STEM Games 2018 - Engineering Arena

Ante Orešković, Ivan Pavić, Filip Šklebar, Marko Švec

Introduction

Since the old times when humans tamed horses, transportation was one of most important parts of any advanced civilization. Having roads and efficient transportation meant that you could send information faster, build bigger buildings and expand further. With the invention of the steam engine, the world became smaller - long routes became minor obstacles and distant places became within reach. Traveling and sending material became faster and cheaper.

This trend was only amplified by the invention of internal combustion engines. Today we are transporting more stuff than ever. With technological advancements, we are on the brink of a new revolution - electrical and self-driving vehicles. In the last few years, we have made big advancements, thanks to famous entrepreneurs promoting electric cars and trucks, desire to stop using fossil fuels and powerful computers that are capable of processing neural networks.

We have decided to give you an idea of what goes into designing an electric truck. Your creative thinking and problem-solving skills will be tested. The whole task will be split into 3 days, and it will consist of picking parts, calculating loads, technical exam and tests. Roughly split, the tasks will be:

Day 1 - Truck mechanics, static tests

Day 2 - Electric vehicle powertrain, vehicle dynamics

Day 3 - Embedded system design

Along with checking your calculations, we will be using a simulation that takes your design and runs tests - from basic sanity tests to check that your design makes sense to obstacle courses that will grade speed and endurance.

Keep in mind that although not everything is specified in the tasks you get, real-life phenomena still may be tested in our simulation. It is your task to catch as many of real life traps as you can!

We hope you find tasks we have given you interesting, and good luck!

Rules

- **Each day will be graded separately!** You will be graded at the end of each day, and you start again next from the configuration we give you. You will gain no benefits from working outside given time!
- You are allowed to use any software packet you are familiar with (Octave, Python/NumPy, spreadsheet - Excell/Google docs...) or you can choose to solve the given problems by hand.
- Whatever you choose, you need to submit your papers/code, and it needs to be well documented. Presentation of the solution, creativity, and readability will also be graded.
- You are not allowed to share the tasks and ask for help from non-participants. Any attempt to do so will result in disqualification.

Day 1

1.1 Introduction

The task for Day 1 is to design a truck using parts from the catalogue and to choose how to combine them. A journey of a thousand miles begins with a single step. The first step of designing a truck is choosing a truck body. The truck body has slots for the engine, slots for cargo and miscellaneous slots for the battery pack, controllers, and other parts. Each truck body differs in some way from another. They have different sizes, different aerodynamic resistances and a different number of slots. For now, we will ignore the cargo that needs to be transported.

After choosing the body, you will choose one of the chassis and some wheels. Along the way, you will need to solve tasks and calculations which will help you choose the best combination possible. Additionally, you will have the chance to solve two side tasks which can get you more points and improve your score.

You will also need to fill out a form in an exact manner. The form is a way to define your truck configuration, and it will be used within our simulation to test the truck performance.

In chapter 1.2 all today's tasks are explained in detail and in chapter 1.3 you are given a catalogue from which you can choose parts for your truck.

Friendly advice: Read everything before solving the tasks.

1.2 Tasks

As previously mentioned, your main task is to design a truck. In Section 1.3 you can find all the availabe parts which you will combine to achieve that. The catalogue contains:

1. Truck body list
2. Chassis list
3. Wheel list

Things your truck has to have:

- One of the truck bodies offered.
- One of the chassis offered. Chassis axle positions have to correspond to the body axle positions.
- Up to two wheel types. All the wheels have to have the same radius and the wheels on both rear axles have to be the same(have to have the same code). Of course, it is assumed that one axle has the same wheels both on right and left sides(the truck actually has six wheels).

Additional instructions:

- Assume that fenders are high and deep enough for any wheel.
- Pay attention to the price. Since there are more feasible solutions, the cheapest one will be considered the best one.

Technical inspection

Techincal inspection is the most important test your team will face today. Therefore, you better be prepared for it. Tasks which will help you do that are:

1. (**6 pts**) Sketch the setup(truck + chassis + wheels). Everything has to be visible, easily distinguished and placed in the coordinate system as in the Figure1(yes, unfortunately alongside negative x axis). All the important coordinates have to be marked in (negative)meters(mark them as, for instance, (x_1, y_1, z_1) and label those values in a table). In essence, the sketch has to be a combination of figures1 and 3, with markings the same as on 3.
You don't have to sketch the motors and the batteries, only the positions of the slots. You don' have to sketch the fenders(but keep them in mind). Sketch only side view, you don't have to skecth the truck's top or front/rear view!

2. (**7 pts**) Using the sketch from the task 1 and masses of all the elements, derive an expression and calculate the truck's center of mass[1]. Mark the total mass as well.

   Additional information:

   - During the first day you don't have to take electric systems into consideration. However, every vehicle needs propulsion and every part has some mass. We have, therefore, chosen these default values for the first day.
     If body you have chosen has six empty slots:
     (a) Slot1 : Motor with mass $m = 196kg$
     (b) Slot2 : Battery pack with mass $m = 413kg$
     (c) Slot3 : Empty
     (d) Slot4 : Empty
     (e) Slot5 : Battery pack with mass $m = 872kg$
     (f) Slot6 : Motor with mass $m = 210kg$

     If body has five empty slots:
     (a) Slot1 : Motor with mass $m = 196kg$
     (b) Slot2 : Battery pack with mass $m = 413kg$
     (c) Slot3 : Empty
     (d) Slot4 : Battery pack with mass $m = 872kg$
     (e) Slot5 : Motor with mass $m = 210kg$

3. (**7 pts**) After having summed up all the masses into single mass point, draw a free body diagram of your truck on a slope α traveling at a speed v_x and acceleration a. The forces you should take into consideration:

   - Gravity
   - Response of the road on each axle(in vertical direction)
   - Air drag (acting at the hight h_{air}[2])
   - Inertial force caused by linear longitudinal acceleration

   You should mark all the important dimensions as well. They will be useful later.

   Additional information:

   - Draw only the above-mentioned forces and pay attention to the point where they act.
   - You don't have to take rolling resistance into account.[3]
   - Positive slope means the vehicle is going uphill.

4. (**12 pts**) Using the free body diagram from task 3, derive the equations for load transfer on all of the axles. Write down the equation for air resistance force before.

   Additional information:

   - Once again, for today's task you don't need rolling resistance, since it does not affect load transfer. However, you will need it tomorrow.
   - Suspention is stiff (there is no elasticity/damping) and the longitudinal load transfer is instantaneous.
   - Load on the rear axles is equally distributed between them.

[1] It is assumed that motor's and battery's center of mass correspond to the center of a chosen slot.

[2] One of the parameters in the catalogue.

[3] You won't need it today, but it is important that you understand what it represents since you will need it for tomorrow's tasks.

5. (**FAIL/PASS**)Using derived equations, design your truck so that the wheels are able to carry its load when driving with a speed of $50km/h$, without acceleration, on slopes of:

   (a) 0%,

   (b) ±10%,

   (c) ±45%,

   (d) ±65%.

   Additional information:

   - Mentioned slopes do not define exclusive tasks. A single truck should be able to withstand all of them.
   - Tires are stiff and wheels don't have inertia.
   - Gravitational acceleration $g = 9.81m/s^2$
   - Air density $\rho = 0.12041dag/dm^3$. There is no wind.
   - $1inch = 2.54cm$
   - If you find some mass data in pounds, use $1lbs = 0.4536kg$ for conversion.

Side task 1 - Max velocity

This is the first side task. It is not crucial for passing the technical inspection, but shows that you understand your truck's dynamics and are capable of imagining abstract concepts. And, of course, it will be graded.

