
Progress meeting 10 — 24 October, 2022

1. Progress

- **Teleoperation:** Last time, I mentioned that a velocity reference results in more responsive tracking of the VR device. This was paired with the safety limits of the robot being exceeded in edge cases (mostly with fast rotations). As a solution, it was suggested that I look into the task weights, however these weights only make a difference by prioritizing certain tasks when not all tasks can be executed perfectly. This was not the problem however; even when the impedance task can be executed perfectly, rotational errors cause the controller to apply a virtual torque, which often is paired with the fast translational displacements. Instead of altering weights, I solved the problem by limiting the velocity reference to reference augmentation. An alternative approach from literature uses a force sensor to decouple translational and rotational impedance DoF's, though I decided that further investigation of this was not a priority.

Furthermore I added a feature where the VR device vibrates to indicate how much force is detected at the end effector. Personally I had difficulty with correlating the vibration with a force when teleoperating, and it also appeared that the vibrations were affecting the velocity reference extracted from the VR device. Overall I prefer teleoperation without the vibration.

- **Impact detection:** Previously I mentioned that force-based impact detection had some flaws with false positives, and that I wanted to implement position-based impact detection with jump aware filtering as was described by Mark Rijnen. It turns out that this approach also has some limitations. Since the approach is model free, it not possible to tell accelerations caused by impact apart from accelerations driven by the controller in free motion. This becomes a problem for my situation where the impacts have lower accelerations due to the soft covered end effector, and the controller gives high accelerations to track the VR device's velocity reference.

I realized that impacts cannot be detected from only force or only position estimations. In Section 1, I define sufficient conditions for an impact, and based on this I propose an impact detection scheme that uses both position and force. So far this impact detector has worked perfectly for my applications with no false positives.

- **Reference spreading with impedance control:** With impact detection, I could finally implement reference spreading on the setup. I performed a stamping experiment similar to Sven, where the end effector moves downward and collides with a fixed surface. In varying experiments, the fixed surface was lowered by 10mm or raised by 30mm, and the tests were ran with- and without reference spreading. The results are shown in Section 2.

It turns out that reference spreading does not provide any significant value to impedance control. The reference itself does not experience any jumps, meaning that the velocity tracking error does not peak. Nevertheless, this tracking error does contain a jump once the estimated system velocity jumps. An impact map could be used to match the post-impact velocity reference to the post-impact velocity of the system. Initially I envisioned that an impact map would not be necessary since the velocity jump could be estimated from the demonstration, however the current implementation only looks at the VR device. Some decisions must be made here since RS currently has no value, and without it, the controller cannot be considered impact-aware.

2. Agenda

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

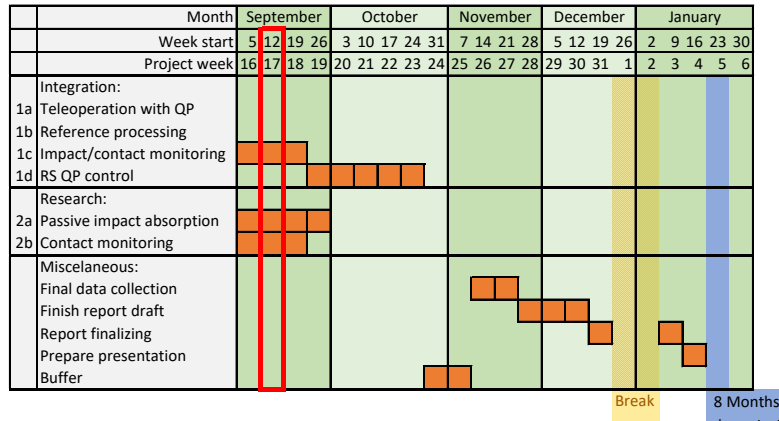
-
-

3. Next steps

-
-

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

I performed an experiment with position-based impact detection as described by Mark Rijnen, and force-based impact detection as was implemented by Benn Proper. Firstly I will give a brief explanation of how these schemes work.

1.1 Position-based impact detection

This scheme is model free: it only requires position measurements q_k where k indicates the timestep. A polynomial of degree K is fitted to the previous $M + 1$ measured positions contained in $\bar{q}_k = [q_{k-M} \dots q_k]$. This polynomial can be used to find a predicted position \tilde{q}_k during a future time t_{k+1} , i.e., $\tilde{q}_{k+1} = P(t_{k+1}, \bar{q}_k)$. A large prediction error $e_k = \|\tilde{q}_k - q_k\|$ can be the result of a large acceleration, possibly caused by an impact. An impact is detected when the error e_k exceeds a bound.

