MF2007 - Workshop A

Adam Lang 861110-3956
Gabriel Andersson Santiago 910706-4538
Andreas Fröderberg 880730-7577

April 7, 2016

Parameter identification

This section covers the methods behind finding the parameters to enter into the model to get the biggest similarity to the physical plant

Level 1

To identify the parameters of the DC motor, firstly an idealized model is studied theoretically to understand which physical parameters that affect the performance of the motor. In the Laplace frequency domain, a model of the motor without inductance can be expressed as

$$X = \frac{k(1+\frac{1}{Rd})}{s\frac{J_t}{\frac{k^2}{R}+d}+1}U, \tag{1}$$

where X is the rotational speed, U is the input voltage, k, R and d are the electric motor constant, internal resistance and the viscous friction. The variable J_t represents the total inertia of the motor and the load. With the load inertia after a gear with gear ratio n, the total inertia is

$$J_t = J + \frac{J_{load}}{n^2}. \tag{2}$$

Here, the load inertia and the inertia of the motor are separated and the motor inertia from the datasheet is presumed to be correct, leaving only the load inertia as an unknown. Now the relations between the motor parameters and the time performance can be seen. The time constant is given by the term

$$\frac{J_t}{\frac{k^2}{R}+d} \tag{3}$$

and as such contains all the parameters of the motor. According to the final value theorem, the steady state value from a step input is given by setting $s = 0$ and therefore the steady state gain is given by

$$k(1+\frac{1}{Rd}). \tag{4}$$

This contains only one unknown parameter. Therefore, to estimate the viscous friction parameter d, it is suitable to examine and match the steady state rotational speed of the real system and the model. Since this model contains no static friction which is probably present in the real system, the test is conducted at maximum permissable input voltage to minimize the disturbances in the test data. Since the desired value does not depend on frequency, only a square wave input is used to simulate steps in both directions.

Now that the friction parameter is presumed to be correct and fixed, the time constant in Equation (3) only has one unknown, the load inertia. This is estimated from taking the time constant from step inputs, simulated by a square wave signal. Again, to minimize model disturbances caused by the static

friction present in the real system, the square wave is run at a full 24 V.
The tuning can be seen in Figure 1 and Figure 2. In Figure 1, the model been tuned to take the viscous friction into account.

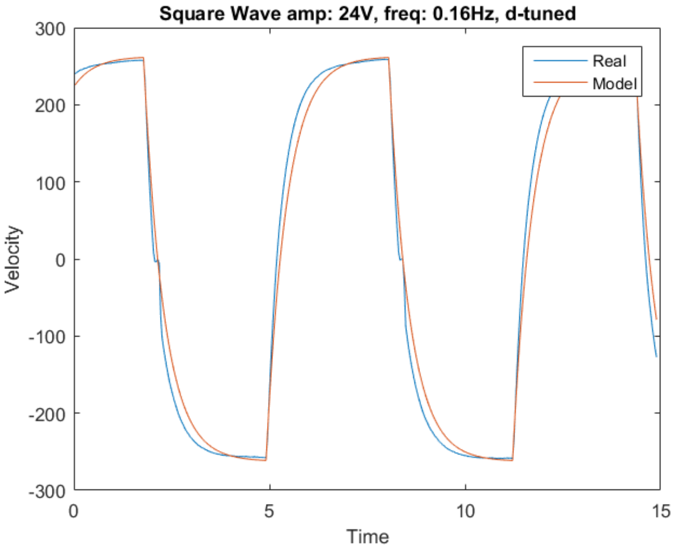


Figure 1: Velocity output from the motor and the model when tuned t for viscous friction.

After the viscous friction is modeled, values on the inertia of the system is adjusted to fit the real motor. The end result after the parameters have been tuned can be seen in Figure 2.

