human environment is constantly growing and has led to the development of human-robot interaction (HRI). In particular, when the interaction is physical, we refer to the physical human-robot interaction (pHRI) [5].

In this context, new and possibly adaptive controllers are needed to assist human operators in performing shared tasks.

Among the possible control schema, Game-Theory (GT) based controllers are widely used to model the interaction between different users [4]. GT offers a solution to the problem of a multi-agent system adopting a cooperative scenario [1] focused on how to maximize the interests of the participants in the game.

One of the assumptions underlying the GT problem is that each player knows the opponent's goal. Therefore, in pHRI, the knowledge of human intention assumes a lot of importance. In this work, human intention is defined as the human's desire to move an object from a starting to a target point, with the assistance of a robot, following a trajectory, over a finite rolling prediction horizon. The method used for human intention estimation is based on recurrent neural networks (RNNs), which are very effective when dealing with sequence-to-sequence learning applications, [3]. In particular, the structure is composed of cascaded Long-Short-Term Memory (LSTM), used to solve various problems where the sequential data and Fully Connected (FC) layers (RNN+FC). The prediction of the RNN+FC is planned to use information about a portion of the trajectory before the current state to predict a future portion of the trajectory. Iterative training is proposed to adapt the model, but it is time-consuming. To resolve this and to adapt to new users/objects, a Transfer Learning (TL) approach has been proposed to transfer knowledge from a related task that has already been learned.

Real experiments are carried out on a UR5 robotic arm, in two dimensions (x–y plain) with a force sensor installed. The behavior of the dMPC framework proposed is analyzed to tune parameters. After adapting the model to a different trajectory, users, and a large object, an application scenario is proposed for co-manipulating two different objects and comparing the obtained results with other controllers.

2. Cooperative Game Theoretic formulation of the dMPC

The robot motion at the end effector is modeled as a Cartesian impedance, and can be described by the equation of mechanic impedance implemented in the Cartesian space:

$$M_i\, a(t) + D_i\, v(t) + K_i\, \Delta x(t) = u_h(t) + u_r(t) \tag{1}$$

where M_i, D_i and $K_i \in R^{6\times 6}$ are the desired inertia, damping, and stiffness matrices, respectively; $a(t)$, $v(t)$ and $\Delta x(t) \in R^6$ are the Cartesian accelerations, velocities and delta positions at the end-effector, with $\Delta x(t) = x(t) - x_0(t)$ with $x_0(t)$ the equilibrium position of the virtual spring, and $u_h(t) \in R^6$ and $u_r(t) \in R^6$ represent the measured human and virtual robot effort applied to the system.

We now describe the structure and the formulation of the dMPC, as expressed in [2], using the cooperative scenario.

$$\begin{aligned} z_{gt}(k+1) &= A_{gt} z_{gt}(k) + B_{h,gt} u_h + B_{r,gt} u_r \\ y_{gt}(k) &= C_{gt}\, z_{gt}(k) \end{aligned} \tag{2}$$

with $z_{gt} = \begin{bmatrix} z \\ z \end{bmatrix}$, with $z = [\Delta x^T \;\; v^T]^T$ containing the state of the system, the matrix $A_{gt} = \begin{bmatrix} A_d & 0^{12\times 12} \\ 0^{12\times 12} & A_d \end{bmatrix}$ containing the state matrix A, the matrices $B_{h,gt} = \begin{bmatrix} B_h \\ B_h \end{bmatrix}$, $B_{r,gt} = \begin{bmatrix} B_r \\ B_r \end{bmatrix}$ containing the input matrices B_h and B_r and the output matrix, and $C_{gt} \in R^{m\times 24}$ is defined according to the desired output.

