Semester Project:
Human robot impedance matching towards unexpected external forces

Author:
Michael Hodara

Teacher:
Prof. Aude Billard
Supervisor:
Mahdi Khoramshahi

Spring Semester 2019

Contents

1 Introduction

1.1 Motivation

Human robot interaction has always been a topic of interest. More than replacing a human's work, robots hold the possibility to empower humans to reach higher possibilities, by cooperating with them. An example would be using robots for inspection in confined places or moving objects around in warehouses. A robot, on one side, can use sensors, to apprehend the situations it faces. To enable proper human robot interaction, the human has to take benefit from the robot's help, and the robot has to use the intelligence, experience, and reaction strategy from the human, whom will most probably act accordingly to the situation they are facing together.

As a lot of applications of human robot cooperation concern intervention either with heavy loads, hazardous situations, or complicated structures, safety has to be ensured. Regardless of the complexity of the task, as soon as a human operator is involved in it, the robot will have to ensure that the human is at all time safe and that he/she will not be harmed if an unexpected event surfaces. An example that we will assess during this semester project is how a robot would cope with the application of an unexpected force on the system that is composed of the robot, the human and the object they are both holding.

To ensure that robots have the right reaction towards these events and operate properly, it has to base his actions on observations. These information can either come from the robot and its sensors, such as using a force sensor giving information about the impact of the unexpected event on the system or by using a camera that would allow it to understand better the scene it is in thanks to computer vision. From this standpoint, the robot can also learn from the behaviour of the human that is standing in front of it, at the other end of the object. Indeed, the human will usually have a proper reaction when he/she notices an unexpected event, such as a force that would try to disturb him/her from completing the task he/she wanted to achieve.

This semester project focuses first on how the robot can effectively detect how the human is reacting towards external forces. Through the case study of transportation of a heavy object (see Section 3.1), we will simulate how the robot will replicate a model of the force the human is applying on the object, by constantly updating his estimation of the human's parameters. To that end, we will present a model of the action of the human (see Section 3.2) and robot's one, based on these estimations. The discussion that will arise from the results of this simulation (see Section 3.3) will tell us whether the robot can base his behaviour on these estimations and lead to a successful cooperation.

In a second approach, we will use machine learning techniques over data gathered from real scenarios with the robot (see Section 4.1) to find a way for the robot to learn (see Section 4.3), based on the human force, whether he/she is being compliant with the external force or he/she is rejecting it (see Section 4.2). The use of PCA, and SVM allows us to perform a classification that will help the robot vary its stiffness depending on the human reaction. Finally, to find the best implementation of this machine learning model, we will perform a hyperparameter tuning (see Section 4.4).

(a) Human Robot Interaction, holding an object together [1]

(b) Human Robot Interaction in car industry [2]

Figure 1: Example of human robot interactions in the industry. By cooperating with a robot, the human operator can lift heavier objects and be helped on manual and laborious tasks such as maintenance, transportation or guarding.

1.2 State of the art

Designing a robot that could represent the best pair of helping hands for a human is a goal scientists from some of the best European universities (Ecole Polytechnique Fédérale de Lausanne (EPFL), Karlsruhe Institute of Technology (KIT), University College London (UCL) and Sapienza University of Rome) have gathered around. They created the SecondHands [3] project, part of Horizon 2020 European program, that is aiming at producing a robot that could perceive the needs of the human worker, by assisting him in a pro-active manner. Tasks such as guarding, or transporting, will be made easier if robots are following models based on providing assistance to automation maintenance technicians. To that end, a prototype of this collaborative robot has been developed at KIT [4]. This humanoid robot, named ARMAR-6, embodies this will of building the perfect collaborative humanoid assistant robot for industrial environments. It is designed to be able to use tools that human usually use such as drills or hammers, and recognizes when his collaboration partner needs help. To control it, methods such as DS-based impedance control are used [5]. Fig 2 shows a way of collaborating with this robot to lift an object and place it. From there, we want this robot to also help the human react towards unexpected external forces. In this case, if the human asked another human for help lifting the object, the robot would need to be compliant with this new force and not reject it, therefore decrease its stiffness. On the other end, if during this same task, an item fell down on the top of the object, the human would want the robot to increase its stiffness to help rejecting this unexpected force.

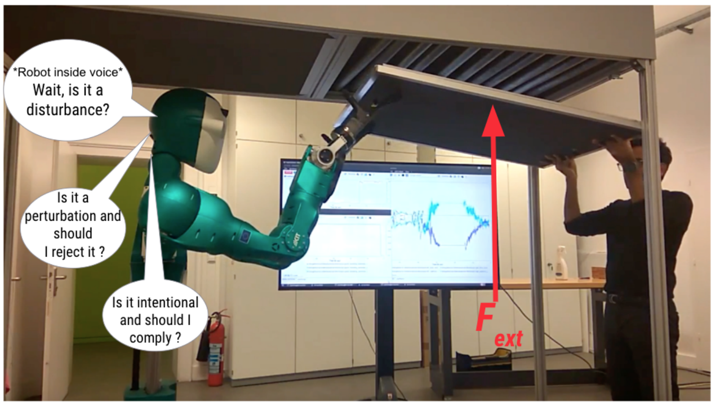


Figure 2: Human cooperating with the ARMAR-6 robot to achieve a task. They work together to lift an object up. The robot has been thought and made as an assistant to the human but it only manages to do so when no unexpected external force is introduced.

This project work uses concepts of Impedance control and Dynamical Systems described through Neville Hogan's work [6], and the notion of intention estimation through stiffness assessed in Klas Kronander and Prof. Aude Billard paper about Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction [7].

1.3 Work Support

Data acquisition for this project was done using the Kuka LWR 4+ along with a ATI force-torque sensor and an end-effector. To pilot this robot abd create logs, ROS (Robotics Operating System) has been used. Aside from that, all the simulations and plots were done through MATLAB v.2019a, run on a Macbook Pro (Early 2015) under macOS v.10.14.4 with a 3,1 GHz Intel Core i7 CPU.

2 Problem statement

2.1 Impedance Control

Here we will focus on a specific case of moving an object in 1 dimension by cooperating with a robot that is holding the object on the other end. For that, we will work with impedance control which consists in driving a mechanism with force-controlled actuators. Therefore, the control law of our system will hold a stiffness term K and a damping term D, allowing us to have a smooth trajectory.

