ETH MIKE Characterization

Jannis Bähler

23^{rd} of December 2021

1 Introduction

The ETH MIKE is a robotic assessment & therapy platform for hand proprioceptive and sensorimotor functions [1]. Since the initial design of the device was not according to the medical device regulations of Switzerland, the electronics of the system were redesigned [2]. The new electronics are reduced in cost and size compared to the old version. For this report, the most important difference is the motor. Initially, it was run with 48V but now only 24V. Also the torque constant is reduced from $0.137\frac{Nm}{A}$ to $0.0302\frac{Nm}{A}$, therefore the system needs more current to generate the same mechanical power as in the initial version. More information about the redesign can be found in Julian's report [2]. The redesigned electronics have not yet been tested and evaluated, which was done for this report.

This document is a description of the characterization for the ETH MIKE, which allows for a comparison of the performance between different devices. Two devices have already been evaluated: ETH MIKE #3 with the initial design and ETH MIKE #6 with the redesigned electronics. The different tests are described in the different chapters in the following order: maximum acceleration, static friction, dynamic friction, transparency planes, KB-plot & virtual wall rendering and position bandwidth.

Detailed Manuals of how to implement the measurements and analysis can also be found in the backend gitlab of ETH MIKE under *eth-mike-back-end\ETH_MIKE_Characterisation\Doc*. The scripts for Data Analysis can be found in *eth-mike-back-end\ETH_MIKE_Characterisation\Data Analysis*. I wrote the scripts in python, there is always one version as python executable and as jupyter notebook.

I recommend using jupyter-lab, installation instructions can be found under this link:
`https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.`

`html`.
It is essential, that python is already installed.

Further python libraries that are needed are:

- npTDMS: `https://nptdms.readthedocs.io/en/stable/`
- Matplotlib: `https://matplotlib.org/stable/users/getting_started/index.html#installation-quick-start`
- numpy: `https://numpy.org/install/`
- scipy: `https://scipy.org/install/`

2 Maximum Acceleration

In this test we want to measure the maximum acceleration of our device. This lets us compare the limits of the different devices as well as the performance of the electronics (e.g. MIKE #3 and MIKE #6).

2.1 Implementation

In order to determine the maximum acceleration, the motor is fed with its maximum current for a short time. We want to know the positive as well as the negative maximum acceleration, this can be set in the frontend of the vi by selecting from cases "positive" and "negative". Keep in mind, that positive is a clockwise movement and negative is a counter-clockwise movement. The process can be started by pressing *start*, however, before you do that you should move the end-effector to the end of its range of motion (i.e. for the clockwise movement, move the end effector to -90 degrees for the start).

2.2 Data Analysis

To determine the maximum acceleration, the measured position signal is recorded over time. The recorded signal is cut, so that only the measured points while the end-effector is moving are taken into account. This "shortened" signal is fitted with a polyfit in order to gain a time-dependant position function, which can then be derived two times in order to calculate the acceleration signal over time. The degree for the polyfit can be set in the code (try and error until it is just not over-fitted). The maximum acceleration can then be read from that signal. But one has to keep in mind that it is only an estimated signal, so it is advised to implement the measurement a few times and to compare the results.

2.3 Results

I did the measurements for MIKE #6 and MIKE #3. For each device I made three measurements for positive and negative acceleration. The results are shown in the following table.

It is visible that the maximum acceleration of MIKE #3 is larger, this is due to the the fact that the new motor of MIKE #6 provides less torque. What is strange though is that in the ICORR paper from 2019 [1] a max acceleration of 52'000 $\frac{rad}{s^2}$, which is higher than the one I measured for MIKE #3. A possible reason for that could be the different method used to derive the acceleration.

	MIKE 6		MIKE 3	
	measurements $[\frac{rad}{s^2}]$	**avg** $[\frac{rad}{s^2}]$	**measurements** $[\frac{rad}{s^2}]$	**avg** $[\frac{rad}{s^2}]$
positive	28'941	26'281	31'950	31'439
	23'995		27'145	
	25'907		35'222	
negative	-28'042	-26'858	-33'013	-32'280
	-27'683		-30'033	
	-24'850		-33'795	

The following plot shows an example of the analysed position and velocity signals:

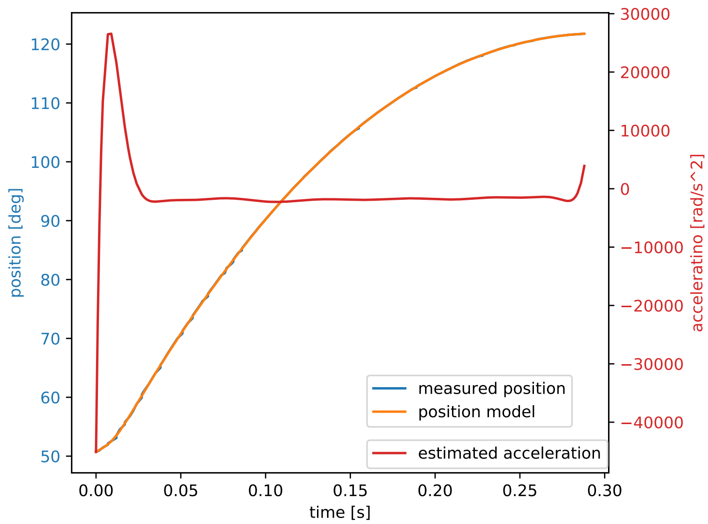


Figure 1: The plot shows the measurement of the maximum acceleration test of MIKE #6. As one can see the model of the position fits the measurements.

3 Maximum Torque

The measurement for the maximum torque is very straightforward. One just takes the measurements of the maximum acceleration test, records the measured current and extracts the largest measured current. By multiplying this current with the torque constant of the motor, the maximum torque can be calculated. The results are summarized in the following table:

	MIKE 6	**MIKE 3**
Torque Constant [$\frac{Nm}{A}$]	0.0302	0.137
max Torque [Nm]	0.312	0.847
max Current [A]	10.331	6.179
min Torque [Nm]	-0.293	-0.858
min Current [A]	-9.695	-6.265

As explained before, the new motor of MIKE #6 generates less torque, but it sinks more current than the old version. The maximum current, which the motor can consume is set by the ESCON. Currently, the maximum is ca. 10A (as can also be seen in the table). If the torque generation is needed to be higher in the future, it can be adjusted in the settings of ESCON (see *eth-mike-hardware/ETH MIKE/05 Setting up/Set up ESCON*. The overall maximum current at the moment is 12A. This one can also be increased by adding a second power source as Julian described in his report [2].

4 Static Friction

The goal of this test is to determine the static friction over the whole range of motion. The static friction torque is the torque which is needed to move the end-effector from standstill. This test allows to make a statement about the transparency of the system without any compensation control implemented.

4.1 Implementation

The range of motion is defined between $\pm 90°$ and it is driven in steps of $5°$. The user has to define the current direction (positive is clockwise and negative is counter-clockwise). The profile is then also driven in this direction. The program runs full-automatic. In the end however, it does not stop automatically but has to be stopped manually. This can be done as soon as the last measurement is taken (the end-effector will go back to $0°$ and continue to do measurements for $0°$ forever). After a position setpoint is reached, the system starts to apply a current (by default in 0.001 A steps). Meanwhile, the position difference between two consecutive measurements is observed and as soon as the difference exceeds $0.015°$, the current setpoint is saved as friction torque and is reset to zero. Afterwards, the system drives to a new position setpoint. Between the reaching of a position setpoint and the increasing of the current, the system waits a small amount of time and does nothing, since there could be some rest-movement, which would falsify the measurements of the friction torque.

4.2 Data Analysis

In order to analyse the results, the final current setpoints are recorded and assigend to the correct position. Since there can always be some measurement errors (mostly too low measurements), it is best to have multiple data sets. These different data sets can all be fed into the script at once. For each position, the largest friction torque measurement is taken. These measurements are then multiplied with the torque constant of the motor version in order to calculate the torque.

4.3 Results

In the following two plots, results for MIKE #6 and MIKE #3 can be seen. It is visible that the friction for MIKE #6 seems to be slightly larger than for MIKE #3. What is a little strange is that in the ICORR paper from 2019, the measurements are all higher. Maybe then the sealings of the bearings were not removed, increasing the friction a lot. The measurements presented here are all pretty consistent over the whole range of motion, there are a few outliers, which are probably due to small measurement errors.