Defining with Np the time instants considered in the prediction horizon and with Nc the time instants considered for the control horizon, the future steps can be computed as:

$$\begin{aligned} y(k+1) =& C_{gt} A_{gt} x(k) + \\ & C_{gt} B_h u_h(k) + C_{gt} B_r u_r(k) \\ & \vdots \\ y(k+N_p) =& C_{gt} A_{gt}^{N_p} x(k) + \cdots + \\ & C_{gt} A_{gt}^{N_p - N_c} B_h u_h(k + N_c - 1) + \\ & C_{gt} A_{gt}^{N_p - N_c} B_r u_r(k + N_c - 1) \end{aligned}$$

Now we formulate the equations of MPC in a cooperative control case. In the cooperative game, players communicate with each other and share a common objective expressed by a parameter $\alpha \in (0,1)$. Define $Q_{gt} = \alpha\, \tilde{Q}_h + (1-\alpha)\, \tilde{Q}_r$, $R_{gt,h} = \tilde{R}_h$ and $R_{gt,r} = \tilde{R}_r$, with $\tilde{Q}_h = \text{blkdiag}(Q_h ... Q_h)$ and $\tilde{Q}_r = \text{blkdiag}(Q_r ... Q_r)$. Q_h and Q_r, define the weight that the human assigns to their own and the robot's reference tracking.

The cost functions that the two players aim at minimizing in the CGT are:

$$\begin{aligned} J_i(k) &= E_{gt}(k)^T\, \tilde{Q}_i\, E_{gt}(k) + U_i(k)^T\, \tilde{R}_i\, U_i(k) \\ &= E_{gt}(k)^T\, Q_{gt}\, E_{gt}(k) + U_i(k)^T\, R_{gt,i}\, U_i(k) \end{aligned} \tag{3}$$

with $i = \{h, r\}$. The two vectors $U_i \in R^6$ are the input vectors along the horizon, $E_{gt}(k) = y(k + N - 1) - y_{ref,i}(k+N)$, $\phi \in R^{mNp\times 6N_c}$ the matrix representing the forced response, $F \in R^{mN_p \times 24}$ is the free response matrix.

So the dMPC problem for the CGT pHRI can then be summarized by minimizing the two cost functions (3) in u_r and u_h. In particular, the solution to this can be computed as:

$$U^* = \begin{bmatrix} U_h^* \\ U_r^* \end{bmatrix} = \begin{bmatrix} I & K_h \\ K_r & I \end{bmatrix}^{-1} \begin{bmatrix} L_h & 0 \\ 0 & L_r \end{bmatrix} \begin{bmatrix} Z_h \\ Z_r \end{bmatrix}$$

with $K_i = ((\Phi_i^T Q_{gt} \Phi_i + R_{gt,i})^{-1} \Phi_i^T Q_{gt}) \Phi_i$ and

$$Z_i = \begin{bmatrix} z_{gt}(k) \\ y_{ref,i}(k+1) \\ \vdots \\ y_{ref,i}(k+N) \end{bmatrix} \text{ with } i = \{h, r\}$$

The two matrices L_i depends on Φ_i^T, that is the typical forced response matrix, Q_{gt} and $R_{gt,i}$.

