Haptic Human-Robot Interfaces - LAB 3

Yves Martin - 262946
Luca Rezzonico - 274269

11th May 2021

Table of contents

1 Friction and gravity compensation

1.1 Dry friction estimation

First, the static dry friction of the paddle is roughly estimated. In order to do so, the motor torque is gradually increased, until the paddle moves. When the paddle moves, the value of the torque is reduced such that the paddle doesn't move anymore. That way, the paddle switches from static friction to dynamic friction almost every second, and the exact applied motor torque at the shift time is recorded.

The dynamic dry friction of the paddle is also estimated. The motor torque is gradually decreased, starting from the static friction value measured previously. For each value of the motor torque, the paddle doesn't move as it is blocked by static friction. To know whether the applied torque is above or below the dynamic friction value, a very small poke is applied manually on the paddle, such that it gets into movement. If the paddle keeps on turning, the torque compensate the dynamic friction. Otherwise, it doesn't. Therefore, the dichotomy algorithm is applied to estimate dynamic dry friction value.

In order to get the most significant results, both experiments were run 5 times. Figure 1 shows the distribution of the measurements.

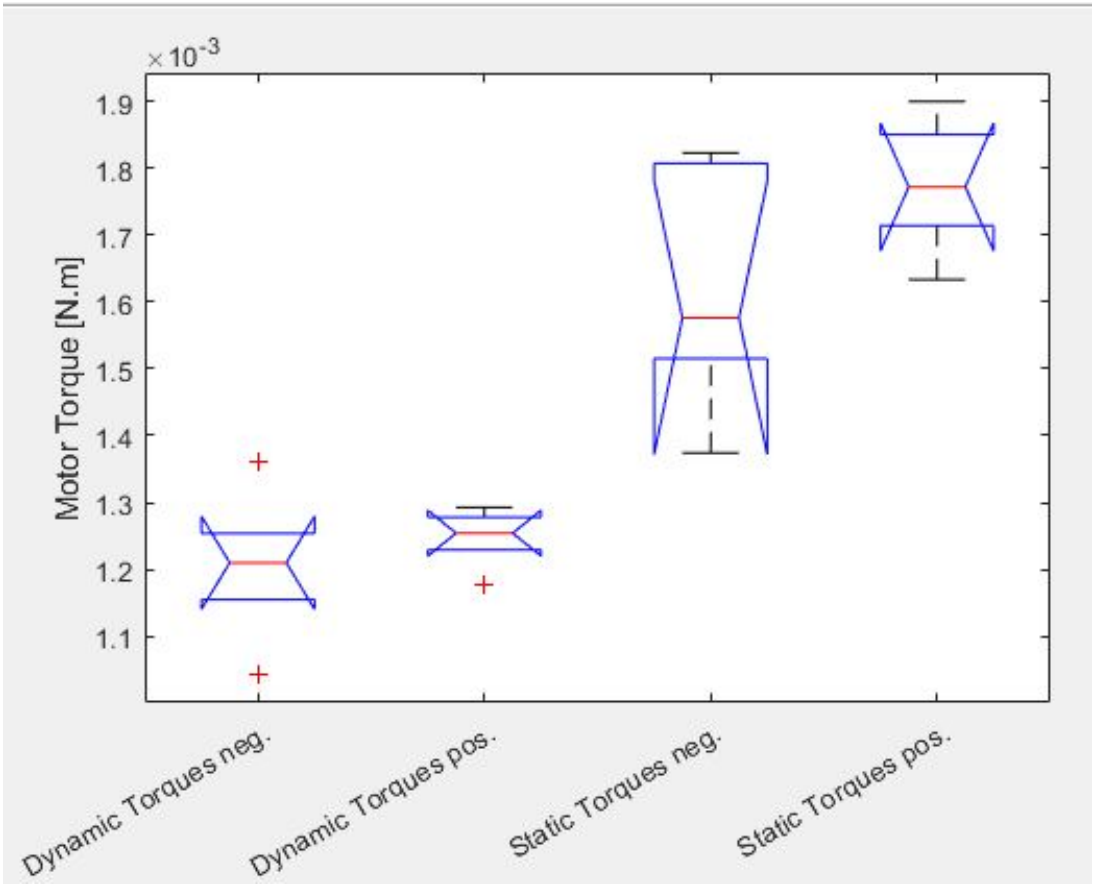

Figure 1: Measurement of the absolute value of friction torques of the paddle, in positive and negative direction

1.2 Theory

The previously identified static and dry friction moments are used to compensate for the effect of dry friction. The viscous friction coefficient of the ball bearings B_m, the mass of the paddle m_p and the distance between the center of mass and the center of rotation l can be found in the hardware documentation and are used to compensate for the

effect of viscous friction and gravity.