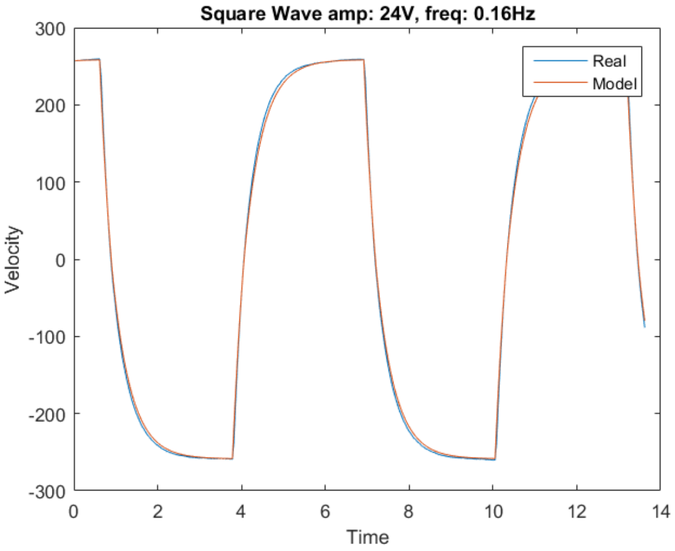


Figure 2: Velocity output from the motor and the model when tuned for both viscous friction and inertia.

We can clearly see that the model is a much better approximation of the real motor after the model parameters have been tuned to fit the real motor. Even though they are not perfect, will the model give a good sense of the real motor.

Level 2

As could be seen in the earlier friction model, it did not incorporate the effects of static friction which led to inaccuracy in the model when varying the rotational speed. A model which captures the real behaviour of the system is used here, called the Karnop friction model. In addition to the linear friction, the static friction is taken into account by adding a stick-slip zone where the friction tourque is equal to the applied torque. The model is given by the equation

$$M_f = d\dot{\phi} + F_c sgn(\dot{\phi}), \tag{5}$$

with the friction torque M_f and the static friction F_c. This friction is the maximum static friction torque that can be applied in the stick slip zone. There is now one more design parameter that needs to be tuned, F_c. The linear friction is still most prominent when at high velocities and the static friction is the strongest when at low velocities. The inertia is decided solely from the rise time and should not differ much from the previous model. Firstly, the step response to a 5 V square wave is examined to determine the static friction. When the static friction is set, the response to a square wave at 24 V is examined and d is adjusted to fit the static friction. A sine of low amplitude is then run to test the behaviour around the stick slip area, i.e. at low speeds.

In Figure 3 the model have been tuned for static friction.

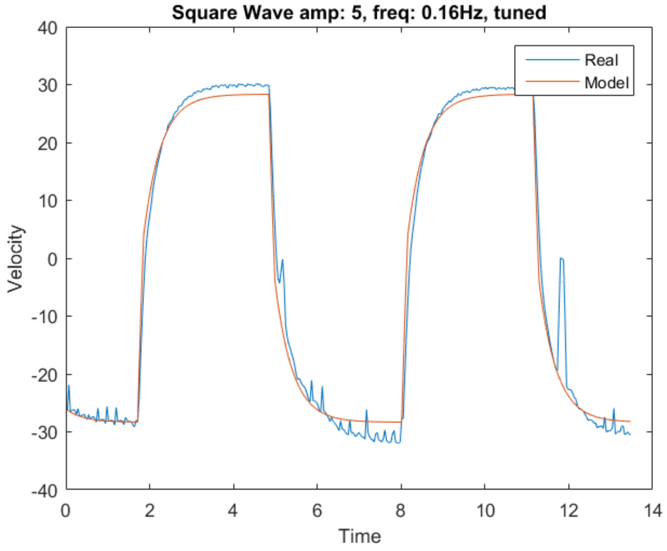


Figure 3: Velocity response of motor and model at 5V amplitude when tuned for static friction.

In Figure 4 the amplitude has been raised to 24V and the model also have been tuned for static friction.

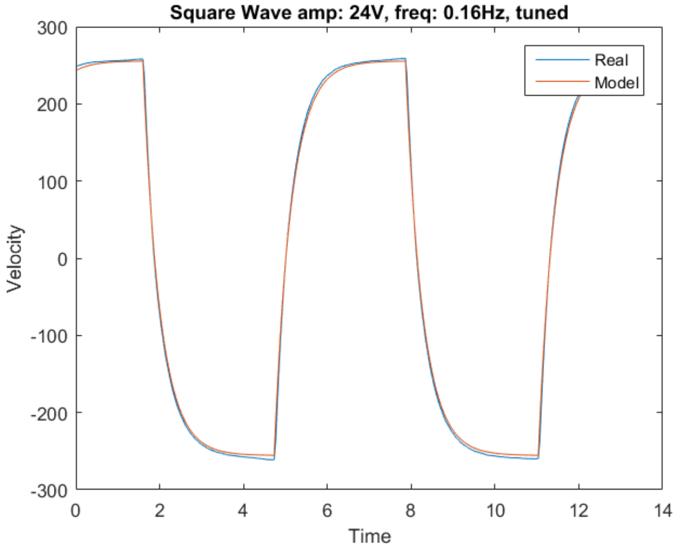


Figure 4: Velocity response of motor and model at 24V amplitude when tuned for both static and dynamic friction.