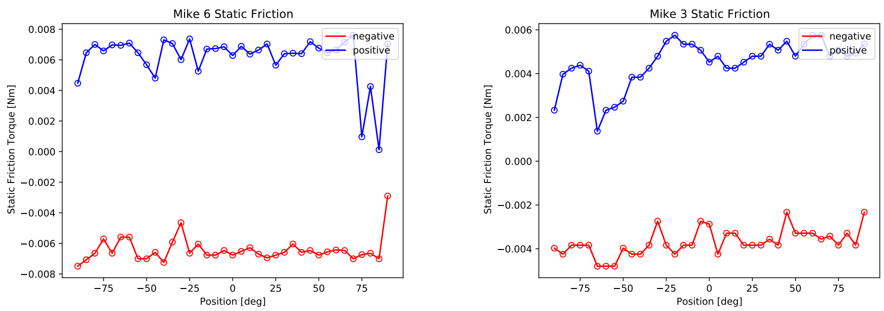


Figure 2: Positive current equals clockwise and negative equals counter-clockwise; The profile was driven in the same direction as the current was applied, since this provides more friction. As can be seen, the static friction is low and therefore the system is very transparent without any control.

It is also important to mention that the current setpoint is used to evaluate the friction torque and not the estimated current. The next plot shows the current setpoint and the measured current in a cutout from a static friction dataset:

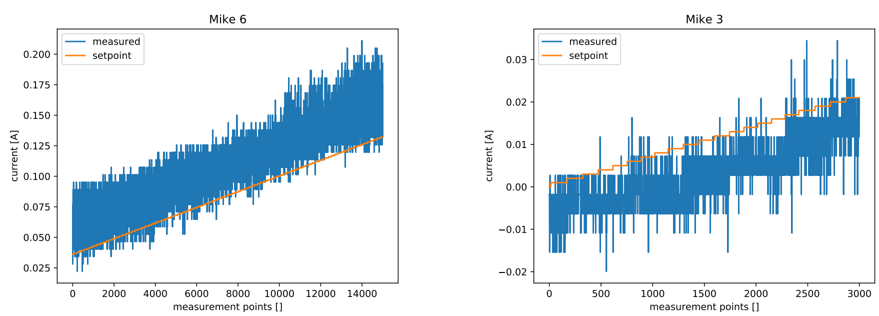


Figure 3: The plot shows the measured as well as the desired current.

It is visible that the measured current is noisy and always seems to have an offset from the setpoint, which has to be taken into account when analysing the results.

Another observation was that there is a strong dependency on the direction in which the position profile is driven. In more detail, the friction torque is much higher when the profile is driven in the same direction as the current is applied. For example if a negative (counter-clockwise) current is applied, the

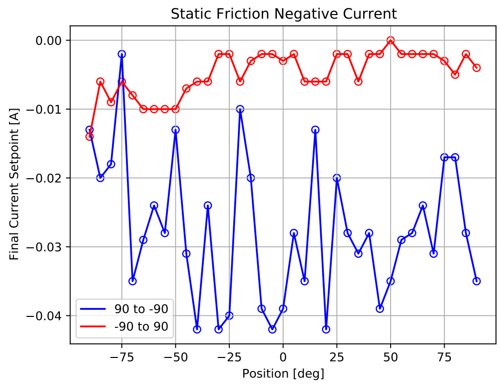


Figure 4: Here the direction dependency is visualized. The profile was driven in clockwise and counter-clockwise direction, while the applied current was negative (CCW) in both cases.

friction torque is higher if the profile is driven counter-clockwise (90° to -90°) than when it is driven clockwise (-90° to 90°). This behaviour was observed for both devices MIKE #6 and MIKE #3 and is visualized in the following plot (recordings of MIKE #3). This is why it is enough to run the profile only in the direction with more friction as described earlier.

5 Viscous Friction

The goal of this test is to determine the viscous friction depending on the velocity, meaning the torque that is needed to maintain a constant velocity. This is also needed to compare the transparency of the different devices. In order to calculate the viscous friction we have to use the following equation:

$$Inertia * \ddot{\theta} = \tau_{motor} - \tau_{visc} = k_t * i_{meas} - \tau_{visc} \quad (1)$$

with k_t being the torque constant and i_{meas} being the motor current. We only investigate steady states after a constant velocity is reached and the acceleration is zero. Therefore we can calculate the viscous friction as follows:

$$\tau_{visc} = k_t * i_{meas} \quad (2)$$

5.1 Implementation

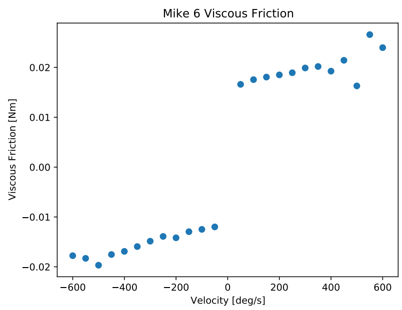
We want to measure the torque produced by the motor while a constant velocity is maintained. Therefore, the robot needs to move with a desired constant velocity, this is especially difficult because our range of motion is relatively small. The desired velocity can be set by setting a desired distance and a desired duration in which the distance has to be overcome. For example if you wish to have 200 $\frac{rad}{s}$ you could set a distance of 200° and a duration of 1s. It is best to have a distance as large as possible or else large velocities will be difficult to maintain. I always took 200° and started at the mechanical stop. The measurement has to be taken separately for each velocity. Also the data analysis can only be done for one velocity at a time, so it is not automatically done.