The equation of motion of the haptic paddle on the paddle side is reminded:

$$J_{eq} \cdot \ddot{\phi} = R \cdot T_m - R^2 \cdot B_m \cdot \dot{\phi} - m_p \cdot g \cdot l \cdot sin(\phi) - r_{p_2} \cdot F_{ext} \tag{1}$$

The viscous friction and the gravity are compensated, if the following motor torque is applied:

$$T_m = \frac{1}{R} \cdot (R^2 \cdot B_m \cdot \dot{\phi} + m_p \cdot g \cdot l \cdot sin(\phi)) \tag{2}$$

1.3 Implementation

In order to compensate the dynamic dry friction of the paddle, the hysteresis in figure 2 is implemented. The dynamic dry friction moments were found previously and the static velocity threshold was set to three times the level of noise of the filtered speed. The torque is reversed every time the speed crosses the opposite threshold, this way the paddle should not stop at the extremity of the swing. If the paddle really stops, the dynamic friction moment compensation is too low to recover the movement. For this reason, if the paddle angle is outside a small interval around the zero position, the static friction compensation moment is applied, and the paddle moves again.

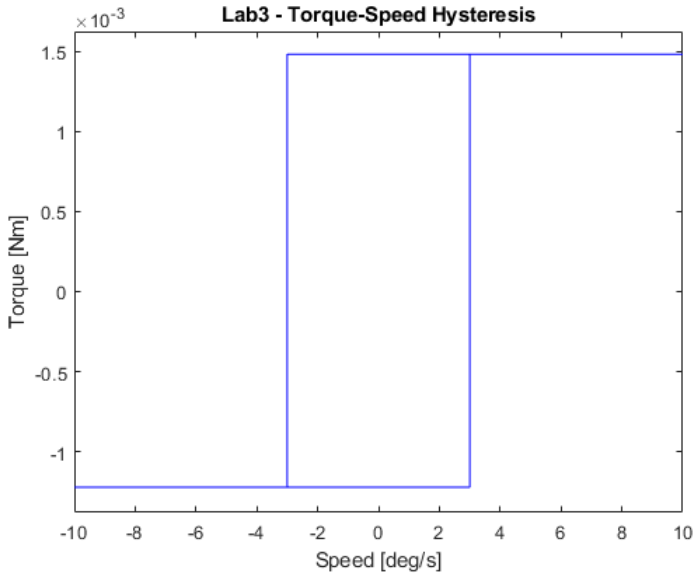

Figure 2: Torque-speed hysteresis for dynamic dry friction compensation implemented on the board

The viscous friction and the gravity compensations were implemented according to the formulas of the previous section.

1.4 Compensation Setup

The compensation is deactivated by default when the board is starting and the position measured by the encoder is used as the angle ϕ. The velocity $\dot{\phi}$ is calculated by differentiating the angle ϕ over one sampling period and is filtered at 50Hz by default, but the cutoff frequency can be changed from the user interface. The three compensations mentioned can be selected separately in the user interface, the compensation can then be activated to see the influence of each component on the motion of the paddle.

1.5 Measurements of the compensated Haptic Paddle

After compensating the dry friction, the position of the haptic paddle is plotted in figure 3. The paddle is pulled by hand to a position of 15° and then released again, three consecutive times. It can be seen that the paddle returns automatically to the angle of 0°, this is because until now the gravity and the viscous friction haven't been compensated. The oscillation damps rapidly, this indicates that maybe the friction was not perfectly compensated, but this is the best achieved result. The dry friction was not easy to compensate, because the friction torque applied varied with the angle and over the repetitions. Note that the oscillation is around 0°, but the position settles at around 0.1°, which is the position threshold chosen to make the paddle move again, by applying the static friction compensation moment if it stopped. We resume that the gravity, which acts like a spring, and the viscous friction, which acts like a damper are not compensated yet.

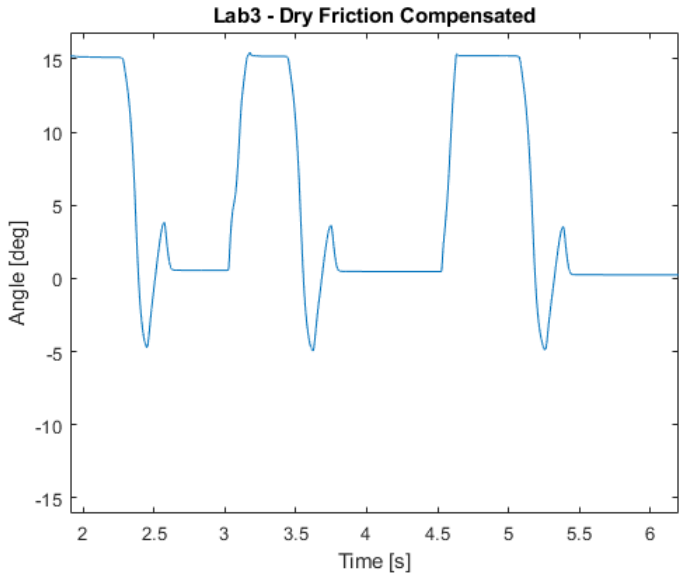

Figure 3: Position of the Haptic Paddle [deg] when released at an angle of 15°, and dry friction compensation is active

Now the paddle is pulled to a lower position of around 12° and released again. It is visible that after compensating the viscous friction, the paddle's oscillations are ongoing. They are not perfectly constant in time, this may be due to the previously not perfectly compensated dry friction.

