

---

## Progress meeting 9 — 7 October, 2022

### 1. Progress

- **Impact detection:** As discussed last time, I was taking a closer look at Benn's impact detector as this supposedly was what Sven also ended up using. While doing so, I could not find any reason for why Benn looks at force estimations for impact detection rather than velocity measurements. Sven provides the reason that, during some timesteps, velocity measurements are missing. This is not the case for the force estimations due to the first order filter. I don't find this a valid argument as a similar filter could also be applied to the velocity measurements. Furthermore there are other benefits to impact detection using velocity, which I explain in section 1. I will get in touch with Benn to see if he (or perhaps you) can convince me that force-based impact detection is more suitable for reference spreading than velocity-based impact detection.
- **Learning from demonstration:** In section 2, I propose two controllers that track ante- and post impact references learned from the demonstration: an impedance controller, and a force/position controller as was used by Sven. The impedance controller is better at replicating the demonstration in a scenario with high repeatability (i.e., impact occurs at the same time as during demonstration) and can be used for a large variety of tasks (grabbing/twisting/etc). The force position controller is better at adjusting to objects of different sizes, though it does lose some versatility in which tasks it can perform. I figure that, for the goal of depelatinization that I have in mind, the geometry and inertia of the boxes does not differ between demonstration and replication (though their position may differ slightly). Furthermore I would like to not only perform grabbing tasks where the force mostly acts in orthogonal direction, but also a sliding task with shear forces. Therefore, the impedance controller is more appropriate than the position/force controller.
- **Velocity reference:** Until now, the teleoperation controller received position and orientation data from the VR device. Performing fast object manipulation tasks could benefit from also using a velocity reference, as this makes the controller more responsive. Therefore I added this feature to the dual arm controller. This change is barely noticeable for rotations, but transnational tracking is now much better, and I was also able to decrease the impedance stiffness which I previously put very high to overcome friction. Some more tuning is still required, as currently the robot's safety limits are sometimes exceeded when the VR device is moved to quickly. This seems to be
- **Custom end effector:** After some silicone delivery delays, the second custom end effector is now completed. As previously requested by Alessandro, I will try to record some videos where I pick up heavy objects with two soft end effectors before the upcoming meeting.

## 2. Agenda

During the meeting, I would like to discuss the following points:

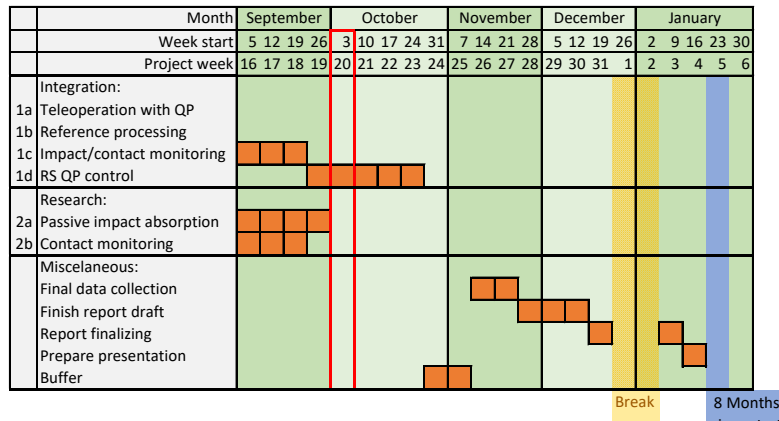
- Do you agree with my arguments for using velocity-based impact detection instead of force-based impact detection?
- Given the usecase of boxes with constant geometry/inertia between demonstration and replication, the impedance controller is more suitable than the force/position controller. Should I also consider other options?
- The safety limit issues occur mostly with rotation, which had me wondering what kind of stiffness Sven was using for rotation. I was unable to find anything regarding control of the rotational DoF's in Sven report. Do you more about his approach in this regard? Did he perhaps prescribe a reference in rotation?

## 3. Next steps

- Ensure that controller with velocity reference does not exceed safety limits
- Apply LfD controller on dual-arm setup
- Write background section for the final report

## 4. Long-term planning

Below is the current long-term planning for the project phase. The description of the labeled subjects are given in the preparation phase report and are also provided on the next page.