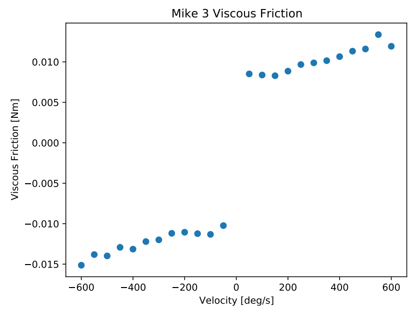
5.2 Data Analysis

For the data analysis, the measured current as well as the velocity are recorded over time and written to a tdms file. As mentioned before, one tdms file is generated for each velocity and they also have to be analysed separately. The two signals can be plotted over time. Afterwards one has to determine a section of the signals where the velocity is as constant as possible, this has to be done visually. One can then calculate the average consumed current over this section and calculate the viscous friction.

5.3 Results

The results for MIKE 3 and MIKE 6 are depicted in the following plots:





It is visible that the viscous friction increases with higher velocities, which was also observed in Monika's ICORR paper and Julian's report. The measurements of the different MIKE versions are close to each other, again MIKE #3 seems to have less friction, which was also observed in the static friction. The obtained results are also similar to Monika's ICORR paper and Marc's final report. The observed values were $< 0.04Nm$ (Monika's ICORR paper [1]) and $< 0.02Nm$ (Marc's Report [3]).

6 Transparency Plane

If a user is moving his finger, while it is strapped in the device, he/she should perceive the device as little as possible. This is called *transparency*. In this test we are interested to measure the torque, that has to be exerted by the user, in order to move the end-effector at a certain velocity or acceleration. This allows for a calculation of apparent inertia and damping according to the following equation:

$$\tau_{user} = i * \dot{\theta} + b * \ddot{\theta} \tag{3}$$

We want these values to be as small as possible.

6.1 Implementation

In this test, we want to measure the torque exerted by the user, therefore the system has to remain unactuated during the process. It is a very simple program that just records all measured data by the system. The velocity is measured once from the tacho and once as the derivation of the encoder signal.

To simulate a real usage of the device, the test-person should attach the index-finger and also use the normal handle. After starting the recording of the signals, the user has to move the finger in sine-form with different frequencies, in order to cover a broad spectrum. Alternatively, one can also move the end-effector by hand, which results in nicer, more even plots.

6.2 Data Analysis

For data analysis we use the encoder derivative for the velocity and adjust the force signal with a leverarm of 7cm to determine the user torque. The same strategy was also used by Marc in his master's thesis [3]. Afterwards, the recorded signals are filtered with a butterworth low-pass filter with cutoff frequency 20Hz. Based on the filtered velocity signal, the acceleration is calculated by dividing the time-steps and velocity steps between two consecutive measurements.

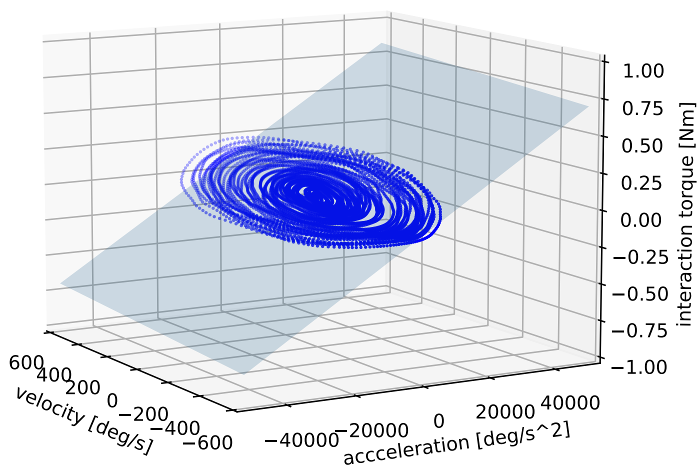
After the data preprocessing, the measured user torques can be plotted over velocity and acceleration in a 3D plot as introduced in [4]. These points are then approximated with a 3D-Plane. The plane is calculated with a minimize function, which minimizes the z-error between the plane and the measured torques. The plane has the following form:

$$\tau_{user} = i_{app} * \ddot{\theta} + b_{app} * \dot{\theta} \tag{4}$$

Therefore, the apparent inertia (i_{app}) and the apparent damping (b_{app}) can be directly read from the plane. Additionally, the average z-residual (error between measured points and plane) and the maximum z-residual are calculated to make a statement about the accuracy of the estimated inertia and damping.