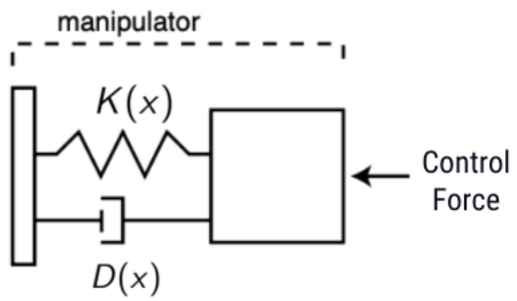


Figure 3: Modelization of an impedance controlled robot, holding an object with his end effector. The robot is modelized as a spring damper system which value will depend on the position.

Similarly to admittance control, **impedance control** puts into relation force and position. From N.Hogan's work [6], we can implement a controller such as the following one:

$$\begin{cases} M\ddot{x} = F_c + F_{ext} \\ F_{ext} = M_d(\ddot{x} - \ddot{x_d}) + d_d(\dot{x} - \dot{x_d}) + k_d(x - x_d) \end{cases} \tag{1}$$

where F_{ext} is an external force that comes and disturbs the system, M_d is the desired inertia of the system, D_d represents the desired damping matrix and K_d the desired stiffness matrix. We will now make the assumption that the desired inertia is equal to the one of the system, and that its acceleration is also negligible :

$$\begin{cases} M = M_d \\ \ddot{x_d} \approx 0 \end{cases} \tag{2}$$

2.2 Perturbations

The system we presented in the previous section is ideal, as $F_{ext} = 0$, which means that no external force would come and perturb our system. Nevertheless, we can assess that the probability that an unexpected event which would change the initial situation is likely (for example in factory, a case where an object falls onto the surface of the object they are manipulating happens often). The simplest case would concern a force exerted on a single point either by an external system (which can as much be another person, or an object). Therefore, we would need our system to **adapt** his behaviour to remain always coherent with the one that the human will have towards this unexpected event.

The approaches we will take in this project will take into account the reaction of the human towards this event and the way he/she **rejects it** or **complies with it** so the robot can adapt its behaviour accordingly. On one hand, if the robot senses through his force sensor that the human is compliant with this external force, the robot will need to decrease his stiffness, to allow the system to adapt to it. On the other end, if the robot senses that the human had not expected this external event, he needs to put everything in his power to reject it, so that the human will not be affected by it, which could lead to injuries in some extreme cases; therefore, the robot will increase his stiffness.

3 Approach

3.1 Case Study

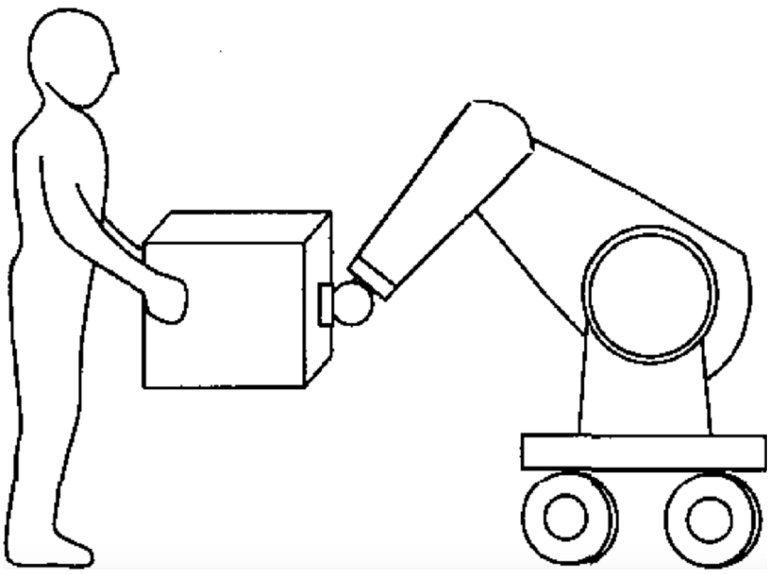


Figure 4: Moving an object together with a robot (from [8]). The control method used considers the human motion characteristics and tries to estimate it, such as goal position, stiffness, and damping. The human can be modeled as a spring-damper model, which damping ratio and stiffness change to minimize a cost function.

To understand how a robot should observe the intention of the human during the task they share, process it to apply a coherent behaviour, we will take a simplistic 1D case study during which the following situation will happen :

- A human and a robot will share a common task: bringing an object together from point A to point B, following a linear trajectory, similar to the one in Fig 4;
- The human has a clear objective : bringing the object $2m$ further and he wants the robot to help achieving it;
- The human uses a certain force to move the object, which the robot has to detect, estimate, and process the force it has to provide to help the human. At the same time, the robot will estimate the goal of the human depending on the variations of force he will apply on the object;

- When the goal is achieved, the human will apply a force to show his intention of bringing back the object to its original place;
- The same way it did it before, the robot will estimate this force, the position of the goal and the stiffness and damping that the human is using.

We will then explain in the next section how the parameter estimation is done and which performances it achieves.

3.2 Online estimation of human impedance

To simulate this case study, we will look at the following system :

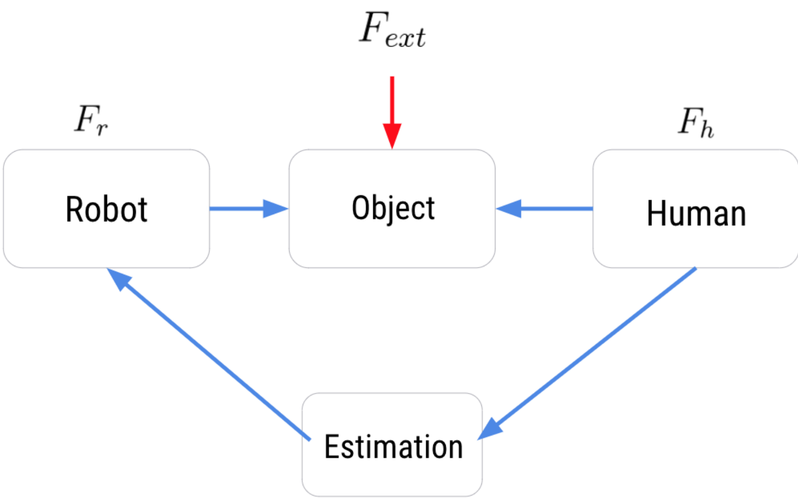


Figure 5: System representation of the cooperative task of transporting an object. The robot estimates the human impedance, his stiffness, and damping ratio to update how it should assist him/her in this task.