- 1a **Translating dual-arm teleoperation to the physical setup.** The existing implementation in simulation already uses the mc\_rtc interface, meaning that the switch to reality shouldn't pose an issue. Nevertheless, this step also involves getting familiar with the software, which increases the anticipated time for this step.
- 1b **Extracting references from the demonstration data.** The demonstrated trajectories should be split into ante-impact and post-impact sections, and extended to facilitate RS. Furthermore multiple measurements should be used to fit ProMPs, after which a reference can be generated. It is also key to identify which data should be learned from the demonstration. This is not limited to choosing between a force or

---

position reference, but can also consists of learning properties of the environment, e.g. friction cones or box inertia, that are crucial to a dual-arm box grabbing scenario.

- 1c **Integrating impact detection and contact monitoring in mc\_rtc.** The majority of the impact detector’s complexity resides in the momentum observer; however, Franka Emika’s software already has an integrated momentum observer. This still leaves tuning of the impact detecting algorithms which might be time consuming. Furthermore an analysis comparing the available methods could be worthwhile. Factors which determines the effectivity of the impact detection algorithm include speed of detection, as well as reliability, i.e. the rate of false positives. The addition of objects that cause unexpected impacts is not considered a part of the research scope.
- 1d **Configuring QP controllers for the ante-impact, intermediate, and post-impact phase.** For each of the phases, it is important to address the redundancy in the arms’ degrees of freedom. After that, control for the ante-impact phase should be trivial. For the intermediate phase, it is expected that ante-impact reference tracking without velocity feedback should be applicable on a dual-arm robot, though this might prove to be false, in which case other methods should be investigated. During post-impact control, the challenge will be maintaining non-slip contact with the box. It is difficult to say how well the results from simulations can be repeated with torque control, where the state of the box can not be sensed to be used in the QP controller.
- 2a **Passive impact absorption:** A soft cover for the end effector will be designed. Such a cover can be connected to the Panda by connecting bolts to the so-called flange interface. A mold will be created using 3D printing to allow for casting of various silicone soft covers. Design parameters – i.e. material properties (controlled by choosing different kinds of silicone) and soft cover thickness – will be analyzed experimentally. A systematic comparison between various designs will require an experiment plan including a realistic testing scenario. Evaluation of performance can be based on the oscillatory response in position and force after establishing contact. Furthermore multiple scenarios with various box surface properties and robot poses should be considered.
- 2b **Contact monitoring:** When investigating contact monitoring, two approaches can be taken: either using proprioceptive or exteroceptive sensors. A possible improvement for contact monitoring using proprioceptive sensors could be to wait a fixed time starting from the last detected impact, rather than waiting a fixed time from the first impact. As for exteroceptive sensors, they can be a hurdle for large-scale commercial applications as they are not integrated in the robot. However, if a soft cover is to be mounted to the end effector, including tactile sensors for impact and contact monitoring becomes more feasible. Practical questions such as which tactile sensor to use and how to integrate it could be addressed, though this is not absolutely necessary for completing the research goals, and therefore has a low priority.

---

## 1 Impact detection using contact force vs velocity

For impact detection, Benn and Sven considered two options: looking at jumps in velocity, or jumps in force. Ultimately, Benn and Sven both ended up using the forces, whereas I think looking at velocities makes more sense in the context of reference spreading. This section explains my reasoning.

Looking at jumps in force for impact detection could result in false positives. Consider an end effector that is already exerting a contact force on a fixed surface using impedance control. A jump in the reference would result in a jump in the contact force. Even if contact is maintained throughout the event, the jump in contact force would trigger the impact monitor while no impact has actually occurred. In the described situation, the velocity remains unchanged, meaning that a velocity-based impact detector would not return a false positive.

Additionally, contact forces cannot be directly measured, and the process of estimating them with a momentum observer is quite complex and involves a delay when compared to determining the velocity.

Given the downsides of looking at force rather than velocity, there must be significant benefits to warrant using force estimates for impact detection. I am not able to fully identify Benn's reasoning here (he states that "The most common method of impact detection is through the use of external joint torque estimators", though I suspect this is in the context of human safety). Sven, however, does provide an argument. In his experiments, he noticed that velocity data was missing during some timesteps, which resulted in poor impact detection. The impact detector using force did not suffer from this issue.