After the model have been tuned, is the model compared to the motor when the input is a 5V sine wave which can be seen in Figure 5.

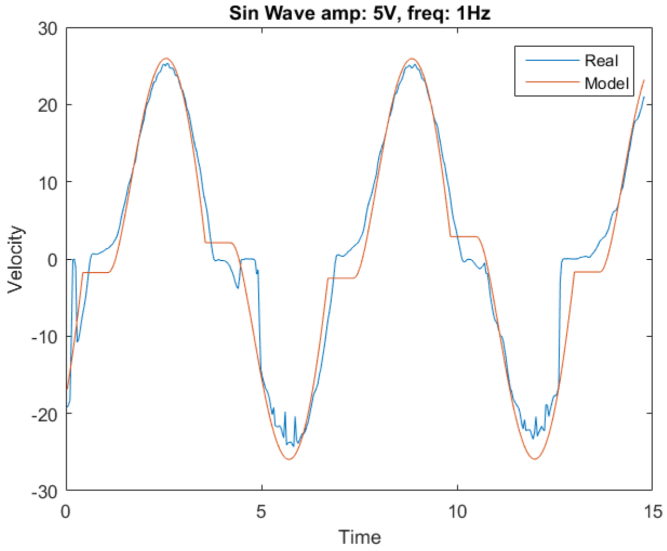


Figure 5: Velocity response of a sine wave at 5V amplitude of the motor and tuned model

The lack of a good model for the static friction made the motor in level 1 be much less accurate when changing the amplitude. Therefor it had to retuned when a change was made. In level 2 when introducing the static friction this has now been improved and the model will now give a much better approximation when changing amplitude.

Velocity control

The parameters generated in the previous section are used to design the controller for the motor. For the design, a simplified model without motor inductance is used in MATLAB to place the poles and estimate the behaviour of the system. Furthermore, a non-linear model with Karnop friction is used in SIMULINK to simulate the behaviour of the system with the controller implemented. Finally, the design is run on the actual motor for verification of plant model and controller.

Level 1

Controlling the velocity is done with a PI controller. To best control the performance, the controller is designed in discrete time.

The rule of thumb for sample time of a system states that the system should be sampled at 4-10 times the rise time of the plant in open loop. Using the simplified model in MATLAB and the commando `stepinfo`, the rise time is extracted and the sampling time is set to 10 times the rise time.

To get the desired performance of the closed loop system, the poles are placed directly in the discrete plane. First, the plant is converted from continuous to discrete space by Zero Order Hold (ZOH), with the sampling time derived earlier.The new sytem matrices are given by the equations

$$\Phi = e^{AT_s} \tag{6}$$

and

$$\Gamma = \int_0^{T_s} e^{A(t-s)} B \mathrm{ds}. \tag{7}$$

Output feedback is used to create the closed loop system. The model of the plant is given in (1) and is here rewritten as

$$\frac{B}{A} = \frac{a}{s+b} \tag{8}$$

,to simplify the equations. Firstly, the feedback polynomial $\frac{S}{R}$ is derived. This is given by

$$\frac{S}{R} = \frac{s_1 z + s_0}{z - 1}. \tag{9}$$

The system is then closed, yielding the pole polynomial

$$AR + BS = z^2 + (as_1 + b - 1)z + as_0 - b. \tag{10}$$

The system is of order one so one closed loop pole given by

$$A_m = z - p_1 \tag{11}$$

is placed with an additional pole given by the observer polynomial

$$A_o = z - p_2 \tag{12}$$

This yields the Diophantine equation

$$A_m A_o = AR + BS. \tag{13}$$

Solving for (13) the unknown controller parameters s_0 and s_1 are found to be

$$s_0 = \frac{p1p2 + b}{a} \tag{14}$$

and

$$s_1 = -\frac{b + p_1 + p_2 - 1}{a}. \tag{15}$$

The T polynomial is given by

$$T = t_0 A_o \tag{16}$$

where t_0 is a static value so that the steady state gain of the closed loop system is 1. Using the final-value theorem for discrete systems, one yields