1. (**1 pt**) Derive an expression for calculating maximum velocity(stable state) of a truck on a road of slope α, starting from $v_x = 0m/s$ and neglecting all resistance forces except air drag(there is no friction and/or rolling resistance).[4]
2. (**1 pt**) Using the derived equation, calculate your truck's maximum velocity for the slope $\alpha = 25$.
3. (**1 pt**) Plot velocity v_x as a function of time(transient behaviour of a system has to be visible).
4. (**1 pt**) Draw stable state velocity v_x as a function of road slope $\alpha \in [0, 90]$.

Side task 2 - Understeering gradient

This is the second side task. Just as the first one, it is not crucial for passing the technical inspection. However, unlike the first one it requires you to do an online research.

1. (**2 pts**) Derive the expression for understeering gradient.
2. (**1 pt**) Calculate understeering gradient for the truck you designed. When calculating load distribution, use $\alpha = 0$, $a = 0m/s^2$ and $v_x = 25km/h$.
3. (**1 pt**) Classify your truck as being understeered, neutral steered or oversteered.
4. (**2 pts**) Calculate critical or characteristic speed, state which one you have calculated and why.

Additional information:

- The convention when calculating understeering gradient is to use bicycle model of a vehicle. You can merge two rear wheels into single wheel. Calculate the distance between this new imaginary axle and CoG[5] as an arithmetic mean of first and second rear axle to CoG distances. (This information is useful for previous tasks as well.)
- Notation is not always the same so please state which marking corresponds to which physical quantity.

[4] In this case, for the sake of simplicity, positive slope means the vehicle is moving downhill.

[5] CoG stands for the center of gravity. In this case it is the same as the center of mass.

- There are a few versions of understeering gradient and they differ by measurement unit.The most common are:
    - K_u $[rad/N]$,
    - K_{u2} $[rad]$,
    - K_{u3} $[rad/(m/s^2)]$.

  You have to calculate the first one (K_u $[rad/N]$), connecting steering angle with centripetal force.

- You will need cornering stiffness, which can be found in tire graphs given in the catalogue.
- It is highly unlikely that your numerical solution will be the same as ours and we will keep that in mind. It is more important to derive correct equations.

1.3 Catalogue

Catalogue is divided into several sections. Each of them contains a list of specific part variations with additional information. We kindly encourage you to examine them carefully before starting the calculations.

1.3.1 Truck body

Truck body dimensions are marked on Figure 1 and Figure 2. The origin of a coordinate system is placed in rear left corner of a truck body. Coordinate axes form a right handed coordinate system, as shown in figures.

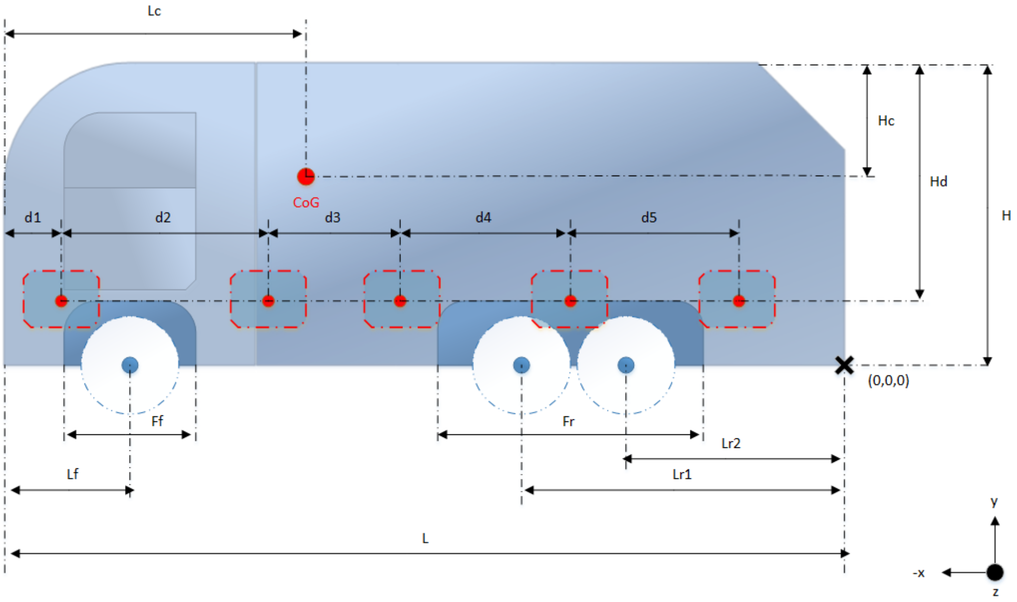

Figure 1: Truck body side view

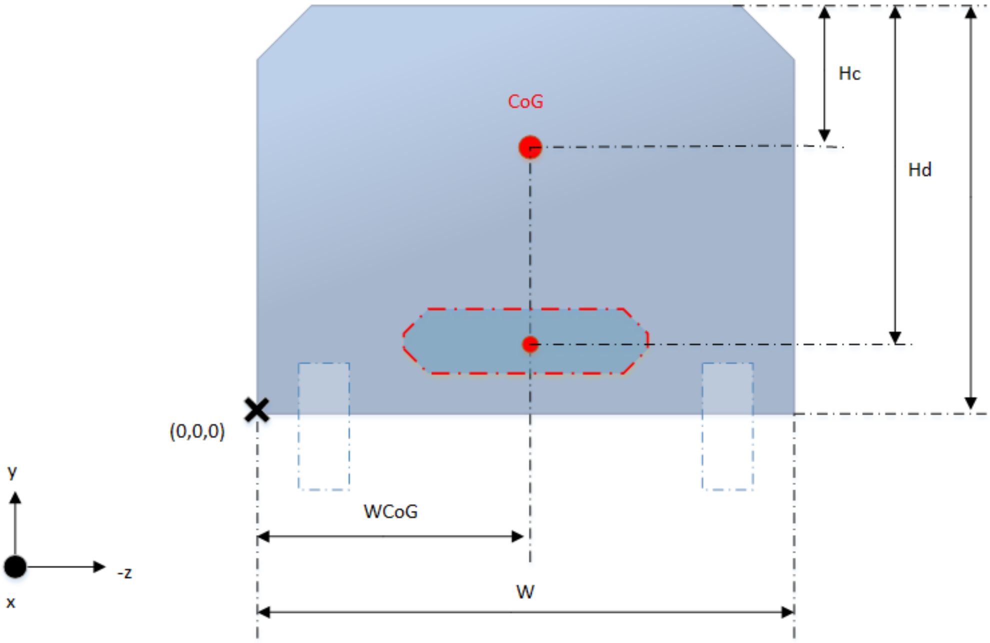

Figure 2: Truck body rear view

Table 1: Body aerodynamics, mass and rear view dimensions data

Code	Price[EUR]	Drag coefficient[1]	Drag area[m^2]	Drag height[m]	Mass[kg]	W	W_{CoG}
1000	40000	0.45	5.5	1.3	4250	2.25	1.125
1001	45000	0.45	5.5	1.5	4000	2.25	1.125
1002	45000	0.4	5.5	1.5	4000	2.25	1.125
1003	30000	0.4	5	1.5	5000	2.25	1.125
1004	50000	0.4	5.5	1.5	3500	2.25	1.125
1005	50000	0.5	6.25	1.5	3500	2.25	1.125
1006	27000	0.35	4.3	1.4	5500	2.25	1.125
1007	45000	0.45	4	1.2	4000	2.25	1.125
1008	60000	0.55	5.5	1.5	3000	2.25	1.125
1009	50000	0.5	4.25	1.15	3500	2.25	1.125