1.2 Force-based impact detection

For this method, external force estimate \tilde{F}_k derived using a momentum observer. A force-rate $\dot{\tilde{F}}$ is determined following $\dot{\tilde{F}}_k = \frac{\tilde{F}_k - \tilde{F}_{k-1}}{t_k - t_{k-1}}$. If the absolute value of this force-rate exceeds a bound, an impact is detected.

1.3 Evaluation

Figure 1.1 shows the results of an experiment where both impact detection schemes were applied. The end effector moves downward and impacts table, after which the end effector is lifted and then moved downward for a second impact. For the position-based impact detector, window length $M = 10$ and polynomial order $K = 2$ were used.

The position-based impact detector error e shows peaks while in free motion. These peaks are more prominent than the peak that happens right before the moment of impact. This is a result of aggressive tracking of the VR device and small acceleration during impacts due to the soft end effector. No bound can be defined so that impacts can be isolated, meaning that this approach is not useful for the scenario.

The force-based impact detector does not suffer from large peaks during free motion. Still, a peak can be seen at $t = 14$ s where contact is lost. Similiar peaks will appear when the force is increased over a short time, despite there being no loss of contacts. No bound can be chosen so that the true impacts are isolated from the other events with high force rates.

Conclusively, neither the force nor the position-based impact detector is useful for this situation.