$$t0 = \left(\frac{B}{A_m}\right)^{-1} \tag{17}$$

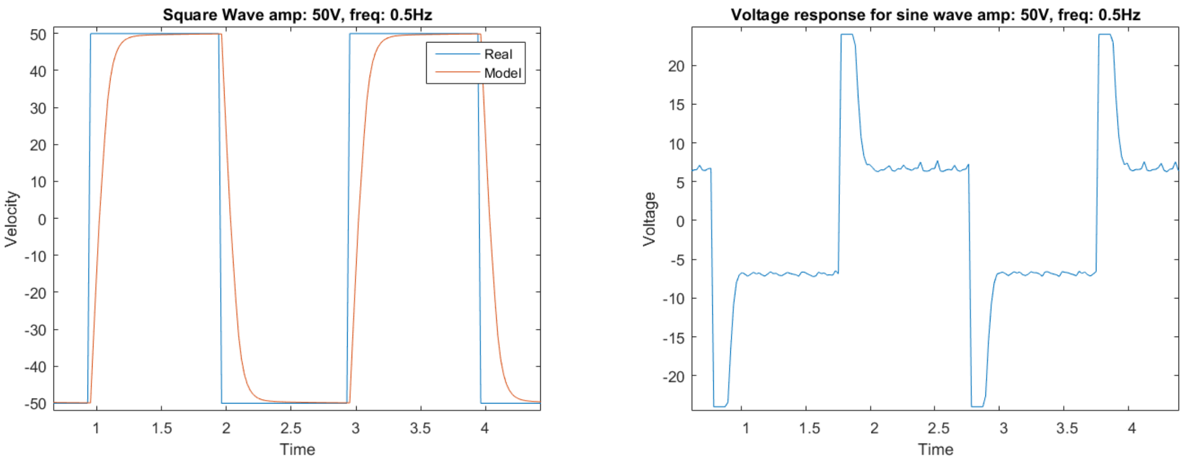
evaluated in $z = 1$. The control law is then found as,

$$R(z)u(z) = T(z)r(z) - S(z)y(z) \tag{18}$$

and in numerical form,

$$(z - 1)u(z) = t_0(z - p2)r(z) - (s_1 z + s_0)y(z) \tag{19}$$

In Figure 6a is the response from a square wave with 50 rad/s and 0.5Hz shown. In Figure 6b the voltage response for the same input is plotted.



(a) Velocity response

(b) Voltage response

Figure 6: Response for square wave 50 rad/s and 0.5Hz

It can also be seen in Figure 7 the plot that the steady state error is less than the specified 0.5 rad/s. And that there is no overshoot.

