Haptic Human Robot interfaces

TP3 Report

Jonathan Bregnard Igor Ayrton

May 13, 2015

Abstract

During this tp we identified the friction torque on the paddle. First we identified dry friction and implemented a compensation. We then implemented gravity compensation. We followed by identifying viscous friction and compensating it as well. Finally a virtual wall was modelised and implemented. We tested our code first only with dry friction compensation but no gravity compensation. Second we added gravity compensation. Third we added viscous friction compensation.

1 Friction torque identification

1.1 Dry friction

To identify the dry friction torque of the paddle we applied a motor current ramp in small steps until the paddle started to move. We did this in both rotating directions. The results can be seen in table 1.

measurement N°	CW [mNm]	CCW [mNm]
1	2.24e-4	-6.32e-4
2	2.073-4	-6.39e-4
3	2.72e-4	-6.64e-4
4	2.46e-4	-6.46e-4
5	2.72e.4	-6.64e-4
mean	2.44e-4	-6.49e-4

Table 1: Dry friction torque measurement

1.2 Viscous friction

To calculate the damping B_m of the motor we used the data sheet of the motor and the relation in equ.1. We obtain $B_m = 0.0145e^{-3}[mNm * sec * deg^{-1}]$.

$$i_m * k_\tau = B_m * \omega_m \tag{1}$$

1.3 Implementation

To detect which way the paddle is going we measure the direction of the speed. Idealy we would have a strain gauge to tell which way the user torque is going. As we are measuring speed by deriving the position we get with the encoder we filter it. The filter is a moving average over 10 data points. This filter is implement via the code in listing1 with the code in listing 2.

Listing 1: moving average filter

```
void ctrl_RegulatePosition()
{
...
ctrl_motorTorque_mNm_c=20*sin((ctrl_timestamp/1000000.0)*2*PI/3);
...
}
```

Listing 2: Friction compensation implementation

```
void ctrl_RegulatePosition()
{
...
ctrl_motorTorque_mNm_c=20*sin((ctrl_timestamp/1000000.0)*2*PI/3);
...
}
```

2 Virtual wall implementation

2.1 Implemetation

Listing 3: Hall sensor running mean implementation

```
  virtual wall implementation
```

2.2 K-B plot

2.3 Results

3 Conclusion

Wall is realistic