Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Adaptive cruise control algorithm development

Jan Švancar

Supervisor: Ing. Tomáš Haniš, PhD
Field of study: Cybernetics and robotics
May 2020

Acknowledgements

I want to thanks to CTU for being a good alma mater to me and my family and my girlfriend for being so supportive to me.

Declaration

I declare that I have prepared the submitted work independently and that I have listed all the used literature.

In Prague, 22. May 2020

Abstract

Purpose of this work was to develop testing framework for Adaptive Cruise Control algorithm in virtual environment and then design Adaptive Cruise Control algorithm with demands on safety and comfort of passengers to be tested.
To create testing framework was used vehicle dynamics model to simulate real vehicle behaviour, Automated Driving Toolbox for Matlab and Simulink and ROS2 network.
Adaptive Cruise Control algorithm was designed with speacial focus on safety and comfort of the passengers. It consists of three subsystems, Target Vehicle Follow system, which leads vehicle to desired spacing between vehicles and keeps it, Cruise Control system, which limit force output of Target Vehicle Follow system to prevent crossing maximal velocity and Cut-in system to handle dangerous situation in traffic.

Keywords: testing framework, virtual environment, ROS2 network, Adaptive Cruise Control, ACC, Cruise Control, CC, Cut-in, automotive

Supervisor: Ing. Tomáš Haniš, PhD
Katedra řídící techniky,
Karlovo náměstí 293/13,
Praha 2

Abstrakt

Účelem této práce bylo vyvinout testovací rozhraní pro Adaptivní tempomat ve virtuálním prostředí a vyvinout algoritmus Adaptivního tempomatu s požadavky na bezpečnost a komfort pasažérů k otestování.
K vytvoření testovacího rozhraní bylo použito modelu dynamiky vozu pro simulaci reálného vozidla, Automated Driving Toolboxu pro Matlab a Simulink a sítě ROS2.
Algoritmus adaptivního tempomatu byl navrhován, aby se zvláštním zaměřením na bezpečnost a komfort pasažérů. Skládá se tří podsystémů, a to systém Sledovaní zaměřeného vozidla, který vede vozidlo k požadované vzdálenosti mezi vozidly, tempomatu, který omezuje výstupní sílu systému Sledování zaměřeného vozidla pro předejití překračovaní maximální rychlosti, a systému Cut-in, který je určený pro zvládnutí nebezpečných situací v provozu.

Klíčová slova: testovací rozhraní, virtuální prostředí, síť ROS2, Adaptivní tempomat, ACC, tempomat, CC, Cut-in, automobilita

Překlad názvu: Vývoj algoritmů adaptivního tempomatu

Contents

Figures

Tables

Chapter 1

Introduction

There were two main motivations to do this work. At first it was developmnet of testing framework for testing Adaptive Cruise Control algorithm in virtual environment. Framework uses ROS2 network to comunicate between nodes. This preparation of distributional network can be in future used to connect multiple vehicles into the network and share driving information between each other vehicle or deploy the testing framework to vehicle simulator.
Testing framework uses ROS2 network to distribute information between various parts of system. ROS2 network is second generation of widely used DDS platform ROS. Thanks to ROS2 network testing framework could be easily modified and extended. To implement it the ROS Toolbox ([Inc20c]) was used.
To create testing framework vehicle environment, which for its function uses vehicle dynamics model and Scenario and Radar blocks from Matlab and Simulink Automated Driving Toolbox ([Inc20a]), was designed.
Vehicle model used in vehicle environment was adopted from Franklin [FPEN10]. Vehicle model is first order model with transfer function from froce to vehicle velocity. The additional integrator was added to vehicle model. The additional integrator integrates velocity to get position of the vehicle.
Next part is Scenario reader block, which gets infromation from predefined scenarios to create testing environment for vehicles. Only Ego vehicle can be directly controlled and any other vehicle participating in the scenario have predefined velocities and trajectories.
Radar block generates detections accordingly to infromation, which Radar block gets from Scenario reader block. Radar is used to detect vehicle and provide information to control system. Radar unit also measures relative velocities of vehicles, which is important fot the ACC case.
Second motivation of the thesis was to produce Adaptive Cruise Control system, which will fulfil both, safety requirements and provide passengers

comfort even in dangerous traffic maneuvers.
Adapative Cruise Control system is in fact simple mechanism, which keeps predefined distance between vehicles. However, to guarantee safety of passengers and certian level of comfort, algorithm starts to be complex. The ACC system is divived into three subsystems, Target Vehicle Follow system, Cruise Control system and Cut-in system.
Target Vehicle Follow system is used to lead vehicle into desired spacing between vehicles and then keep the distance. Accordingly to Rajamani [Raj11a] Constant Time Gap policy was used as spacing policy. Purpose of CTG policy is to control spacing in dependency on velocity of Ego vehicle.
To keep passengers comfort, while approaching desired spacing, the relative speed reference signal is calculated based on Ego to Target vehicle distance. This method was adopted from Rajamani [Raj11b].
Next part of ACC system is Cruise Control system. The Cruise Control system provide tracing of vehicle constant velocity once the Target Vehicle Follow system is not active.
Last part of ACC system is Cut-in system. This system is designed to handle potentionally dangerous situations with preservation of comfort of passengers. The Cut-in system is active once the Ego vehicle to Target vehicle is significantly higher compared to Target Vehicle Follow system reference value. The Cut-in system provide safety decelaration of Ego vehicle with minimal possible deceleration value to maintain passenger comfort.
Part of Cut-in system is Emergency breaking system, which act only in situation, when distance between vehicles is lower than desired spacing between vehicles.

Chapter 2

Testing Framework for Adaptive Cruise Control System

2.1 Introduction

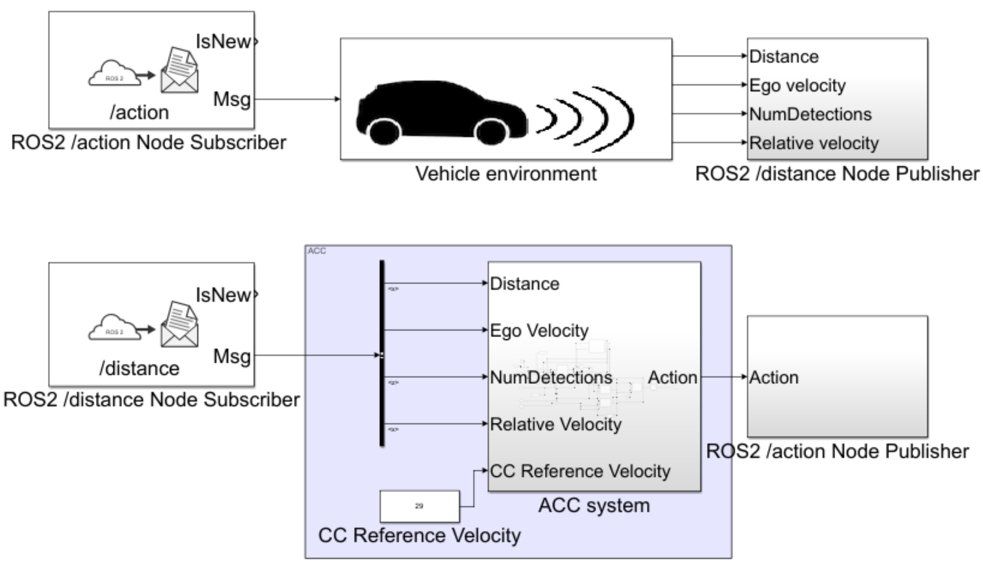

Figure 2.1: Diagram of testing framework

Testing framework was designed for simulation and testing of Adapative Cruise Control system in virtual environment. It consists of three parts (Fig. 2.1).

- **Vehicle environment** - This part is used to simulate vehicle dynamics, read predefined scenarios and simulate behaviour of vehicle sensors (radar, camera, etc.).
- **ROS2 network** - It's Data Distribution Service platform, which is used to distribute data into certain parts of Testing Framework.
- **ACC system** - It's one of ADAS systems. It is a control algorithm used to keep safe distance between vehicles on the road.

For implementation of a testing framework in MATLAB and Simulink environment was used Automated Driving Toolbox [Inc20a] and ROS Toolbox [Inc20c].

There are few therms used further in text explained bellow

- **Ego vehicle** - Vehicle with radar attached to it and which velocity and position is controled by ACC. This is the tested object.
- **Target vehicle** - Vehicle which executes testing maneuvers to test functionality of ACC, namely Target Vehicle Follow and Cut-in maneuvers. Velocity and position of Target vehicle can't be changed externaly and is given by prepared scenarios.

2.2 Vehicle Environment

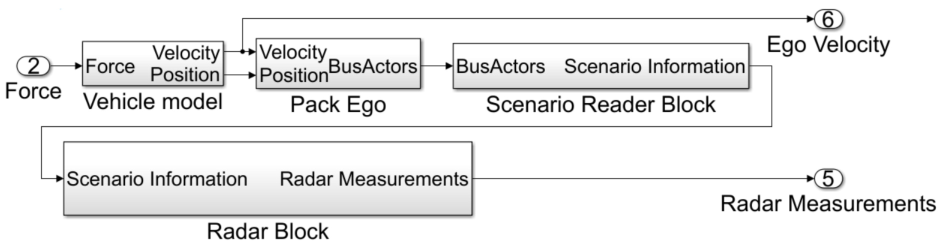

Figure 2.2: Diagram of vehicle environment

Vehicle environment is composed of several parts.

- **Vehicle model** - Ego vehicle dynamics
- **Pack Ego Actor** - Creates structure of information from vehicle model.
- **Scenario Reader** - It process information from predefined scenarios in time.

- **Radar** - Generates radar detection according to setup of radar and scenarios infromation from Scenario Reader.