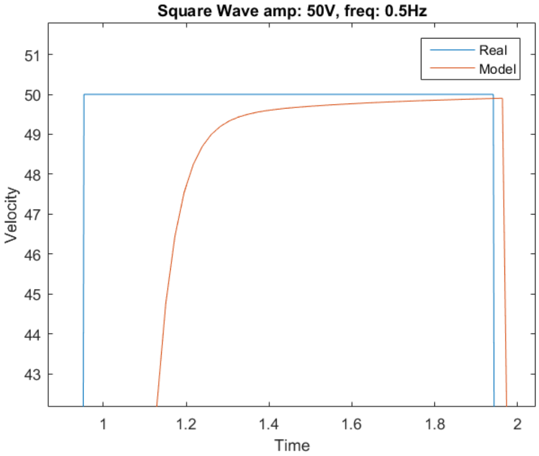
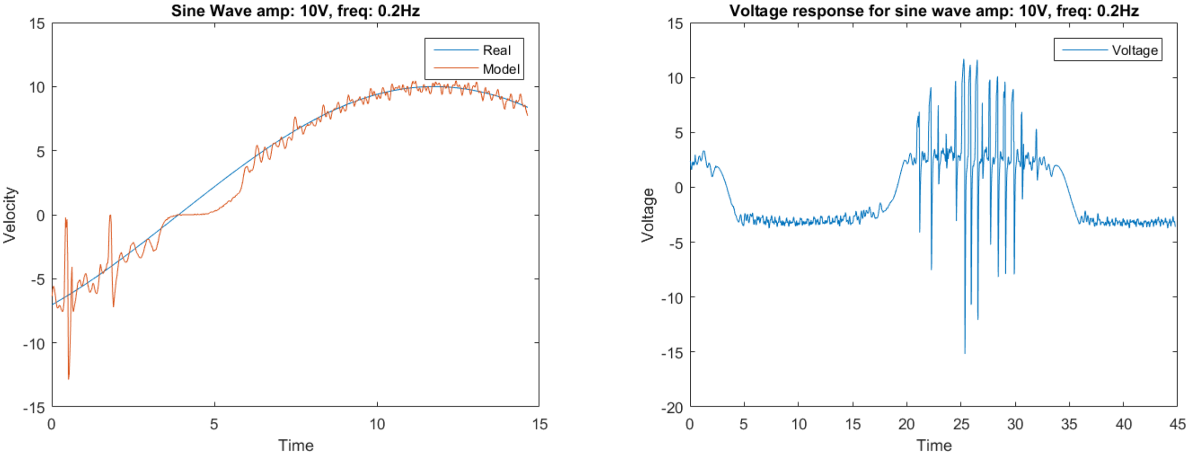


Figure 7: Zoomed plot of the Velocity response

In Figure 8a is the response from input of $\varphi_{ref} = 10sin(2\phi0.2t)$ shown.

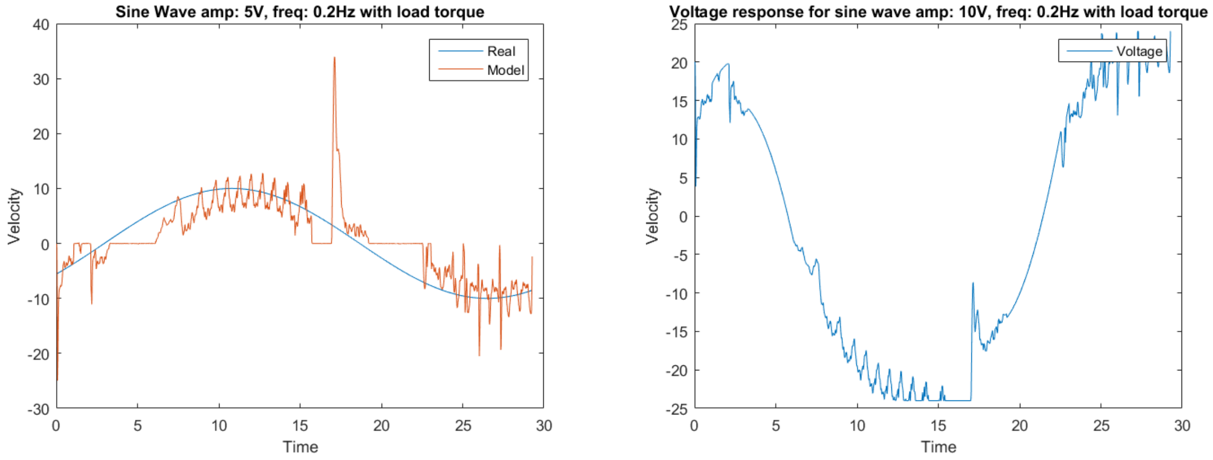


(a) Velocity response

(b) Voltage response

Figure 8: Response for sine wave 10 rad/s and 0.2Hz

In Figure 9a is the response from the reference input of $\varphi_{ref} = 10sin(2\pi0.2t)$ shown. In this plot is the rotor loaded with a torque by holding it.



