SOLUTIONS - EXERCISE 6

Microcontrollers and dynamics optimizing

1. MICROCONTROLLER

Q1. The system flashes the red and blue LEDs at 0.5 Hz frequency i.e. they are 1 second on and 1 second off but not at the same time. First, the red LED is on and the blue is off for one second. Then the blue LED is on and the red is off for one second. This program is repeated over and over again until the controller is switched off.

The resistor is in the circuit to limit the current through the LEDs and thus to avoid destroying them.

Q2. Maybe the easiest way is to just comment out the row of code on line 3 in the loop()-function which turns the blue led off:

```
1.  void loop() {
2.    digitalWrite(red_led, HIGH);
3.    //digitalWrite(blue_led, LOW);
4.    delay(1000);

5.    digitalWrite(red_led, LOW);
6.    digitalWrite(blue_led, HIGH);
7.    delay(1000);
8.  }
```

Also you could remove line 3 or change the parameter LOW->HIGH.
A more sophisticated way to implement this is to set the blue led ON in the void setup() part and remove it from the loop part. This would remove the periodic execution of commands that have no function.

Q3. The voltage drop V_f across an LED is virtually independent of the current through the led. When one LED is on, the rest of the supply voltage U is lost in the current limiting resistor and the current through the LED and the resistor is defined by the resistance R of the resistor according to Ohm's law.
When both LEDs are on, voltage drop across the resistor stays the same and thus also the current through the resistor stays the same (assuming similar forward voltages). However, now that current is divided between the two LEDs meaning that the current through one LED is only half of the current compared to the situation with only one LED on. The amount of light emitted by an LED depends on the current through it and thus the brightness of the LED is lower when both LEDs are on. So as the red LED blinks, also the brightness of the blue LED changes.

When both LEDs are on

$$U = V_f + R(I_{blue} + I_{red}), \tag{1}$$

and since

$$I_{blue} = I_{red}, \tag{2}$$

equation 1 equals

$$U = V_f + 2RI, \tag{3}$$

and from equation 3 we get the current I through one led:

$$I = \frac{U - V_f}{2R}. \tag{4}$$

With only one LED on we get equivalently:

$$I = \frac{U - V_f}{R}. \tag{5}$$

From these equations we can see that the current through the LED is smaller, when two LEDs are on, which leads to LED dimming.

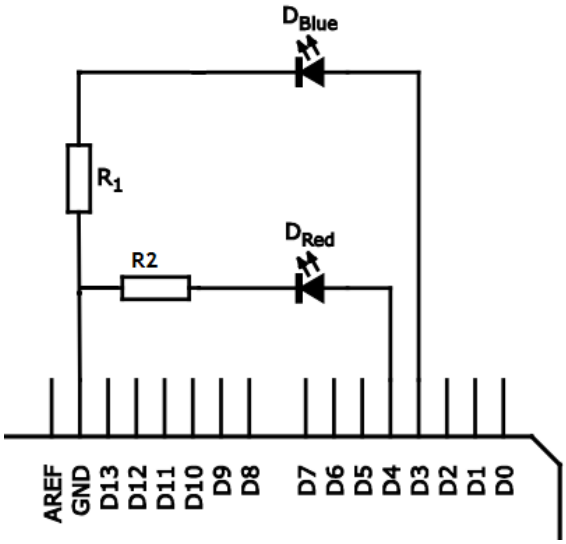

FIGURE 1. MODIFIED CIRCUIT.

A separate current limiting resistor for both LEDs should fix the problem. Now the current through one LED circuit doesn't have an effect on the other circuit.

2. POTENTIOMETER – MICROCONTROLLER – DC MOTOR

The task describes an efficient procedure for controlling a high current with a small amount of current, which is often the case. The microcontrollers are meant to send control signals; not to act as power supplies. Usually microcontrollers are not even capable of producing the current the application needs. Moreover, this kind of solution does not drive the DC motor through a resistor or a potentiometer to control the velocity, which would lead to large amount of energy loss in the resistor.

Q1.