with F_r being the force applied by the robot on the object, F_h the force applied by the human on the object, F_{ext} being an external force coming to disturb the system.
From this situation, the control laws are the following :

$$\begin{cases} F_h = -K_vel(\dot{x} - \dot{x_d}) \\ \dot{x_d} = -K_pos(x - x_{goal}) \end{cases} \tag{3}$$

where $\dot{x_d}$ is the desired velocity, derived from the difference of position between the object and the goal, which is at $x_{goal} = 2$, and $K_velandK_pos$ are constants. In the real implementation we will also add some random noise to have a more realistic movement. On the other side, the robot will try to estimate the impedance of the human to adapt the force he should apply. Therefore the model for human forces is the following [6]:

$$\tilde{F} = -k_{est} * (x - x_{est}) - d_{est}\dot{x} + f_{est} \tag{4}$$

where k_{est}, d_{est} and f_{est} are respectively the estimated stiffness, damping and force of the system. These estimations are derivated from minimizing the cost function J with respect to every parameter, in the following manner:

$$J = \frac{1}{2}(\tilde{F} - F_h)^2 \tag{5}$$

$$\begin{cases} -\epsilon \frac{\partial J}{\partial k_{est}} = -\epsilon(\tilde{F} - F_h)x \end{cases} \tag{6}$$

3.3 Discussion

From the simulation described in the previous sections, we were able to obtain an interesting set of results, allowing us to see a clear cooperation between the robot and the human, as the robot tries to take over the human's task by matching it.

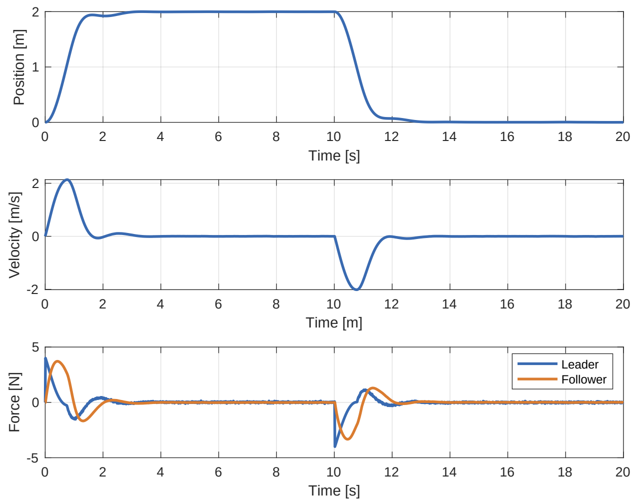


Figure 6: Simulation of an object moved together by a human and a robot in 1D. On the first plot, we see that the system (composed of the robot and the human) manages to bring twice the object at the goal position. It does it with a robust response, characterised by a low overshoot and a rise time of less than 2 seconds, which is highly performant for this task. The graph in the middle shows us that the velocity of the system increases and decreases almost as the square of x, which brings us to a smooth trajectory, not being abruptly interrupted by control issues, or non-syncronization between the human and the robot. Lastly, the third plot shows us how the force applied by the follower (e.g. the robot) tends to follow the one applied by the human, and how it interprets the trajectory the human is trying to achieve. Both forces do not exceed $4N$, which is realistic with the situation we are trying to simulate.

Parameter estimation is the key of our solution. From this estimation, the robot is able to apprehend how the human is moving and to deduce a motion model to help achieving his task. Therefore, of Fig 7, we can see that the robot quickly picks up the fact that the object has to be moved and approximates where it should be moved. Indeed, as 70% of the movement is done within the first second, the model quickly estimates that the damping and stiffness have to increase so the motion can stop at the right place. We can directly see both of them rise at the same time it estimates that the position of the object is rising rapidly.

On the other side, we can observe the estimated damping decrease at the moment the object has began its movement towards its initial position right after the 10^{th} second, which allows it to increase by the same amount ($\simeq 1N.s/m$) when the motion has to be stopped because the object has reached its goal position.

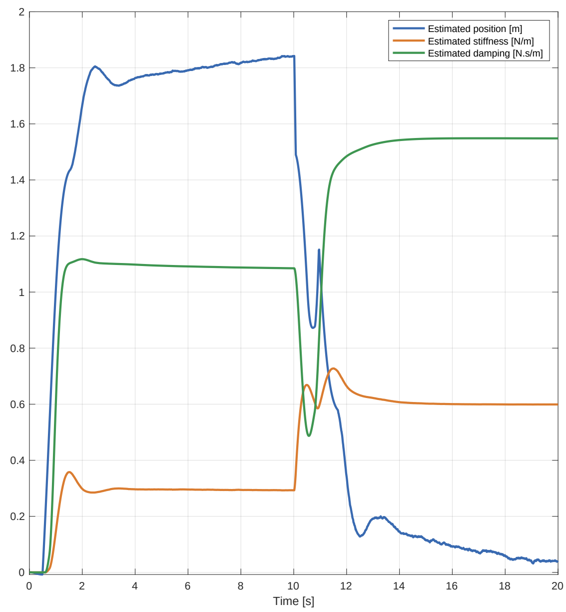


Figure 7: Parameter estimation while doing a 1D transportation task; bringing an object two meters forward, and then, at the 10^{th} second, bring it back to its original position. Estimated position is quickly found and gets close to the actual goal the human sets. Damping and stiffness estimations give robust step responses and make clear that their variation makes the trajectory smooth when coming close to the goal.

4 Impedance matching

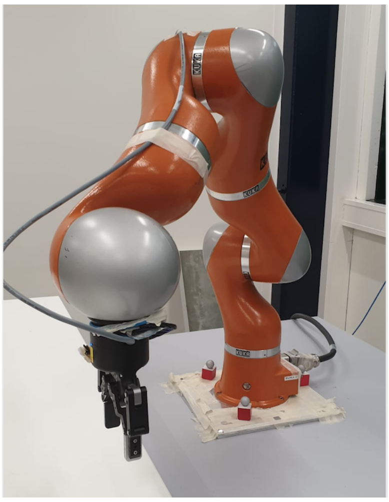
As our first approach successfully managed to estimate a model of the human impedance, and its main parameters such as stiffness and damping in a simulation, a lot of questions remain whether it will be the same in the real world. As the aim of the project is to assess whether the robot can detect and use the human's impedance to adapt his, we want to find a way to measure this change in impedance so that the robot can take decisions, and increase or decrease its stiffness. To that end, we experimented different situations with the Kuka, within the idea of gathering data.