This architecture (see Fig. 2.2) allows to simulate and dynamically change a behaviour of an EGO vehicle. However, applied tools (Scenario Reader and Radar from Automated Driving Toolbox) allows only to directly control EGO vehicle and unfortunately behaviour of Target vehicle is given by designed source scenario and can't be dynamically changed. This functionality is suitable to simulate a behaviour of ACC driven Ego vehicle with one other vehicle (Target vehicle) on the road.
Parts of the vehicle environment will individually described in next sections.

2.2.1 Ego vehicle dynamics

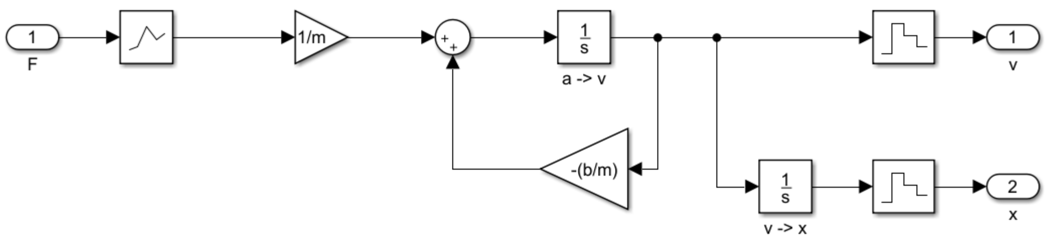

Figure 2.3: Ego vehicle dynamics model

Continous model of vehicle longitudal dynamics (Fig. 2.3) was used to simulate behaviour of Ego vehicle like a real vehicle. This model was also used to design longitudal velocity regulators for implementation of ACC functionality. The vehicle model is a simple car model, which was adopted from Franklin [FPEN10] in this form

$$m\ddot{x} + b\dot{x} = u, \tag{2.1}$$

$$\ddot{x} + \frac{b}{m} = \frac{1}{m}u, \tag{2.2}$$

where x is position, m is vehicle mass, b resistance coefficient and u is system input representing a force taking effect on vehicle. For purposes of designing ACC functionality were the equations reworked to this form

$$\dot{v} + \frac{b}{m}v = \frac{1}{m}u \tag{2.3}$$

using $\dot{x} = v$, where v is a vehicle velocity. Equation 2.3 then fit the used vehicle model. Constant parameters used are:

- $m = 1000$ kg
- $b = 50$ Nsm^{-1}

These constants were also adopted from [FPEN10].
The second integrator was also used in the model to integrate velocity to get vehicle position. Ego vehicle velocity and position are used by Scenario Reader to identify Ego vehicle position and velocity, for proper simulation and visualization of the given scenario.
Vehicle model input had to be saturated to simulate real vehicle. Maximal motor force was choosed as $F_max = 15000$ N and maximal breaking force $F_min = -30000$ N. As can be seen in Fig. 2.3, there are used Zero-Order Hold and First-Order Hold blocks. Zero-Order Hold blocks are used to dicretize output signals and First-Order Hold blocks are used to convert discrete signal on the input port of continous model, as the environment includes discrete and continous parts.

2.2.2 Pack Ego Actor

BusActors	
'ActorID'	1
'Position'	[0 pos 0]
'Velocity'	[0 vel 0]
'Roll'	0
'Pitch'	0
'Yaw'	90
'AngularVelocity'	[0 0 0]

Table 2.1: BusActors structure definition

Pack Ego Actor block was adopted from Automated Driving Toolbox example [Inc20b]. Pack Ego Actor block gets information from vehicle model and "packs" them into a structure (bus), which is supported by Scenario Reader block as an input.
In the Fig. 2.1 can be seen function inside the Pack Ego Actor block. All parameters of packed structure used in function apply to Ego vehicle, because, as was told earlier, through the scenario reader the Ego vehicle could be directly controlled.
Explanation of used parameters:

- **ActorID** - ID of Ego vehicle in prepared scenario.
- **Position** - Position of Ego vehicle in Carthesian coordinates. In our case x-coordinate and z-coordinate are constant values, because all roads in our scenarios lead from origin of coordinate system straightly in direction of positive values of y-axis.
- **Velocity** - Velocity of Ego vehicle in direction of x- , y- and z- axis.
- **Yaw, Pitch, Roll** - Rotation of car in space given in Euler angles. In this case Yaw is constant value, becuase used roads and trajecteries of Ego vehicle are straight towards positive values of y-axis.
- **AngularVelocity** - Angular velocity of Ego vehicle. In this case it's zero due to shape of used roads and Ego vehicle trajectories.

Variable *pos* is y-part of position of Ego vehicle in Carthesian coordinates given by output of vehicle model and variable *vel* is y-part of Ego vehicle velocity in Carthesian coordinates.

2.2.3 Scenario Reader

Scenario reader block is reading prepared scenario files with trajectory and setup of Target vehicle. Trajectory, velocities and other setup of Ego vehicle are changed via block input from Pack Ego Actor block, which provides information from vehicle model.
Creation of scenarios and used scenarios for simulations of ACC features and utilities will be explained in Chapter 4.

2.2.4 Use and setup of vehicle radar

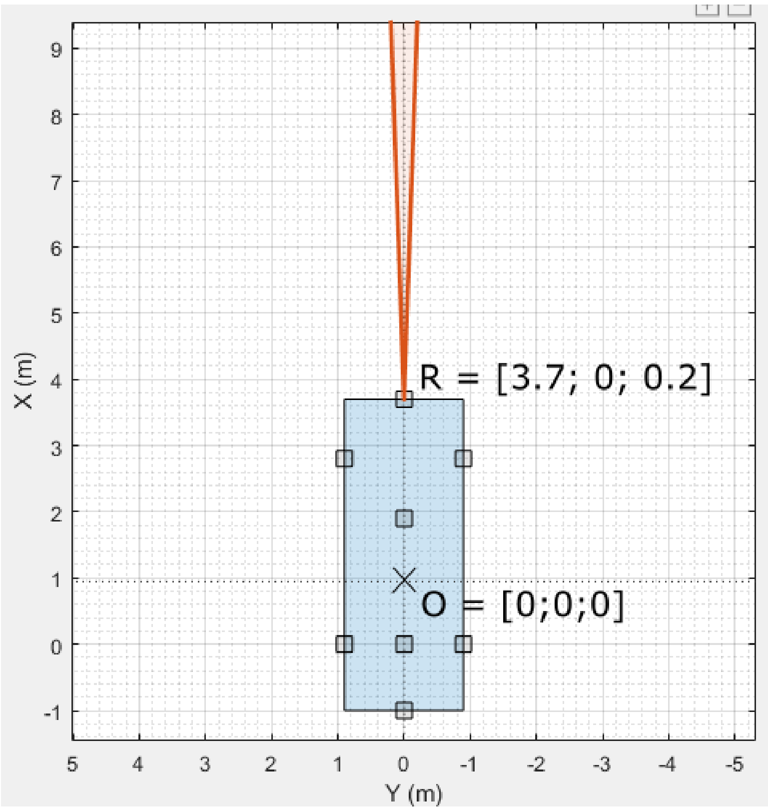

Figure 2.4: Ego vehicle with radar

Radar block is used to get information about simulated scenario in 'real time' and generates detection of vehicles on the road according to information from Scenario Reader. In fact Radar block simulates behaviour and detection capabilities of real radar attached to vehicle.
For its function, radar uses expression of measurement in vehicle coordinate system. Vehicle coordinate system and its origin is connected to Ego vehicle as can be seen in Fig. 2.4. That means that all measurements done by radar

are relative to position and velocity of Ego vehicle.

Radar placement and setup

Radar is attached to Ego vehicle at point $R = [R_x, R_y, R_z] = [3.7, 0, 0.2]$ in vehicle coordinates. Maximal range of radar is $d_{max} = 150$ m and field of view azimuth of radar is $\alpha = 0.9°$.
All measures done by radar are in Ego vehicle coordinate system. It means that position of Target vehicle is evaluated as a relative distance to Ego vehicle (same same for velocities measured by radar). Origin of vehicle coordinates is point $[0, 0, 0]$ as can be seen in Fig. 2.4. Radar has also capability to measure multiple points on surface of Target vehicle, but this function was simplified only to one detection, because it isn't necessery to measure multiple points for purpose of ACC testing.

Radar Measurment

Radar is capable of measuring relative position and relative velocity of Target vehicle in all three axes in Ego vehicle coordinate system.
Distance measured by radar is relative to point R, where the radar is attached to Ego vehicle. That means that radar output has offset from Ego vehicle coordinate system origin, where the offset is by position of radar attachment point. Radar measures distance in meters.
Relative velocity measured by radar is defined as

$$v_{mrel} = v_t - v_e, \tag{2.4}$$

where v_{mrel} is relative velocity of Target vehicle measured by radar, v_e is absolute Ego vehicle velocity and v_t is absolute Target vehicle velocity. Measured velocities are in meters per second.
All the values measured by radar are vectors (velocity in all three axes and same for distance), but due to ACC functionality we care only about x-parts of all measured values in vehicle coordinates.

