AUTOMATED DRIFT CONTROL OF A FOUR-WHEEL DRIVE VEHICLE

The goal of this challenge is to drive a given vehicle on a short, curved track in such a way that the vehicle drifts as *impressively* as possible. In the current context, drifts correspond to open-loop unstable equilibria in which the rear tires are saturated and are typically characterized by large vehicle sideslip angles. In order to achieve this, contestants must design a control algorithm which induces and maintains these unstable equilibria throughout the course of the given track. The vehicle model provided, described in detail below, reflects a drift car powered by four electric motors.

1. VEHICLE MODEL

The vehicle model has been simplified to limit the scope of the challenge. The suspension consists of lumped mass kinematic linkages which allow each wheel to translate vertically with respect to the chassis body. Vehicle mass and inertia are represented by lumped values at the center of gravity. Parameters of interest are given in the table below.

Parameter	Value	Unit
Mass	1300	kg
I_{xx}	200	kg.m^2
I_{yy}	1300	kg.m^2
I_{zz}	1400	kg.m2
Wheelbase	2.745	m
Front Track	1.7	m
Rear Track	1.684	m
CoG Distance from Front Axle	1.4	m
CoG Height	0.35	m
Steering Gain[1]	12	-
Wheel Radius[2]	0.254	m
Wheel Inertia about Spin Axis	1	kg.m^2

[1] ratio of steering wheel angle to wheel angle

[2] radius of front and rear wheels is equal

Table 1: Vehicle Parameters

1.1 Actuators

The actuators available for control are four motors, one for each wheel, and the steering wheel. Torques are applied directly to the wheel and steering wheel angle is transmitted to the front wheels through a constant steering ratio. The motors can apply both accelerating and braking torque. All actuators have a realistic range of operation and a maximum slew rate as listed in the table below.

Motor	Torque		Maximum Power
	Maximum	Slew Rate	
Front	1200 Nm	60 Nm/ms	150 kW
Rear	2000 Nm	60 Nm/ms	200 kW

(a)

Maximum Value	Maximum Slew Rate
720°	500°/s

(b)

Table 3: Actuator Limits; (a) Motor, (b) Steering

1.2 Tire Model

Since tires are the major source of non-linearity and instability in a vehicle, the detailed Magic Formula model as described in *Pacejka, H., & Besselink, I. J. (2012). Chapter 4, Tire and Vehicle Dynamics* is used in the vehicle model. The tire coefficients are provided in the Adams/Tire standard .tir format. The file is human-readable and can be parsed by many simulation tools. The tire model is also provided as an FMU (*described in section 4*) which accepts the slip ratio and slip angle as input and produces longitudinal and lateral forces. The normal load and wheel centre velocity are available as parameters. Details are provided in the table below.

Name	Description	Type	Unit
lambda	Slip Ratio[1]	Input	-
alpha	Slip Angle[1]	Input	°
F_z	Normal Load	Parameter	N
V_x	Wheel Center velocity	Parameter	m/s
F_x	Longitudinal Force	Output	N
F_y	Lateral Force	Output	N

[1] *as defined in Pacejka, H., & Besselink, I. J. (2012). Tire and Vehicle Dynamics*

Table 2: Tire fmu variables

2. TRACK GEOMETRY

The track consists of four segments. It starts with a 50m straight, followed by 180° left and right-hand constant radius turns. Radii have been chosen such that at 60km/h, tire lateral forces are in the linear and saturation regions for the first and second turns respectively. The circuit ends with a 50m straight and has a constant width of 12m.

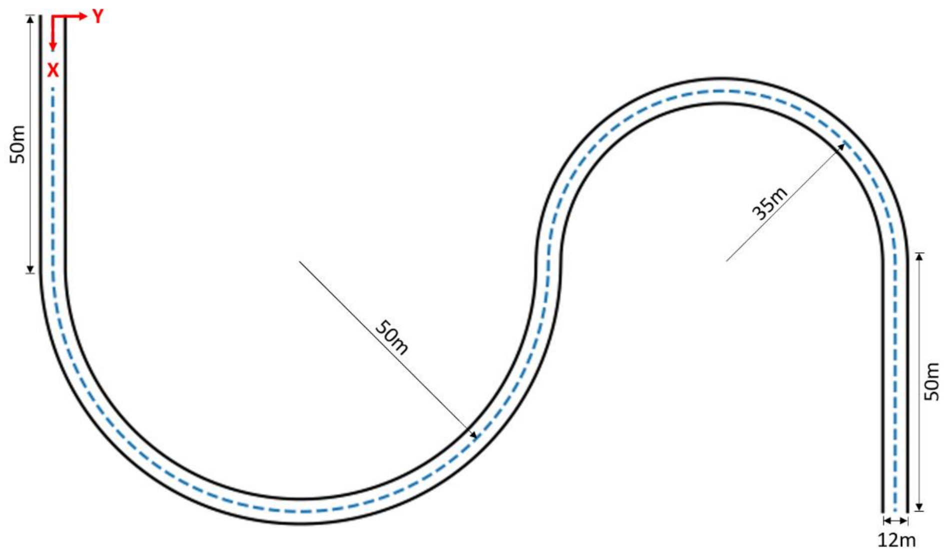

Fig. 1: Track geometry

The first straight segment is meant to allow the vehicle to accelerate to a higher speed if desired. It also allows the vehicle to initiate drift – to increase the sideslip angle until the vehicle settles in an unstable equilibrium.