Table 2: Side view dimensions data

Code	L	H	L_c	H_c	d_1	d_2	d_3	d_4	d_5	d_6	H_d	L_f	L_{r1}	L_{r2}	F_f	F_r
1000	5	2.25	1.75	1.25	0.5	0.7	0.8	0.9	1.1	x	1.9	1	2	1.2	1	2.5
1001	6	2.5	2.5	1.3	0.8	0.7	1	1.1	1	0.8	2.15	1	2.2	1.2	1	2.5
1002	6	2.5	2.75	1.3	0.8	0.8	0.9	1.3	1.1	0.6	2.15	1	2	1.2	1	1.5
1003	6	2.5	2.75	1.3	1.2	0.9	0.9	0.6	1.4	x	2.15	0.5	2.2	1.2	0.95	2.5
1004	6	2.5	2.5	1.3	0.8	0.7	1	1.1	1.3	0.5	2.15	1	2.2	1.2	1	2.5
1005	6	2.5	2.5	1.35	1.2	0.9	0.8	1.1	1	x	2.15	0.5	2.2	1.2	0.8	2
1006	5	2.25	1.5	1.25	0.5	0.6	1	0.8	1.1	x	1.9	1	2.4	1.2	1.1	2.5
1007	5	2.25	1.25	1.25	0.2	0.9	1	0.7	1	x	1.9	1	2.4	1.2	1.1	3.1
1008	6	2.5	2.75	1.3	0.8	0.8	0.9	1.3	1	0.7	2.15	1	2	1.2	1	2
1009	5	2.25	1.25	1.25	0.2	0.9	1	0.8	0.9	0.6	1.9	1	2	1.2	0.9	3.1

1. All the data without measurement unit in the tables is in meters [m].
2. If the value of d_6 equals "x", that means that the chosen truck body has only five item slots. Otherwise, it has six of them.

1.3.2 Chassis

Truck chassis coordinates are marked on Figure 3 and Figure 4. The origin of a coordinate system is placed in rear left corner of a truck chassis and corresponds to the origin on Figure1. Coordinate axes form a right handed coordinate system, as shown in figures.

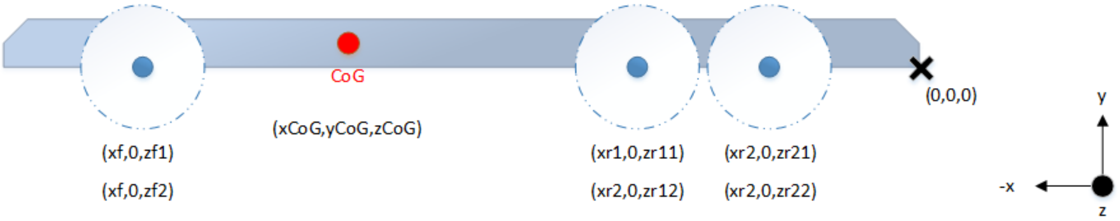

Figure 3: Truck chassis side view

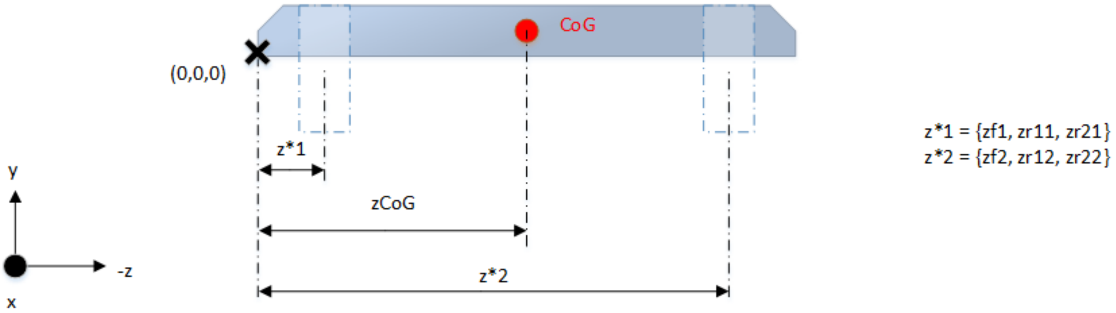

Figure 4: Truck chassis rear view

Table 3: Chassis dimensions data

Code	Price[EUR]	Mass[kg]	x_{CoG}	y_{CoG}	z_{CoG}	x_f	y_f	x_{r1}	y_{r1}	x_{r2}	y_{r2}	z_{*1}	z_{*2}
2000	7500	500	-3	0.2	-1.125	-5	0	-2	0	-1.2	0	-0.2	-2.05
2001	5000	600	-3	0.2	-1.125	-5	0	-2	0	-1.2	0	-0.3	-1.95
2002	7500	500	-2.5	0.2	-1.125	-4	0	-2	0	-1.2	0	-0.2	-2.05
2003	5000	600	-2.5	0.2	-1.125	-4	0	-2	0	-1.2	0	-0.2	-2.05
2004	7500	500	-3	0.2	-1.125	-5	0	-2.2	0	-1.2	0	-0.35	-1.9
2005	4500	650	-3	0.2	-1.125	-5	0	-2.2	0	-1.2	0	-0.2	-2.05
2006	7500	500	-3	0.2	-1.125	-5.5	0	-2.2	0	-1.2	0	-0.35	-1.9
2007	4500	650	-3	0.2	-1.125	-5.5	0	-2.2	0	-1.2	0	-0.2	-2.05
2008	7500	500	-2.5	0.2	-1.125	-4	0	-2.4	0	-1.2	0	-0.2	-2.05
2009	5000	600	-2.5	0.2	-1.125	-4	0	-2.4	0	-1.2	0	-0.25	-2

1. All the coordinates in the table are in meters [m].
2. In the notation z_{*1} and z_{*2}, $*$ stands for $f, r1, r2$ (e.g. z_{f1} for front axle or z_{r22} for second rear axle). In essence, it means that this data is the same for all three axles.

1.3.3 Wheels

Wheel parameters are defined in Table 4. Force profiles are shown on figures 5, 6, 7 and 8.

Table 4: Wheel data

Code	Price[EUR]	Mass	Rolling resistance	Force profile	Tire specification
3000	3000	50	0.005	1	385/55 R 22.5 140 Q
3001	3000	60	0.01	2	385/55 R 22.5 140 R
3002	1500	50	0.005	1	235/75 R 17.5 123 Q
3003	1500	60	0.01	2	235/75 R 17.5 123 R
3004	2000	50	0.005	1	215/75 R 19.5 125 Q
3005	2000	60	0.01	2	215/75 R 19.5 125 R
3006	1500	50	0.005	1	295/60 R 22.5 120 Q
3007	1500	60	0.01	2	295/60 R 22.5 120 R
3008	2000	50	0.005	1	285/70 R 19.5 125 Q
3009	2000	60	0.01	2	285/70 R 19.5 125 R
3010	1200	40	0.003	4	205/55 R 16 100 S
3011	1200	55	0.025	3	205/55 R 16 100 R
3012	1100	40	0.003	4	225/40 R 18 105 S
3013	1100	55	0.025	3	225/40 R 18 105 R
3014	1100	40	0.003	4	255/40 R 17 105 S
3015	1100	55	0.025	3	255/40 R 17 105 R
3016	1000	40	0.003	4	265/35 R 17 90 S
3017	1000	55	0.025	3	265/35 R 17 90 R
3018	2500	40	0.003	4	275/35 R 19 100 Y
3019	2500	55	0.025	3	275/35 R 19 100 Y