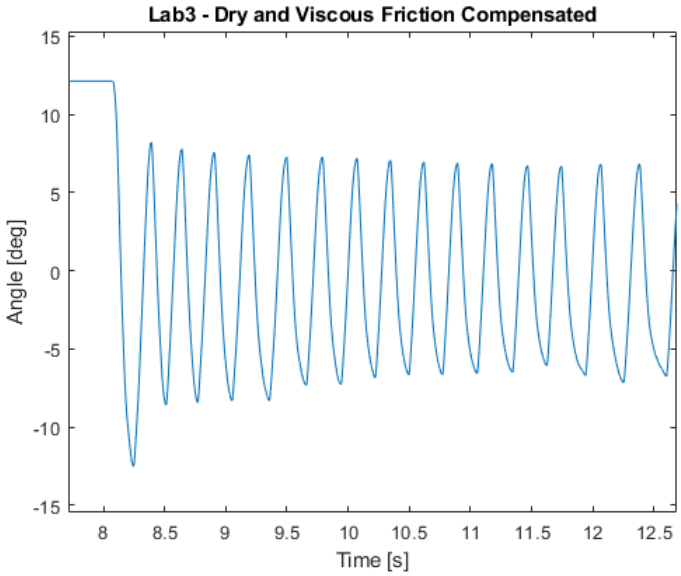

Figure 4: Position of the Haptic Paddle [deg] when released at a certain angle, and dry friction and viscous friction compensation are activated

After compensating both friction components the gravity compensation is added, and a manipulation is shown in figure 5. In the beginning the paddle is at rest and after approximately 0.5s an external force was applied to disturb the system. The paddle turns and bounces off the virtual wall already implemented at -15°, in order to avoid any damage. It can now only be stopped by an external force. The motion feels much smoother than when the compensation was disabled, this was shown in the previous plots.

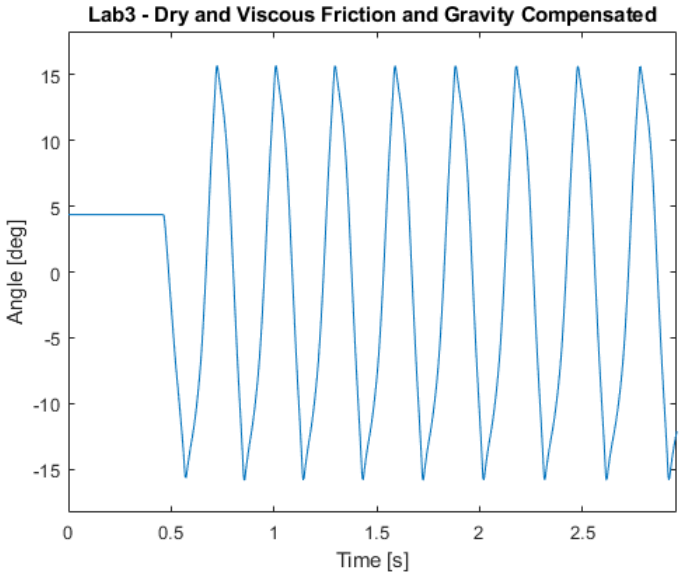

Figure 5: Position of the Haptic Paddle [deg] with a small impulse at the start, and dry friction, viscous friction and gravity compensation activated

The compensation is exact in closed-loop, since the velocity an the position measurements are used to recompute

the torque applied by the motor at each iteration. To illustrate the closed-loop system, figure 6 shows the bloc diagram taken from the course of the feed-back compensated paddle.

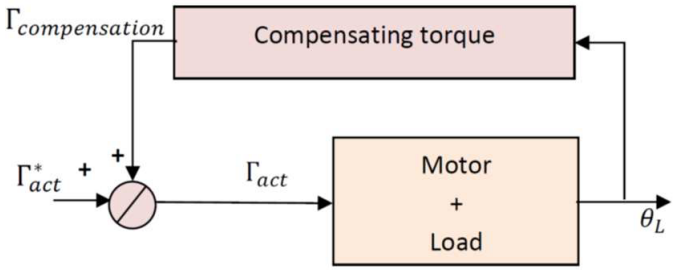

Figure 6: Feed-back compensation bloc diagram

1.6 Compensation considering multiple degrees of freedom

Here will be discussed how the compensation would be implemented in a case where the robot would be using multiple degrees of freedom.