3. Learning human intention for trajectory prediction

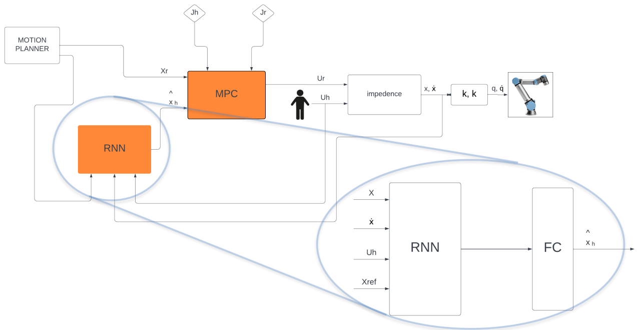


Figure 1: Representation of RNN+FC inside the control scheme

In order to develop a suitable controller, it is very important to know the desired trajectory of the human. The solution adopted is to use the RNN cascaded with a fully connected(FC) part that learns human behavior and provides the robot with the necessary information to assist the human. To tune the parameters of the RNN, the machine learning approach requires training the model on real data. So before using the model we have to acquire the data needed. The procedure is done not using any predictive model and assuming that the human-predicted trajectory is equal to the robot one. The dataset we create is called D_0(the "0" indicate that it is the iteration done with no model loaded) and it is done assuming that this equation applies $\hat{x}_{ref,h} = x_{ref,r}$. where $\hat{x}_{ref,h}$ stands for the predicted trajectory of the human and $x_{ref,r}$ the actual trajectory done by the robot.
With this data collected, we can train the model we called M_0 that depends on the first data set only $M_0 = M_0(D_0)$. This new model is used for collecting new data and the robot can predict $\hat{x}_{ref,h}$ and can assist better the human during its task.
The process that is being created follows an iterative procedure where the first step is the one already explained and so the next step is to collect a new dataset D_1, and train a second model on this data. $M_1 = M_1(M_0, D_0)$ This procedure is done to find the best model possible and this process can be done K times: $M_K = M_K(M_{k-1}, D_{k-1})$
The iteration process can be stopped by a stop criterion *i.e.* the law that indicates at which iteration the model doesn't improve itself. An example of this could be the average of the Root Mean Square Error (RMS), computed as

$$e_{RMS} = \frac{1}{L}\sum_{T=1}^{L}\sqrt{\frac{1}{N}\sum_{K=T}^{T+N}(\|\hat{x}_{ref,h} - x_k\|^2)} \qquad (4)$$

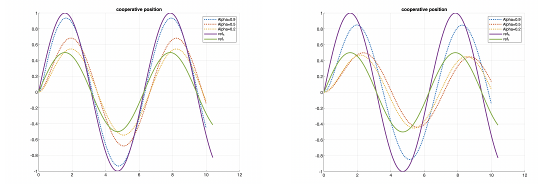
where $\hat{x}_{ref,h}$ is the predicted human intention, x_k the measured poses, L is the length of the trajectory, and N is the prediction horizon. The stopping criterion can be expressed with this assumption $\|e^{k+1}_{RMS} - e^{k}_{RMS}\| < toll$
To make the model as general as possible and reduce the processing time, we use the theory of *transfer learning*, through a model, based on RNN+FC for the new user/object. This procedure allows a considerable reduction in process time. This is because we only perform a single iteration, due to the fact that we start from a model that has already been trained and fewer data are required. Specifically, in our case, we freeze the part of RNN, and the part we consider the FC is the part we train again. So using the transfer learning approach, we define the M_{TL} model equation as follows: $M_{TL} = M_{TL}(M_k, D_k)$
The full control schema including the various modules is visible in figure 1.

4. Experimental Results

The presented method is evaluated with simulations and real experiments. The robotic platform is a UR5 robot from Universal Robots, with 6 dof, equipped with a Robotiq FT300 sensor mounted at the tip for measuring the human interaction force. A gripper allows being grasped by the human directly and a suction cap is used explicitly for carrying large/heavy objects. The robot's nominal trajectory, defined by a motion planner, is defined offline by an external computer at 125 Hz. The RNN+FC model experiments are performed on the x–y plane, involving only two dimensions. The impedance control parameters in (1) are set as follows: $M_i = diag([10, 10])$, $Ci = diag([100, 100])$ and $K_i = diag([0, 0])$. The two cost functions parameters in (3) of the two players are set as: $Q_{h,h} = Q_{r,r} = diag([1, 1, 0.0001, 0.0001])$, $Q_{h,r} = Q_{r,h} = 0^{2\times2}$, and $R_h = diag(0.0005)$ In particular, the human cost function parameters $Q_{h,h}$,$Q_{h,r}$ andR_h are recovered via Inverse Optimal Control (IOC) and an average value is used. The robot parameters $Q_{r,r}$ and $Q_{r,h}$ are set equal to the human's to mimic a person except for R_r, which together with α and Prediction Horizon $\mathcal{H}$, are designed through experiments.

