SOLUTIONS - EXERCISE 3

Control systems and modelling

1. PID CONTROLLER

Your answer could contain following parts:

- PID controller, Proportional-integral-derivative controller is a feedback control loop mechanism, which is commonly used in pretty much any kind of applications. It is used to minimize the error between a desired output value of the system, i.e. a reference value, and a measured output value of the system (speed, displacement, temperature etc.).
- The mathematical form for the controller command signal is:

$$C = K_p e(t) + K_d \frac{d}{dt} e(t) + K_i \int e(t) dt, \qquad (1)$$

where K_p is the proportional gain, K_d is the derivative gain and K_i is the integral gain. e(t) is the error signal: how far the measured value is from the reference.

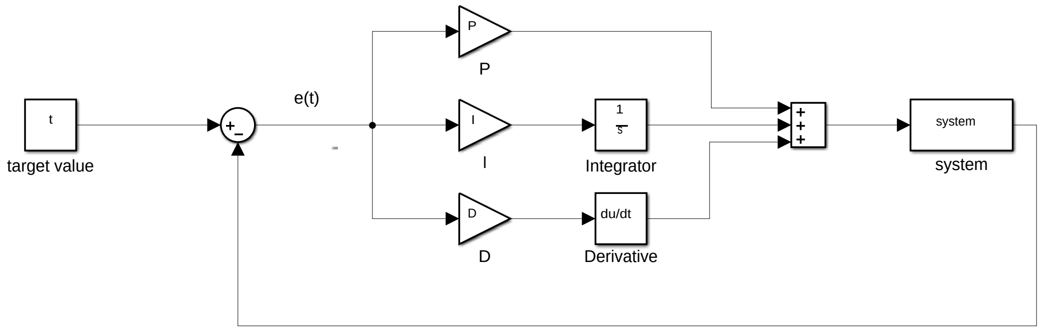

FIGURE 1. PID CONTROLLED SYSTEM

- The proportional control part is easy to understand: proportional control part is proportional to the error signal. The larger the error is, the larger the control signal. Proportional part causes fast system response, but may lead to overshooting and oscillation due to delays in the controlled system, measurement or the controller.
- The integral part removes steady state errors. The integrator sums up the error signal over time. The longer the error is present, the larger the correcting signal is.
- The derivative part responds to changes in error signal. This may help to prevent oscillation and overshoot by predicting the future errors based on current signal change. This may also cause problems if the feedback signal has noise (electromagnetic interference) or stepwise changes (digital signals).
- PID controller is very easy to implement, because it doesn't demand any knowledge of the system.
- Tuning the different gains can be done manually (testing, simulating), with certain mathematical methods or with tuning software (Matlab has also one).

2. DC SERVO MOTOR

Find the solution model and script in MyCourses.

The DC motor is represented by a subsystem which has only the voltage as input and angular velocity as output.