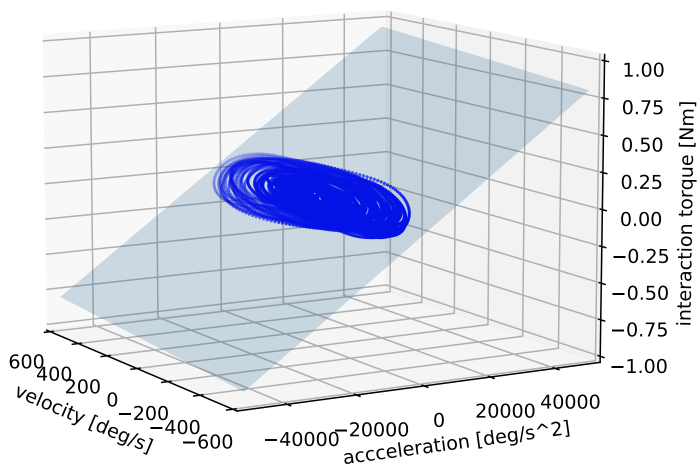
6.3 Results

In the following, the transparency planes for MIKE #6 and MIKE #3 are shown. The velocity and acceleration range was chosen as Marc did in his report, in order to have a comparable operating range for all MIKE versions. In addition to the plots, the apparent inertia (i_{app}), the apparent damping (b_{app}) as well as the average (r_{avg}) and the maximum (r_{max}) z-residual are indicated next to the plot. Both plots were recorded by moving the end-effector by hand and not with the finger attached. They show similar results to what was presented in Marc's report ($i_{app} = 0.0164 \frac{gm^2}{deg}$ and $b_{app} = 0.242 \frac{mNm}{deg/s}$).



$$\begin{aligned} i_{app} &= 0.0149 \ \frac{gm^2}{deg} \\ b_{app} &= 0.1032 \ \frac{mNm}{deg/s} \\ r_{avg} &= 0.011 \text{ Nm} \\ r_{max} &= 0.0489 \text{ Nm} \end{aligned}$$

Figure 5: Transparency Plane of ETH MIKE #6



$$
\begin{aligned}
i_{app} &= 0.0171\ \frac{gm^2}{deg} \\
b_{app} &= 0.113\ \frac{mNm}{deg/s} \\
r_{avg} &= 0.0125\ \text{Nm} \\
r_{max} &= 0.0716\ \text{Nm}
\end{aligned}
$$

Figure 6: Transparency Plane of ETH MIKE #3

7 KB-Plot/Virtual Wall Rendering

To evaluate the capability of the system to render virtual dynamics, a KB-plot is created. In our case we investigate the rendering of a virtual wall. To do so, we search for KB-pairs as large as possible with the system still being stable. In case that the wall is placed at an angle $\theta = 0$, the motor torque is calculated as follows:

$$\tau = K * \Delta\theta - B * \dot{\theta} \tag{5}$$

With the results, we can also measure how well the wall is rendered by measuring the created force on the user.

7.1 Implementation

As mentioned before, we want to find KB-pairs with a stable wall rendering. The program is simple, one can set the values for K, B and the wall position manually. After starting the simulation, the tester has to move the end-effector into the wall with his/her own hand. One can then move it around a little to test if one can find any instability. Even if a small instability is found, the values have to be reduced!

Attention! The program only generates counter-clockwise torques and the wall always goes from you angle to clockwise direction. Therefore you have to make sure that the end-effector is on the left side of the wall when you start the simulation.

In the program one can also choose the velocity estimation method. There are three possibilities:

1. Tacho (standard)
2. Encoder Derivation (Finite Differentiation Method (FDM))
3. First Order Adaptive Windowing (FOAW) with window size of 20 samples and d=0.05

They should all be used once in order to compare the different methods.

After the measurement for the KB-Plot is done, one can apply the maximum KB-pair found and measure the force in the force sensor by pushing the end-effector into the wall.

7.2 Data Analysis

In this case the data analysis is very simple. No data has to be recorded, since the stable KB-pairs have to be determined manually. The KB-pairs have to be written to a separate excel file, with which they can be read into python and can then be plotted.