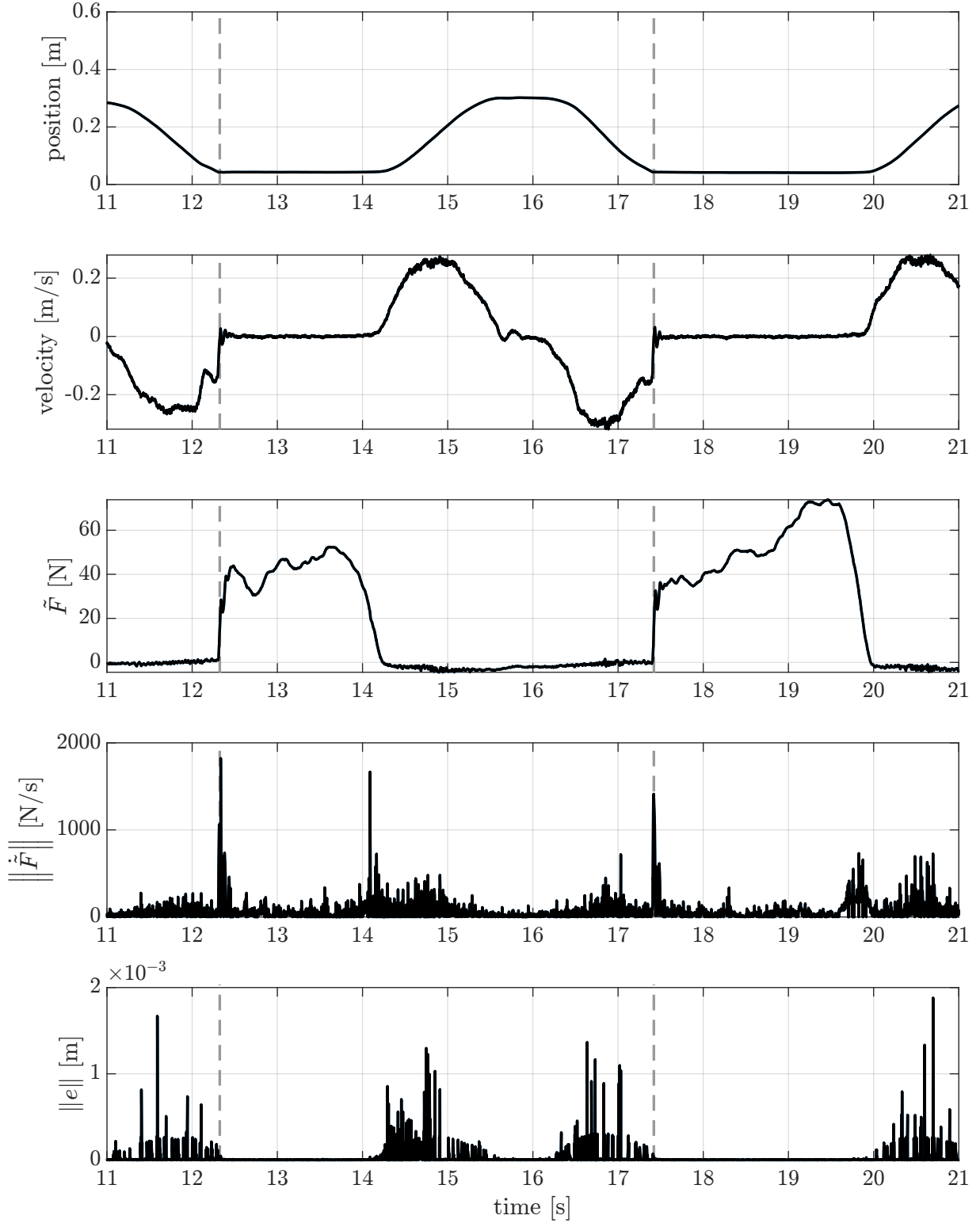


Figure 1.1: Result of an experiment featuring two impacts. Only results in downward direction are shown. The vertical dashed lines are detected with a newly-proposed method.

1.4 Hybrid impact detection

Relatively large accelerations and force rates may be caused by impacts, though they are not sufficient conditions. To formulate an alternative impact detection scheme, firstly an impact is formally defined.

Let there be two bodies named A and B. The relative velocity of body A with respect to B is $\vec{v}_{A,B}$. Furthermore, when in contact, the bodies exert a force \vec{F} on each other. We can say that an impact has occurred between time-steps t_1 and t_2 , with $t_2 > t_1$, if the following sufficient conditions are met:

1. $\|\vec{F}_{t_1}\| = 0$
2. $\|\vec{F}_{t_2}\| > 0$
3. $(\vec{v}_{A,B})_{t_1} \cdot \vec{F}_{t_2} < 0$.

The third conditions evaluates the velocity in direction of the force vector to ensure that contact was not already previously established. Without it, an impact may be detected between two bodies that do not move.

In practice, these conditions are hard to evaluate. We do not know the velocity of the end-effector to the contact surface, and the force estimations are hardly ever exactly equal to zero. Therefore, the conditions are changed to

1. $\|\vec{F}_{t_1}\| < b_F$
2. $\|\vec{F}_{t_2}\| > 2b_F$
3. $(\vec{v}_{E,W})_{t_1} \cdot \vec{F}_{t_2} < -b_v$

where b_F and b_v are bounds that should be larger than errors in the force and velocity estimations respectively. Furthermore $\vec{v}_{E,W}$ is the velocity of the end effector with respect to the world frame.

These more practical conditions are not perfect. Low-velocity impacts or impacts with low forces will not be detected. Furthermore, when the robot is static and another object bumps into it, this will not be classified as impact. Nevertheless, this is not a problem in the desired usecase, and more importantly, there will be no false-positive impact detections.

The hybrid impact detection method was evaluated experimentally. The vertical lines in Figure 1.1 are the results of this impact detection scheme, with $b_F = 8$ N, and $b_v = 0.05$ m/s.