4.1 Acquisition of the data

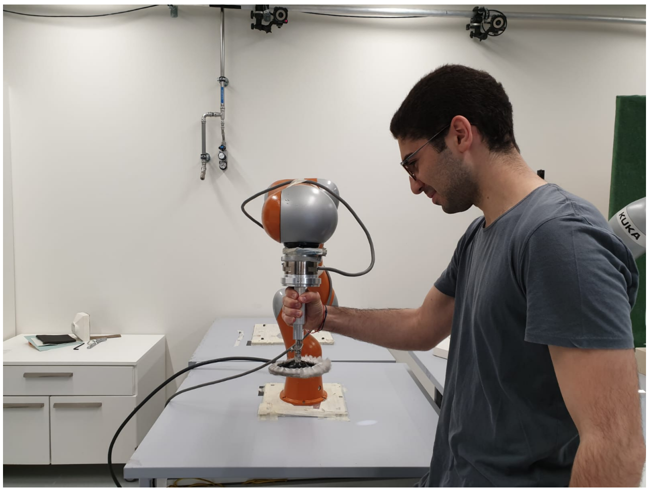
Using the force/torque sensor available on the Kuka, we were able to simulate different cases during which the human would either stay compliant with an external force coming to disturb the movement, or reject it because the human wants to reject this disturbance, as it was not expected. Through these measurements, we aim at finding a real metric so that the robot can learn how to differentiate these two cases.

During this experiment, we gathered data on the following scenarios :

1. Human and Robot are moving up together, an external force comes and the human is compliant with it
2. Human and Robot are moving up together, an external force comes and the human is rejecting it
3. Human and Robot are moving up together, an external force comes to push the system to the side but the human brings back the system to the its original position and goes up again



(a) Kuka LBR iiwa on which we conducted the tests



(b) Human is applying a force on the end effector

Figure 8: Setup of data acquisition. The human achieves a certain trajectory when suddenly an external force comes to get him out of that trajectory by applying a force on the end effector. The human chooses whether he/she rejects it or if is compliant with it. Therefore, in the first case, the force sensor will detect a variation in force, linked to an increase of stiffness. In the second case, this increase won't be noticed as much, but only variations due the fact that the human is coping with external force.

4. Human and Robot are moving up together without any disturbance
5. Human and Robot are moving circularly together when an external force is applied and the human complies with it
6. Human and Robot are moving circularly together when an external force is applied and the human rejects it
7. Human and Robot are moving circularly together and every 5 seconds the human increases his stiffness as an external force comes to disturb the system. Human's compliance is therefore variable in this case.

The last case will constitute the dataset of our visualization analysis (see next Section 4.2) as it presents both cases we are interested in and we can probably find a way to differentiate the times during which the human increases his stiffness, and the times during which he/she does not.

4.2 Visualization

As we visualize on Fig 9, that the force in the z-direction felt by the sensor, we are able to clearly see that every 5 seconds, there is a big change of force, as the human rejects the disturbance that is coming to the system.

As this increase in force is directly related to an increase in stiffness, we will aim at making the robot use this variation in stiffness to know whether the human is complying to or rejecting the disturbance. We also observe that the noise generated by these increases helps us distinguish the stiff case from the soft case (during which we comply to the disturbance). To observe it on a graph, we conducted a Fast Fourier Transform on a 2 seconds- moving window, a fast and efficient process to convert our time-domain approach to a frequency domain visualization. We then plotted the power generated by this noise depending on the frequency of the noise to be able to check the difference between the stiff

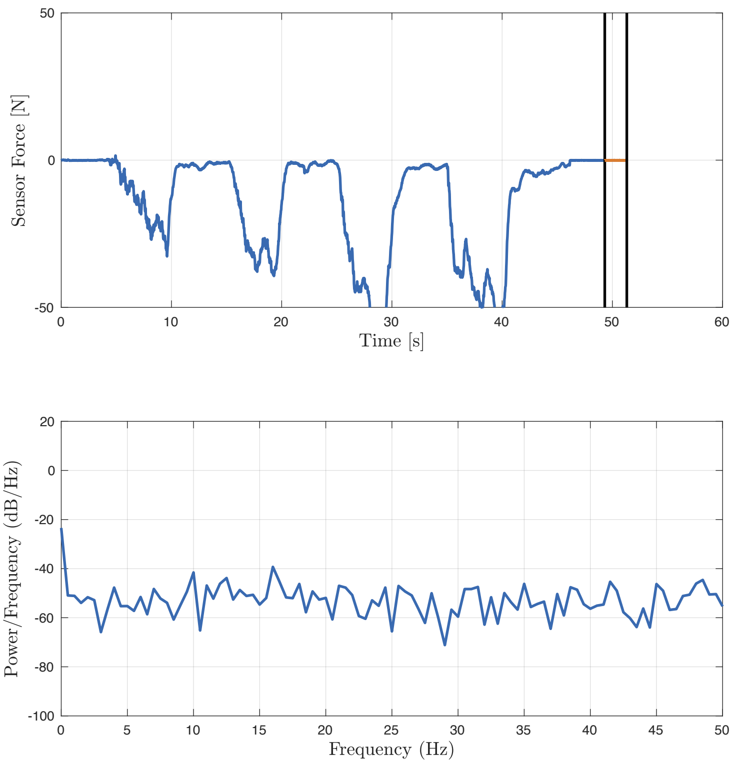


Figure 9: Visualization of the human force in the z-direction and Fast Fourier Transform conducted on a 2 s timeframe. On the first plot, the sensor detects that the force in the z-direction increases quickly. At the same time the $2s$ window passes on the first plot, the MATLAB program processes the Fast Fourier Transform over this sample and the noise power related to it, which is shown on the second plot. This noise power varies whether we pass onto a region with increased force or not.

case and the soft case. As one can compare between Fig 10 and Fig 11, the power of the noise is much higher (all values are above -20 dB/Hz) on Fig 10 when the window is passing over a portion of time where a higher force was applied to reject the disturbance, while it will have a lower amplitude (see Fig 11) when the window will be on a portion of time when we are complying with the external force.