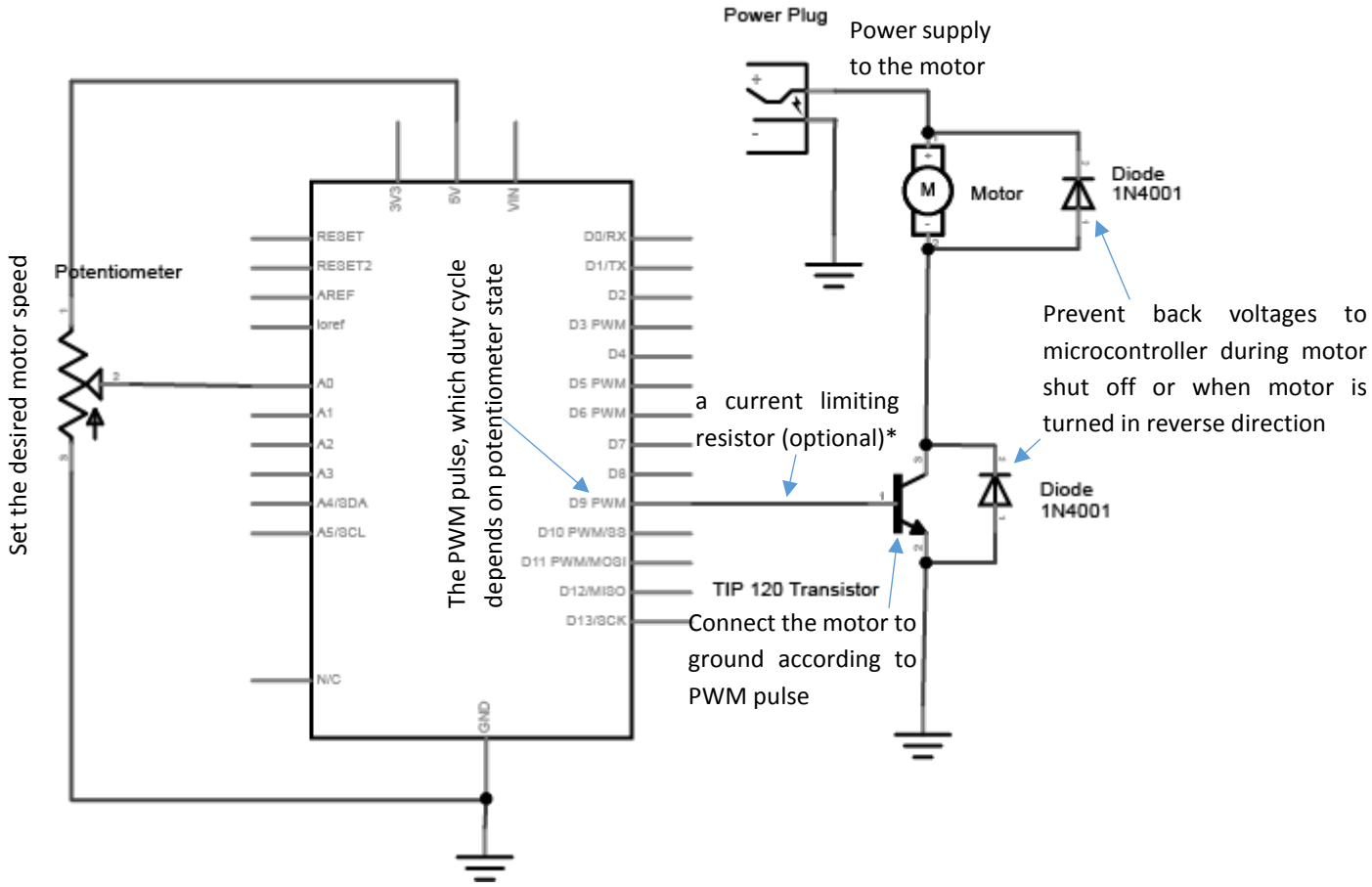

FIGURE 2. EXAMPLE CIRCUIT.

* Often it is possible to connect the transistor base to a microcontroller pin directly, since the microcontroller currents are usually low enough. However, that depends on the microcontroller used and using a resistor is recommended.