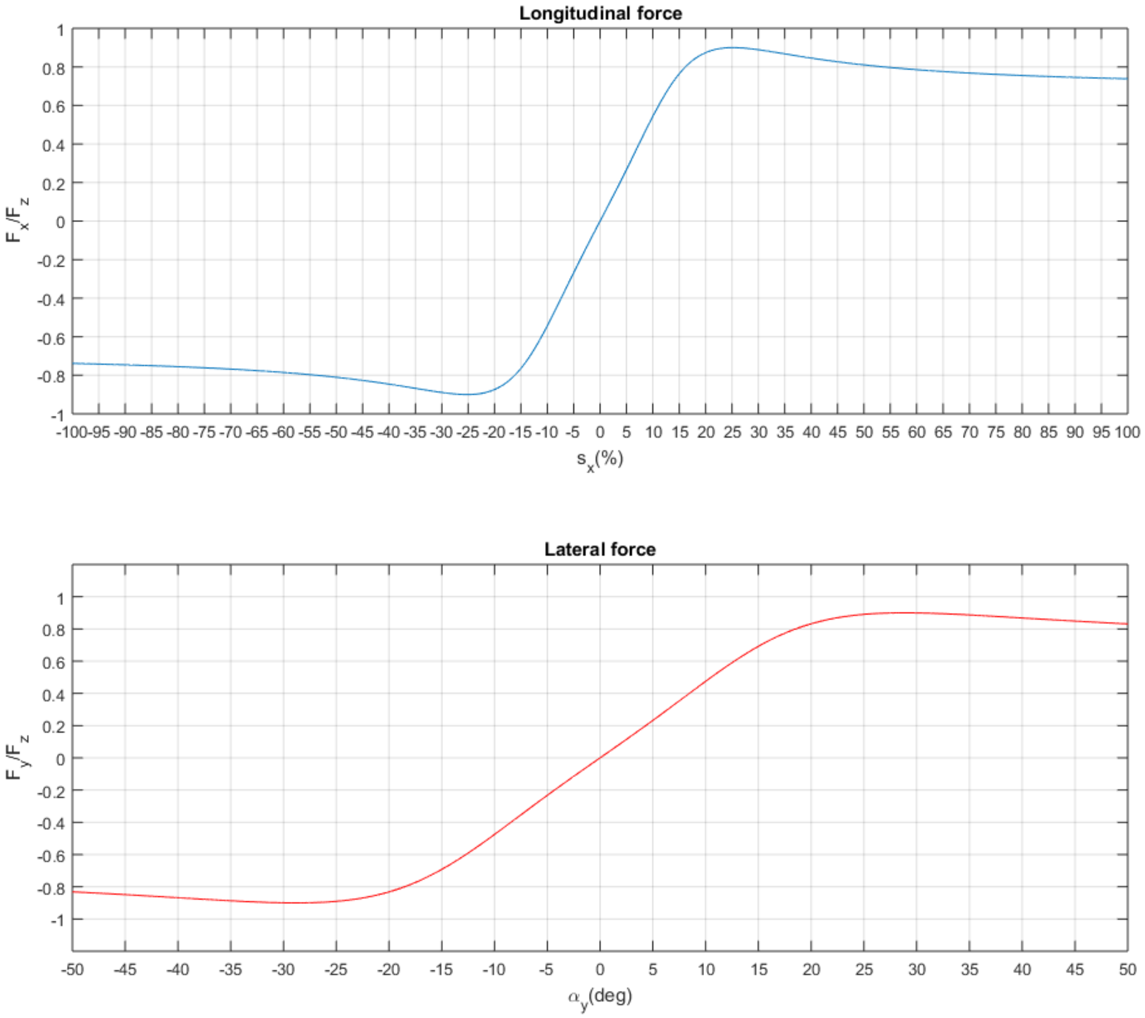

Figure 5: Tire force profile 1

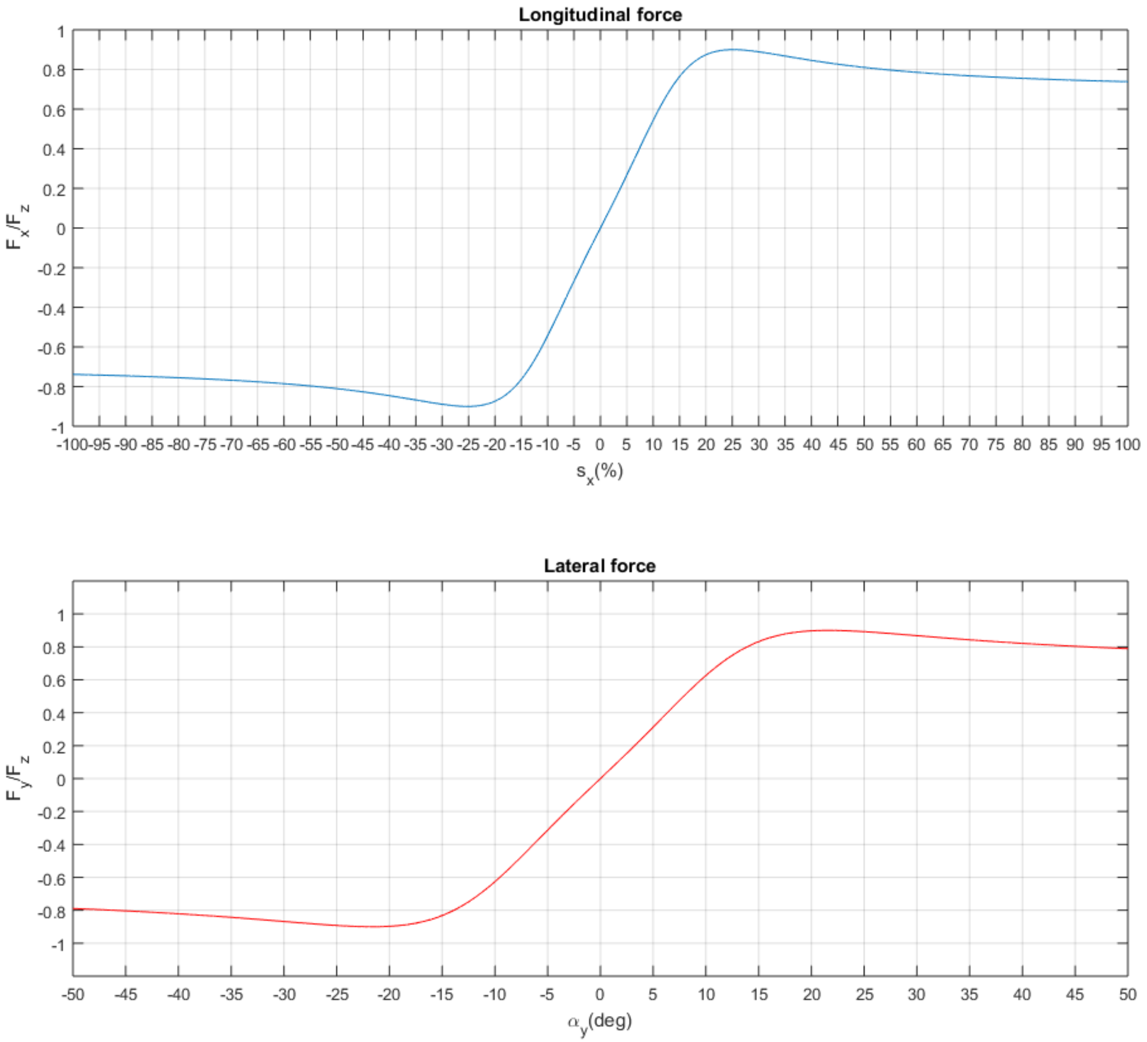

Figure 6: Tire force profile 2

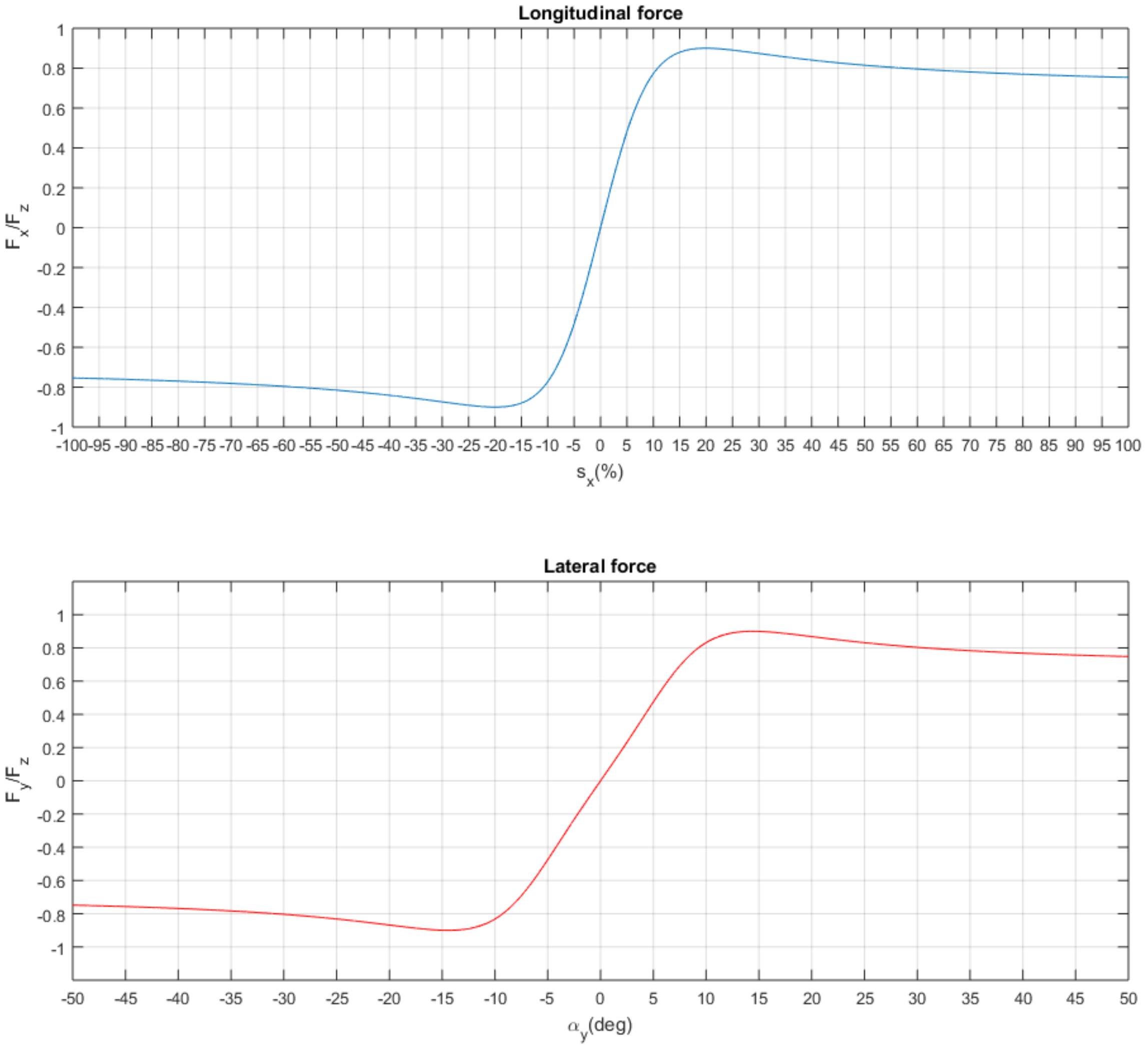

Figure 7: Tire force profile 3

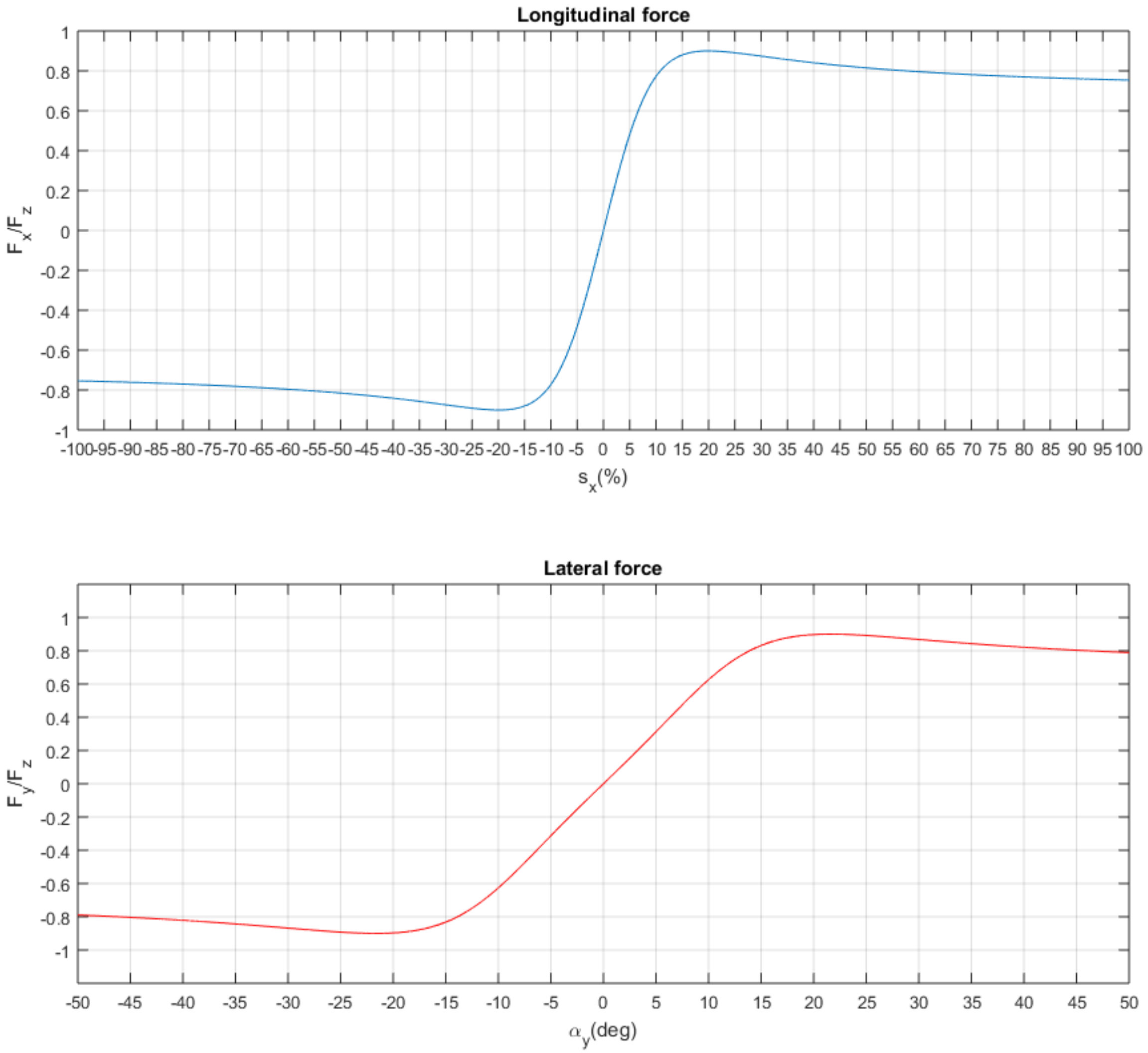

Figure 8: Tire force profile 4

1.4 Grading scheme

1. Tasks 1.-4. in section 1.2 carry up to **32 pts**. You have to have at least 75% (**24 pts**) to be able to pass technical inspection.
2. Task 5. in section 1.2 is graded as **FAIL/PASS** depending on whether your truck satisfies mentioned conditions.
3. If the design is infeasible(unable to be assembled), the truck fails technical inspection.
4. If the team has passed technical inspection, it can earn additional money by solving side tasks 1. and 2. The money earned is calculated as 1000eur * (number of points gained on those tasks).
5. Team which spent less money is considered better. If some of the teams have equal spending, the one with most points gained on compulsory tasks is considered better.
6. Teams whose design has not passed technical inspection are ranked by percentage of points gained(for all the tasks, not just compulsory part).