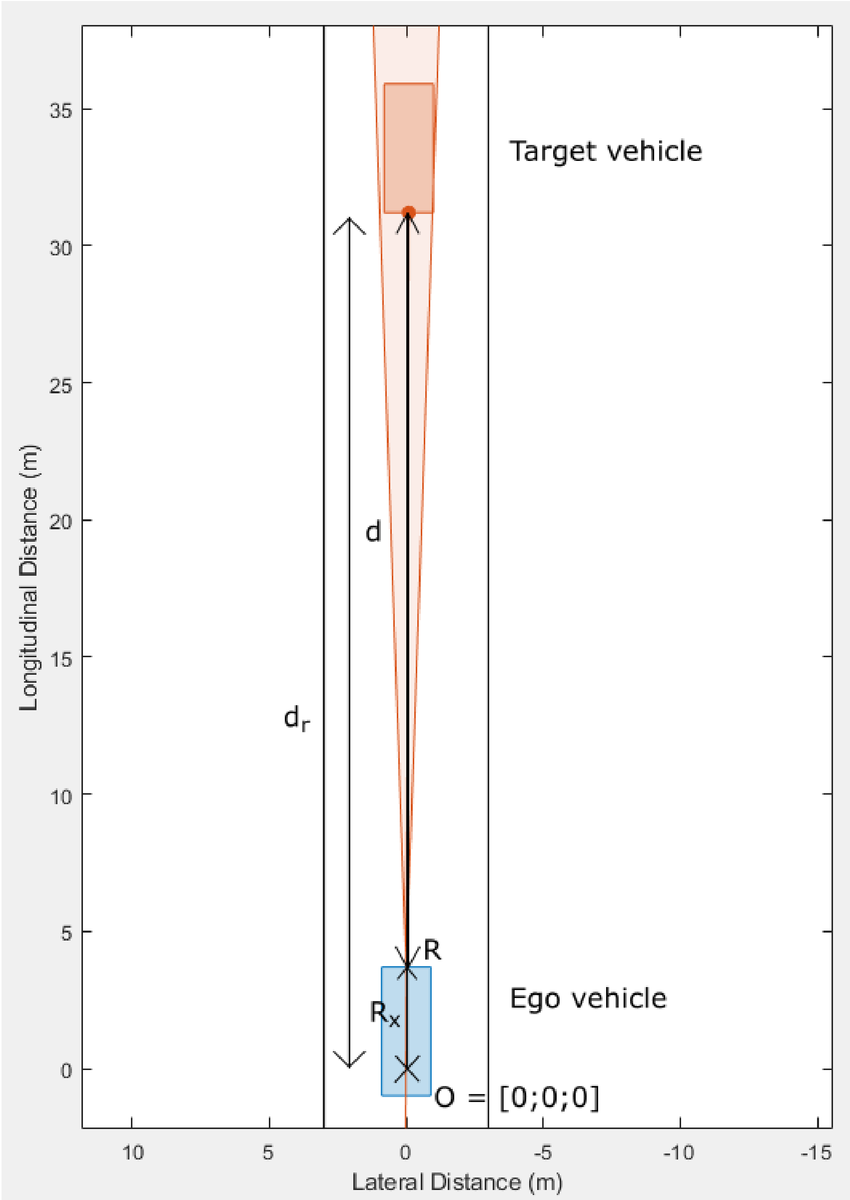

Figure 2.5: Radar measurment modifications

Radar Measurement Modification

Because position measured by radar is in vehicle coordinates, measured distance has to be modified to have the disatnce from front bumper of Ego vehicle to Target vehicle and not from Ego vehicle coordinate system origin to Target vehicle. The modified distance is

$$d = d_r - R_x, \tag{2.5}$$

where d is distance from Ego vehicles front bumper to detection point on Target vehicle in meters, d_r is distance measured by radar in vehicle coordinates and R_x is x-coordinate of radar attachment point R (see Fig. 2.5).
When radar doesn't get any detection, that means, there isn't any car in detection area, measured distance is $d_r = 0$ m. This has to be modified due to ACC algoritm, which started to break, when there wasn't any detection on radar. Radar has full detection range at $d_{max} = 150$ m in direction of x-axis from radar attachment point.
For switching between radar measurement, when Target vehicle is detected and application of maximal radar range when there isn't any detection. Signal *NumDetections*, which says how many detections was detected by radar (in this case where only one detection is enabled, state of *NumDetections* can only be 1 or 0), was used as condition for switching between two distance measurements. Algorithm is explained bellow (Alg. 1).

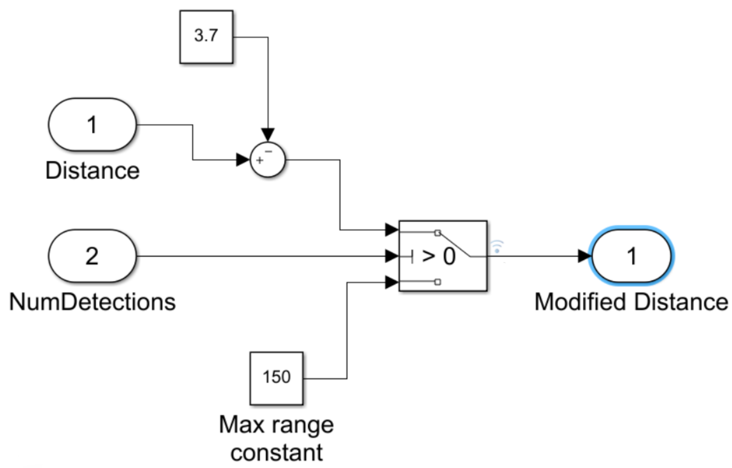

Figure 2.6: Radar detection switching

Algorithm 1 Switch detections

$d = RadarMeasurement$
$d_{max} = 150$
$c = NumDetections$
if $c == 1$ **then**
  return d
else
  return d_{max}
end if

2.3 ROS2 network

ROS (Robot Operating System) is DDS platform widely used in robotics for control, navigation, etc. of robotic systems. ROS2 is second iteration of ROS development.

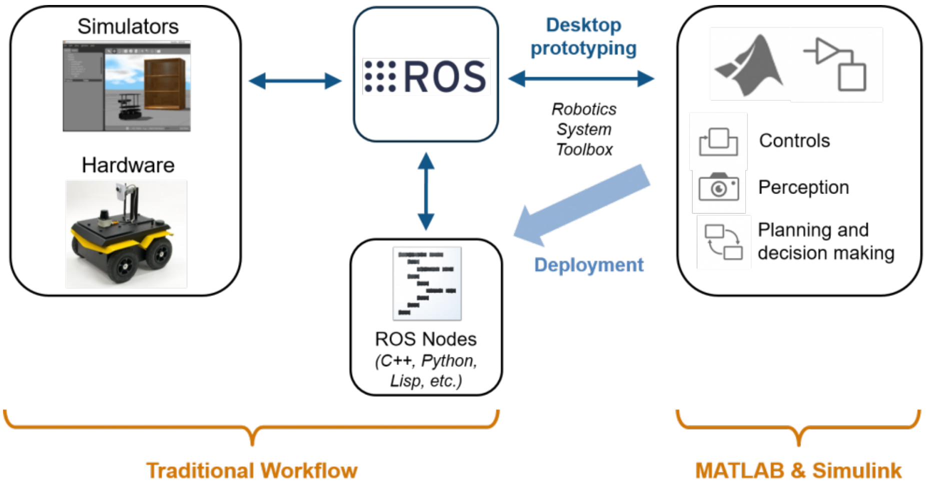

Figure 2.7: Diagram of ROS workflow

Fig. 2.7 was adopted from [Cas20].
As can be seen in Fig. 2.1 there are two ROS2 nodes in testing environment. ROS2 network in this case has the functionality of distributional bus. In future experiments ROS2 network could be used also to connect Testing framework to real system or vehicle dynamics simulators.
To build ROS2 network the Simulink toolboxes for ROS networks and relevant blocks for ROS2 Subscribers and Publishers was used.
Node named */distance* gets output from radar and Ego vehicle a sends them into ACC subsystem to further process. Data sent by */distance* node publisher are relative velocity in x-axis of vehicle coordinates, distance in x-axis in vehicle coordinates, absolute Ego vehicle velocity and number of detection done by radar.
Node named */action* gets value of action intervention calculated by ACC algorithm and sends it to */action* Subscriber, which output is used as input force of Ego vehicle model.

Chapter 3

Implementation of Adaptive Cruise Control Algorithms

3.1 Introduction

Adaptive Cruise Control algorithm is in principle simple algorithm, which lead vehicles to desired distance between them and keeps the given distance. However, with requirments on passengers safety and comfort it requires more complex algorithms and utilities to be part of the system. For example Cut-in system, which guarantees safety even in dangerous maneuvers on the road and still keeps certain level of comfort of passengers.

3.2 Overall Architecture

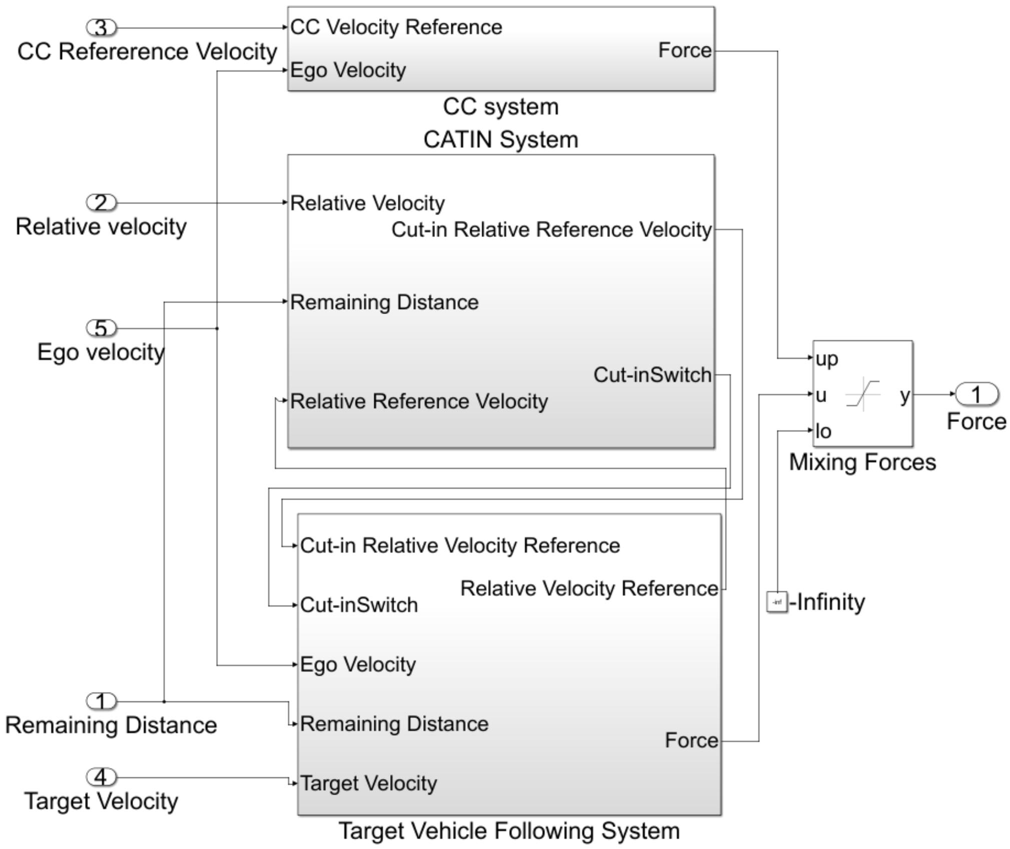

Figure 3.1: ACC system block diagram

Implementation of ACC algorithm consists of following parts.