The simplest way to achieve this is by “power oversteer”. While cornering, applying a large amount of torque to the rear wheels will cause the rear wheels to overslip and the lateral force produced by the rear axle will decrease. The vehicle sideslip angle will then increase past the normal cornering value. A quicker and more complex way is the “Scandinavian flick”, which takes advantage of longitudinal weight transfer to unload the rear axle and reduce the rear lateral force more quickly.

The first cornering segment is a test of the control algorithm’s ability to maintain drift. Ideally, the vehicle would drive at a high sideslip angle with its rear wheels nearly touching the outer edge of the track.

In the short road segment where the left-hand turn changes into the right-hand turn, the vehicle will need to reverse the sign of its sideslip angle to successfully complete the second corner. This can be achieved by stabilizing the vehicle, then initiating drift again on the right-hand turn. However, this will result in a low reward. A higher reward can be achieved by taking advantage of the vehicle’s inertia to spend as little time as possible in the stable equilibrium.

The second cornering segment is much like the first, but the smaller radius means that the vehicle will need to move at a lower speed to maintain its trajectory.

3. REWARD FUNCTION

To judge the drift maneuver objectively, the following reward function will be used:

$$\int_0^t (a_1\beta^2 + a_2\Delta y^2 + a_3 {v_x}^2)dt$$

β – vehicle sideslip angle (rad)

Δy – lateral offset of Center of Gravity from track centerline (m)

v_x – vehicle speed (m/s)

Weight	Value
a_1	1
a_2	0.125
a_3	0.0075

Table 4: Reward function weights

The reward function is structured to reflect real-world drift competitions. It is composed of three components; one each for vehicle sideslip angle, lateral offset of center of gravity from track centerline and vehicle speed, with a_1, a_2 and a_3 being the corresponding weights. A higher reward function value indicates better control algorithm performance.

Vehicle sideslip angle is of primary interest during a drift and equilibria at higher sideslip angles are harder to maintain. Additionally, a higher lateral offset is rewarded as the margin for yaw oscillations is reduced when the vehicle is closer to the edge of the track, making the control task more challenging.

Vehicle speed is included for similar reasons as sideslip angle - equilibria at higher speeds are less stable. Weights have been chosen to respect this order of importance and relative magnitudes of the metrics. For example, driving with maximum lateral offset and zero sideslip angle, although much easier to achieve, would produce a lower reward compared to maintaining a low lateral offset and high sideslip angle.

The reward function value is considered valid only under the following conditions:

1. All wheels must be on the track
2. Longitudinal speed must be above a threshold of 30km/h

If these conditions are violated at any time during the simulation, the entire simulation run is considered invalid for the purposes of evaluating the control algorithm performance.

4. USING THE VEHICLE MODEL

The vehicle model is provided as a Functional Mockup Unit (FMU), which is an exchange format based on the Functional Mockup Interface (FMI) standard. FMUs can be imported into various simulation environments (*see http://fmi-standard.org/tools/ for a list of supported tools*). To re-use your work in Stage B of the challenge, Matlab/Simulink is recommended.

With release R2017b and beyond, Simulink supports simulation and integration workflows using the Functional Mockup Interface (FMI). The FMI import block allows you to import Functional Mockup Units (FMU) into a Simulink model. If you are using an earlier release and it is not possible for you to upgrade your licence, you might still be able to use Simulink as your simulation environment. For more information, please refer to the MATLAB Central answer on this topic here. When importing the provided FMU, we recommend using the Co-Simulation mode as it allows the model to be simulated with a Fixed-step Discrete solver (which would be required to run the model on the RDP).

The simulation is initialized with the vehicle in the middle of the road with its rear axle along the track Y axis and longitudinal speed of 60km/h.

4.1 Model Outputs

The model output consists of relevant vehicle states, value of the reward function and the simulation state. A list of outputs along with description and units is provided in the table below.