Day 2

2.1 Introduction

The task for day 2 is to make your truck move! Yesterday (hopefully) you designed a truck that passes technical inspection. The technical inspection consisted mostly of static tests. Today you will choose motors with reductions, design a battery pack and choose motor controllers. The truck will be tested on few different tracks and will be driven trough endurance tests to see the autonomy of the vehicle.

You will have our hand-picked basic truck configuration (body, chassis, and tires) to start you off.

Keep in mind that since today there are many valid configurations, grading will be mostly relative to other teams performance. We will compare each trucks price, weight, endurance, braking distance and top speed. Grading will be such that the best team will be one who does the best overall, not the one who is #1 in a single category. More on that in grading section of this task.

To submit your results you will fill out a spreadsheet and submit your results by filling out the form `Day2_SolutionForm.xlsx` in the same Google drive folder as yesterday.

You might want to read all the tasks before you start solving because some tasks impact the results from the others.

Good luck!

2.2 Tasks

Motor - mechanical constraints and gearbox

Your first task for today is to choose appropriate motor and gearbox according to mechanical constraints for the truck. Motor mechanical constraints depend on truck recommended requirements. General guidelines for the truck are:

- truck needs to be able to drive at the continuous speed of $130km/h$ on a flat surface
- truck needs to be able to drive at the continuous speed of $80km/h$ on a $10°$ incline
- truck needs to be able to decelerate from $100km/h$ to $0km/h$ in 300 meters on a downwards slope of 5%. Assume that only regenerative braking is applied.

Keep in mind that rolling resistance is taken in account, and downforce is ignored.

Each truck body can have up to 3 motors. Those are:

- front motor
- rear motor 1 - rear motor closer to the front
- rear motor 2 - rear motor closer to the back

Each motor is linked to its respective axle through corresponding reduction (one-speed gearbox). Gearbox can be designed as an 2 integer number fraction - a/b. For example you can design your gearbox to be $20 : 1$, $45 : 8$, $1 : 1$, $1 : 20$... Neither numerator or denominator cant be larger than 100. Assume that gearbox is massless and there are no gearing losses.

You do not need to place all of the motors. If you don't want to place a particular motor, write 0 in code and slot fields for the given motor, in the spreadsheet on Google drive.

You can choose any motor from table 5 for any axle. There are, however some restrictions:

- front motor can be in slots 1 or 2
- rear motor 1 can be in:
  - **5 slot body:** Slots 3 or 4
  - **6 slot body:** Slots 3, 4 or 5
- Rear motor 2 can be in:
  - **5 slot body:** Slots 4 or 5

- **6 slot body:** Slots 5 or 6

Note: You need to provide your calculations (either handwritten or in a form of a of an computer aided calculation). Choosing some motor without a valid and reasonable calculation will be graded with 0 points.

Motor - electromechanical parameters

As mentioned before, the propulsion system for designed truck consists of electric motors. Find motor parameters (R_s, K_t, K_e) based on motor catalogue of motors of your choice. Relation between some of these parameters can be found by solving additional side task 2.2.1.

2.2.1 Side task - electromechanical parameters

Basic electromechanical characteristics of the motor can be expressed with equation 1.

$$V_s(t) = R_s \cdot i_s(t) + L_s \frac{di_s(t)}{dt} + e(t) \tag{1}$$

Equation 1 describes voltage V_s which is applied on motor stator over time as the function of stator current $i_s(t)$ and electromotor force $e(t)$, while R_s is stator coil resistance and L_s stator coil inductance. Additionally, $e(t)$ depends on motor angular speed 2:

$$e(t) = K_e \cdot \omega_r(t); \tag{2}$$

Solve equation 1 considering equations 3 and 4.[6]

$$V_s = V_m \cdot \mu(t) \tag{3}$$

$$\omega_r(t) = \omega_n \cdot (1 - e^{-\frac{t}{k_m}}) \cdot \mu(t) \tag{4}$$

V_m is the maximum voltage of the power supply, while ω_n corresponds to nominal angular speed of a motor which can be found in catalogue. $k_m = 2[s]$ is the mechanical constant of the first order approximation of the system. It is worth noting that $\omega_r(t)$ in the real system is far more complex and can't be expressed as the first order system response. Determine K_e so that current in stationary state satisfies equation:

$$\lim_{t \to \infty} i_s(t) \leq 115A \tag{5}$$

[6] $\mu(t)$ is the step function

Table 5: Table of motors to choose from. "R" is short for rated

Code	R power [kW]	Rated voltage [V]	R. speed [1/min]	Max speed [1/min]	R. torque [Nm]	Max torque [Nm]	Max power [kW]	R. current [A]	Efficiency [%]	Mass [kg]	Price [EUR]
5000	45	320	3200	4000	134	268	90	165	96.8	83	3700
5001	45	320	3500	4000	123	260	100	171	94.8	152	4100
5002	45	430	6000	6500	70	140	100	105	94.8	122	4360
5003	50	320	3500	4000	136	300	110	190.3	94.8	160	4500
5004	50	320	6000	6500	78	300	160	168	94.9	126	4230
5005	55	320	6000	6500	86	180	120	168	94.9	130	4570
5006	60	200	3500	4000	164	360	130	300	95	170	5000
5007	60	320	6000	6500	94	200	130	189	95	136	4805
5008	80	320	3200	4000	238	477	160	294	97	196	5400
5009	80	360	3500	4000	218	450	180	244	97	210	5555
5010	80	200	2800	3300	273	600	160	400	97.2	178	5450
5011	85	430	4500	5300	180	400	170	198	95	214	5640
5012	90	360	3500	4000	246	540	200	275	97	230	5820
5013	90	320	3200	4000	268	537	180	330	97	312	6130
5014	95	240	3200	3800	284	700	210	396	96.3	217	6210
5015	100	400	7000	8300	136	300	200	250	96.7	187	6340
5016	100	360	3500	4000	273	600	230	305	96	240	6500
5017	120	320	3200	4000	358	716	240	441	97.2	230	6860
5018	120	360	3500	4000	327	700	260	366	96.3	280	6980
5019	130	360	3500	4000	382	850	300	428	96.4	310	7215
5020	140	240	3300	3900	405	900	280	583	97.1	279	7300
5021	140	430	7400	8700	181	400	270	326	96.3	235	7290
5022	150	360	3500	4000	409	900	350	458	96.6	330	7465
5023	160	360	3500	4000	437	1000	360	488	96.6	350	7800
5024	180	240	3400	4000	506	1200	380	750	96	348	7950
5025	180	440	7600	9000	226	500	350	409	95	285	8190
5026	200	360	4500	5000	424	1000	420	578	96.7	350	8555
5027	240	400	4000	4700	573	1300	480	600	96.3	412	9400
5028	280	400	5000	5900	535	1200	560	700	96.3	440	9990
5029	320	360	3800	4500	804	1800	640	889	96.3	460	10670
5030	360	400	3500	4100	982	2300	740	900	96.3	500	11300

Motor controllers

In addition to the gearbox, each motor has to have a motor controller. The controller determines the maximum voltage and power output to a given motor and the efficiency of regenerative braking. Below is the table of the controllers you can choose from.