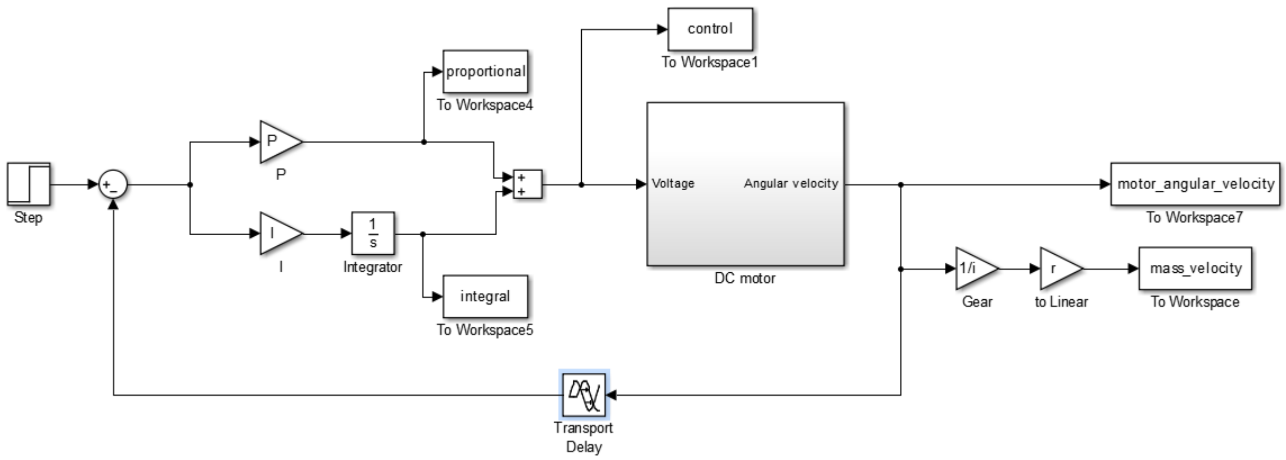

FIGURE 2. SOLUTION MODEL

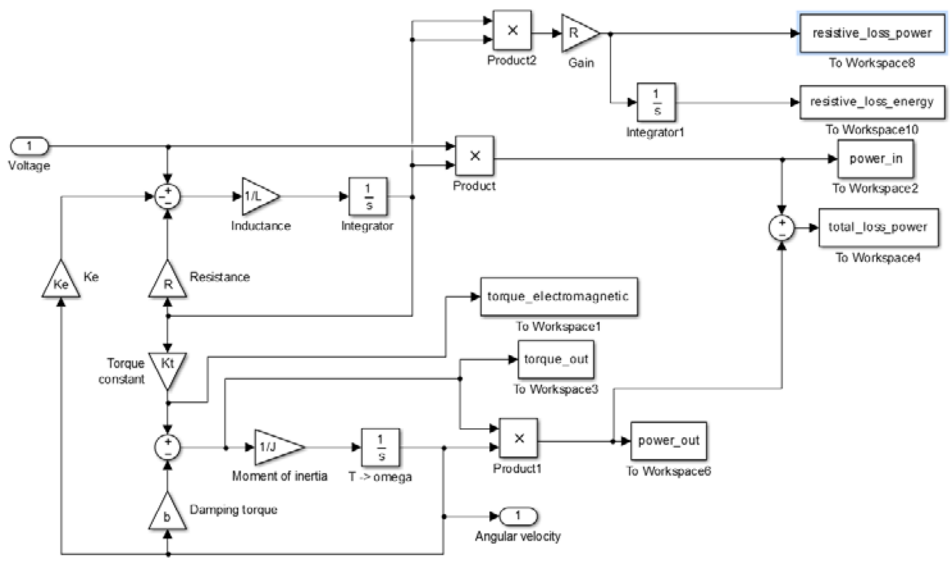

FIGURE 3. DC MOTOR SUBSYSTEM

Q1. PID controller works as described in task 1. The output of the controlled system, in this case the angular velocity of the motor, is subtracted from the reference signal i.e. the desired output of the system, in this case the target angular velocity, and the error signal is fed to the PI controller. The transport delay representing the measurement delays is initially zero. Step block outputs the target angular velocity. The proportional and integral control parts are calculated from the error signal and then summed to form the total controller output, which in this case represents the input voltage to the motor.

Q2. With controller gains P=0.2 and I=1.4 the following results are achieved:

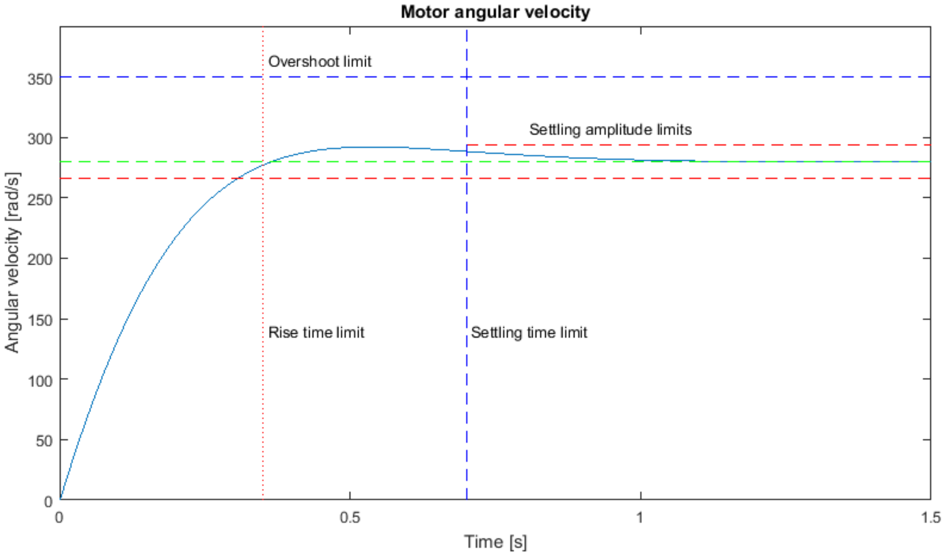

FIGURE 4. VELOCITY

With these parameters, the step response fulfills all the set requirements. The 0-95 % rise time is approximately 0,3 seconds and the settling time is the same since the overshoot stays below 5 % i.e. the response does not go outside the defined +-5 % limits after the response time. Also the maximum input voltage to the motor stays below 60 volts. There are several combinations of gains which produce response that satisfies all the previously mentioned requirements. These gains can be found by trial and error, which in this case was fairly easy, or by more sophisticated methods.

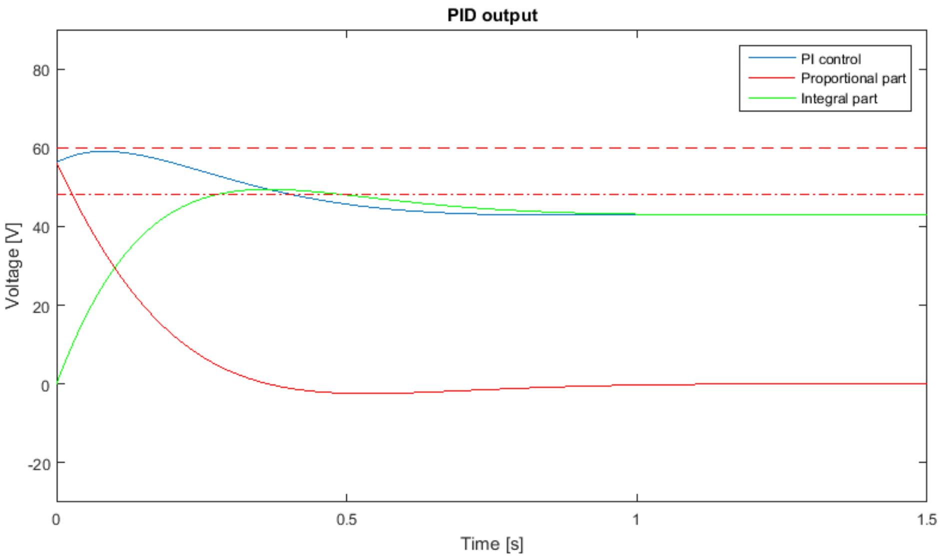

FIGURE 5. PID VOLTAGES

Initially the proportional part is large because of the big difference between target velocity and the output angular velocity. The integral part is however zero, since no time has passed and thus the integrated error has not accumulated. As the error signal grows smaller, so does also the proportional part respectively. In steady state, the error is nearly zero and so is proportional part of the control. The integral sums up the error and ends up approximately to the reading calculated in Exercise 2 (43.1 V) to maintain the desired velocity. Altogether, the control signal is a sum of both signals P and I.

In this simulation, the output of the PI controller is fed directly to the voltage input of the motor model. In real life, the PI controller is an algorithm that runs in for example a microcontroller. The output of the controller is some value, which in turn controls for example the PWM or analog voltage output of the microcontroller. This output is then connected to some type of power electronics that provides the large voltage and current required by the motor.

Q3.

Because the time constant of the simulated motor is tens or hundreds of milliseconds, the 10 millisecond transport delay does not have a large impact on the system's response. However, it slightly increases the overshoot of the angular velocity.

Increasing the delay further makes the overshoot larger. The effect of a larger delay is also clearly visible in the controller output. The proportional gain stays constant during the delay even though the angular velocity of the motor is getting closer to the reference.

The following figure shows the step responses for the motor's angular velocity with 0, 10 and 100 milliseconds of transport delay. With a 100 ms delay the system oscillates a lot more than without the delay. The controller output is also clearly not following the speed of the motor during the first 0.1 seconds.