- **Cruise Control system** - This system is used to control Ego velocity v_e, when radar has no detection or claculated relative velocity v_{rel} is lower than calculated relative reference velocity v_{rref}. Its active region is section marked by number 1 (see Fig.3.2). Its output creates upper limit for output from Target Vehicle Following system.
- **Target Vehicle Follow system** - It's used to lead Ego vehicle to desired distance between Ego and Target vehicle. It has two parts, approach and follow. Approach means that system decreasing relative reference velocity is based on distance between vehicles. Its active region is linear function numbered 2a (see Fig.3.2). Follow function keeps given distance between Ego and Target vehicle. Its active region is point marked 2b (see Fig.3.2).

- **Cut-in system** - Purpose of Cut-in system is to keep passengers safe with certain level of comfort even in potentionaly dangerous maneuvers on the road. Cut-in active regions are sections marked by numbers 3 and 4 (see Fig.3.2), whereas section 4 is for Emergency breaking Cut-in utility.

ACC algorithm takes several inputs, which are further processed. The inputs are absolute velocity of Ego vehicle v_e, relative velocity measured by radar v_{mrel}, distance between cars measured by radar d_r, reference velocity for Cruise Control regulator v_{eref} and *NumDetections* signal from radar. All velocities, ACC system works with, are in meters per second and distances are in meters.
ACC gets raw input from radar and, as was mentioned in section 2.2.4, distance measured by radar have to be modified. All other inputs are raw inputs from radar. Output of ACC system is force taking effect on vehicle in newtons.
In Fig. 3.2 there is velocity depending on distance between Ego and Target

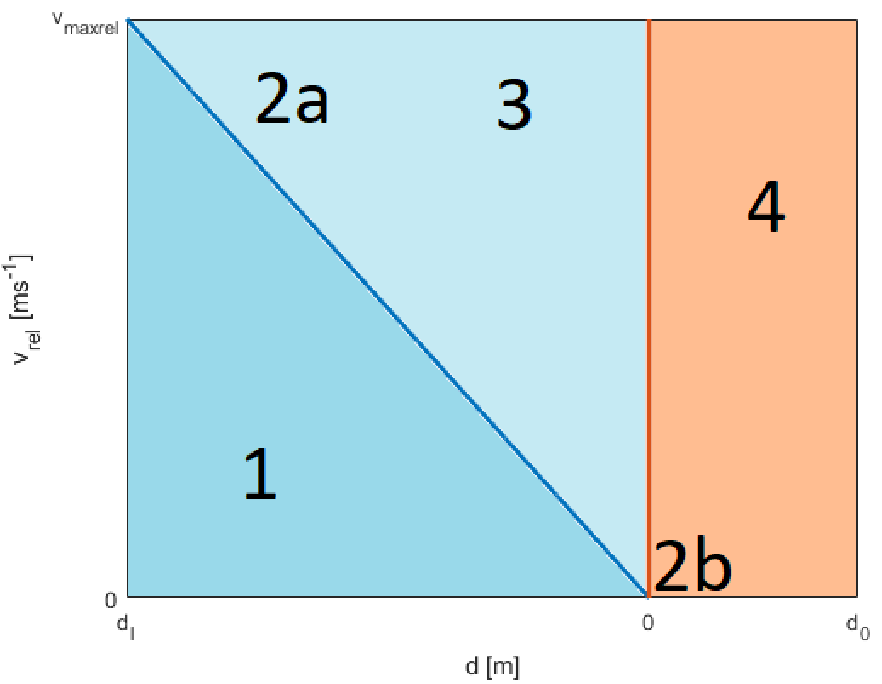

Figure 3.2: Graph of ACC functionality sections in distance domain

vehicle. Distance here is divided into two parts. Towards positive direction of x-axis is desired spacing between vehicles d_0 and negative part of axis is remaining distance d_l of Ego vehicle to reach desired spacing d_0.
The Constant Time Gap policy [Raj11a] (further CTG policy) was used for spacing calculation. Accordingly to CTG calculation of space between two

vehicles is based on Ego vehicle velocity v_e. It means that desired spacing d_0 and remaining distance to reach desired spacing d_l are changing with Ego vehicle velocity.
Relative velocity used in Fig. 3.2 is defined as

$$v_{rel} = v_e - v_t = -v_{mrel} \tag{3.1}$$

As was mentioned before, theACC system is divided into several parts which switching between each other according to behaviour of Target vehicle during maneuvers on the road. Graph above (Fig. 3.2) display sections where certain parts of ACC system are active. This will be further explained in next sections.

3.3 Target Vehicle Follow System

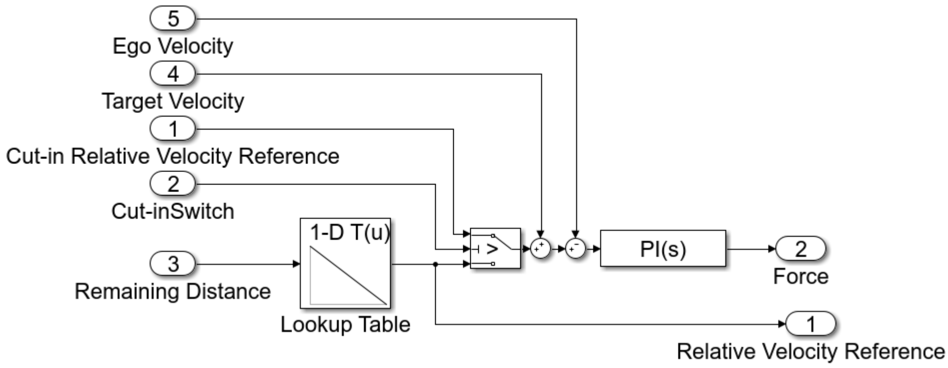

Figure 3.3: Target Vehicle Follow system block diagram

Regions, where Target Vehicle Follow system is active, are linear function marked as 2a and point $O = [0; 0]$ marked as 2b in Fig.3.2.

- Approach maneuver (section 2a) - System is used to lead vehicle to desired spacing of d_0 by decreasing relative reference velocity v_{rref} depending on remaining distance d_l. System, basicaly, brings v_{rel} and d_l to zero. Caluculation of v_{rref} is realized by Lookup Table (Fig.3.3).
- Follow feature (section 2b) - System should keep distance between two vehicles. Relative velocity v_e and remaining distance d_l should be 0.

In Fig. 3.3 could be seen also inputs from Cut-in system for switching between reference velocities. This will be explained in section 3.5.

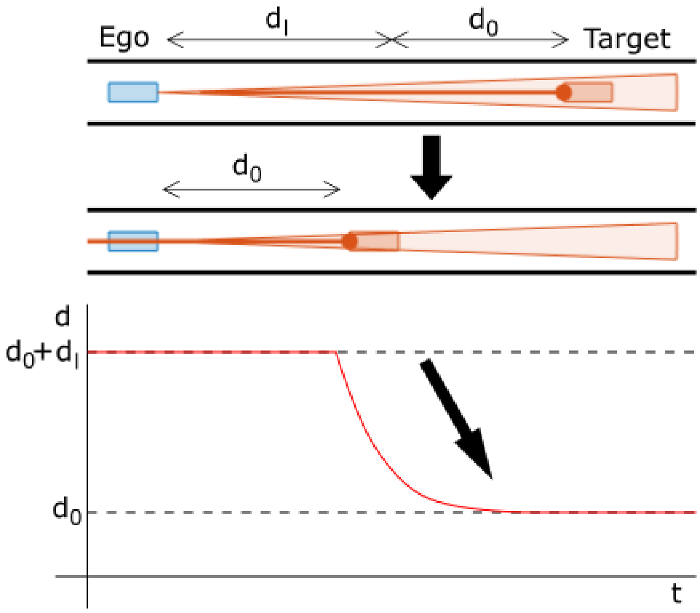

Figure 3.4: Diagram of following system functionality

3.3.1 Spacing Calculation

As was mentioned above in section 3.2 for spacing control was used CTG policy (Constant Time Spacing policy) adopted from Rajamani [Raj11a]. It means that space between two vehicles is changed with Ego vehicle velocity v_e given by equation

$$d_0 = l + v_e t, \tag{3.2}$$

where l is distance between vehicles, when aboslute velocity of Ego vehicle is zero, t is time gap, v_e is absolute velocity of Ego vehicle and d_0 is desired spacing.
The recomended time gap between two vheicles in Czech Republic is $t = 2$ s. This value was also used as constant time gap to calculate desired spacing d_0. Calm distance between vehicles l was specified as $l = 3$ m. Calculation of desired spacing d_0 is specified by equation

$$d_0 = 3 + 2v_e. \tag{3.3}$$

For example, if velocity of Ego vehicle is $v_e = 20\ \mathrm{ms}^{-1}$, desired spacing is $d_0 = 43$ m.
Remaining distance d_l is then given by

$$d_l = -(d - d_0) = -(d - (3 + 2v_e)) = -d + 3 + 2v_e, \tag{3.4}$$

where d is modified distance from radar.
Example of dividing distance can be seen in Fig. 3.5.