With the potentiometer, the user can provide a variable voltage to the input of microcontroller's the AD converter.

The power supply is able to provide the large voltage and current required by the DC motor.

The transistor acts as a current controller switch. The transistor conducts through its collector pin the large current from the motor to ground when the microcontroller sources a small current through the base pin of the transistor to ground.

In practice, the microcontroller outputs a PWM signal. During the high part of the PWM, the voltage in the output pin of the microcontroller is high (e.g. 5 V) and therefore there is a current going from the microcontroller's pin through the base of the transistor to ground. During the low part of the PWM signal there is no potential difference between the output pin of the microcontroller and ground and thus there is no current and the transistor does not conduct the current from the motor either.

Because of the inductance in the windings of the motor, the current through the motor cannot be suddenly stopped. Therefore, when the transistor does not conduct the current to ground, the current is circulated through the freewheeling diode next to the motor back to the high voltage side of the motor.

Q2. *For example in "Arduino language":*

```
int potValue = 0;                       //initialize the pot value to 0

void setup() {
    pinMode(9, OUTPUT);                 // set the transistor pin as output
}

void loop() {
    potValue = analogRead(A0);          // read the potentiometer
    potValue = potValue/4;              // convert the pot value to 0-255
    analogWrite(9, potValue);           // control the pulse width of the PWM
}
```

AnalogWrite() function controls the pulse width of the PWM output with 256 steps.

Q3. The resolution of the control is dictated by the lowest amount of bits in ADC or DAC conversion. In this case it is 8-bit DAC:

$$R = \frac{5000\ rpm}{2^8} = 19.53\ rpm. \tag{6}$$

Q4. The PWM duty cycle is the percentage of time that the output pin is written to HIGH, i.e., how many percent of time the output pin commands the transistor to let current through. With always HIGH, the current would pass all the time and the motor voltage effective value would be 12V and since the corresponding velocity 5000 rpm. With a duty cycle *D* of

$$D = \frac{1750}{5000} \cdot 100\% = 35\% \tag{7}$$

the effective voltage would be 4.2V and the velocity correspondingly 1750 rpm.

3. DYNAMICS OPTIMIZING

Find the solution files also in MyCourses.

The Simulink model for the task is illustrated in figure 3.

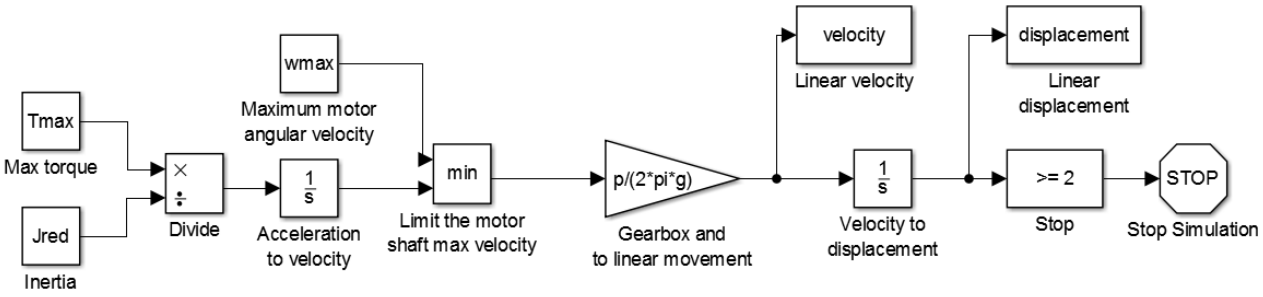

FIGURE 3. THE SIMULINK MODEL OF THE TASK.

The motor is modeled as a constant torque source. Angular acceleration of the motor is defined by the torque and the sum of all the inertia components reduced on the motor axis. Angular velocity is integrated from the acceleration, limited to the maximum rotating speed and converted to linear velocity. The min-block in figure 3 limits the max motor shaft velocity. Limiting the velocity can be done with several blocks such as Saturation or directly in the parameters of the velocity integrator block. The simulation is stopped, when the displacement reaches 2 m.