Variable	Description	Unit
pos_X	X coordinate of center of gravity in global frame[1]	m
pos_Y	Y coordinate of center of gravity in global frame	m
vel_Long	Longitudinal velocity of vehicle in vehicle frame[2]	m/s
vel_Lat	Lateral velocity of vehicle in vehicle frame	m/s
acc_Long	Longitudinal acceleration of vehicle in vehicle frame	m/s^2
acc_Lat	Lateral acceleration of vehicle in vehicle frame	m/s^2
acc_Vert	Vertical acceleration of vehicle in vehicle frame	m/s^2
angRate_Roll	Roll rate of vehicle in vehicle frame	rad/s
angRate_Pitch	Pitch rate of vehicle in vehicle frame	rad/s
angRate_Yaw	Yaw rate of vehicle in vehicle frame	rad/s
offset_Lateral	Lateral offset of center of gravity from track centerline	M
wheelTorque_act_IJ	Actual torque applied to the corresponding wheel[3]	Nm
wheelSpeed_IJ	Angular speed of the corresponding wheel	rad/s
function_Reward	Reward as described in section 3	-
Simulation_State	Simulation state as defined in section 4.2	-

[1] *global frame is identical to track frame indicated in Fig. 1*
[2] *vehicle frame is oriented according to the ISO 8855 standard, with x, y, and z axes pointing forward, leftward, and upward respectively and origin at the vehicle center of gravity*
[3] *torque produced after slew rate limit and power limit is applied*

Table 5: List of Model Outputs

4.2 Simulation State

The following table defines the simulation states:

Code	State	Trigger
0	Simulation is running	-
1	Vehicle has completed the circuit	Center of gravity crosses finish line
2	Vehicle has left the road	Projection along Z axis of at least one of the four wheel centers is outside the track
3	Vehicle speed too low	Vehicle longitudinal speed falls below 30 km/h

Table 6: Simulation State

To allow contestants to use the vehicle model for testing, the simulation does not stop when the simulation state becomes nonzero. Instead, the reward function output is frozen and the simulation keeps running.

If the road geometry is ignored, the simulation model can be used for model fitting, controller tuning etc. When using the model for this purpose, the vehicle can be assumed to ply on an infinite flat surface.

4.3 Example Model

An example Simulink model is provided to demonstrate the usage of the vehicle model and help contestants get acquainted with model inputs and outputs. It can be simulated in three modes, each corresponding to a simulation state.

In the first mode, the vehicle follows the centerline at a fixed speed. As the simulation runs, the reward function continually increases. When the vehicle completes the circuit, the simulation enters state 1 and the reward function value stops increasing. The value of the reward function at this point would be considered the control algorithm’s score for grading purposes.

In the second mode, the vehicle drives straight and leaves the track. When the vehicle’s front-right wheel leaves the track, the simulation enters state 2 and the reward function stops increasing. The value of the reward function at this point would be disregarded for grading purposes – the simulation did not terminate with state 1 so the simulation run is considered invalid.

In the third mode, the vehicle brakes with all four wheels. Soon after simulation start, the vehicle speed falls below the minimum, the simulation enters state 3 and the reward function stops increasing. This simulation run would also be considered invalid for grading purposes.

5. CHALLENGE RULES

Only students enrolled in an institute of higher education (B.Sc, M.Sc or Ph.D. level) are eligible to participate in the challenge. Students can form teams and the number of members per team is not restricted. Contestants will be asked to provide a proof of student status issued by their institution.

The challenge will consist of two stages. This document describes Stage A, in which the performance of a control algorithm as applied to the vehicle model provided will be judged in accordance with the reward function described above. The top three teams will proceed to Stage B, which will take place during the MED2018 conference in Zadar. Travel and accommodation costs will be covered for up to 2 members per team.

Control algorithms will be ranked according to the value of '*function_Reward*' produced by the FMU when '*simulation_State*' changes from 0 to 1, with larger reward function values ranked higher. Simulations where '*simulation_State*' output assumes any value other than 0 and 1 will not be evaluated.

To participate, teams must register using the registration link listed on the "Industrial Challenges" section of the Conference website. Teams will be asked to provide a contact e-mail, which will be used to communicate which repository belongs to which team.

All resources related to the challenge are available in the following Github repository: https://github.com/rimac-technology/med2018-automotive-challenge

To start working on Stage A of the challenge, teams must fork this repository. Control algorithms will be evaluated by running the *initialize.m* script, followed by the *solution.slx* Simulink model, both located in the root of the repository. The last commit in the fork's **master** branch before the deadline will be evaluated. Results will be made public no later than 2 weeks after the completion of Stage A and the top three teams will proceed to Stage B.

Stage B will consist of implementing the control algorithm in a format suitable for execution on the dSpace MicroAutoBox II Rapid Development Platform (RDP). The Stage B teams will be provided with a software package which will allow them to adapt their control algorithm in Simulink and generate code for the RDP. In addition, teams will receive a specification for the CAN communication between the RDP and the vehicle, which the control algorithm will need to adhere to.

At the conference proper, teams will be provided with a differently parametrized vehicle model and a different track geometry description. In the time available, contestants will need to re-tune their control algorithm and run it on the RDP, with the vehicle implemented on a dSpace SCALEXIO Hardware In the Loop (HIL) test rig and connected to the RDP via CAN Bus. The same evaluation criteria as Stage A will be used and the highest-scoring team will be declared the challenge winner during the conference gala dinner.