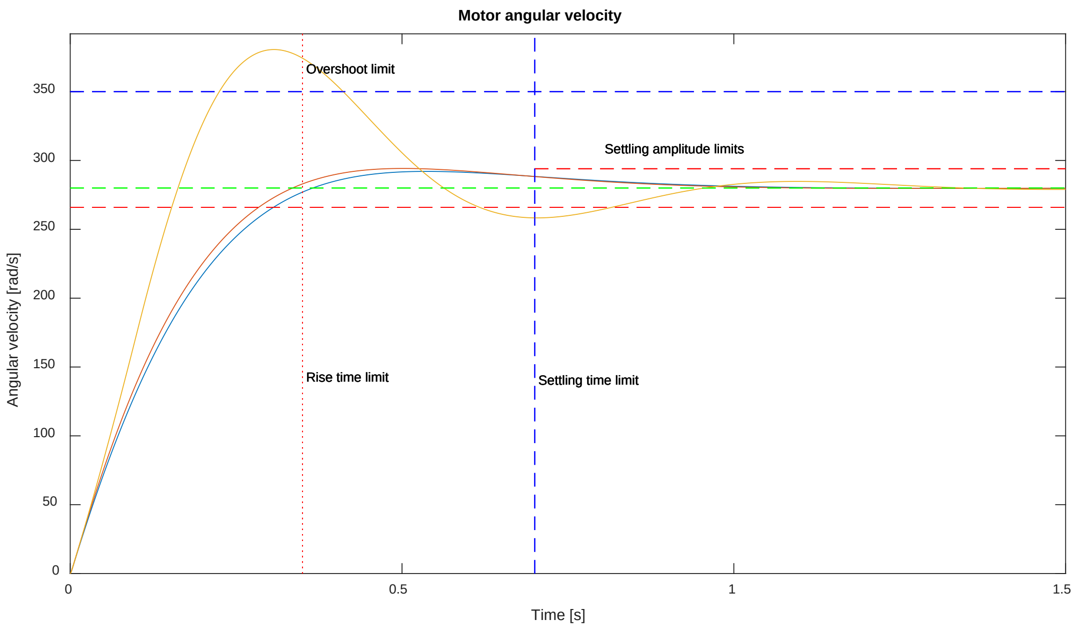

FIGURE 6. STEP RESPONSES WITH TRANSPORT DELAY 0, 10 AND 100 MILLISECONDS. BLUE = 0, RED = 10, YELLOW = 100

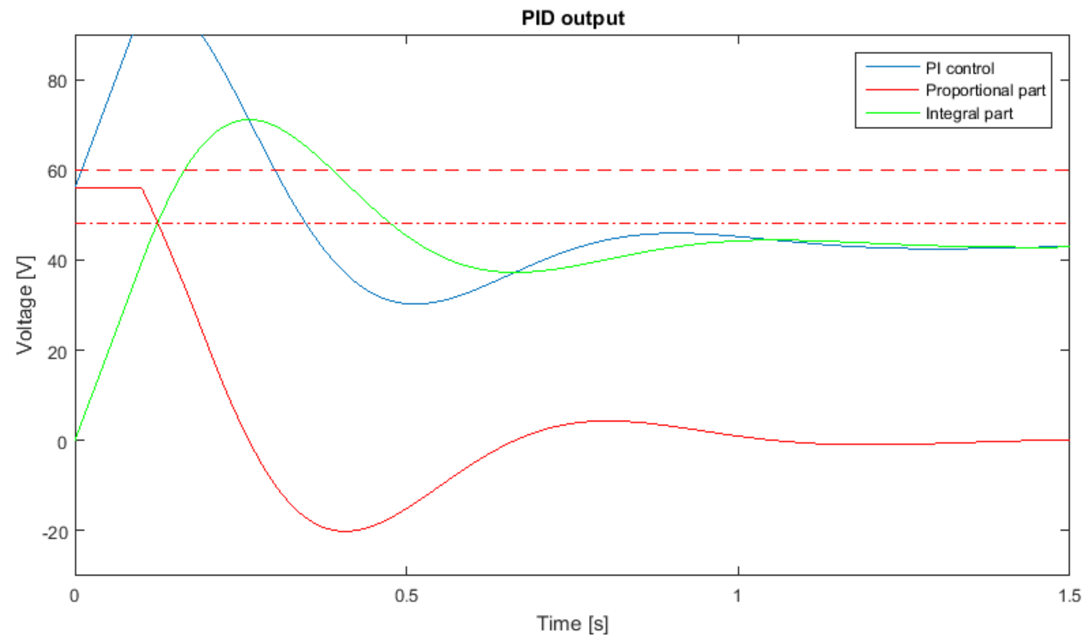

FIGURE 7 CONTROLLER OUTPUT WITH 0.1 SECOND DELAY

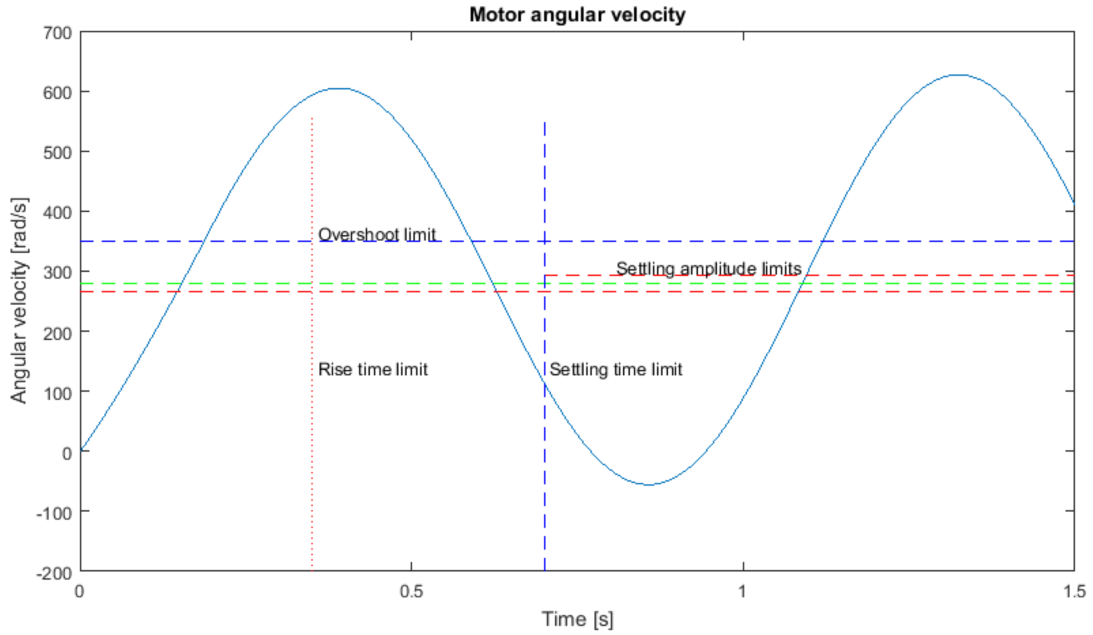

FIGURE 8 UNSTABLE SYSTEM WITH 0.2 SECOND DELAY

When the delay is increased to 0.2 seconds, the system becomes unstable i.e. it oscillates with an increasing amplitude. Therefore, the system never reaches a steady state. The amount of delay required to make the system unstable depends somewhat on the gains you chose. Larger P and I gains usually mean that the system becomes unstable more easily.

Q4. The windings of the coil use the current to generate a magnetic field which generates the torque. The inductance of the motor's coil windings slows down the change in the winding's current, because energy gets stored in the magnetic field. The larger the inductance, the more energy gets stored in the magnetic field with a certain current. Because the torque produced by the motor is proportional to the current, the inductance also slows down the change in the torque. This means in practice that some delay is added to the system.