Table 6: My caption

Code	R. power	Voltage in		Voltage out		Reg. efficiency	Price
	[kW]	Min. [V]	Max. [V]	Min. [V]	Max. [V]	[%]	[EUR]
6000	80	300	400	200	350	50	3230
6002	80	300	400	200	350	80	3570
6003	80	450	600	320	500	60	3350
6004	80	450	600	320	500	90	3740
6005	100	300	400	200	350	50	4440
6006	100	300	400	200	350	80	4800
6007	100	450	600	320	500	60	4590
6008	100	450	600	320	500	90	5000
6009	140	300	400	200	350	50	5710
6010	140	300	400	200	350	80	6090
6011	140	450	600	320	500	60	5890
6012	140	450	600	320	500	90	6320
6013	180	300	400	200	350	50	7040
6014	180	300	400	200	350	80	7440
6015	180	450	600	320	500	60	7250
6016	180	450	600	320	500	90	7700
6017	220	300	400	200	350	50	8430
6018	220	300	400	200	350	80	8850
6019	220	450	600	320	500	60	8670
6020	220	450	600	320	500	90	9140
6021	300	300	400	200	350	50	9880
6022	300	300	400	200	350	80	10320
6023	300	450	600	320	500	60	10150
6024	300	450	600	320	500	90	10640

Battery pack

For the next part, you will have to design battery pack for the truck. Below in the table 7 is the list of the 18650[7] cells you can choose from.

Design a battery pack such that during the conditions given in the previous task (motor test conditions) the battery doesn't get above $60^\circ C$ and that the vehicle autonomy on a flat road is at least $200km$ given the ambiental temperature of $20^\circ C$.

In the spreadsheet, you can define which cell you will be using and number of series and parallel cells. Battery pack can be split in up to 3 slots. There is limit of $1000kg$ per slot. If you go over that limit, you will be punished with 20% reduction of the final points you get. This gives you flexibility to distribute weight more evenly across tires. All the slots need to have the same number of series batteries, but the number of parallel cells can be changed.

Battery thermal performance is modeled as if each cell was standing in ambiental temperature air, in an infinite array of cells standing up - only top and bottom surfaces are dissipating heat. There are 2 modes of cooling, convective dissipation, and radiative dissipation.

The coefficient of convective heat transfer is $15W/m^2K$, and the emissivity of the cell surface is 0.8. Ambiental temperature is $20^\circ C$

If your batteries get over $60^\circ C$ during the endurance tests, you will be punished with 20% reduction of the final points you get.

[7] 18650 is the name of a specific cell type

Table 7: Table of battery specifications and prices

Code	Mass [g]	Price [EUR]	Rated capacity [mAh]	C rate	Cycles Cap >80%	Internal resistance [mOhm]
4000	48	3.77	2500	8	250	30
4001	47	4.26	3000	7	250	30
4002	48.5	3.00	3000	2	150	110
4003	46	3.69	2850	3	400	40
4004	44	2.05	2600	1	450	100
4005	49	3.95	3500	3	250	60
4006	30	1.82	1000	1	100	150
4007	45	4.05	2500	8	135	22
4008	48	2.51	2900	3	300	70
4009	46	3.69	3000	3	200	20
4010	47	2.79	1600	19	150	19
4011	50	4.25	2100	14	350	19
4012	44.3	4.10	2600	8	200	20
4013	46	5.44	3000	10	100	20
4014	48	2.96	3180	1	300	60
4015	48.5	4.04	3400	2	175	50
4016	47	3.01	2900	3	210	40

Endurance test

In the endurance there are 2 tests, we will tests autonomy of your truck. In these tests we will determine how far can your truck drive before the battery is empty.

Endurance 1 Flat road at $100km/h$

Endurance 2 Hill road - the truck drives on a road defined by equation below

$$angle = 5 * \sin(2 * pi * 1/4000 * x)$$

Where the x is the distance traveled in meters.

2.3 Grading

As it was mentioned before, we will measure the performance of your truck against other teams, and they will be graded based on their relative performance to other team's trucks. There are 7 categories, in which you can get up to 100 points for first place.

Depending on the on the ranking, the number of points is determined via formula:

$$\text{Points}(rank) = \text{round}\left(e^{(21-rank)/4.3429}\right)$$

That calculated gives the next grading scheme:

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Points	100	79	63	50	40	32	25	20	16	13	10	8	6	5	4	3	3	2	2	1

The categories and their respective weights are::

- 20% Price of all components used
- 20% Endurance on a hill road
- 15% Truck mass
- 15% Breaking distance on a $-10°$ road from $100km/h$
- 10% Endurance on a flat road
- 5% Top speed on flat road
- 5% Top speed on 5° slope
- 10% solution of side task 2.2.1

After each rank and points are given and scaled based on percentage given category earns, points are summed up and final team ranking is calculated.

Day 3

3.1 Introduction

Center of the truck control system is embedded system based on microcontroller with very simple architecture. Your goal is to study the truck control system architecture and provide solution for system control. Some of the problems you will encounter are:

- sketching block diagrams of embedded systems
- developing discrete models for PI controllers
- writing controller submodules in C programming language
- amending control parameters based on simulation results

3.1.1 Microcontroller unit (MCU)

Microcontroller is based on 32-bit STEM processor architecture which is typical RISC processor architecture. Additionally, microcontroller is equipped with following peripherals:

- general purpose timer (GPT)
- general purpose input output (GPIO)
- accelerometer peripheral unit (APU)
- analog-digital converter (ADC)
- simple interrupt controller (SIC)
- digital-analog converter (DAC)
- static random access memory (SRAM)

STEM processor has FPU (*floating-point unit*). Access to peripheral units is done through memory access which is regulated by SMC (*simple memory controller*).

3.1.2 General Purpose Timer (GPT)

GPT is timer with only one **upcounter** mode of operation. Value of GPT_CNT register increases until it reaches value of GPT_LOAD register. In case of:

$$GPT\_CNT >= GPT\_LOAD \tag{6}$$

timer expired event is triggered which causes following:

- value of the GPT_CNT register is 0x00000000
- timer signals interrupt to SIC (through `GPT_EVT` signal, if GPT_CTRL[1] bit is 0
- timer is stopped and masked (i.e. GPT_CTRL = 0x00000002)

It is worth noting that counter keeps on counting while GPT_CTRL[0] equals 1.

Table 8: General Purpose Timer register bank

Register	Type	Initial state	Description
GPT_ID	R	0x0200beef	Contains ID of GPT silicon
GPT_VER	R	0x00000001	Contains version of GPT silicon
GPT_LOAD	R/W	0x00000000	Contains load value for counter
GPT_CNT	R/W	0x00000000	Contains current value of counter
GPT_CTRL	R/W	0x00000000	If least significant bit (GPT_CTRL[0]) is 1, timer is enabled. Additionally bit GPT_CTRL[1] masks interrupt.
GPT_FREQ	R	0x000003E8	Frequency of timer input signal in Hz.

Note. In tables type of register can be read-only (**R**) or read and write (**R/W**).