4.1. dMPC performance analysis

We first discuss the choice of the dMPC parameters, analyzing its performances varying α, R_r, and $\mathcal{H}$. The test is conducted assuming a sinusoidal signal as a reference ($x_{ref,h} = sin(t)$ and $x_{ref,r} = 0.5\,sin(t)$) and we evaluated the model on $\alpha = \{0.2,\ 0.5,\ 0.9\}$, $R_r = \{0.01,\ 0.0005,\ 0.0001\}$ and with $\mathcal{H} = \{0.04,\ 0.16,\ 0.4\}$ seconds. Low values of α correspond to the case where the shared cost approximates the robot's cost and high values correspond to the case where the shared cost approximates the human cost. In the case of R_r, the lower it is, the higher the robot's assistance is. With $\alpha = 0.9$ the curve follows the human reference more, so we have a more assistive controller that follows the human intention. With $\alpha = 0.2$, according to GT, it should be the human who puts much effort into helping the robot track its reference. This case is not applicable to the presented method because human does not know the reference of the robot and it is unnatural for a human to assist the robot. The difference between the $R_r = 0.0001$ and $R_r = 0.0005$ is that the lower it is, the more responsive the robot behavior becomes. Varying the prediction horizon, we see better results when we can predict as far ahead as possible in time. The analysis is done on all the parameters and to give a representation of this, figure 2 shows an example of two specific cases.



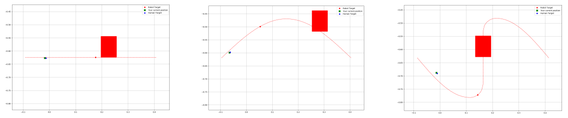
(a) $R_r = 0.0001$.

(b) $R_r = 0.0005$.

Figure 2: The α variation seen in just two cases with $\mathcal{H} = 0.4$

4.2. Human Intention prediction evaluation

All the datasets contain data on the robot's actual poses, velocities, reference robot's trajectory, and interactive force. The human force is measured at the robot tip via the FT sensor. The robot's nominal trajectory defined by the motion planner is defined offline and commanded in real-time. The data collected are sampled at 0.008 seconds, as this is the sampling time of the robot's controller. A single trial consists of following a given trajectory that appears on a monitor, and conducting the robot from an initial point to an endpoint, avoiding a virtual obstacle. The trajectory are represented in figure 3a, 3b, and 3c, respectively. A complete dataset, for the iterative phase, is composed of 60 trials, 20 for each of the three trajectories, and is performed 4 times by the author.



(a) Linear.

(b) Curve.

(c) Sine.

Figure 3: Trajectories used in the training phase

The LSTM model used is composed of 3 layers with 250 hidden nodes, and the FC is composed of two connected layers. The model is trained for 25 epochs with a batch size of 64 and a learning rate initially set at 0.001. The neural network used works with 125 times instant precedent step as input. The model predicts 50 instant steps ahead of the current state. To evaluate the prediction model, we set the parameters with $\alpha = 0,8$ to allow sufficient assistance and $Rr = diag(0.0005)$, set equal to the human's to mimic collaboration with another person. For the training phase, we decide to dedicate 20% to the test part and the remaining 80% to the training part. Define the first model train as M_0, and with D_0 the dataset collected. With the subscript $_0$ we denote the model trained and the dataset collected with no model loaded. The following takes the name of the iteration we proceeded. The new dataset collects D_{N+1} are created with the model M_N loaded. The collection procedure is done in the same way as described in the first case. Unlike the first data collection, the assumption $x_{ref} = x_h$ does not hold anymore, but the RNN+FC model now predicts the x_h reference. From now the train will be based on the model created in the previous step.
Despite this, the iteration is done only by the same user and on three trajectories. To adapt the model quickly, TL performances are evaluated. The procedure of TL is done on different users and adapts to a co-manipulated object and new trajectory. In all cases, the first data are collected based on the last model trained by the author, which here we called M_{prev} The difference is that only 15 trials are performed in total, 5 for each trajectory.