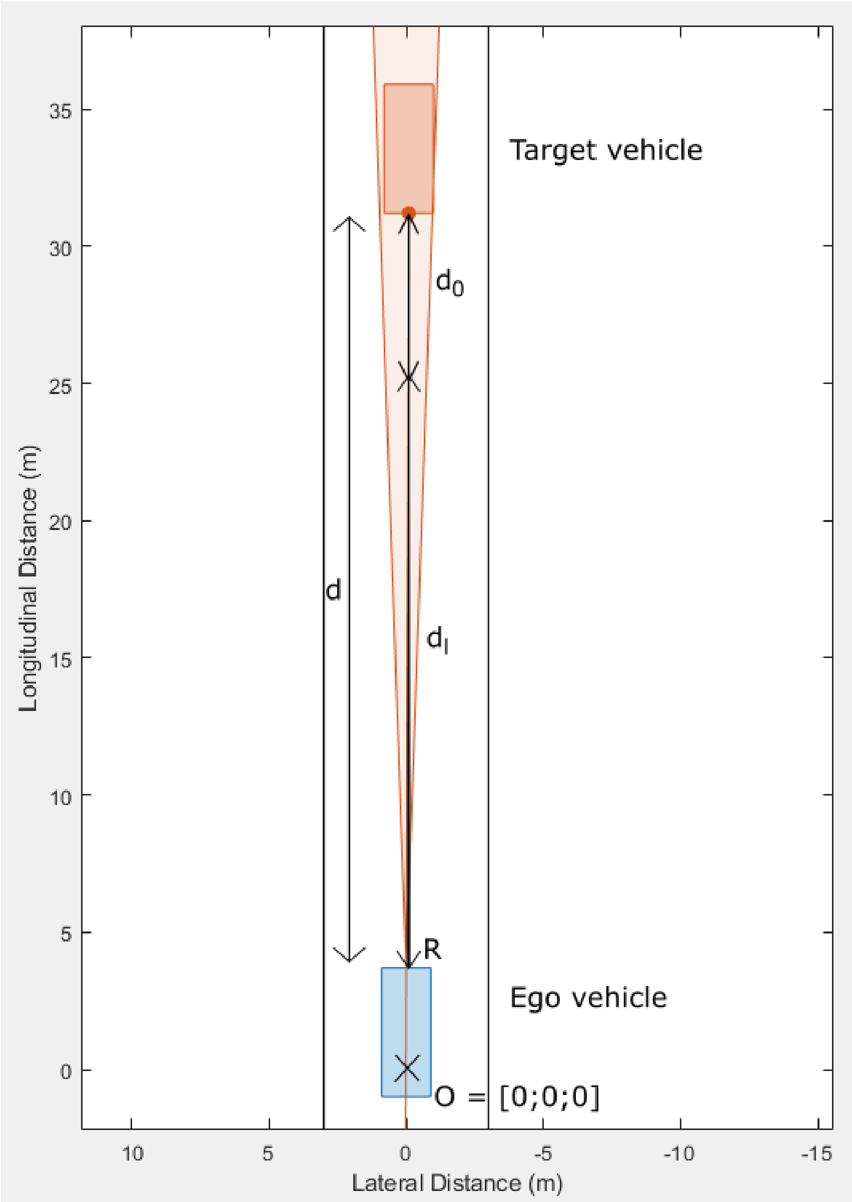

Figure 3.5: Distance dividing

3.3.2 Velocity Reference Calculation

The change of reference velocity v_{ref} based on remaining distance d_l (Rajamani [Raj11b]), was used to lead Ego vehicle desired spacing d_0. To calculate v_{ref} in certain remaining distance d_l was used descresing linear function with equation

$$v_{rref} = -kd_l, \tag{3.5}$$

where v_{rref} is relative reference velocity, d_l is distance remaining to desired spacing and k is constant. Value of k was specified as $k = \frac{10}{150} = \frac{1}{15}$. That means that on remaining distance of $d_l = -150$ m is $v_{rref} = 10\,\mathrm{ms}^{-1}$. It's very unlikely that two vehicles will meet with relative velocity v_{rel} higher than $10\,\mathrm{ms}^{-1}$.
In simulation this calculation is done by Look Up Table Simulink block, which gets remaining distance d_l as input and provide value of v_{rref} as output.
However, v_{rref} can't be driectly used as reference for linear regulator. It has to be recalculated to absolute reference velocity v_{ref} using Target vehicle velocity v_t. Target vehicle velocity is calculated using velocity of Ego vehicle v_e and relative velocity measured by radar v_{mrel} as can be seen in equation 3.6.

$$v_t = v_{mrel} + v_e \tag{3.6}$$

Then absolute Ego vehicle velocity reference v_{ref} can be calculated using

$$v_{ref} = v_{rref} + v_t. \tag{3.7}$$

3.3.3 Target Vehicle Follow System Regulator

Linear regulator used to lead Ego vehicle to desired spacing d_0 takes velocity reference v_{ref} and velocity of Ego vehicle v_e as an input and it's output is force taking effect on vehicle F. Type of designed regulator is PI, which can provide zero steady state error.
Regulator was designed for vehicle model mentioned in section 2.2.1, which is first order system.
Transfer function of vehicle system (transfer function is derived from equation 2.3 and used constants are listed in section 2.2.1)

$$P(s) = \frac{b(s)}{a(s)} = \frac{\frac{1}{m}}{s + \frac{b}{m}} = \frac{0.001}{s + 0.05} \tag{3.8}$$

and equation of PI regulator is

$$C(s) = \frac{q(s)}{p(s)} = \frac{K_p s + K_i}{s}. \tag{3.9}$$

Closed-loop characteristic polynom is

$$c_{CL}(s) = a(s)p(s) + b(s)q(s), \tag{3.10}$$

$$c_{CL}(s) = s^2 + \frac{K_p + b}{m}s + \frac{K_i}{m}. \tag{3.11}$$

Then desired second order polynom equation is

$$c_{CLd}(s) = s^2 + 2\zeta\omega_n + \omega_n^2 \tag{3.12}$$

where ζ is damping and ω_n is natural frequency. Then is valid that

$$\begin{aligned} \zeta &= \frac{-log(\frac{\%OS}{100})}{\sqrt{\pi^2 + (log(\frac{\%OS}{100}))^2}}, && (3.13) \\ \omega_n &= \frac{4}{\zeta T_s}, && (3.14) \end{aligned}$$

where T_s is settling time and $\%OS$ is percentual overshoot.
Regulator was designed with condition to have $\%OS = 0\%$ and $T_s = 5$ s. Values of damping and natural frequency are then $\zeta = 1$ and $\omega_n = 0.8$. Desired closed-loop polynom is

$$c_{CLd}(s) = s^2 + 1.6s + 0.64. \quad (3.15)$$

By comparing polynoms (3.16) two equation system (3.17, 3.18) is achieved.

$$s^2 + \frac{K_p + b}{m}s + \frac{K_i}{m} = s^2 + 1.6s + 0.64 \quad (3.16)$$

$$\begin{aligned} K_p &= 1.6m - b = 1550 && (3.17) \\ K_i &= 0.64m = 640 && (3.18) \end{aligned}$$

Regulator transfer function is

$$C(s) = \frac{1550s + 640}{s}. \quad (3.19)$$

Performance of regulator was tested on step signal response with these results. As can be seen in Fig. 3.6 even if regulator was designed without overshoot, there is overshoot aproximately $\%OS = 12\%$ and settling time $T_s = 6.7$ s. Control is complicated by system zeros.
Closed-loop poles were placed in positions on real axis $s_{1,2} = -\frac{4}{5}$. Both poles was placed to same location by feedback control. Pole placement was verified in Root Locus (Fig. 3.7).
Because of integration part in regulator and saturation in vehicle model it was neccesary to apply anti-windup. Type of applied antiwindup is clamping. As can be seen in Fig. 3.3 regulator is used also for Cut-in system control.