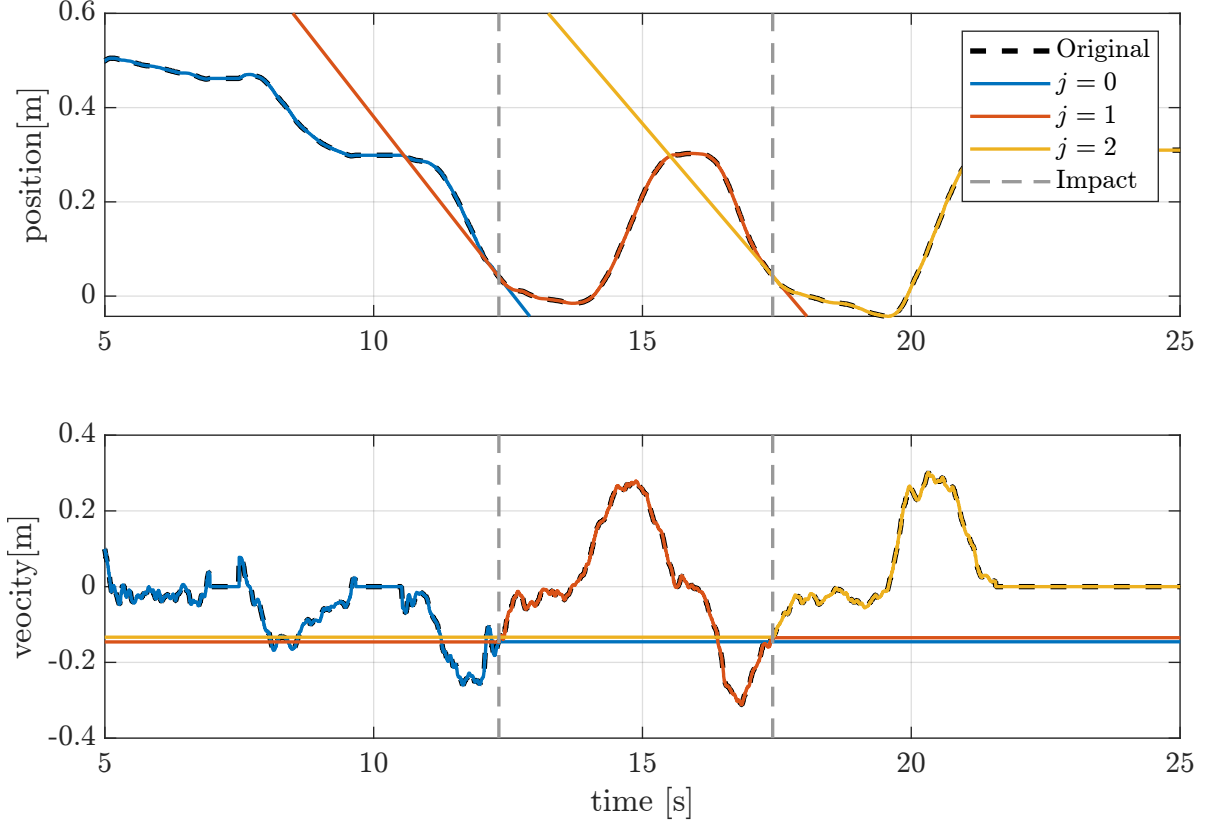


Figure 2.1: Reference extracted for stamping task, together with the extended references. The value of j relates to how many jumps have occurred.

2 Impedance control with reference spreading

To evaluate impedance control with reference spreading, firstly a demonstration was performed where the end effector stamps the table twice. A metal plate with a thickness of approximately 10mm was placed on the table during the demonstration, so that it may later be removed to move the impact surface. The extracted and extended references in z-direction are shown in Figure 2.1.

The generated references were applied in configurations where the impact surface was 10 mm lower, the same height, or 30 mm higher. Furthermore each of these configurations was evaluated with- and without reference spreading. The results are shown in Figure 2.3-2.5.

Looking at the figures, there are no significant improvements when using reference spreading as opposed to not using reference spreading when looking at tracking errors, external forces, or motor torques. Since the reference is based on the motions of the VR device, the reference does not contain any jumps, and therefore reference spreading does add the value of aligning jumps in the reference and in reality. This is visualized in Figure 2.2.

There might still be some opportunities with reference spreading, however this would require a jump to be introduced in the reference. This could be done based on the jumps in the tracked velocity during the demonstration. It should be considered, however, whether introducing jumps into a reference is actually beneficial. Another approach could be briefly reducing damping after impact, or setting the velocity reference to zero for a short while.