5. Results

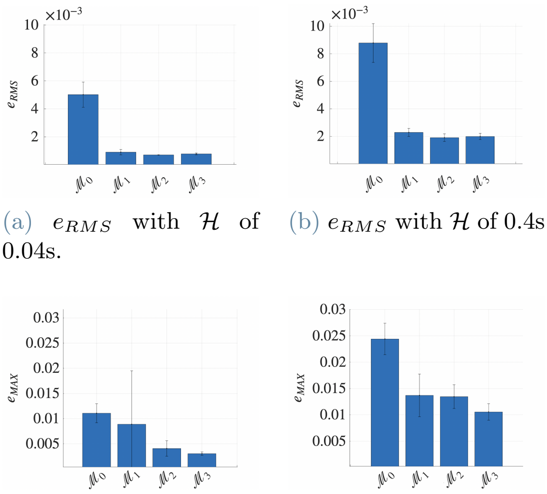
The performances are measured on e_{RMS}, already define in (4) and e_{MAX} defined as:

$$e_{MAX,i} = \max_{i \in L}\{\|\hat{x}_{ref,h} - x_k\|\} \qquad (5)$$

where $\hat{x}_{ref,h}$ is the predicted human intention and x_k the measured poses. L is the length of the trajectory. We evaluated this value by comparing it with different time horizons to see its dependency and in particular with $\mathcal{H} = \{0.04, 0.08, 0.16, 0.4\}$.

5.1. Model Evaluation

To evaluate the model we compare the four different iterations, and figure 4 give an example of this showing the two index with just $\mathcal{H} = \{0.04, 0.4\}$.



(a) e_{RMS} with $\mathcal{H}$ of 0.04s.

(b) e_{RMS} with $\mathcal{H}$ of 0.4s

(c) e_{MAX} with $\mathcal{H}$ of 0.04s.

(d) e_{MAX} with $\mathcal{H}$ of 0.4s

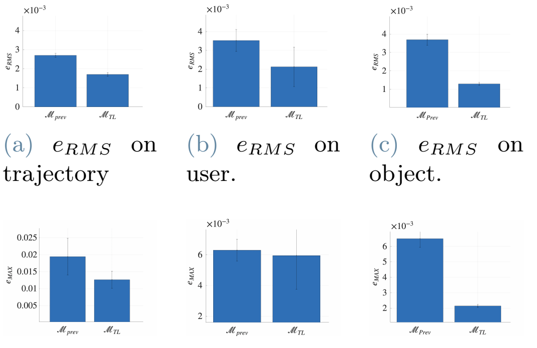
Figure 4: Comparison of the e_{RMS} and e_{MAX} for the four iterations with different $\mathcal{H}$.

In e_{RMS} the improvement is clear between the first iteration M_0 and the subsequent one. In particular, between the first and the second, the difference is huge while from the third, and fourth, it's stabilized. The e_{MAX} index, instead, gives us more information. We can notice that it reduces only after 3/4 iteration and not only just after one. So, this value said that iterating the process multiple times can improve the model. Regarding the different time horizons, we can see that it is more complex to predict long prediction horizons as the value of the error increases when we augment the prediction horizon. This is mainly because it is very complex to predict human deviations from the nominal trajectory in advance.

5.2. TL Evaluation

The same indices are used to analyze the improvements done using transfer learning. The TL is done in the new trajectory, 5 different users, and with a co-manipulated object.

Fig 5 represents the comparison between the model based on the iteration process and the one result after TL. We can see that we have an improvement in every three TL.



(a) e_{RMS} on trajectory

(b) e_{RMS} on user.

(c) e_{RMS} on object.

(d) e_{MAX} on trajectory.

(e) e_{MAX} on user.

(f) e_{MAX} on object.

Figure 5: Comparison based on e_{RMS} and e_{MAX} based on TL approach with $\mathcal{H}$ as 0.4sec

It is interesting because we can see that by using transfer learning we only need one training iteration and fewer data to reach performances comparable to the results obtained using the full iterative training procedure.

Another advantage is shown in the next table 1, where we can see the time for adapting the model to new users/objects dramatically decrease compare to the iterative training phase.