For the virtual wall rendering we have to measure the force (force sensor) as well as the position. We can then plot the force over the the position in order to evaluate the rendering.

7.3 Results

In the following the results are presented. There is always one curve for FDM, tacho and FOAW respectively. As is visible, the results are similar. However, the values are smaller than the ones in Monika's ICORR paper ($K_{max} = 1.3\frac{Nm}{deg}, B_{max} = 0.0012\frac{Nm*s}{deg}$). This could be because the test is not very objective and a lot depends on the feeling of the testing person.

The measured values for MIKE #3 and MIKE #6 are comparable to each other. We can see that using the tacho generates the highest possible values for K, while FDM allows for the largest damping.

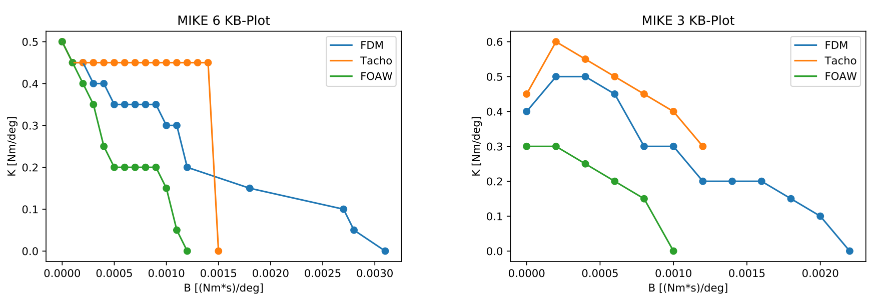
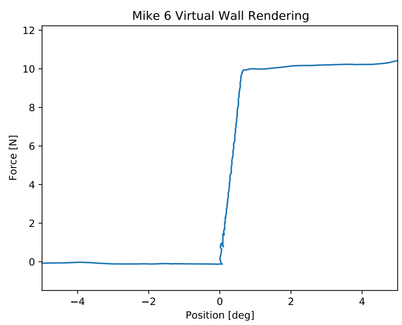


Figure 7: KB-plots of MIKE #6 and MIKE #3.

In the following the force exerted by the wall is visualized. MIKE #3 generates higher forces since its motor can generate more torque. However, if pushed too much, MIKE #3 will collapse and cannot hold the maximum force while MIKE #6 can continuously hold its maximum force. This can either be because of hardware or because of a safety feature implemented in the software, I am not sure about that. As in this test, the end-effector was moved by hand, the influence of B (damping) is small, because the velocity is low. Therefore, the important parameter is K (stiffness).



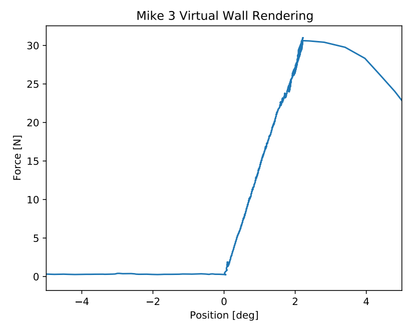


Figure 8: Left: MIKE #6, with K=0.45 and B=0.0014; Right: MIKE #3, with K=0.6 and B=0.0003

8 Position Bandwidth

The goal of this test is to find the maximum frequency with which the end-effector can follow a sine trajectory for a specific controller. In order to evaluate this, we create a *Bode-Plot* with the position setpoint as input and the measured position (of the end-effector) as output. The position bandwidth is defined as the frequency with which the amplitude of the output position is reduced by 3dB compared to the amplitude of the setpoint signal.

8.1 Implementation

We want to measure a sine wave of different frequencies. At the moment, there has to be a separate measurement for each frequency. In the program, one can set the amplitude, the frequency as well as the number of cycles one wants. For my tests I took 5° as amplitude.

In general, one should use one version of the controller gains for all frequencies, since the bode plot strongly relies on the controller. For my measurements I tried to use the standard sine-params, which worked fine for MIKE #6 but not for MIKE #3 (more info in results section).