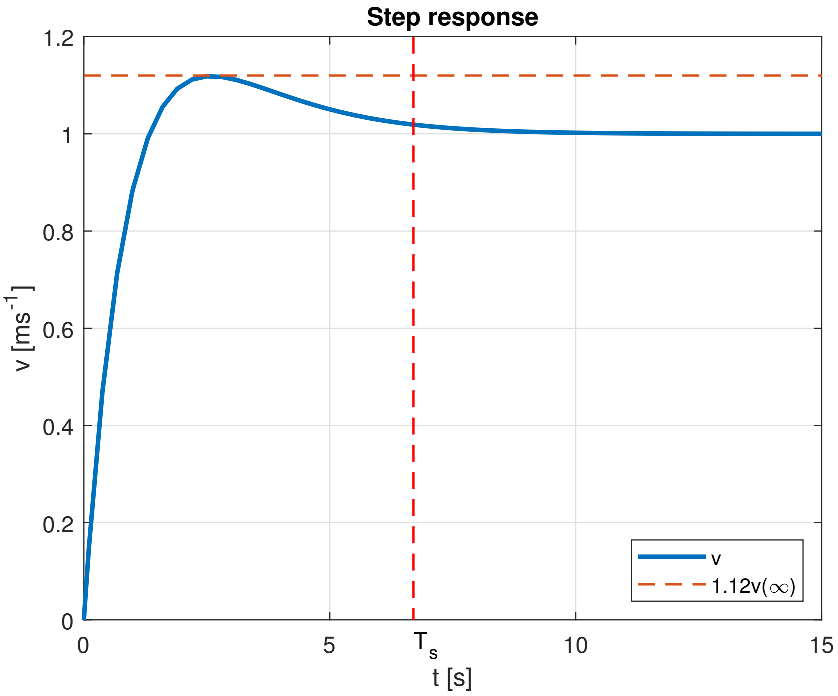

Figure 3.6: Step response of closed-loop from v_{ref} to v

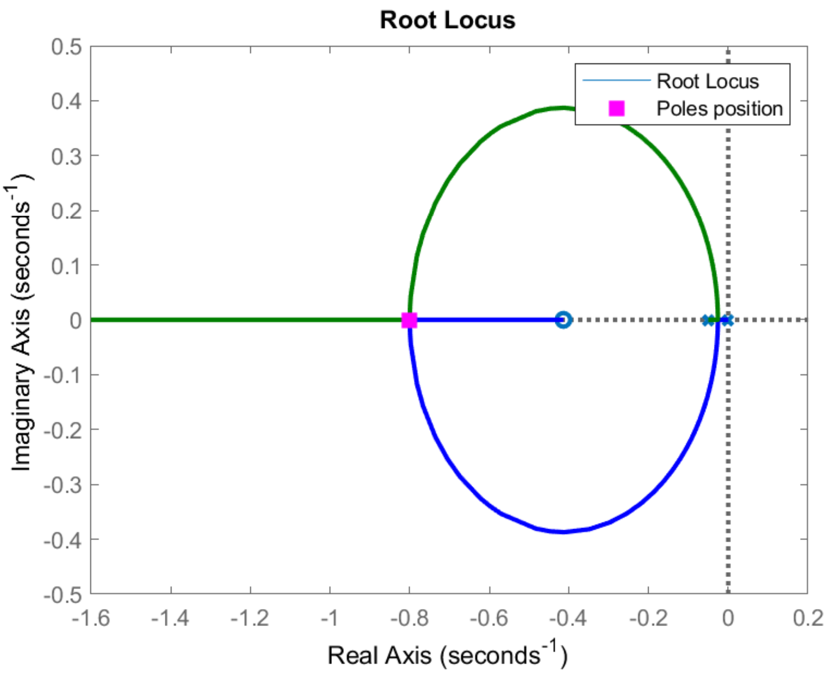

Figure 3.7: Root locus closed-loop vehicle system

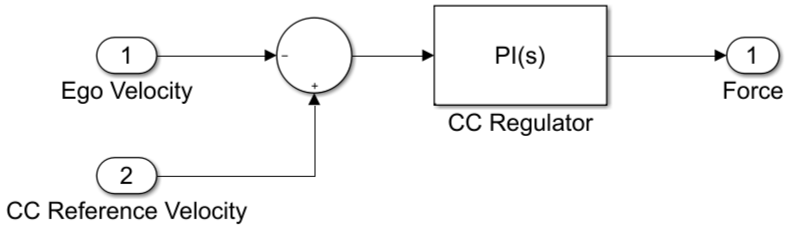

Figure 3.8: CC system block diagram

3.4 Cruise Control system

After creating Target Vehicle Following system appeared problem with crossing maximal velocity given by law or velocity set by user or external system. Because of agresivity of Target Vehicle Follow system regulator, the response of system could be too fast for comfort drive. The second linear regulator with slower dynamics with purpose of Cruise Control system was added. This regulator limits the force output of Target Vehicle Follow system regulator. Active section of Cruise Control system is section in Fig. 3.2 marked by number one. Cruise Control system needs two condition to be satisfied to be active (3.20, 3.21).

$$d_l < 0 \tag{3.20}$$

$$v_{rel} < -\frac{1}{15}d_l \tag{3.21}$$

As can be seen in Fig. 3.9, with no radar detection or detection in distance

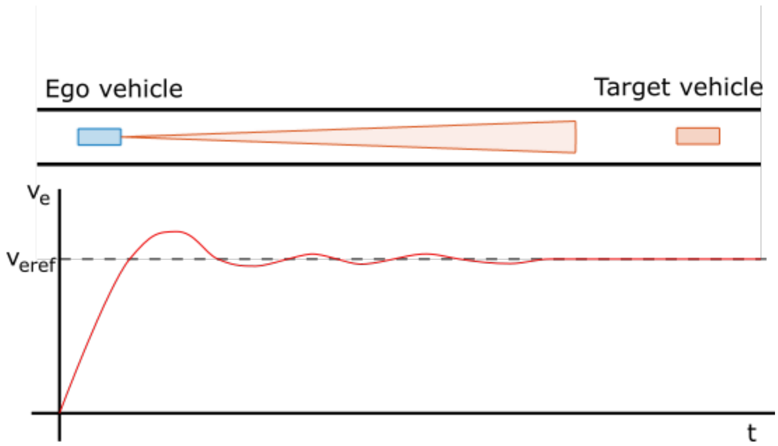

Figure 3.9: CC system functionality diagram

and relative velocity fulfiling inequations 3.20 and 3.21 CC system keeps Ego velocity v_e given by external reference velocity v_{eref}.

3.4.1 Cruise Control Regulator

Design of this linear regulator is very simillar to design of linear regulator used in Target Vehicle Folowing system. Regulator was designed for vehicle model explained in section 2.2.1 therefore design process is the same as in section 3.3.3 only with different demands on performance. Type of regulator used for Cruise Control is PI regulator.
Closed-loop polynom is

$$c_{CL} = s^2 + \frac{K_p + b}{m}s + \frac{K_i}{m}. \tag{3.22}$$

Regulator was designed with demands on overshoot $\%OS = 0$ and settling time $T_s = 30$ s. Demands on performance of this regulator are mainly to achieve comfort of passengers.
Calculation of constants for second order characteristic polynom are explained in section 3.3.3.
Desired characteristic closed-loop polynom is

$$c_{CLd}(s) = s^2 + 0.2667s + 0.0178 \tag{3.23}$$

Comparison of $c_{CLd}(s)$ and c_{CL} creates two equations 3.24 and 3.25.

$$K_p = 0.2667m - b = 216.6667 \tag{3.24}$$

$$K_i = 0.0178m = 17.7778 \tag{3.25}$$

Transfer function of Cruise Control linear regulator is

$$D(s) = \frac{216.6667s + 17.7778}{s}. \tag{3.26}$$

Because of integration part in regulator and saturation in vehicle model it was neccesary to apply anti-windup. Type of applied antiwindup is clamping. As can be seen in Fig. 3.10 closed-loop system with regulator $D(s)$ has overshoot aproximately $\%OS = 5\%$ and settling time $T_s = 35.5$ s. For purposes of Cruise Control system it's acceptable.
For example of usual function of Cruise Control regulator $D(s)$ in Fig. 3.11 is reaction of regulator on step of reference from 0ms^{-1} to 20ms^{-1}.

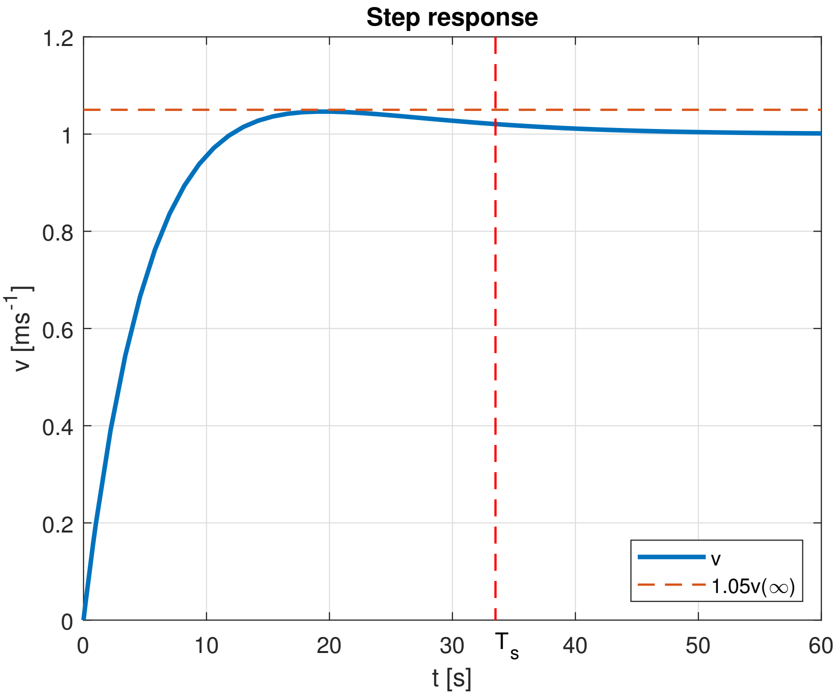

Figure 3.10: Step response of Cruise Control form v_{eref} to v

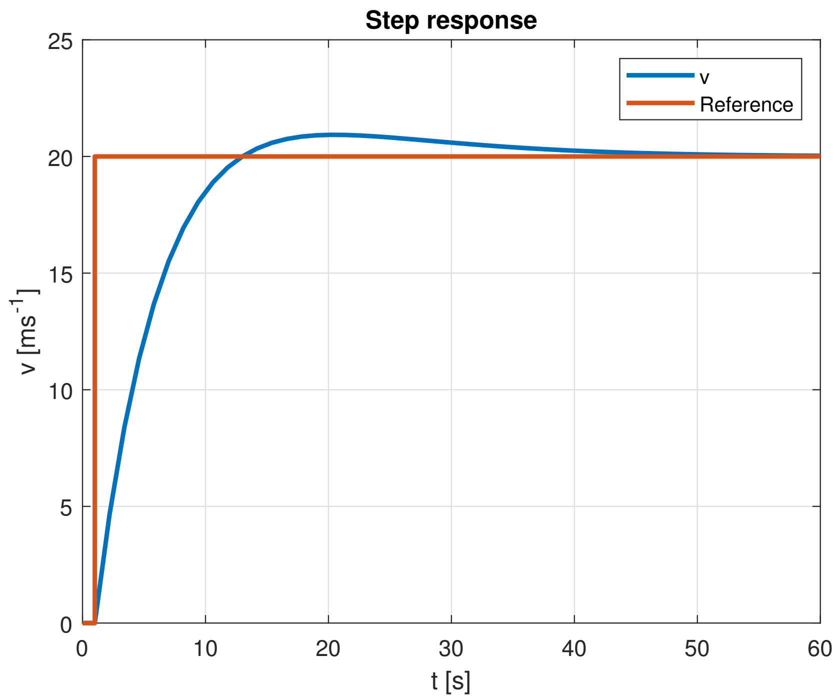

Figure 3.11: Response of regulator $D(s)$ on step of reference

3.5 Cut-in System

Purpose of Cut-in system is to maintain passengers safety at first place and then comfort even in dangerous maneuvers performed by other participants of traffic.