In order to use the fminbnd, the iterative minimizing function to find the optimal gear ratio, a function for simulating the model with varying gear ratios is required. The function should return the time required to reach the target position i.e. the time required to run the simulation because our simulation stops when the displacement of 2 m is reached.

The parameter values defined inside a function are seen only by that function – that's why we have to use simset function to define the workspace used by our function as the source of the simulation parameters. The options also restrict the maximum simulation step to 0.0001 seconds to get more accurate simulation results. The reduced moment of inertia on the motor shaft (*Jred*) depends on the gear ratio (*g*), and thus we have to calculate it inside the function for each *g* value.

Fminbnd function is an iterative optimizing function which evaluates the simulating script at certain points and finally always produces the approximate local minimum point of the function to be optimized. Fminbnd only works for one variable functions but there are also built-in optimizing functions in Matlab for more complex cases.

Check the model scripts for details on how the optimizing and running the simulation is implemented.

The optimal gear ratio in this case is 0.6104 and the time it takes to reach 2 m displacement is 4.78 seconds.

The plots:

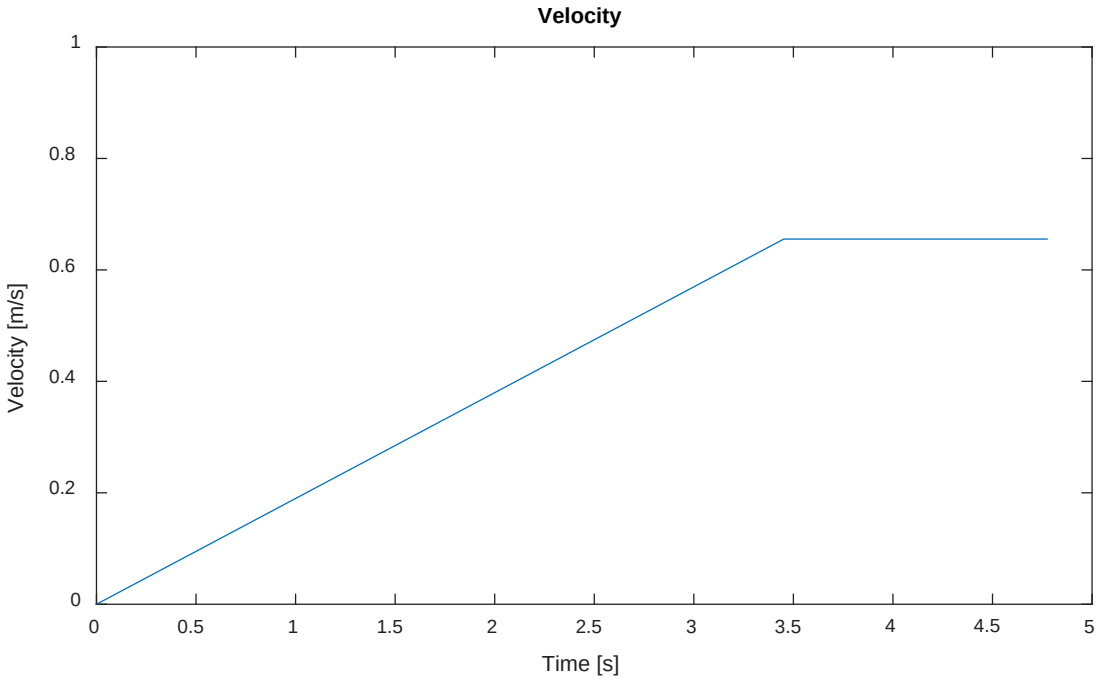

FIGURE 4. VELOCITY WITH OPTIMAL GEAR RATIO.