As discussed during the questions of Sven's defense, the momentum observer acted as a filter which made it so that missing data was not problematic for the impact detector that uses force. Therefore I don't think that missing velocity data is a valid argument for looking at force data; applying a filter to account for missing velocity data makes more sense in my opinion.

As a final argument, the reason for applying reference spreading is diminishing peaks in the velocity tracking error. If a jump in velocity cannot be measured, then it is also not possible for the velocity tracking error to peak (barring a peak in the reference). While there is some connection between the peaking error and contact forces, the connection between measured velocity and the peaking error is much closer.

Conclusively, based on the arguments given above, a velocity-based impact detector is more appropriate for reference spreading than a force-based impact detector.

## 2 Learning impedance vs force/position reference

When considering what reference to extract from demonstrations, two options come to mind. Firstly, it is possible to look at the position trajectory of the robot during demonstration. As the robot position by itself does not fully describe the robot's state, a force reference should also be extracted. This resembles the approach used by Sven. A second option would be to learn an impedance reference, i.e., the position of the VR device, which encapsulates both position and force. This section describes how both approaches would be implemented for a simple testcase. Based on this, a decision is made on whether impedance or force/position control is more suitable for impact-aware learning from teleoperation.

As a disclaimer, it should be noted that neither Sven's implementation nor the current VR implementation utilize a velocity reference. While Sven used reference spreading for the force reference, the value of reference spreading might be more apparent when applied to a velocity reference. The tracking performance is also expected to improve with the addition of a velocity reference. Therefore, the remainder of this section will assume that a velocity reference is included in the control scheme. Obtaining this reference should not prove difficult as a robot's velocity can easily be determined, and the HTC Vive provides velocity (and even acceleration) measurements.

### 2.1 Testcase description

The testcase consists of a 1-dof system with mass  $m = 1$  kg and position  $x$ . The EoM is given by  $m\ddot{x} = F_c + F_e$ , with control force  $F_c$  and environmental contact force  $F_e$  following

$$F_e = \begin{cases} 0 & \text{if } x \geq w \\ -k_e x - d_e \dot{x} & \text{if } x < w. \end{cases}$$

Parameters  $k_e = 10^4$  N/m and  $d_e = 100$  Ns/m represent the environment's contact stiffness and damping, respectively. Furthermore,  $w$  describes the location of the contact surface and is equal to 0 m unless stated otherwise.

The control force during the demonstration,  $F_c^d$ , is generated using an impedance controller with stiffness  $k_c = 40$  N/m and damping  $d_c = 2\sqrt{k_c}$  Ns/m:

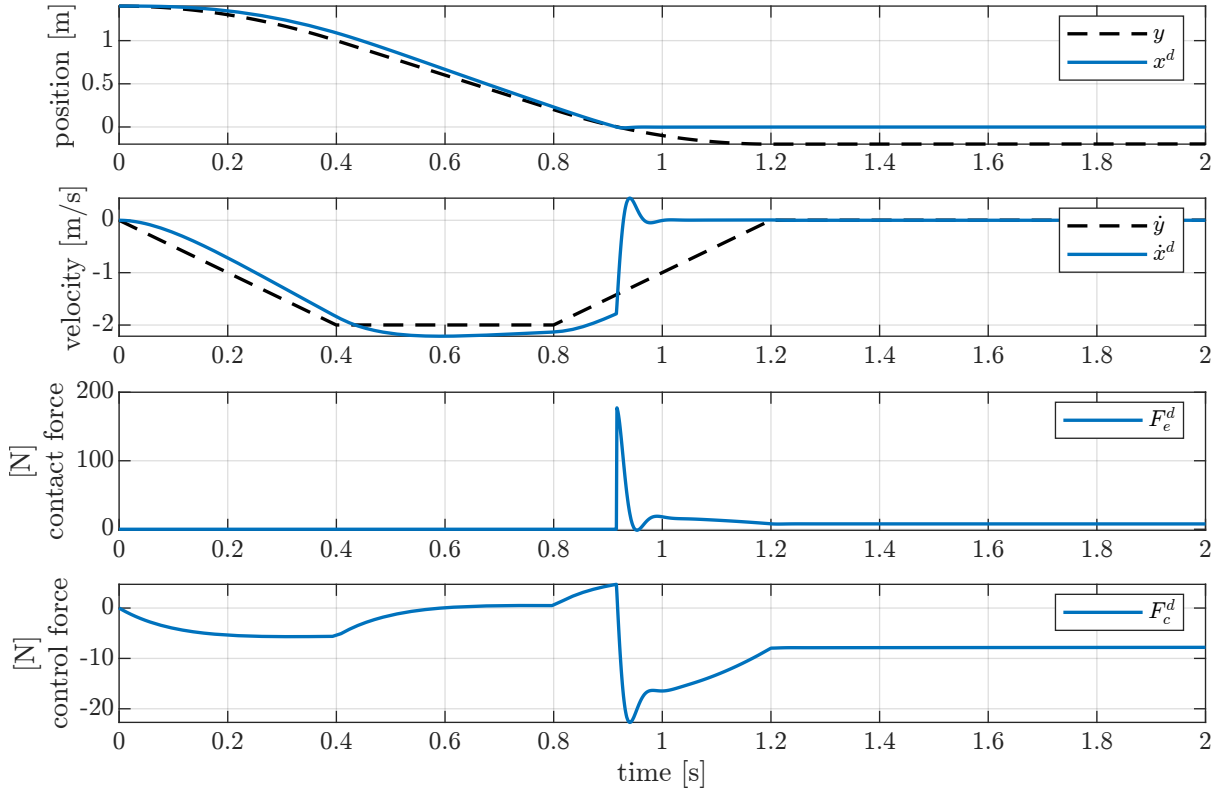
$$F_c^d = k_c(y - x) + d_c(\dot{y} - \dot{x}).$$