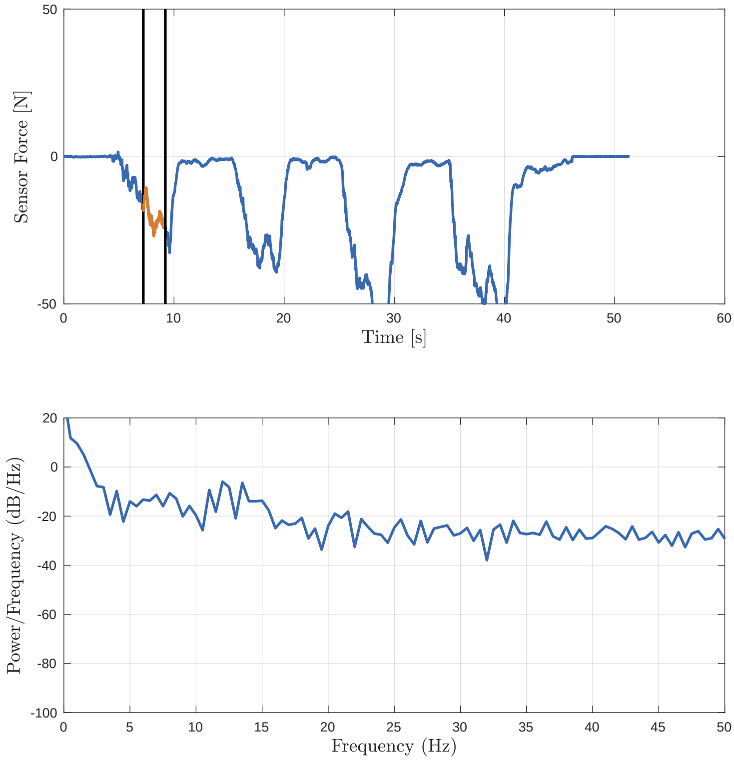


Figure 10: Visualization of the human force in the z-direction and Fast Fourier Transform conducted on a 2 s timeframe. On the first plot, the sensor detects variations on the force applied in the z-direction. The $2s$ time window passes on a portion where force is increased, therefore, during which noise power is amplified (always over $-40dB$).

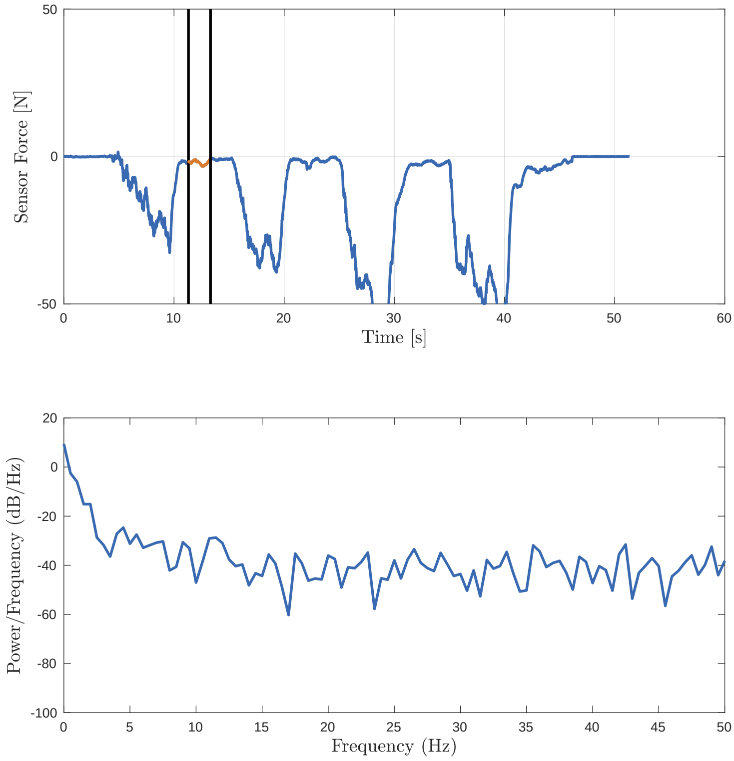


Figure 11: Visualization of the human force in the z-direction and Fast Fourier Transform conducted on a 2 s timeframe. On the first plot, the sensor detects variations on the force applied in the z-direction. The $2s$ time window passes on a portion where force is almost constant, therefore, during which noise power is not amplified (between $0dB$ and $-60dB$ for most frequencies).

4.3 Machine Learning Approach

The goal of this approach is establish a way for the robot to effectively learn from the human whether it has to increase his stiffness or decrease it. In this section we have implemented successively different operations that will lead us to the decision making of the robot on the situation. To collect a comprehensive and significant amount of data, we will gather data from two scenarios mentioned in Section 4.1 :

1. Human and Robot are moving circularly together when an external force is applied and the human is compliant with it.

2. Human and Robot are moving circularly together when an external force is applied and the human rejects it.

Then after storing these data we follow this procedure for each scenario :

1. Store all the FFT values inside the same array (M x N, M being the number of observations and N the features for each observation);
2. We perform PCA on the dataset, to get information on the amount of variance contained in every observations of our dataset;
3. We choose to keep a certain number of principal components that holds at minimum the amount of variance we want to perform our analysis on. We will discuss this amount further in the next section;
4. As we want our model to learn from a part of the data and then to be tested on the rest before validation, we split our dataset between a training set and a testing set. We will discuss the value of the split ratio further in the next section;
5. To be able to classify our datapoints between the two classes (soft, and stiff), we perform Support Vector Machine on the data. Tuning the different hyperparameters such as the type of kernel, its degree and coefficient, and the regularization parameter C will allow us to optimize the classification. We will discuss this tuning in the next part;
6. We then perform validation by assessing if the robot is able to classify well the moments it has to increase his stiffness and when he has to decrease it. Without having information on the ground truth, we put into question this validation that would need to be confirmed with a more mathematical technique such as K-Fold cross validation.

4.4 Hyperparameter tuning

In this approach, we use different machine learning techniques. We will discuss successively the role of each hyperparameter as we go through each step described in the previous section. We will use the classification error on the testing set as a metric to determine the importance of each hyperparameter and give an observation on its impact on the error. The default configuration we will run as we tune every hyperparameter singularly is the following :

- We set the number of principal components to 15, holding almost 80% of the cumulative error;
- We use a 80% splitting ratio between the training set and the testing set;
- We train the SVM Classifier with a Gaussian Radial Basis Function (hereafter, "rbf") Kernel, with a kernel scale factor of 1000;
- We validate our model.