1.6.1 The viscous friction compensation

The viscous friction only need $\dot{q}$ to be computed. A simple formula that can be used, is a Hadamard product with dry friction constants:

$$\vec{\tau}_{Visc} = \vec{\dot{q}} \odot \vec{\mu}_{Visc} \tag{3}$$

1.6.2 The dry friction compensation

To compute the dry friction, one can choose a model where the dry frictions are supposed constant. Let's call $\vec{a}$ a vector where if, for a given joint i, $q_i < \epsilon$, $\vec{a}_i = 0$, and else $\vec{a}_i = 1$. $\vec{a}$ corresponds to whether the joint is moving or not.

$$\vec{\tau}^{Dry} = \vec{\dot{a}} \odot \vec{\mu}^{Dry}_{Static} + \vec{\dot{a}} \odot \vec{\mu}^{Dry}_{Dyn} \tag{4}$$

1.6.3 The gravity compensation

For gravity compensation, one has to use inverse dynamic model to project gravity on each segment of the robot.

$$\tau_P = J^T \cdot F_R \tag{5}$$

2 Virtual Wall

2.1 Implementation of the Virtual Wall

Virtual walls were implemented according to the following pseudo code:

```
WallAngle = 15 [deg]
if |PaddleAngle| > WallAngle
    torque -= sign(PaddleAngle)*Stiffness*|PaddleAngle-WallAngle|
    torque -= sign(PaddleSpeed)*DampingFactor
end
```

2.2 Z-Width Measurement

Using the previous implementation and the GUI interface to increase the virtual stiffness until the paddle vibrates, the K-B plots were computed and plotted in figure 7.

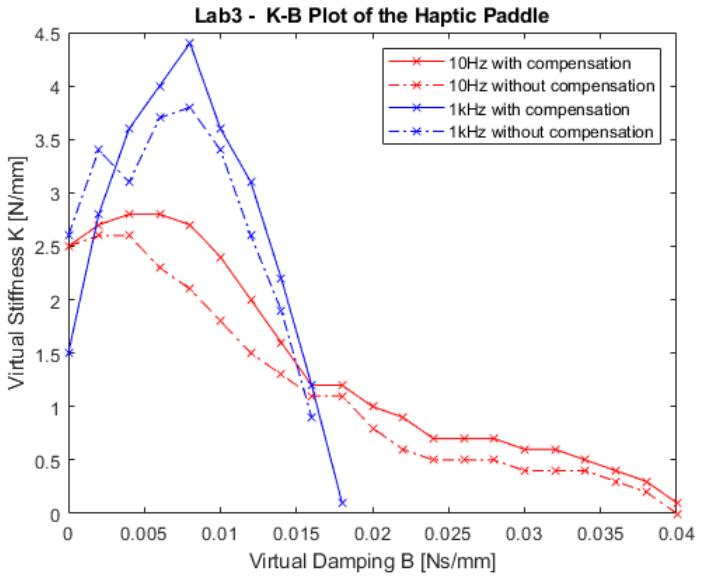

Figure 7: K-B plot of the haptic paddle using the filtered speed at different cutoff frequencies and with the compensation enabled or disabled

Low-pass filtering the speed with a higher cutoff frequency results in the system being stable for a higher virtual stiffness, but the virtual damping is smaller. For a lower cutoff frequency, the maximum virtual stiffness before instability is decreased, but the maximum virtual damping is increased. The bandwidth is smaller and this introduces delay in the system. If the low-pass filter is not strong enough, there is too much noise and then the damping factor makes the motor oscillate at high frequency, when pushed against the wall. Typically, using a Hall-effect sensor would add measurement noise into the system, and more filtering would be necessary. The K-B plot would be much flatter (K values would decrease). Compensation results in the system being stable for a higher virtual stiffness, thus the K-B plot is shifted (K values are increased). The shift corresponds the impedance of the haptic device, which is compensated in closed-loop. For perfect compensation, the closed-loop impedance would be equal to the device impedance. An other way to improve the Z-width is using better filtering. In the course, First-Order Adaptive Windowing (FOAW) shows much better Z-width performance, than the Butterworth filter used in this project. FOAW dynamically adapts the cutoff frequency, to get the best trade-off between precision and reactivity [1].

References

[1] Farrokh Janabi-Sharifi, Vincent Hayward, and Chung-Shin Jason Chen. Novel adaptive discrete-time velocity estimation techniques and control enhancement of haptic interfaces. *IEEE Trans. Control Systems Technology*, 8(6):1003–1009, 2000.