The references  $u$  and  $\dot{y}$  emulate an input coming from the VR device.

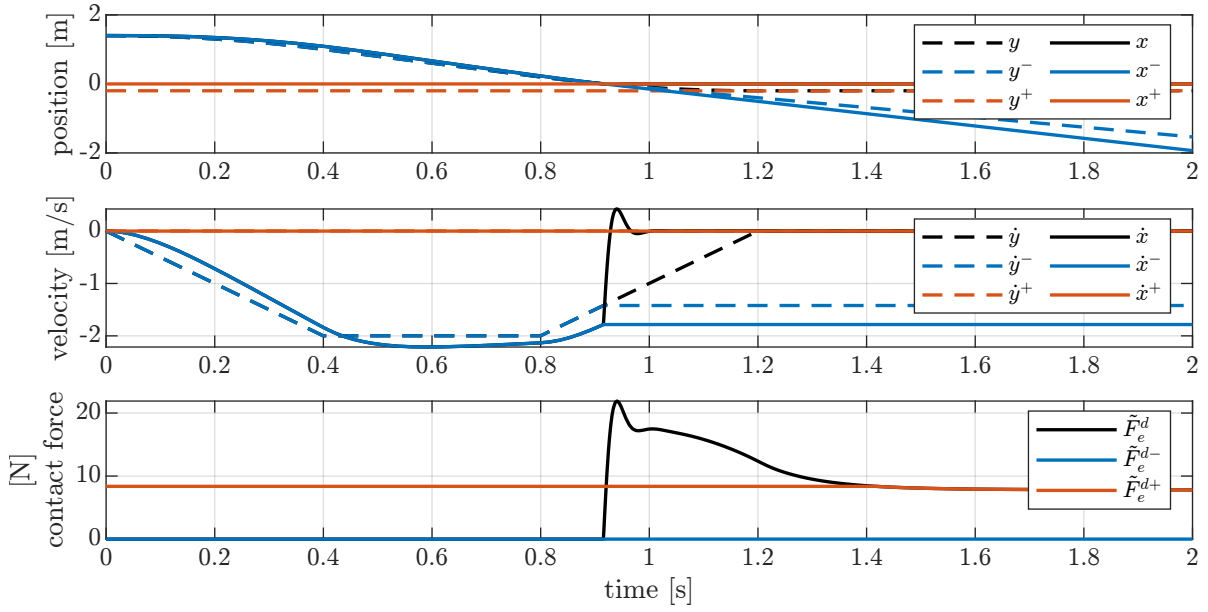
Figure 2.1 shows the reference  $y$ , together with simulation results of the demonstration. Five signals from this simulation are saved, so that they may be used as a reference for the LfD controllers:  $y$ ,  $\dot{y}$ ,  $x^d$ ,  $\dot{x}^d$ , and  $F_e^d$ . Superscript  $d$  denotes a result from the demonstration simulation.