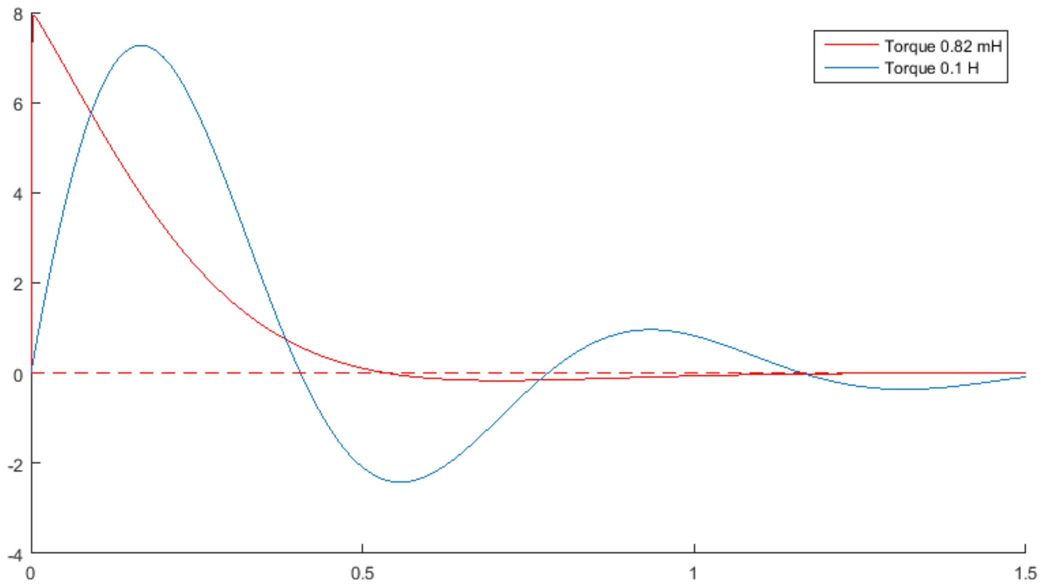

FIGURE 9 TORQUE WITH TWO DIFFERENT INDUCTANCES

Therefore, a higher inductance causes the motor to react more slowly to a change in its control voltage i.e. the motor is not as easy to control as one with a lower inductance. This can be seen in the response of the modelled servo system. With a higher inductance, the system oscillates and overshoots much more, as can be seen in figure 9. In this example the change in inductance is highly exaggerated but a low inductance is an important parameter in servo motors also in real life.

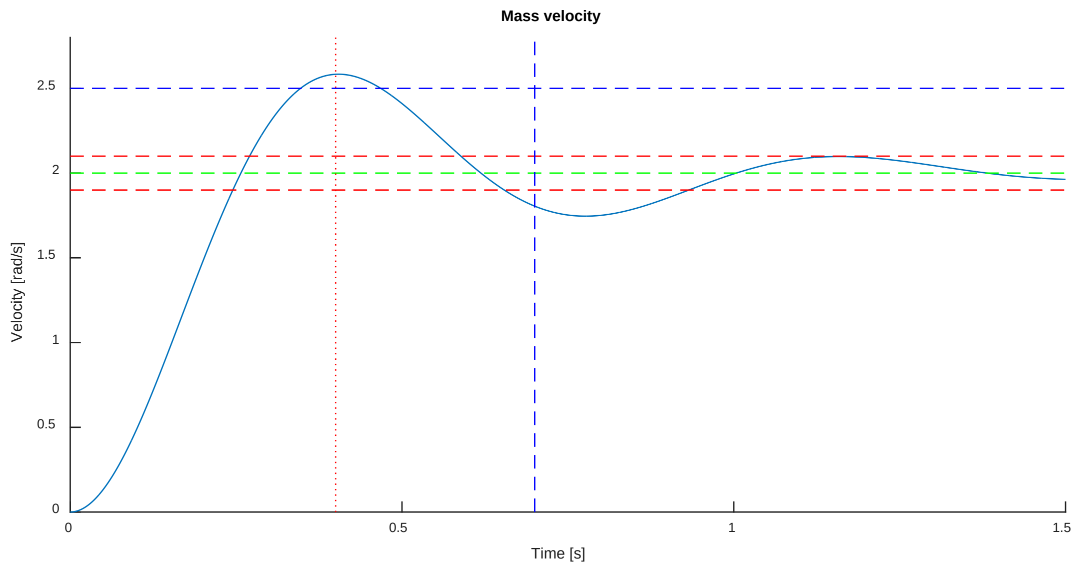

FIGURE 10. ANGULAR VELOCITY WITH INDUCTANCE 0.1 H

Q5. If the motor accelerates past the target speed (there is overshoot), the controller reacts by decelerating the motor i.e. by producing negative torque. With a “lazy” controller i.e. with small P and I gains, the system may never overshoot and therefore there is no negative torque in the simulation.