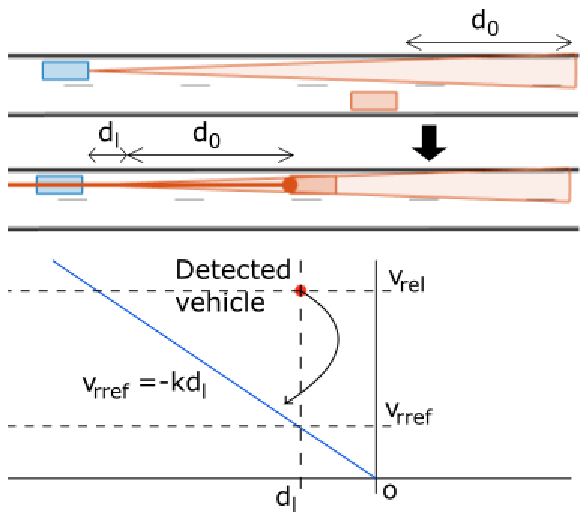

Figure 3.12: Cut-in system functionality diagram

3.5.1 Architecture

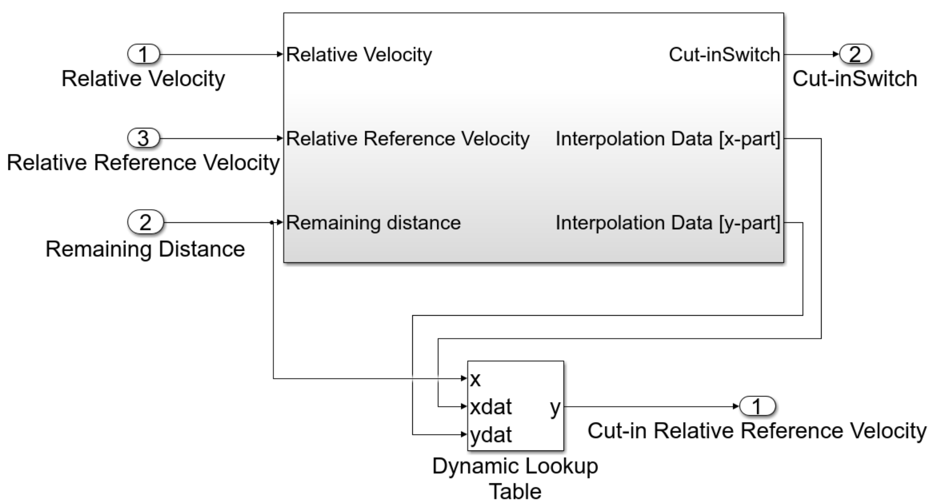

Figure 3.13: Block diagram of Cut-in system

Cut-in is used to take control in dangerous maeuvers. For example in Fig. 3.12 is situation, where Target vehicle with lower velocity switch line in short distance in front of Ego vehicle. In case ther is no Cut-in system, Traget Vehicle Follow system would try to quickly lower difference between v_{rref} and v_{rel} and this will result in aggressive and most likely uncomfortable maneuver of Ego vehicle. Cut-in system should minimize this behaviour and still keep comfort of passengers.
Cut-in system is active in both regions numbered 3 and 4 in Fig. 3.2. Mainly it's consisting of two parts divided by way of interpolation and position, in which is Target vehicle detected. When Target vehicle is detected in distance $d \geq d_0$ with relative velocity $v_{rel} >> v_{rref} = -kd_l$ (section 3 in Fig. 3.2). The second part called Emergency Breaking system is active when Target vehicle is detecteSd in distance $d < d_0$ (section 4 in Fig. 3.2).
Detail functionality will be described in further sections (3.5.2).

3.5.2 Cut-in System Design

Cut-in system was designed to minimize peaks of Ego vehecle decceleration. Ego vehicle control is switched to Cut-in when condition 3.27 is fulfilled.

$$v_{rel} - \epsilon > v_{rref}, \tag{3.27}$$

ϵ is threshold relative to value of v_{rref}. In this case threshold is $\epsilon = 3\text{ms}^{-1}$
Cut-in interpolate points $D = [d_l; v_{rel}]$ and $O = [0; 0]$ with linear function. Signal *Cut-inSwitch* is positive and switch relative velocity reference v_{rref} from Target Vehicle Follow system to its Cut-in velocity reference v_{Cut-in} as can be seen in Fig.3.3. Cut-in hand ver control back to Target Vehicle Follow system, when condition 3.28 is valid for certaian time t_{STOP}.

$$v_{rel} < 0.01 \tag{3.28}$$

Functionality of Cut-in system is described in following pseudocode (Alg.2). However, this method alone can't be used to interpolate points in section 4 (Fig.3.1). This is why there was added Emergency breaking utility.

Algorithm 2 Cut-in switch and interpolation

Input: d_l, v_{rel}, v_{rref}
$\Delta v = v_{rel} - v_{rref}$
$\epsilon = Constant$
if *Cut-inSwitch* $!= 1$ **then**
  if $\Delta v > \epsilon$ **then**
    Cut-inSwitch $= 1$
    $x_{interpolation} = [d_l; 0]$
    $y_{interpolation} = [v_{rel}; 0]$
    return *Cut-inSwitch*
    return $x_{interpolation}$
    return $y_{interpolation}$
  end if
else
  if $\mid v_{rel} \mid < 0.01$ **then**
    $t_{STOP} += 1$
    if $t_{STOP} > Constant$ **then**
      $t_{STOP} = 0$
      Cut-inSwitch $= 0$
      return *Cut-inSwitch*
    end if
  else
    $t_{STOP} = 0$
  end if
end if

3.5.3 Emergency Breaking Utility

Emergenency breaking utility was designed to interpolate trajectory of Cut-in reference velocity v_{Cut-in} in case that

$$d < d_0. \tag{3.29}$$

This system was added to ensure that interpolation linear function will be decreasing function, which Cut-in alone can't guarantee, in way it's implemented.
When Target vehicle is detected in $d < d_0$ or $d_l > 0$ (see section 3.3.1), the Emergency breaking functionality saves interpolation point $D_e = [d_l; -v_{rel}]$. It creates new Cut-in reference velocity by interpolation of points D_e and $O = [0; 0]$ with linear function. This way the decreasing function and thus imediate breaking of Ego vehicle will be assured.
Bellow is modified Cut-in algorithm with Emergency breaking (Alg.3).

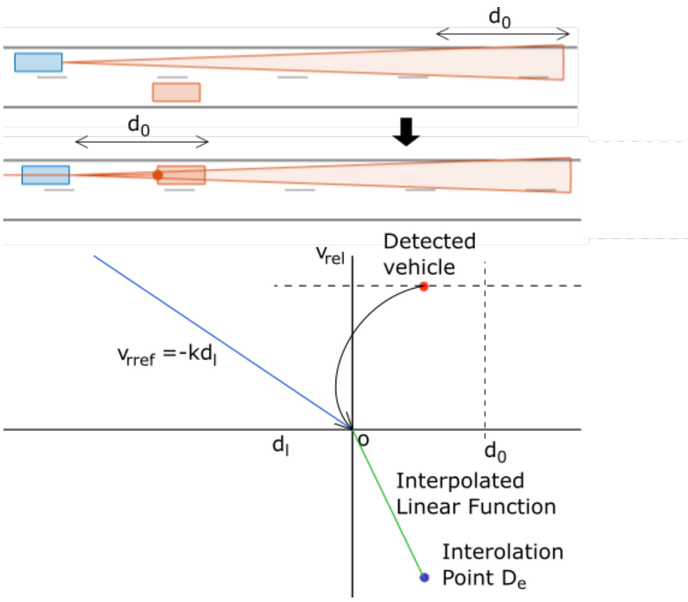

Figure 3.14: Functionality diagram of Emeregency breaking

Algorithm 3 Cut-in switch and interpolation with emergency breaking

```
Input: d_l, v_rel, v_rref
Δv = v_rel − v_rref
ϵ = Constant
if Cut-inSwitch != 1 then
  if Δv > ϵ then
    if d_l < 0 then
      Cut-inSwitch = 1
      x_interpolation = [d_l; 0]
      y_interpolation = [v_rel; 0]
      return Cut-inSwitch
      return x_interpolation
      return y_interpolation
    else
      Cut-inSwitch = 1
      x_interpolation = [0; d_l]
      y_interpolation = [0; −v_rel]
      return Cut-inSwitch
      return x_interpolation
      return y_interpolation
    end if
  end if
else
  if | v_rel |< 0.01 then
    t_STOP+ = 1
    if t_STOP > Constant then
      t_STOP = 0
      Cut-inSwitch = 0
      return Cut-inSwitch
    end if
  else
    t_STOP = 0
  end if
end if
```

Chapter 4

Testing scenarios

4.1 Introduction

For purpose of ACC system testing in virtual environment (Chap.2) several scenarios of different maneuvers to test system functionality were created. Driving Scenario Designer from Automated Driving Toolbox [Inc20a] for Matlab and Simulink was used to create scenarios. This app allows us to define trajectory, velocities and postion of Target vehicle to create predefined scenarios of traffic situations. Also it allows to configure sensors or vehicles itself (dimensions etc.)
Created scenarios are

- **Capture nad Follow scenario** - This scenario tests Target Vehicle Follow system and Cruise Control system.
- **Stop and Go** - This scenario tests ability of ACC system to keep track of an Target vehicle with changing Target velocity.
- **Cut-in scenario** - Scenario, where Target vehicle perform Cut-in maneuver with $d > d_0$.
- **Emergency scenario** - Target vehicle performs Cut-in maneuver with $d < d_0$.