### 2.2 Reference spreading

To apply reference spreading, the references must be split into an ante- and post-impact reference, and extended. The resulting references are shown in Figure 2.2. The ante-impact reference is extended after the time of impact, defined as the first timestep where  $x \leq d$ . Furthermore, the post-impact references are extended up to 0.5 seconds after time of impact. This delay prevents extension during an oscillation. Force and velocities were extended with a constant hold, whereas positions were updated to match the new velocity references.



**Figure 2.1:** Results of the demonstration simulation



**Figure 2.2:** Ante- and post-impact references generated from the demonstration data

### 2.3 LfD controllers

Two controllers will be compared. The impedance controller  $F_c^i$  is identical to the controller used in demonstration, i.e.,

$$F_c^i = k_c(y - x) + d_c(\dot{y} - \dot{x}).$$

Furthermore, the force/position control force  $F_c^f$  is given by

$$F_c^f = k_c(x^d - x) + d_c(\dot{x}^d - \dot{x}) + F_e^d.$$

This force/position controller is identical to what Sven used, with the exception that Sven did not use a velocity reference  $\dot{x}^d$ . (He does fit a reference to the velocity, but it is not used by his controller according to his paper/thesis. Could this perhaps be an oversight?)

These controllers were evaluated with the switching references from the previous subsection. Firstly, it will be assumed that the location of the contact surface matches the location during the demonstration, i.e.,  $w = 0$  m. The results are shown in Figure 2.3.

When comparing the impedance controller during demonstration and during reference spreading, tracking of position and velocity is very similar. There is a notable difference in control force however. Since the velocity reference is set to zero by reference spreading, the resulting peak in control force is smaller.

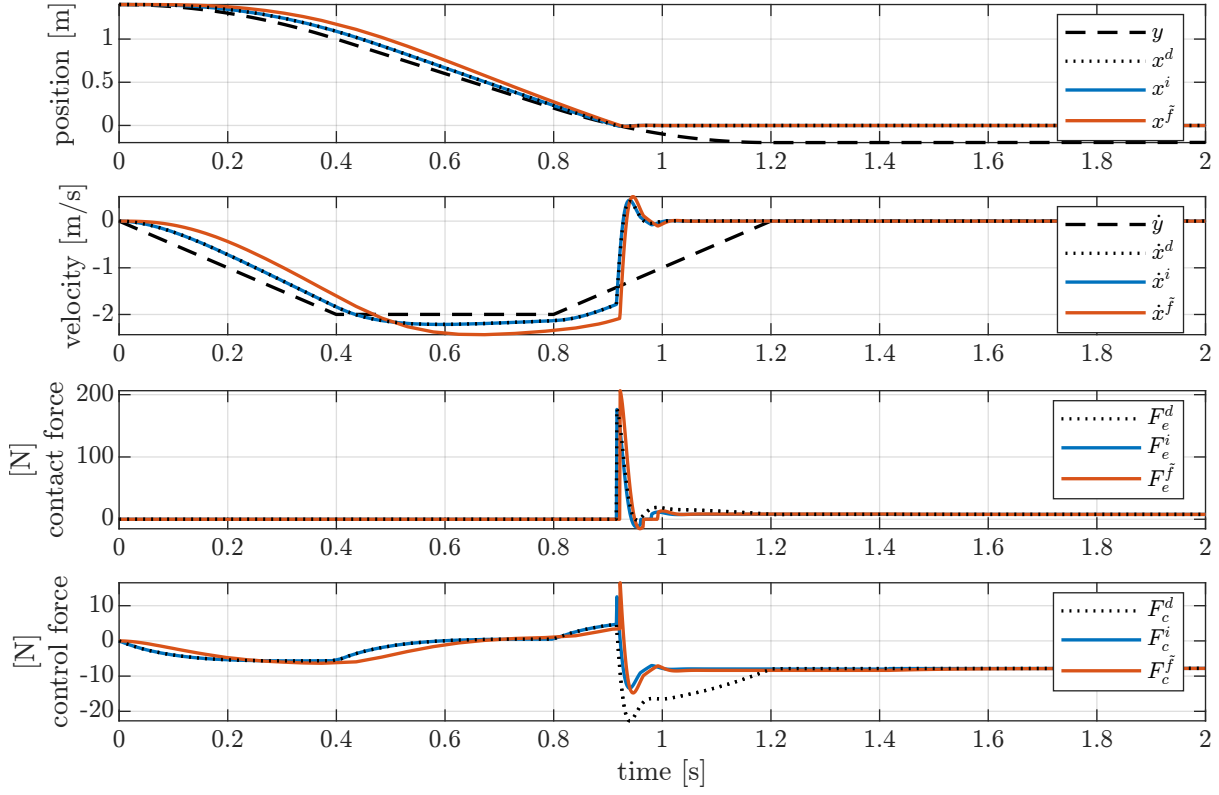
The difference between the force/position controller and impedance controller is very minor. The force/position controller does not directly follow  $u$ , but follows  $x^d$  instead, causing it to slightly lag behind the impedance controller, giving it a slight disadvantage.