8.2 Data Analysis

The data analysis is more difficult than in other cases. The basic working principle is the following: For each frequency, there is one separate measurement from which the magnitude as well as the phase can be determined. In order to do so, the measured position signal is filtered with a butterworth filter. Afterwards the peaks and their time points of the measured signal as well as the setpoint are extracted. The magnitude can then be calculated by dividing the measured peaks through the setpoint peaks. The phase can be determined from the time points according to the following formula:

$$\Delta\phi = f * \Delta t * 360^\circ \tag{6}$$

	$\Delta\phi$	phase [°]
with	f	frequency [Hz]
	Δt	time-difference between setpoint peak and measured peak [s]

However, there are several problems which can occur while reading out the peaks. The most important points which I observed and how I solved them is listed below:

1. In the measured signal not all measured peaks represent the amplitude, but can also occur due to control issues or measurement errors.

   solution: When reading the peaks, I set a certain minimal distance in time between two peaks ($\frac{1}{3*f}$). If the distance between two peaks is smaller, they are neglected.

2. Once I have observed that all the measured minimal-peaks of the setpoint signal are around 0

   solution: I set the minimal peak-height of the setpoint signal to be 4, this means peaks smaller than 4 are neglected.

3. In the beginning of the trajectory following there is an overshoot for $f > 0.2$, resulting in the first measured peak being way too high

   solution: deleting the first peak measurement for frequencies larger than 0.2

4. All measurements in which the setpoint is 0 are neglected. For $f \geq 4$, the last setpoint peak is therefore not reached in the measured signal due to phase shift.

   solution: delete the last minimal setpoint peak

5. For higher frequencies ($f \geq 8$) the system need some time to reach a transient swinging state and the first peak measurements are not reliable.

   solution: For these frequencies, only the last 20 peaks are used.

6. The minimal and maximal peaks are stored in separate variables. For lower frequencies I have observed that in the measured peaks minimal peaks can be in the maximal peaks and vice versa.

   solution: All maximal peaks smaller than zero and all minimal peaks larger than 0 are deleted.

7. I have observed that sometimes a peak is measured before the setpoint peak is reached. This can result in more measured peaks than setpoint peaks and a positive phase (impossible).

   solution: If a measured peak has an earlier time point than the associated setpoint peak, it is deleted.

8. At last I have observed that the phase can increase again for higher frequencies (which is not possible) but can happen due to measurement errors or an insufficient control performance.

   solution: Before plotting, if a phase of a frequency is closer to zero than the phase of the frequency measured before, it is set to the same phase as measured before (only for $f \geq 10$)

8.3 Results

The results are presented in the following. It is important to mention here is that the position bandwidth depends mostly on the controller and the chosen gains. Therefore the interpretation of the results is dependant of the goal one wants to examine. If one wants to investigate the performance of one specific controller, it is sufficient to run the test with this controller and to calculate the bandwidth. One could also look for the highest possible bandwidth by comparing different controllers (different sets of controller gains).

I investigated only the performance with the controller gains defined for the sine-params during the initial tuning of the ETH MIKE's. For MIKE #6 I could use the standard PID-gains and calculated a bandwidth of 6 Hz. For MIKE #3 I had to change the params a little, as I was not able to run the test stably using the standard sine-params[1]. The resulting bandwidth was 10.5 Hz. Compared to Monika's ICORR paper (result: 12Hz), MIKE #6 is far away. However, then only a PD controller was used and not the real controller running on the MIKE. Maybe one could also improve the bandwidth of the other MIKE versions by a PD controller.

The resulting bode plots are displayed in the following:

[1]The sine params of MIKE #3 were adjusted after this experiment

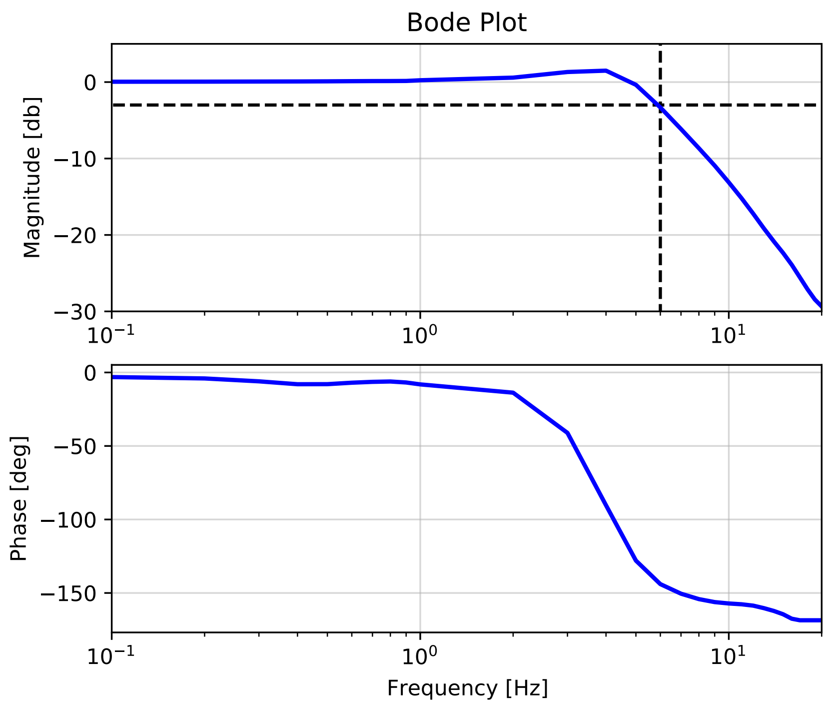


Figure 9: MIKE #6; Kp=1.2; Ki=6; Kd=0.008; Resulting Bandwitdh=6Hz

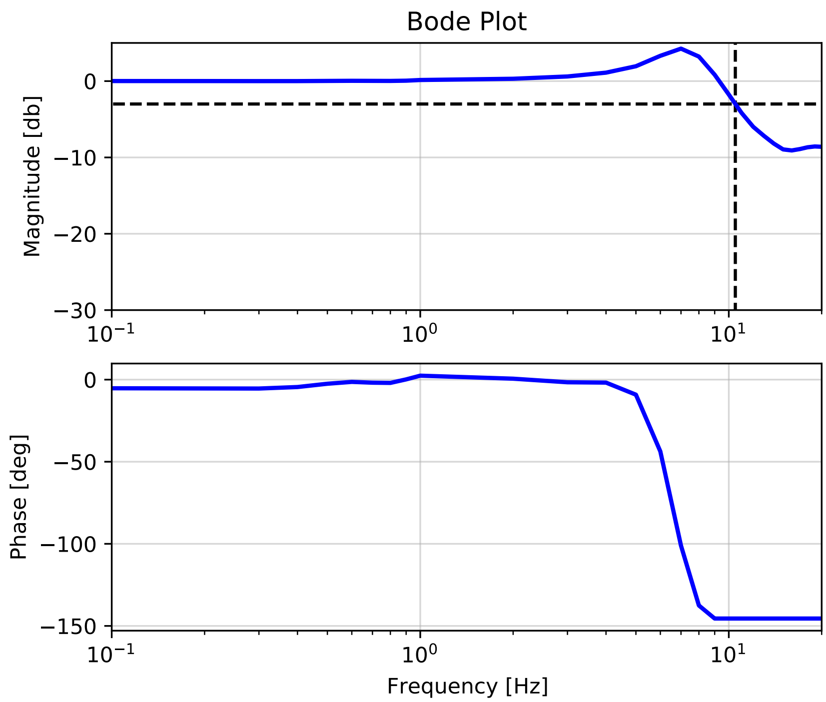


Figure 10: MIKE #3; Kp=0.5; Ki=5; Kd=0.0006; Resulting Bandwitdh=10.5Hz

9 Conclusion

The main results showed, that there are no severe differences in performance between the two versions MIKE #3 and MIKE #6, despite the new electronics. There are some points in which MIKE #6 shows a weaker performance, for example maximum acceleration, maximum torque or the force generation in the virtual wall rendering. This was to be expected due to the new electronics. However, the performance is enough for its application. The device is not used at its torque or acceleration limits. The force generation (virtual wall) is smaller for MIKE #6, but the generation of MIKE #3 is also not sufficient to render a wall, this is why we use the hard-stop for the force test.

All of the scripts to measure and analyse the data have been written in a way, that allows for future usage. Therefore, also other devices can be evaluated.

References

[1] M. Zbytniewska, M. Rinderknecht, O. Lambercy, M. Barnobi, J. Raats, I. Lamers, P. Feys, J. Liepert, and R. Gassert, “Design and characterization of a robotidevice for the assessment of hand proprioceptive motor, and sensorimotor impairments,” in *2019 IEEE 16th International Conferencon Rehabilitation Robotics (ICORR)*, 2019, pp. 441–446.

[2] J. Moosmann, “Electronics redesign for the robotics assessment therapy platform eth mike,” Semester-Thesis, ETH Zürich, 2021.

[3] M. Siegenthaler, “Interaction control strategies and sensorimotor task design for the assessment robot eth mike,” Master’s Thesis, ETH Zürich, 2020.

[4] J.-C. Metzger, O. Lambercy, and R. Gassert, “Performance comparison of interaction control strategies on a hand rehabilitation robot,” in *2015 IEEE International ConferenceonRehabilitation Robotics (ICORR)*, 2015, pp. 846–851.