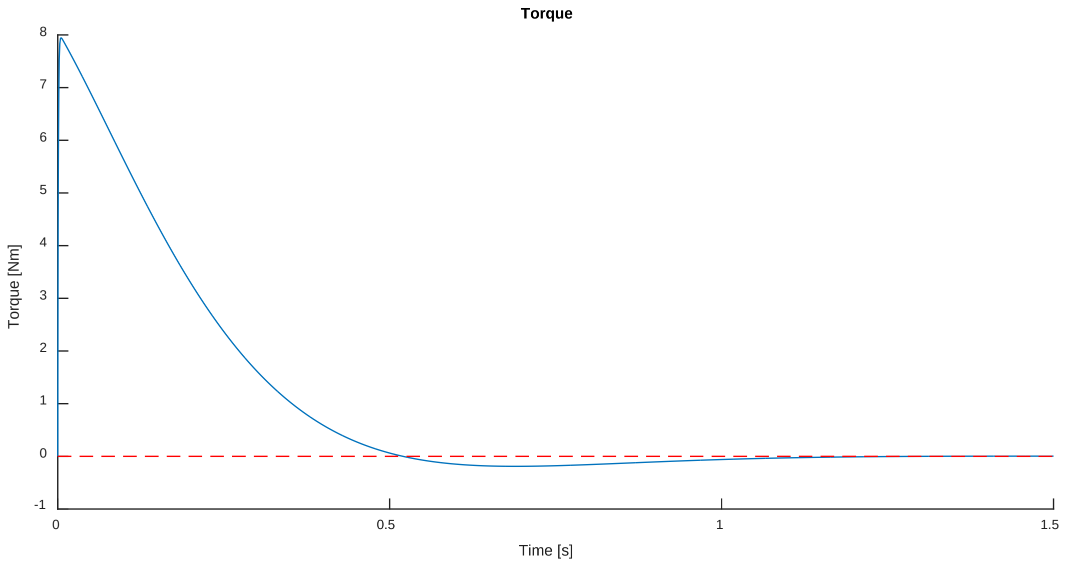

FIGURE 11. TORQUE

We can also see in figure 4 that there is overshoot in velocity during the corresponding negative torque period. Moreover, in figure 5 we can see that the control voltage is during this period also below the voltage demanded for a constant velocity in steady state. So during that period of time, the motor has not produced enough electromagnetic force to overcome its internal friction but instead some of the kinetic energy stored in the moment of inertia of the load has been released to rotate the motor. If the motor must be decelerated even faster, the motor can also actively produce negative torque.

Q6.

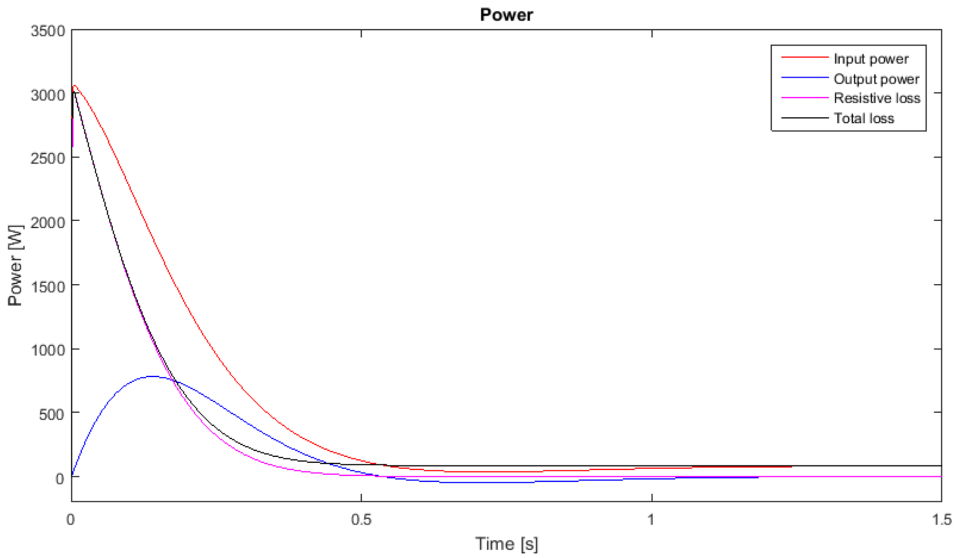

FIGURE 12. POWER CURVES

Electrical input power:

$$P_{in} = U_{in} I_{in} \tag{2}$$

Mechanical output power, which moves the mass:

$$P_{out} = T_{out}\omega \tag{3}$$

Thermal power:

$$P_{therm} = I^2 R \tag{4}$$

Waste power:

$$P_{waste} = P_{in} - P_{out} \tag{5}$$

Initially, the current is at its maximum and therefore resistive loss is very large. Even though that generates also a large torque, the velocity is almost zero, which leads to zero output power.