In another experiment, the contact surface is moved to  $w = 0.1$  m when executing the learned reference. This is shown in Figure 2.4. This experiment gives similar results, in the sense reference spreading results in reduced control force peaks compared to during the demonstration, and the impedance controller outperforms the position/force controller in tracking the position during the demonstration.

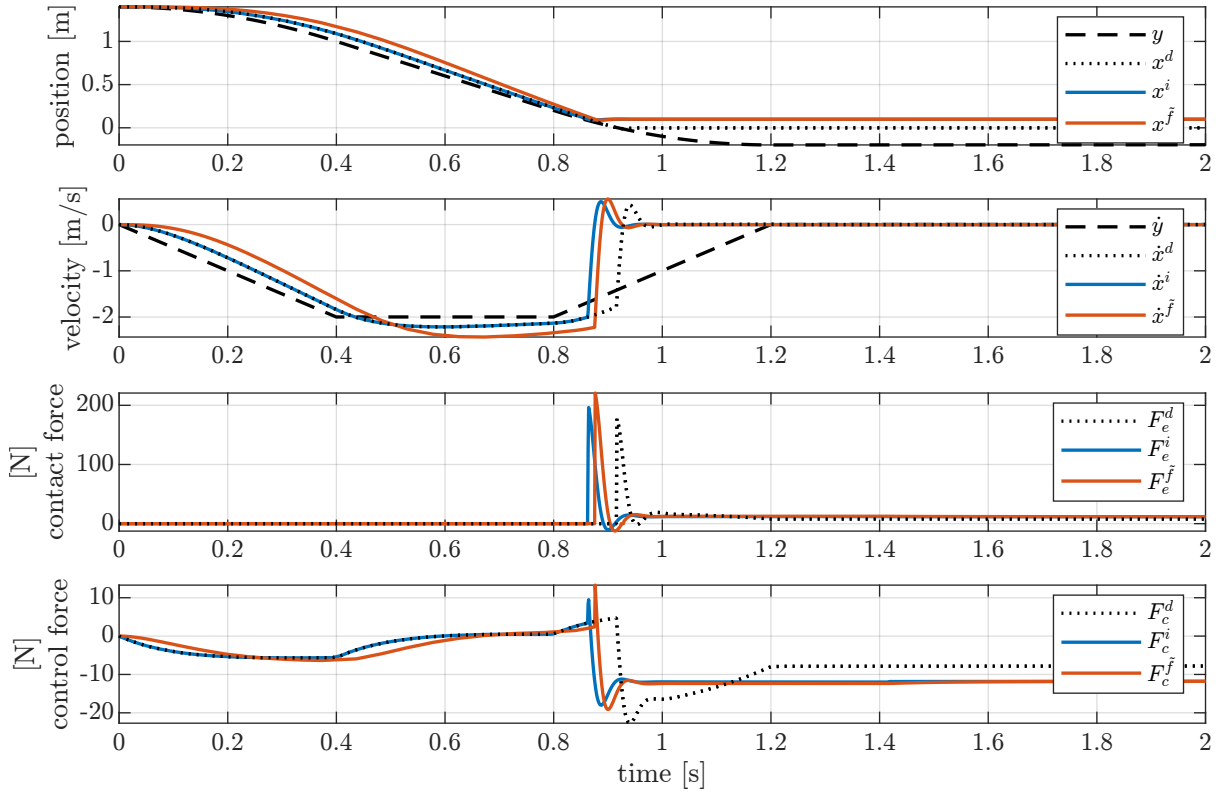
### 2.4 Conclusion/discussion

Given the experimental results, the impedance controller barely outperforms the force/position controller. Combined with the fact that the force/position controller adds complexity – in the sense that a force reference is required and that the controller deviates from the controller used during the demonstration – the impedance controller is deemed more suitable for the impact-aware learning from teleoperation usecase.

Jari lead me to understand that a force/position controller could have benefits with a slightly different implementation. Currently, if the impact surface is closer than expected, this results in a larger contact force. If the position tracking was disabled after impact, the force reference would ensure that the contact force is not affected by the position of the contact surface. In a multidimensional case, position tracking shouldn't necessarily be disabled in all directions. This direction may differ between different tasks however, therefore taking away from the controllers versatility.