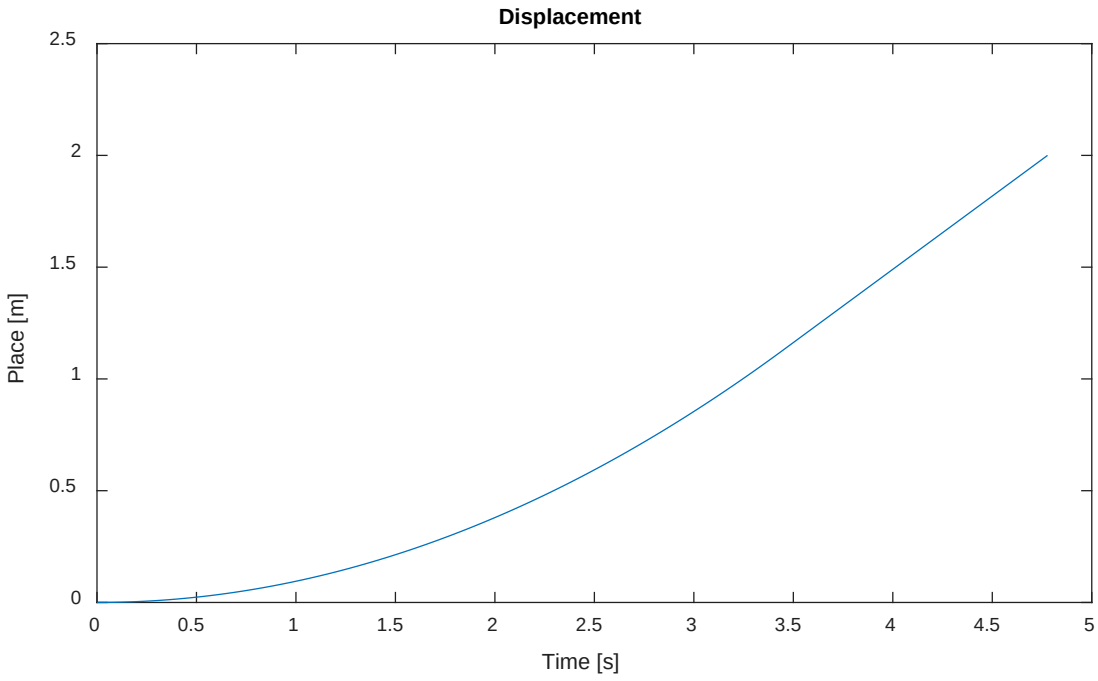

FIGURE 5. DISPLACEMENT WITH OPTIMAL GEAR RATIO.

The following figures may help to understand that there certainly is optimal gear ratio and that it's all about balancing between acceleration rate and maximum speed:

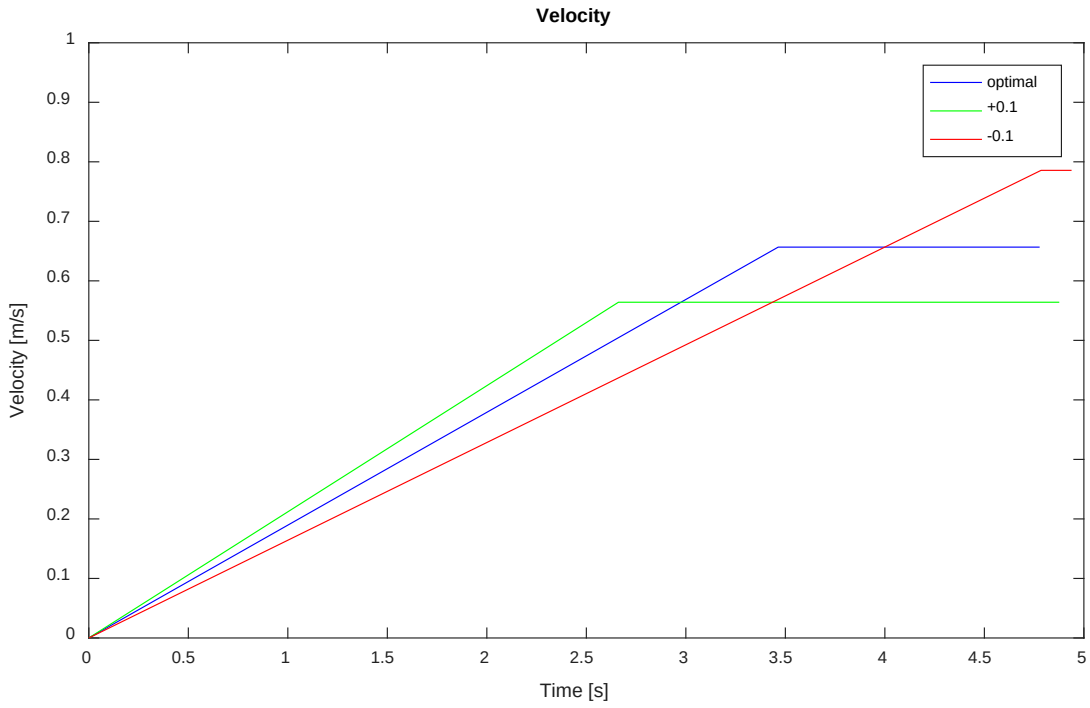

FIGURE 6. VELOCITY WITH SLIGHTLY VARYING GEAR RATIOS.

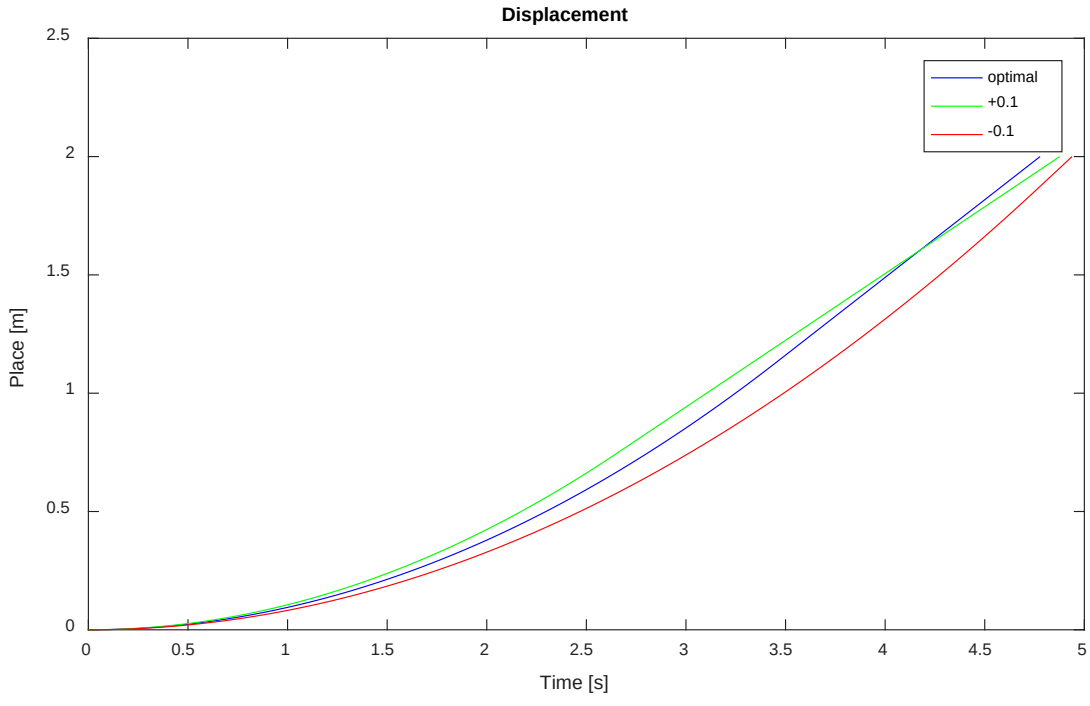

FIGURE 7. DISPLACEMENT WITH SLIGHTLY VARYING GEAR RATIOS.

The motor used in this exercise produces a pretty high torque and the ball screw has a small pitch which acts as a sort of reduction gear. Therefore, instead of using a reduction gear between the motor and the ball screw, it is beneficial to increase the angular velocity of the ball screw with the transmission. If the gear ratio would be high, the motor

would reach the maximum angular velocity quickly but then the maximum velocity would be fairly low. On the other hand decreasing the gear ratio leads to a slower acceleration. The area inside each velocity curve is equal and corresponds the distance travelled.