(a) Velocity response

(b) Voltage response

Figure 9: Response for sine wave 10 rad/s and 0.2Hz

Position Control

The control law for the position controller is derived similar to the velocity controller with the resulting control law in Equation 18 as,

$$((z-1)(z+0.677))u(z) = (71.4z^2 - 105z + 39.9)r(z) - (6.30z^2)y(z) \tag{20}$$

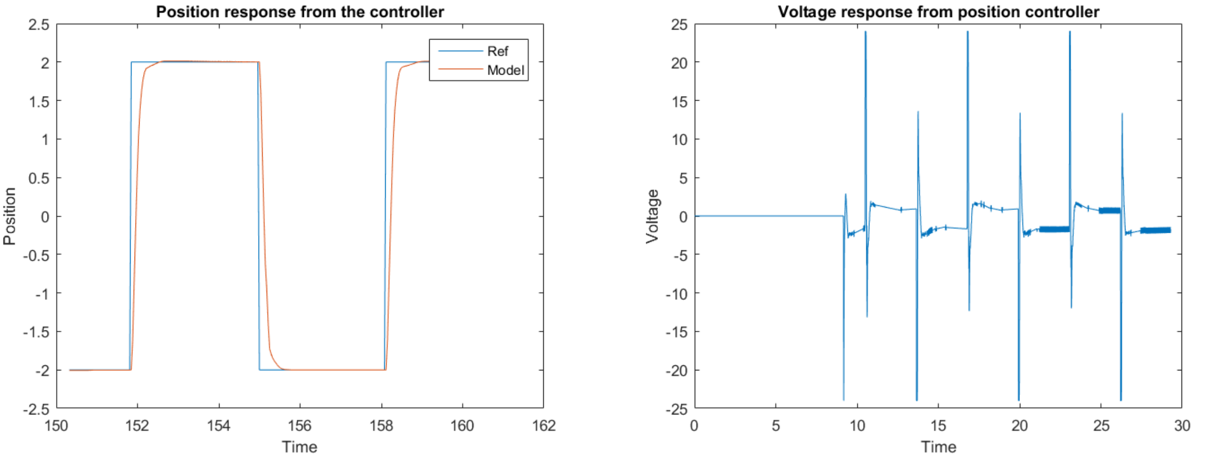
Where the observer polynomial and the closed loop polynomial is,

$$A_m = (z-0.6)(z-0.5)$$
$$A_o = z^2,$$

with sampling time $T_s = 0.0362$.

Level 1

In Figure 10a the response from the controller can be seen and in Figure 10b the voltage response is plotted. The specifications is met as can be seen in the Figure 10a



(a) Position response

(b) Voltage response

Figure 10: Response from position controller

Level 2

The controller used in level 1 is designed directly in the discrete plane. To compare admissable sampling times, a continuous time controller is designed and then discretized with Tustins approximation.

The design steps are the same as the output feedback design in the velocity controller case and are therefore not stated again. For the feedback controller, a continuous PID with low pass filter is chosen,

$$\frac{S}{R} = \frac{s_2 s^2 + s_1 s + s_0}{s(s{+}_0)}. \tag{21}$$

The Diophantine equation yields the parameter values

$$r_0 = 2\omega_3\zeta - b + \omega_1 + \omega_2, \tag{22}$$

$$s_0 = \frac{\omega_1\omega_2\omega_3^2}{a}, \tag{23}$$

$$s_1 = \frac{\omega_3(2\omega_1\omega_2\zeta + \omega_1\omega_3 + \omega_2\omega_3)}{a}, \tag{24}$$

and

$$s_2 = \frac{1}{a}(-2b\omega_3\zeta + 2\omega_1\omega_3\zeta + 2\omega_2\omega_3\zeta + b^2 - b\omega_2\omega_1\omega_2\omega_3^2), \tag{25}$$

where

$$A_m A_o = (s + \omega_1)(s + \omega_2)(s^2 + 2\omega_3\zeta s + \omega_3^2). \tag{26}$$

The feedback polynomial becomes

$$\frac{T}{R} = \frac{t_0(s^2 + 2\omega_3\zeta s + \omega_3^2)}{s(s + r_0)}. \tag{27}$$

Designing the controller is done by iteration and testing. Firstly, the speed and performance of the system is set by moving the A_m polynomial so that a good mix of model disturbance and sensor noice rejection is found. The closed loop poles in A_o are then placed to get the desired performance. The sampling rate is first set by the thumbrule that it should be 10-30 times faster than the fastest pole in the system, then it is tuned to see what the maximum reachable sampling time is while maintaining the perfomance within specifications. The resultsing control law, sampling time and perfomance is displayed below.