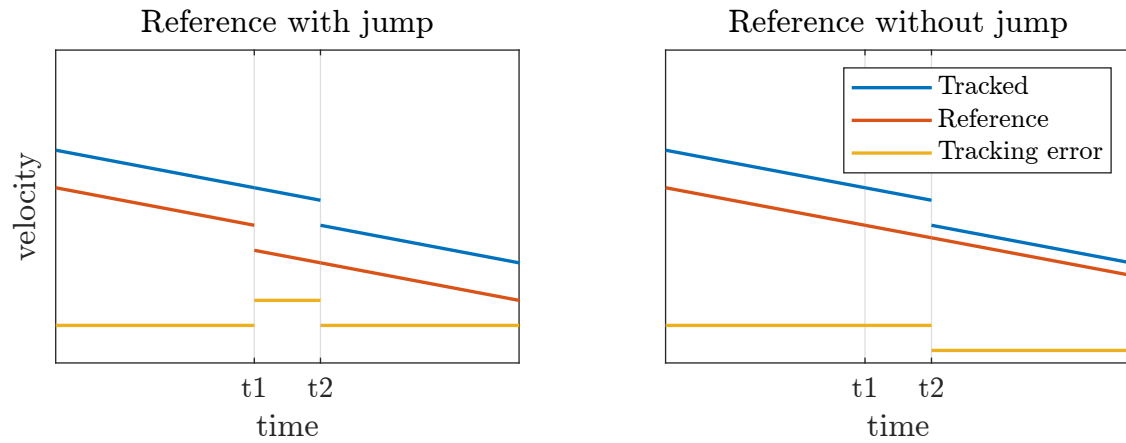


Figure 2.2: When the reference contains a jump that is misaligned with the jump that is actually tracked, the error peaks. When the reference does not contain a jump, as is the case with the reference directly extracted from the VR device during demonstration, there is no peak in the tracking error.

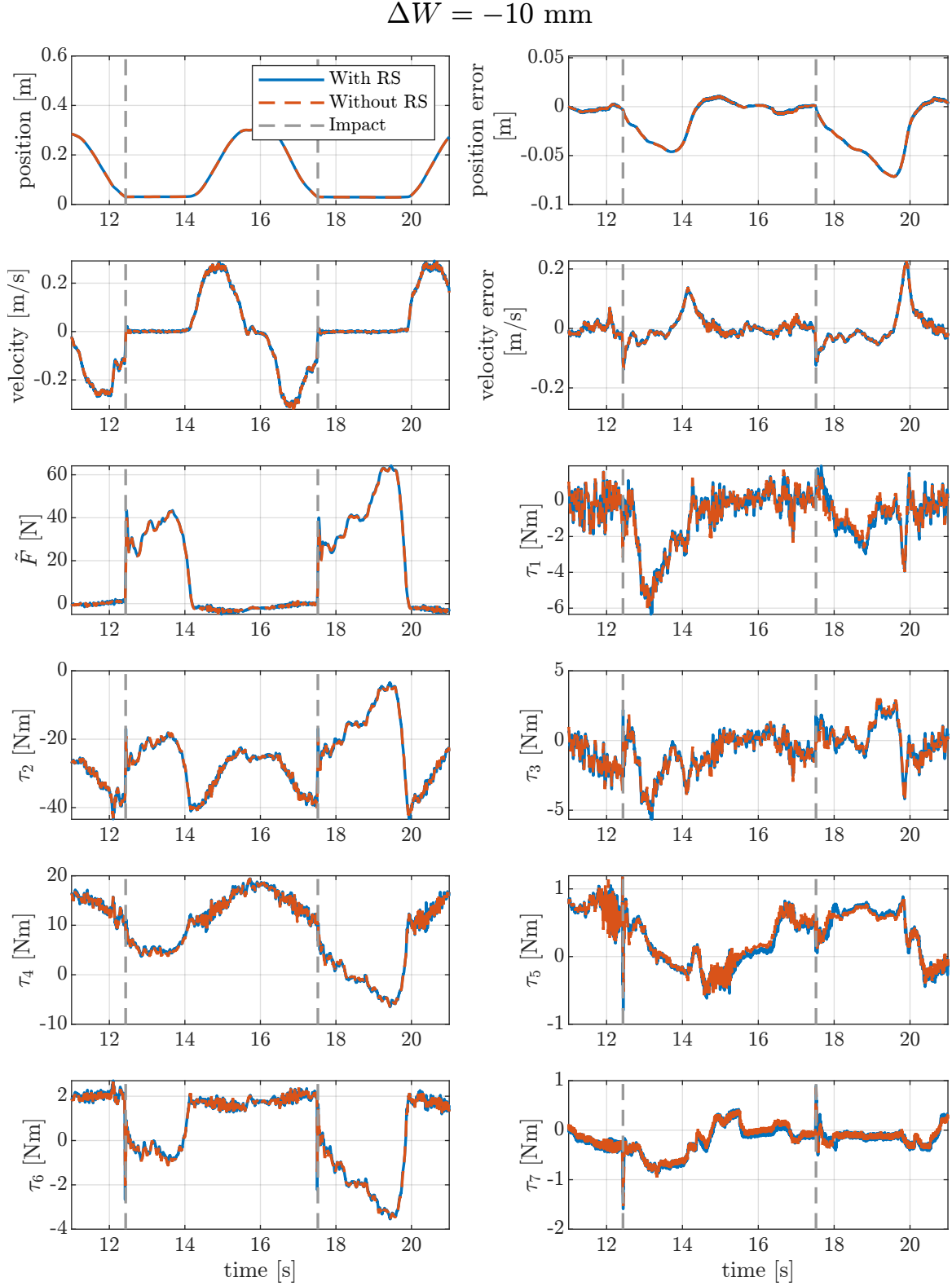


Figure 2.3: Results of stamping experiment with and without reference spreading. Only the z-direction in cartesian space is shown. The contact surface is 10 mm lower than during the demonstration.