3.1.3 General Purpose Input Output port (GPIO)

GPIO port has 32 independent digital input/outputs. It is configured through 2 registers as shown in table 9.

Table 9: General Purpose Input Output

Register	Type	Initial state	Description
GPIO_CTRL	R/W	0x00000000	Digital I/O mode, 0 corresponds to input, 1 corresponds to output
GPIO_DATA	R/W	0x00000000	Digital I/O state

3.1.4 Accelerometer Peripheral Unit (APU)

Accelerometer Peripheral Unit (APU) gives information about truck acceleration in in horizontal and vertical direction with reference to **global coordinate system**. Acceleration is described as 32-bit signed integer. LSB (*least significant bit*) corresponds to acceleration of $0.01ms^{-2}$. APU registers are shown in the table 10.

Table 10: Accelerometer Peripheral Unit register bank

Register	Type	Initial state	Description
APU_CTRL	R/W	0x00000000	control register, APU_CTRL[0] enables accelerometer
APU_X	R	0x00000000	acceleration in horizontal direction as 32-bit signed integer
APU_Y	R	0x00000000	acceleration in vertical direction as 32-bit signed integer

3.1.5 Analog-to-digital converter (ADC)

Analog-to-digital converter (ADC) converts voltage signal to digital representation with respect to reference voltage. In this system ADC is connected to battery temperature sensor. LSB of ADC_DATA register corresponds to $0.1C$.

Table 11: Analog-to-digital converter register bank

Register	Type	Initial state	Description
ADC_CTRL	R/W	0x00000000	control register, ADC_CTRL[0] enables analog-digital converter
ADC_DATA	R/W	0x00000000	converted data with respect to reference voltage

3.1.6 Digital-to-analog converter (DAC)

Digital-to-analog converter (DAC) converts data to analog voltage representation with respect to reference voltage. In this particular system LSB of DAC corresponds to $1mV$.

Table 12: Digital-to-analog converter register bank

Register	Type	Initial state	Description
DAC_CTRL	R/W	0x00000000	control register, DAC_CTRL[0] enables digital-analog converter
DAC_DATA	R/W	0x00000000	data to convert with respect to reference voltage

3.1.7 Simple Interrupt Controler (SIC)

Microcontroller processor has only one output interrupt line (`IRQ`). Processor is connected to SIC with `IRQ` interrupt line. SIC has 4 priority ordered peripheral interrupt lines (`ch_1`, `ch_2`, `ch_3`, `ch_4`). When certain peripheral interrupt line is active, register `SIC_ACT` contains peripheral interrupt line index. Detailed register map of SIC is shown in table 13. The first interrupt line (`ch_1`) is connected to `GPT_EVT` line.

Table 13: Simple interrupt controller register bank

Register	Type	Initial state	Description
SIC_ID	R	0x0100beef	Contains ID of SIC silicon
SIC_VER	R	0x00000001	Contains version of SIC silicon
SIC_ACT	R	0x00000000	Contains index of active interrupt line
SIC_ACK	R/W	0x00000000	Write operation to this register acknowledges interrupt
SIC_CTRL	R/W	0x00000000	Control register. Writing 1 causes interrupt line IRQ to be masked

3.1.8 Static Random Access Memory (SRAM)

This memory should be used as storage for global variables. Persistence of other global variables outside of SRAM is not guaranteed. Size of SRAM is limited to 512 bytes. After reset this memory is initialized automatically with zeros.

3.1.9 Hardware Abstraction Library (HAL)

You will be provided with appropriate HAL for this MCU. It contains header files which will ensure you access to all relevant registers of certain peripheral unit. These files are:

- adc.h
- apu.h
- dac.h
- gpio.h
- gpt.h
- pi_reg.h
- sic.h
- sram.h

Every file contains peripheral unit base address and appropriate offsets for accessing registers. SRAM is byte oriented random access memory and can be used for storage of variables of various types (be careful with pointer arithmetics and data types).

3.2 Tasks

3.2.1 Embedded system scheme

For this task you should provide sketch of the embedded system based on MCU described in previous section. Your sketch should have following properties:

- connections between MCU processor and MCU peripherals (do not forget about memory controller)
- mark system inputs and outputs
- ensure GPIO line 17 is connected to indicator LED (assume **open drain** configuration of GPIO pin 17 and include electrical scheme with adequate external hardware)

3.2.2 Initialization functions and interrupt handling

Create initialization functions for microcontroller peripherals which should enable utilization of system hardware. Your solution should be organized in two files:

- init.c
- init.h (not obligatory, but recommended)

Files above should contain implementation and declaration of `void init_mcu(void)` function. Function `init_mcu` should contain initialization of peripherals based on registers listed in corresponding tables. Additionally, file `init.c` should contain `void irq_handler(void)` function which should handle interrupts and arm timer for subsequent interrupts.

At this stage, system functionality is defined with following list:

- system keeps high logical level on GPIO line 17 (which should drive indicator LED), while other GPIO lines are low
- system receives periodic interrupts (every $5ms$) through IRQ line from SIC, caused by GPT
- system DAC and ADC are ready for operation
- system APU unit should be ready for operation

3.2.3 Handling timer interrupts for PI speed regulation

In this task, you should implement speed regulation with parallel implementation of PI regulator. Output of PI regulator is stored to DAC_DATA register which controls amount of voltage applied on motor. Regulator constants can be found in file `pi_reg.h`. Input of the regulator is angular speed error (in **rpm**) of motor axle. You should implement discrete PI regulator with $5ms$ as discretization time. As part of the solution, you should provide:

- block scheme of PI regulator
- mark inputs and outputs and add connections to system sketch from the **first task** (3.2.1)
- PI regulator discrete and continuous transfer function and appropriate recursive equation.

Your solution should be organized in two files:

- speed_reg.c
- speed_reg.h (not obligatory, but recommended)

Files above should contain implementation and declaration of `void pi_reg(void)`.

Note. Data from APU contains accelerations in horizontal and vertical direction of whole truck. File `pi_reg.h` contains appropriate constants for unit conversion.

3.2.4 Handling timer interrupt for speed control

In this task the high level control of the system should be implemented. So far, you only regulated the speed of the truck. However, the truck route properties were not taken into consideration. Consider various scenarios of roads truck can encounter. Speed regulation from previous task ensures that speed is the same as reference speed. In this task you should ensure that reference speed of the truck is optimal based on its position on some road. This can be done by changing reference speed according to information about position which you is derived from accelerometers.

Your solution should be organized in two files:

- control.c
- control.h (not obligatory)

You should write `float change_ref_speed(float x0, float y0, float x, float y)` function. This function will be periodically called from `irq_handler` every 1000 *ms* and should return new reference speed. You should devise algorithm for changing reference speed. The goal is to optimize distance truck can pass in limited time. Additionally, you should optimize power consumption.

Note. Points you get on this task depend on your rank.

3.3 Grading scheme

You can achieve up to **45 points** in 4 tasks:

- 3.2.1 **5 points**
- 3.2.2 **10 points**
- 3.2.3 **15 points**
- 3.2.4 **15 points**

Every written submodule is worth **10 points** and is graded in the following fashion (code in tasks 3.2.2, 3.2.3):

- 1 point if module can be compiled successfully
- 2 points if module does not interfere with normal operation of MCU (i.e. does not cause hard fault)
- 2 points for description of operation, pseudo-code or block diagram
- 5 points if module function is correct

Correct schemes, formulas and equations for tasks 3.2.1, 3.2.3 bring **5 points**.
Points of 3.2.4 depend on rank by distance (30 %) and battery state of charge (70 %).