	Iterations	TL
data collection	60 ± 10 min	5 ± 2 min
training	45 ± 5 min	4 ± 1 min

Table 1: Time required at various steps

5.3. Application scenario and other controller comparisons

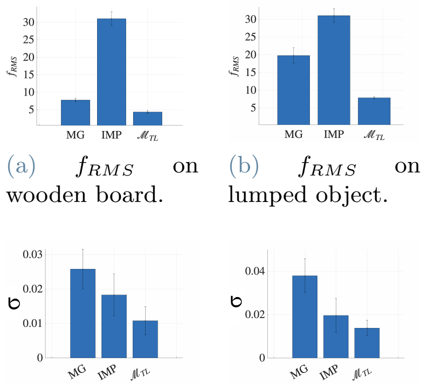
We can see the improvement also apply to a real case application The task was to conduct two different objects, a wooden board, and a lumped heavy object, from a starting point to a target point. We compare it between a Manual Guidance (MG) control and an Impedance control (IMP). First of all, we compare the force applied to the robot as the Root Mean Square, computed as

$$f_{RMS} = \sqrt{(f_x^2 + f_y^2)} \tag{6}$$

. with f_x and f_y, the force along x and y. Then, we measure the precision to reach the target point with the object, expressed with σ, that gives the deviation from the point, measured as:

$$\sigma_i = \|x_{current} - x_{final}\| \quad i \in L \tag{7}$$

with $x_{current}$ the current value, x_{final} the final point, and L the length of the trajectory from starting point to the endpoint.



(a) f_{RMS} on wooden board.

(b) f_{RMS} on lumped object.

(c) σ on wooden board.

(d) σ on lumped object.

Figure 6: Comparison between MG, IMP and M_{TL}, through two objects by f_{RMS} and σ

In figure 6a and 6b is shown that the M_{TL} needs less force than the other two and so the robot is assistive. In figure 6c and 6d is evaluate the σ index expresses in (7). It is calculated for 3 seconds starting from the point where the σ is around ± ε, arbitrarily decided, and equal to 0.025cm. The figures show how the M_{TL} stays and reaches the end-point better than the other.

6. Conclusions

This work presents an assistive controller for pHRI, described by Differential Cooperative Game Theory, and a learning method for predicting the desired human trajectory over a finite time horizon. The model is simulated to tune the different parameters and is validated with real-world experiments done by the author on the UR5 robotic arm. It needs an iterative training procedure, done 4 times, to adapt the model and to reduce the prediction error. To improve the time-consuming problem and to adapt to new situations and users, Transfer Learning is applied. The results obtained are satisfactory as starting from a model trained, after just three iterations, on a single subject, and on a specific task, it is possible to quickly adapt the model to new users and tasks with comparable performances. This method allows to dramatically reduce the time necessary for data collection and training the model compared to the iterative procedure. Finally, the superiority of the assistive controller enhanced by the RNN+FC model, compared to standard controllers typically used in pHRI is shown by measuring the average interaction force, and the precision to reach a target point. Future works will focus on implementing the model with cameras that detect the position of the human and impart a fictitious force to the robot, allowing flexible material co-manipulation. The possibility of varying online assistance will be addressed by feeding the RNN+FC with this additional time-varying parameter.

References

[1] P. Franceschi, N. Pedrocchi, and M. Beschi. Adaptive impedance controller for human-robot arbitration based on cdgt. *ICRA*, 2022.

[2] S. Ko and R. Langari. Shared control between human driver and machine based on gt model predictive control framework. *ICAI Mechatronics, Boston, USA*, July 6-9, 2020.

[3] H.S. Moon and J. Seo. Prediction of human trajectory following a haptic robotic guide using rnn. *WHC, Tokyo*, 9-12 July 2019.

[4] X. Na and D. J. Cole. Game-theoretic modeling of the steering interaction between a human driver and a vehicle collision avoidance controller. *Transactions on Human-Machine System Vol.45 No.1 Feb*, 2015.

[5] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi. An atlas of physical human-robot interaction. *Science Direct, Mechanism and Machine Theory 43*, 2008.