$$\Delta W = 0 \text{ mm}$$

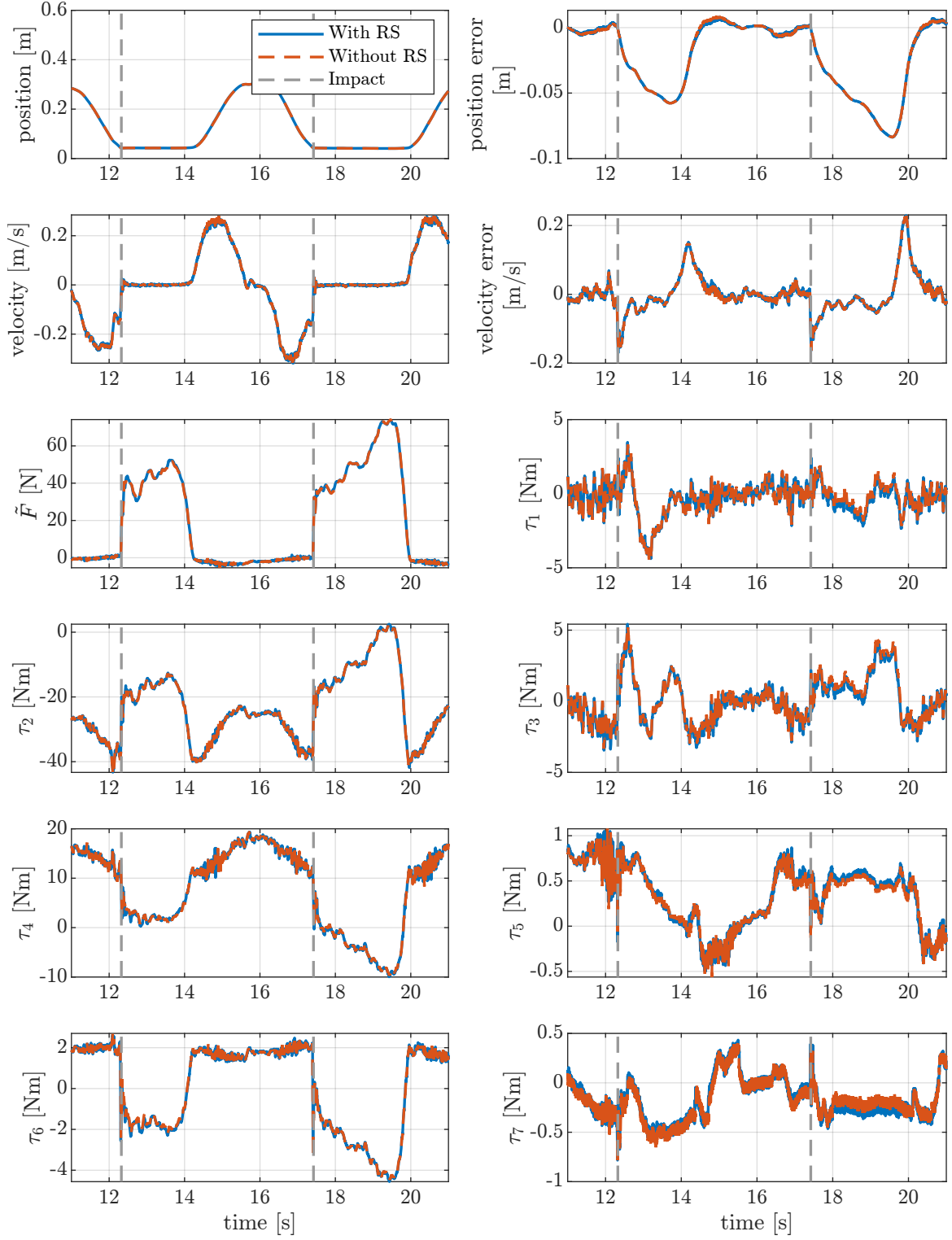


Figure 2.4: Results of stamping experiment with and without reference spreading. Only the z-direction in cartesian space is shown. The contact surface is the same height as during the demonstration.