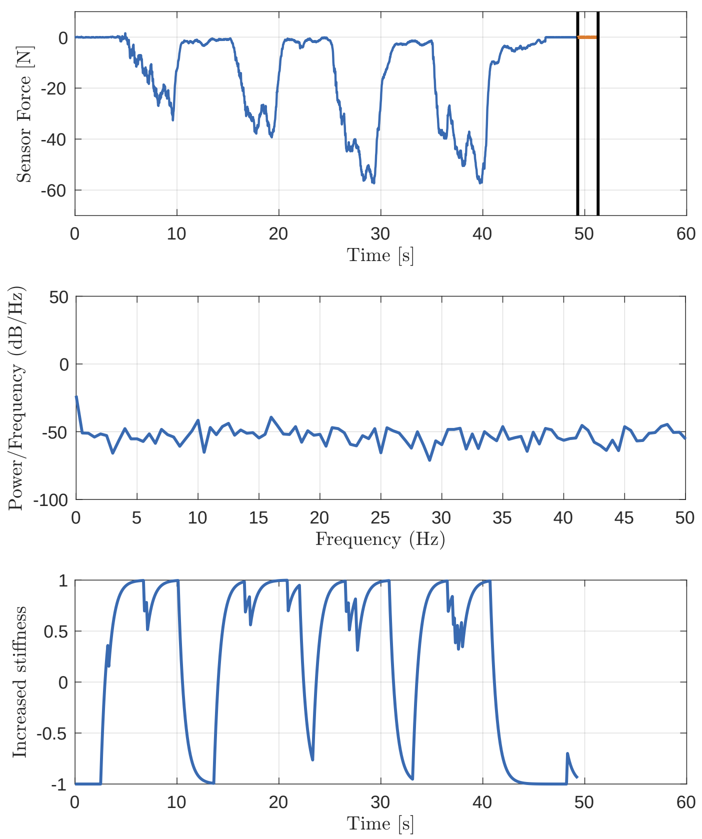


Figure 12: Classification for default set of parameters with 15 principal components, holding 80% of the cumulative error, a 80% splitting ratio between the training set and the testing set and the SVM Classifier is trained with a Gaussian RBF Kernel, with a kernel scale of 1000. The first plot shows the force applied by the human in the z-direction over time. A time window of $2s$ is passing and a FFT is conducted over this timeframe, as the second plot shows the power of the noise over this same timeframe. Finally, the third plot shows the result of classification over time, showing whereas yes (y=1), or no (y=-1) stiffness had to be increased for the human.

Here we obtain a well classified situation with a very low error ($err \approx 4\%$). We can see on Fig 12 that we successfully classify whether the human had a stiff behaviour or a soft one for a scenario during which we alternate between the two every five seconds. Nevertheless, we notice that the performance decreases after the 20$^{\text{th}}$ second, which can be analyzed by the fact that the force applied at these times varies much more than for the 20 first seconds.

4.4.1 Number of principal components

Dimensionality Reduction is a well known problem in Machine Learning as we always try to optimize the complexity of our models by only working with the most significant dimensions. In our case, we want to find a tradeoff between the number of principal components we keep, and the amount of variance related to it while getting rid of the ones which do not hold relevant information. By convention, we try to always keep at least 80% of variance within the number of principal components kept. In our case, we can see on Fig 13 that it accounts for at least 15 principal components.

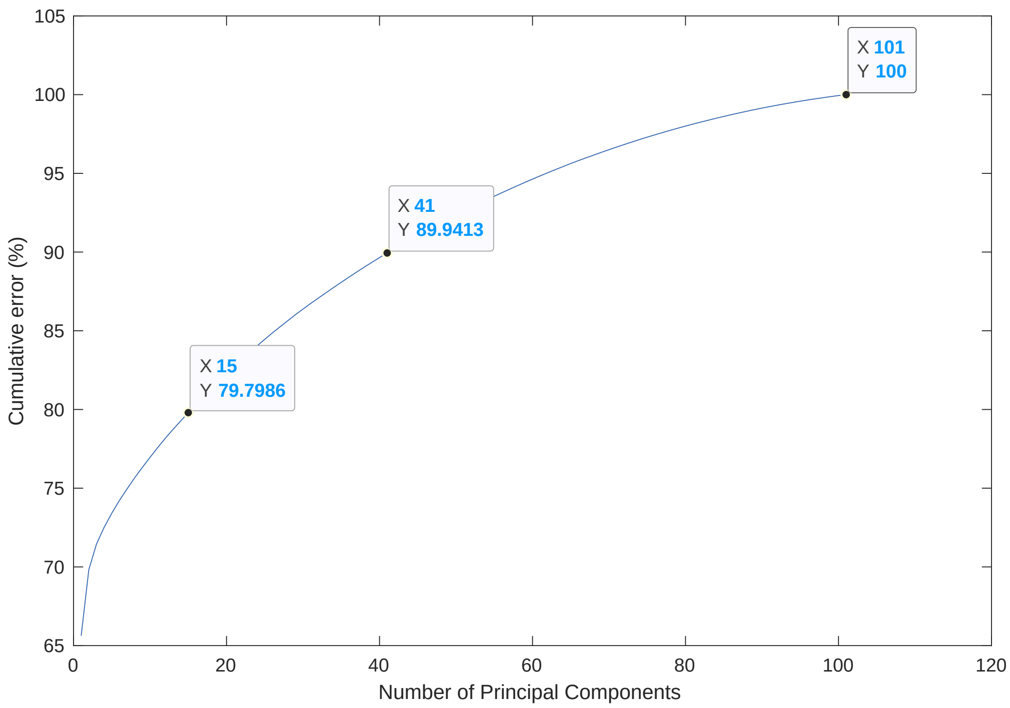


Figure 13: Cumulated error over the number of dimensions of our dataset.

By taking other values we find that for only 65% of error, corresponding to 1 principal component (as we can see in Fig 14), the performance degrades a lot as the model is not able to differentiate correctly whether the human increased his stiffness or not.
Therefore, by only taking a too small amount of principal component, the percentage of classification rises a lot as there are not enough information for the model to learn from, giving an error superior to 60% for the training set and for the testing set.

On the other hand, if we increase the number of principal components (to 41, as we can see in Fig 15) and get a cumulated error above 90% we get an almost perfect classification, with an classification error inferior to 5%, but of course it comes at a price of higher time of computation.
To get a good trade-off, we would need to find a compromise around the default value, which gives us already acceptable results of classification.