The output power is zero in steady state due to our definition: we defined the output power to be the power that moves the mass, not the power to move the mass *and* win the internal damping losses. So the demanded torque to move the mass with constant velocity is zero (we assume no friction force) -> P_{out} is also zero.

Initially most of the loss power consists of resistive losses, i.e. the thermal power generated in the resistance of the rotor windings. Later, when the steady state is reached, the demanded current is small, which leads to small resistive loss. In steady state, angular speed is large and therefore also the damping/friction losses are large. Thus in steady state, total power losses consist almost totally of the power consumed to win the mechanical damping (bω).

Q7.

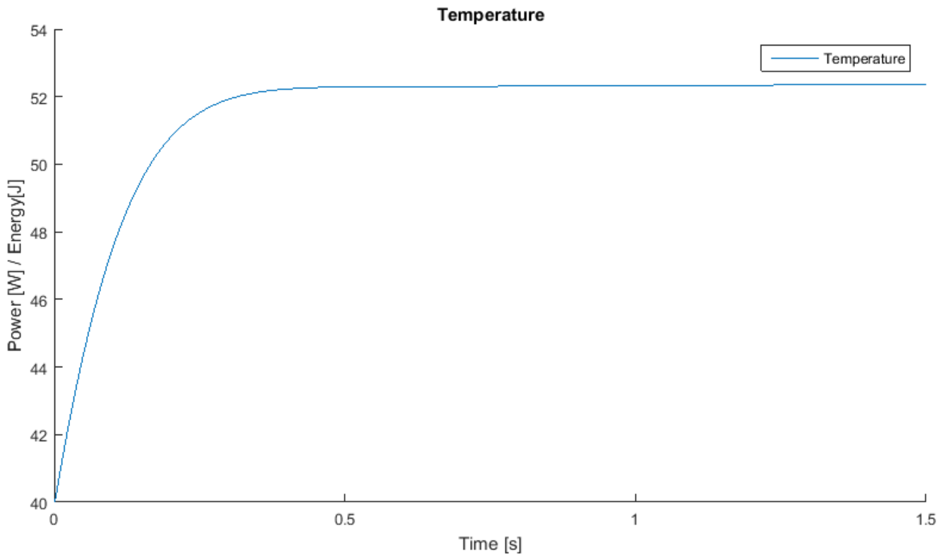

FIGURE 13. ROTOR TEMPERATURE

The resistive loss produces thermal energy (heat) in the rotor and the thermal energy is stored in the rotor material. The accumulated thermal energy can be integrated in the Simulink model as in the provided model files or summed in the script. Heat capacity represents the ratio of heat added or removed from an object and the change in its temperature. The temperature of the rotor is calculated with the following equation where T_0 is the starting temperature of 40 C °, E_t is the produced thermal energy and C is the heat capacity of the rotor.

$$T(t) = T_0 + \frac{E_t(t)}{C}$$

Initially the temperature increases rapidly (large current, large resistive losses), but it stabilizes quickly when steady state is reached. The temperature in the figure increases slowly also in the steady state since motor still requires current to overcome friction and any thermal conduction away from the rotor is not taken to account in the calculations.

With the previously mentioned controller gains, the temperature of the motor rises from 40 C ° to approximately 52 C °. With an approximately 12 C ° temperature rise per acceleration the motor would heat to 88 C ° after four acceleration cycles. In reality the motor would require a sufficient time to cool between accelerations. Cooling time could be reduced with forced air or liquid cooling systems.

If the gain I for example is increased to 4.4, the controller uses much larger output voltages, overshoot increases and therefore the motor works harder. This leads to a larger temperature increase and the final temperature is approximately 66 C° after one acceleration. With a lazier controller, the maximum current does not rise as high which means lower resistive losses and also lower temperature.