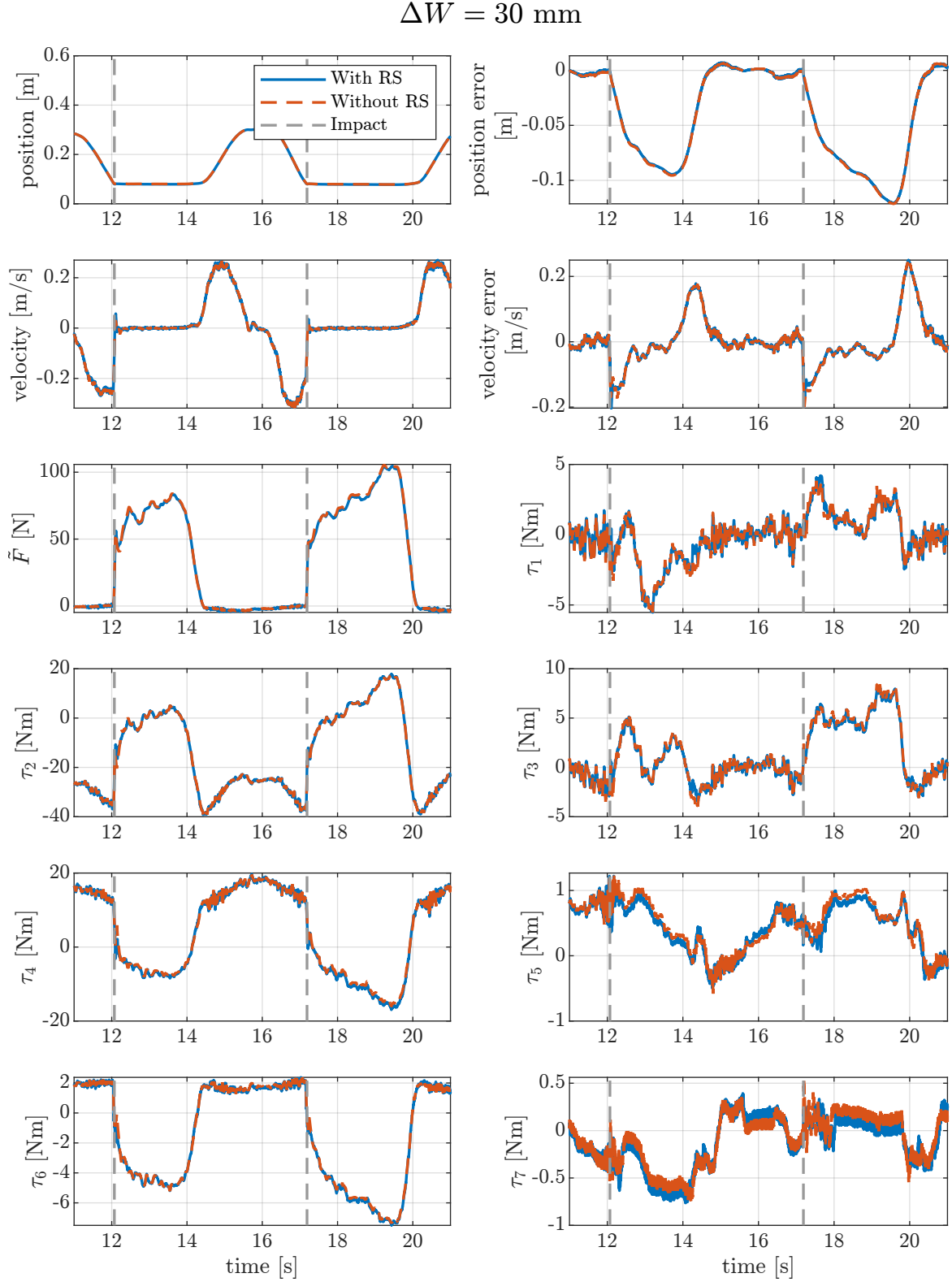


Figure 2.5: Results of stamping experiment with and without reference spreading. Only the z-direction in cartesian space is shown. The contact surface is 30 mm higher than during the demonstration.