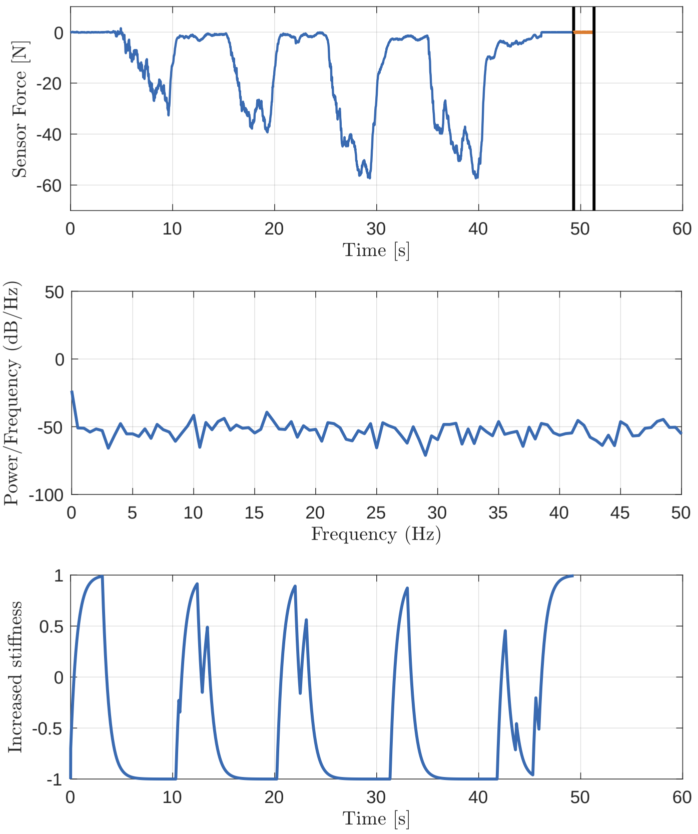


Figure 14: Classification for 1 principal component, holding less than 70% of the cumulative error, a 80% splitting ratio between the training set and the testing set and the SVM Classifier is trained with a Gaussian RBF Kernel, with a kernel scale of 1000. The first plot shows the force applied by the human in the z-direction over time. A time window of $2s$ is passing and a FFT is conducted over this timeframe, as the second plot shows the power of the noise over this same timeframe. Finally, the third plot shows the result of classification over time, showing whereas yes (y=1), or no (y=-1) stiffness had to be increased for the human. Results are bad, as the classification error is above 60% and the third plot presents 6 peaks in stiffness, while there should be only 4.

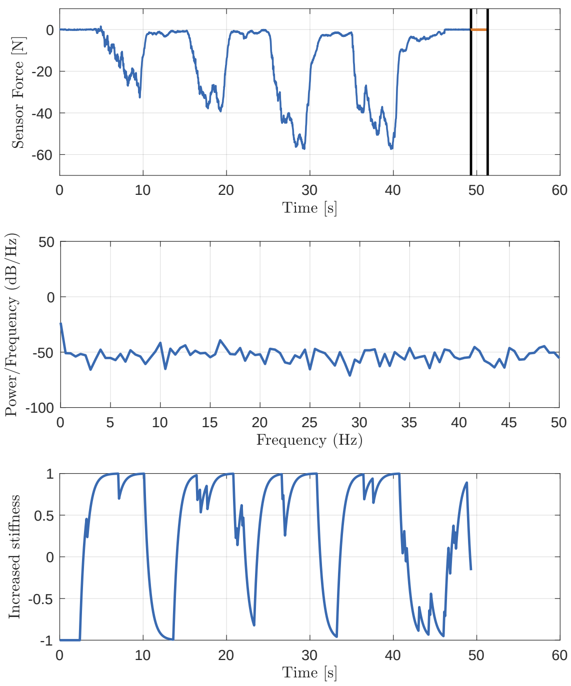


Figure 15: Classification for 41 principal component, holding more than 90% of the cumulative error, a 80% splitting ratio between the training set and the testing set and the SVM Classifier is trained with a Gaussian RBF Kernel, with a kernel scale of 1000. The first plot shows the force applied by the human in the z-direction over time. A time window of $2s$ is passing and a FFT is conducted over this timeframe, as the second plot shows the power of the noise over this same timeframe. Finally, the third plot shows the result of classification over time, showing whereas yes (y=1), or no (y=-1) stiffness had to be increased for the human. Results are acceptable as the classification error is below 5% and the classification is correct on each peak of stiffness from the human.

4.4.2 Train/Test split ratio

By optimizing this parameter we want our training set to hold enough data for our algorithm to provide good generalization, and then good results on the testing set. By convention, we will go using a $2/3^{rd}$ - $1/3^{rd}$ rule, meaning twice more datapoints for training than for testing. By reducing this ratio, we will be able to see if classification is as good as for an conventional ratio. For example, by having a splitting ratio of 60%, leads us to a mean error of 4.15% (and a standard deviation of 0.92, which is not really different from the performance we achieve with a 80% ratio over 10 runs). Whereas by having a smaller testing set, and a splitting ratio of 90%, we obtain similar performances, with a similar mean percentage of classification of 4.74% but with a higher standard deviation of 1.78% (over 10 runs). It shows us that the splitting ratio does not impact that much the performances of the system.

4.4.3 Kernel functions

When employing SVM Classification on a model, having different kernel functions will allows us to observe different hyperplane decision boundary between the two classes. Linear and polynomial kernels are less time consuming but hold usually lower performances than for Gaussian kernel functions which are more accurate. For each kernel function, we will have to tune different parameters, for example for a gaussian kernel, we will tune the variance of the kernel, whilst for a polynomial, we will tune the degree of the polynomial.

Kernel Scaling factor is also an important hyperparameter but it revolves around the range of our data, which is coherent with the fact that performance decrease rapidly when we derive it from the default value of 5000.

Using a linear kernel function brings worse results than with RBF, as the mean percentage of classification error rises to 35.13% with a high standard deviation of 26.36%.

By moving on to a polynomial kernel function, we find a mean percentage of classification above 47% (over 10 runs) for orders above 1 (tested for 2,3 and 4 polynomial orders), as for the training set and for the testing set, leading us to a very bad classification.

Finally, using a sigmoid function as kernel function gives us good performances as well. By tuning its parameters, we obtain a classification error below 6%, which is still worse than using a gaussian rbf. In this case, we need to tune γ and C to find best performances. The best performances are found around the following values : $\gamma = 2e^{-2}$ and $C = -1.2$.

This leads us to concluding that the kernel function has a key role to play in our classification; we will therefore hold on to a gaussian rbf kernel function which leads us to the best classification on our dataset.

4.5 Discussion

From hyperparameter tuning, we were able to see that good performances rests upon optimizing well hyperparameters so our model can be trained on a good amount of data and then be tested and provide acceptable results. Specifically to our case of impedance matching, we could see that by using a dataset composed of either stiff or soft points, we were able to estimate correctly whether the human complied with the disturbance or rejected it. This has to be balanced by the fact that the dataset was not comprehensive enough. Indeed, by taking more datapoints and varying even more the stiffness patterns, it could be possible to implement a way of quantifying stiffness and not only knowing if stiffness was increased or not, which is limited.

The different sets of parameters tested are brought together into Table 1 below.