**Figure 2.3:** Results of both LfD controllers using RS. Impact at  $x = 0$  m

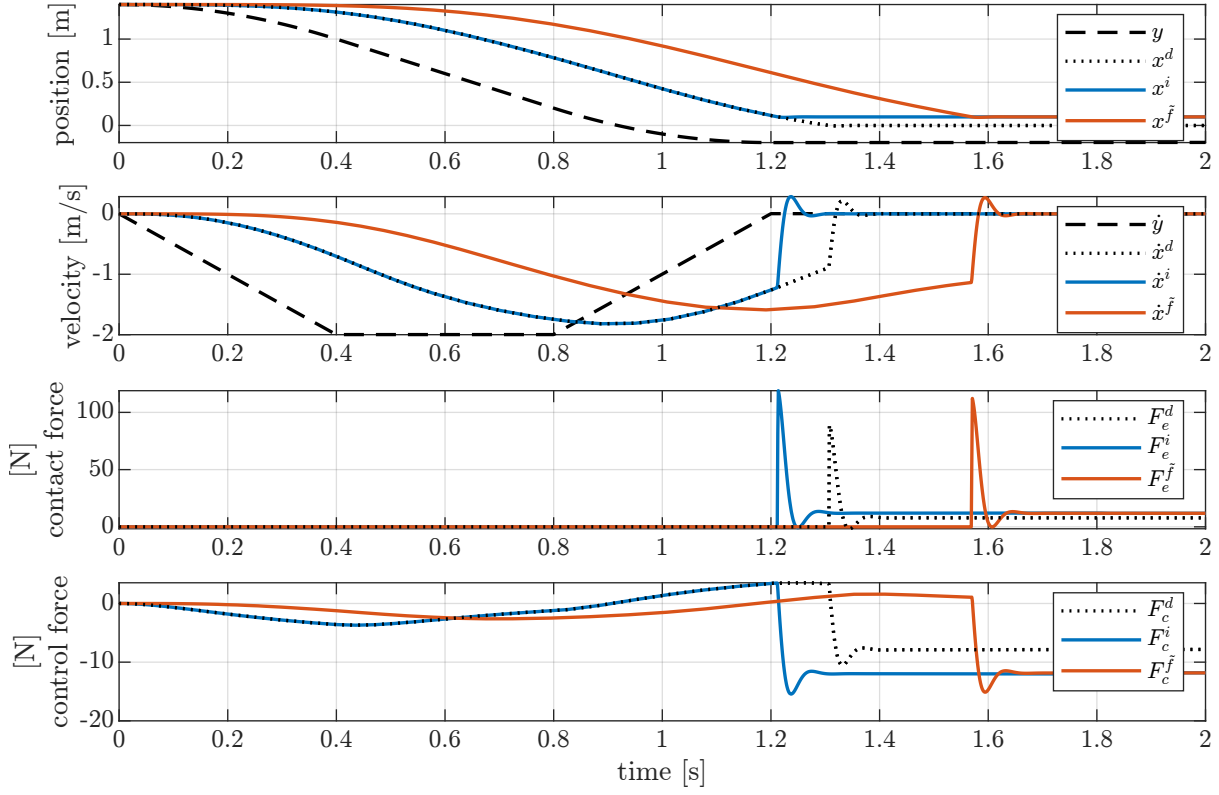


**Figure 2.4:** Results of both LfD controllers using RS. Impact at  $x = 0.1$  m

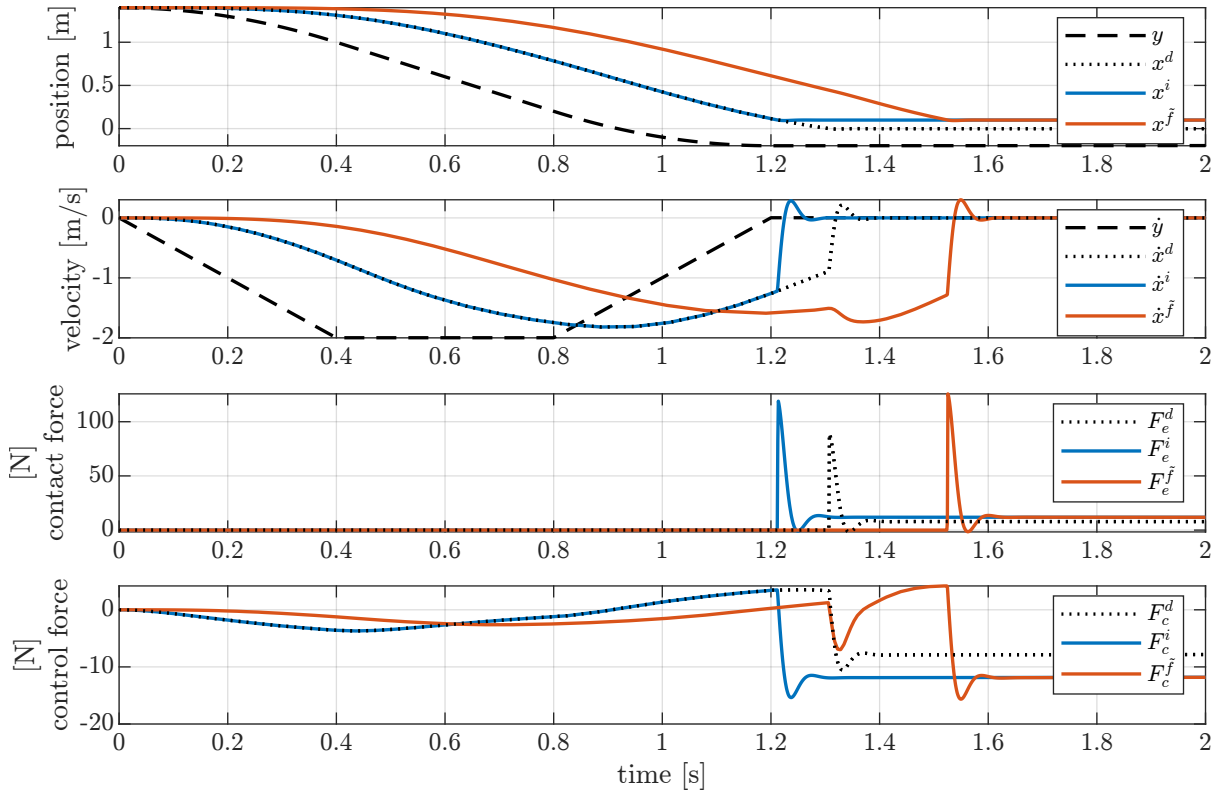


### **2.5 Extra: value of reference spreading in absence of velocity reference**

The experiment was repeated using controllers that do not use a velocity reference. This was done with reference spreading (Figure 2.5) and without reference spreading (Figure 2.6). For the impedance controller, there is no noticeable difference between using reference spreading or not using it. For the force/position controller, reference spreading ensure that the force reference does not activate prior to impact, which can be interpreted as a small improvement. Nevertheless, the effects of reference spreading isn't nearly as significant as it is in the presence of a velocity reference.



**Figure 2.5:** Results using RS without a velocity reference. Impact at  $x = 0.1$  m



**Figure 2.6:** Results without RS and without a velocity reference. Impact at  $x = 0.1$  m