Parameter set	# of principal components	Cumulative error	Splitting Ratio	SVM Kernel Function	Kernel Scaling Factor	**classification error on testing set**
Default	15	79.8 %	80 %	RBF	5000	4 ±0.4%
#1	**1**	65 %	80 %	RBF	5000	>60%
#2	**41**	90 %	80 %	RBF	5000	5 ±0.45%
#3	**101**	100 %	80 %	RBF	5000	4 ±2.57%
#4	15	79.8 %	**60 %**	RBF	5000	4.15 ±0.92%
#5	15	79.8 %	**90 %**	RBF	5000	4.74 ±1.78%
#6	15	79.8 %	80 %	RBF	**1000**	3.95 ±1.12%
#7	15	79.8 %	80 %	RBF	**100**	38.73 ±21.4%
#8	15	79.8 %	80 %	**Linear**	1.5	11.43 ±2.37%
#9	15	79.8 %	80 %	**Polynomial order 2**	150	6.34 ±2.69%
#10	15	79.8 %	80 %	**Polynomial order 3**	0.5	7.32 ±4.53%
#11	15	79.8 %	80 %	**Polynomial order 4**	7500	4.65 ±1.31%
#12	15	79.8%	80 %	**Sigmoid** ($\gamma = 0.02, C = -1.2$)	5	5.71 ±0.94%
#13	15	79.8 %	80 %	**Sigmoid** ($\gamma = 0.1, C = -1.2$)	5	19.53 ±5.99%
#14	15	79.8 %	80 %	**Sigmoid** ($\gamma = 0.02, C = -5$)	5	49.71 ±4.67%
#15	15	79.8%	80 %	**Sigmoid** ($\gamma = 1, C = -5$)	5	25.4 ±2.43%

Table 1: Table gathering the results for every hyperparameter tuning. Best performances are achieved with values coherent with Machine Learning standards (80% of cumulative error, RBF Kernel Function, sufficient splitting ratio). The best parameter set tested is the default parameter test, see Fig 12.

5 Conclusion

The richest source of information for a robot involved in a cooperative task with a human resides within the human's features. Indeed, by correctly estimating his characteristics, and his behaviour, towards unexpected event that come out of the main scope of the task, the robot is able to learn from the human, and adapt his reactions to ensure the human's safety. However, as the term says, these events are really hard to predict as they could come at any time, from any angle and at any amplitude which makes learning difficult for the robot as it has to then generalize his behaviours.

The first step of this project was aiming at using an impedance control method from [6] and previous work from [9] to find a way for the robot to estimate a model for the human it is sharing the task with. By developing simulations of simple cooperative tasks such as the one detailed in Section 3.1, we were able to see that this estimation was possible. Still, as this case is simplistic, it would need to be generalized to 3D to be closer to reality.

From a successful simulation, a great challenge was to be able to use real scenarios to gather data from the robot that could be useful for learning. As stiffness was a great metric to know whether the human was compliant or not with the unexpected force, we focused on finding a way for the robot to know when the human varies it. To get the robot to learn, we had to elaborate a model, and run it on a comprehensive dataset so that the robot could effectively classify whether yes or no, the human complied with the unexpected force, and do the same. Use of machine learning techniques such as Principal Component Analysis, and Support Vector Machine Classifier were key into finding good results for our model. At the end of the day, as great as your model can be, or your data, efficiency depends on how your hyperparameters are tuned. This tuning will be put back into question as we gather higher quality data in the future.

As the robot is now able to increase or decrease its stiffness, by mimicing the human reaction, a great challenge holds over finding a way of quantifying these variations, to make sure the stiffness does not change too rapidly, which could lead to stability issues that could become dangerous for the human sharing the task.

Future work will reside essentially in the implementation of this classification on the robot to test it. Nevertheless, other methods such as the one developed in [10] uses RBF Neural Networks to estimate the impedance of the human during human-robot cooperations. It would be interesting to see this adaptative impedance control method (see Fig 16) being implemented as it was validated on experiments before and a great challenge would consist in further developing it so that the robot could learn to estimate human motion even, smoother, and better and actively follow him while cooperating.

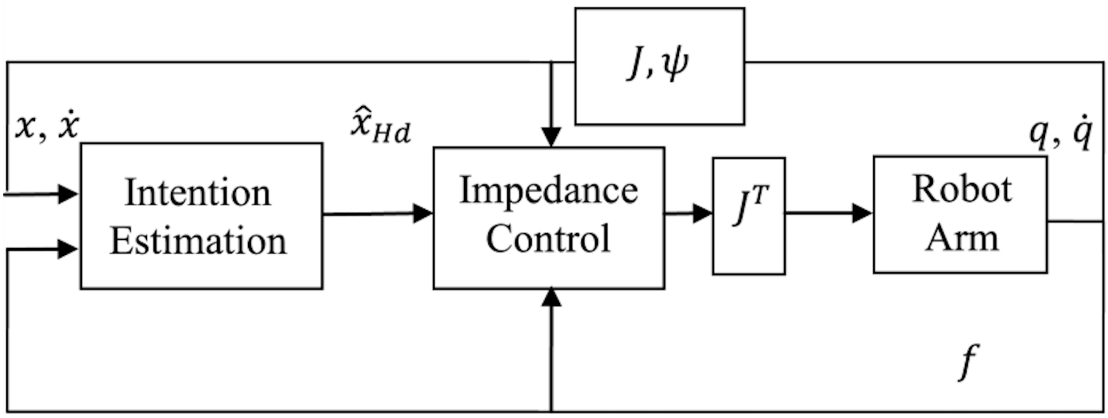


Figure 16: Adaptative impedance control with estimated motion intention using RBFNN from [10]

.

References

[1] Example of cooperation between robot and human. URL http://blog.pal-robotics.com/robotic-help-human-robot-cooperate-transport-goods/. [Online; accessed June 5th, 2019].

[2] Example of cooperation between robot and human in the automobile industry. URL http://www.ai3.uni-bayreuth.de/projects/simero/. [Online; accessed June 5th, 2019].

[3] Horizon 2020. Secondhands. URL https://secondhands.eu/. [Online; accessed June 5th, 2019].

[4] Collaborative robot provides support in automated warehouse environment. URL https://www.vision-systems.com/factory/article/16752061/collaborative-robot-provides-support-in-automated-warehouse-environment. [Online; accessed June 6th, 2019].

[5] Mahdi Khoramshahi and Aude Billard. A dynamical system approach to task-adaptation in physical human–robot interaction. Autonomous Robots, 43(4):927–946, 2019.

[6] Neville Hogan and Stephen Buerger. Impedance and Interaction Control. 10 2004. ISBN 978-0-8493-1804-7. doi: 10.1201/9781420039733.ch19.

[7] Klas Kronander and Aude Billard. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. IEEE transactions on haptics, 7(3):367–380, 2013.

[8] Ryojun Ikeura, Tomoki Moriguchi, and Kazuki Mizutani. Optimal variable impedance control for a robot and its application to lifting an object with a human. pages 500–505, 2002.

[9] Klas Kronander and Aude Billard. Passive interaction control with dynamical systems. IEEE Robotics and Automation Letters, 1(1):106–113, 2015.

[10] Yanan Li and Shuzhi Sam Ge. Human–robot collaboration based on motion intention estimation. IEEE/ASME Transactions on Mechatronics, 19(3):1007–1014, 2013.