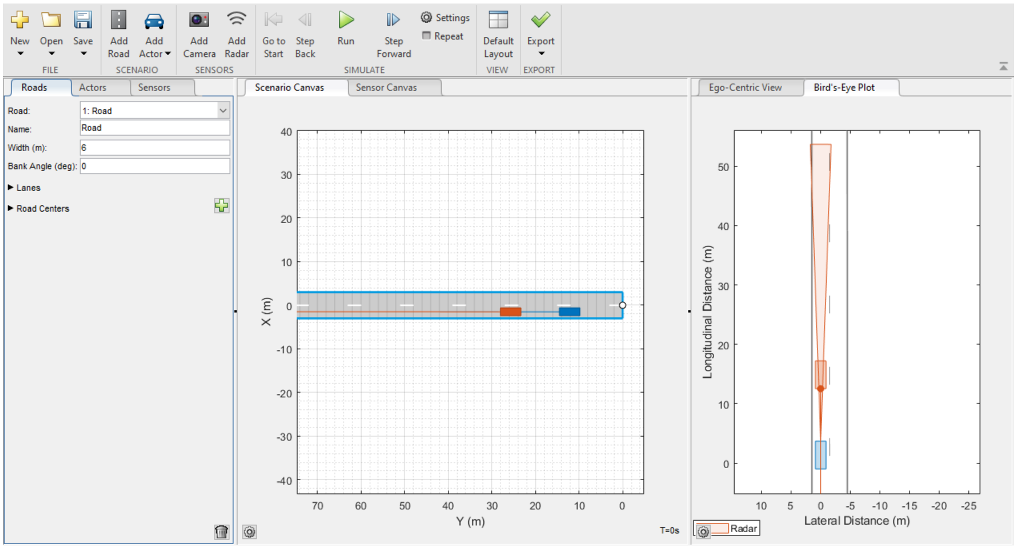

Figure 4.1: Scenario Designer App Example

4.2 Capture and Follow

Capture and Follow scenario is used to test CC system and Target Vehicle Folowing system functionality. Ego vehicle keeps Ego velocity $v_e = 23\,\mathrm{ms}^{-1}$ and in certain moment it detects Target vehicle with Target velocity $v_t = 20\,\mathrm{ms}^{-1}$. In time of detection relative reference velocity $v_{rref} = 6.73\,\mathrm{ms}^{-1}$ is higher then $v_{rel} = 3\,\mathrm{ms}^{-1}$ and CC system is still active. In remaining distance $d_l = -45$ m and distance between vehicles $d = 94$ m Target Vehicle Following system starts decreasing Ego velocity to lead and keep Ego vehicle in desired distance d_0 (see Fig.3.4).

4.3 Stop and Go

Stop and Go scenario was designed to test behaviour of ACC system in situation, where Ego vehicle follow Target vehicle and Target velocity is changing linearly. Target velocity decreasing on length $\Delta x_1 = 500$ m from $v_t = 20\mathrm{ms}^{-1}$ to $v_t = 0\mathrm{ms}^{-1}$ and then on length $\Delta x_2 = 500$ m from $v_t = 0\mathrm{ms}^{-1}$ to $v_t = 20\,\mathrm{ms}^{-1}$. Then Target velocity remains steday on value $v_t = 20\,\mathrm{ms}^{-1}$.

4.4 Cut-in maneuvers

In this maneuver Target vehicle switch lanes in front of Ego vehicle in remaining distance $d_l = -46$ m with Target velocity $v_t = 10\,\mathrm{ms}^{-1}$. Ego velocity is $29\,\mathrm{ms}^{-1}$ in time of maneuver. Relative reference velocity is $v_{rref} = 3.07\mathrm{ms}^{-1}$ and relative velocity is $v_{rel} = 19\,\mathrm{ms}^{-1}$. Interpolation point is then $D = [d_l; v_{rel}]$. In this scenario is sure that Cut-in take control of Ego vehicle due to crossing threshold ϵ.

4.5 Emergency breaking maneuver

In Emergency breaking maneuver Target vehicle switch lanes in front of Ego vehicle in distance $d_0 > d = 47$ m with Target velocity $v_t = 10\,\mathrm{ms}^{-1}$. Ego velocity is in time of maneuver $v_e = 30.3\,\mathrm{ms}^{-1}$ and then desired spacing is $d_0 = 63.6$ m and remaining distance is $d_l = 16.6$. Relative velocity is here $v_{rel} = 20.3\,\mathrm{ms}^{-1}$. Interpolation poitn is then $D_e = [d_l; -v_{rel}]$.

Chapter 5

Testing Results

5.1 Capture and Follow Result

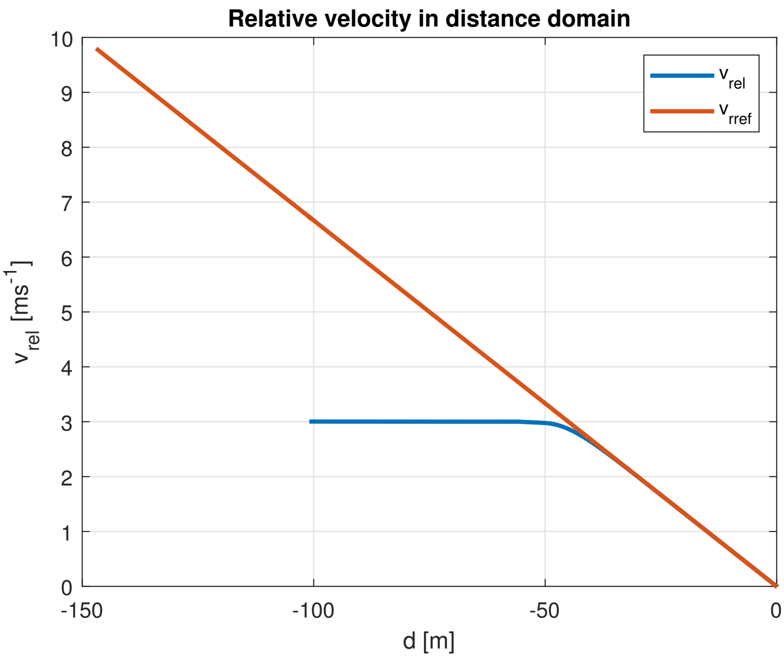

Figure 5.1: Relative Ego velocity in distance domain in Capture and Follow manuever

As can be seen in Fig.5.1 in remaining distance $d_l = < -100; -50)$ m Ego vehicle is controled by Cruise Control system and from $d_l = -50$ m starts to

decreasing Ego velocity v_e to approach to remaining distance $d_l = 0$ m.
In Figs. 5.2 and 5.3 can be seen that radar has no detection approximately to $t = 65$ s.
Fig.5.2 shows continual decreasing of Ego velocity to Target velocity.

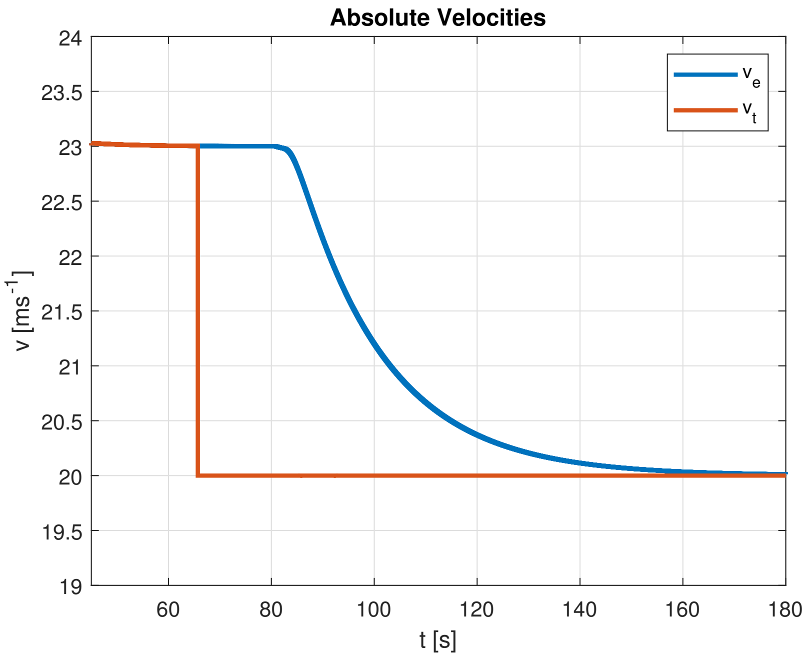

Figure 5.2: Absolute velocities in Capture and Follow

Fig.5.3 shows decreasing distance between vehicle to desired spacing $d_0 = 43$ m.

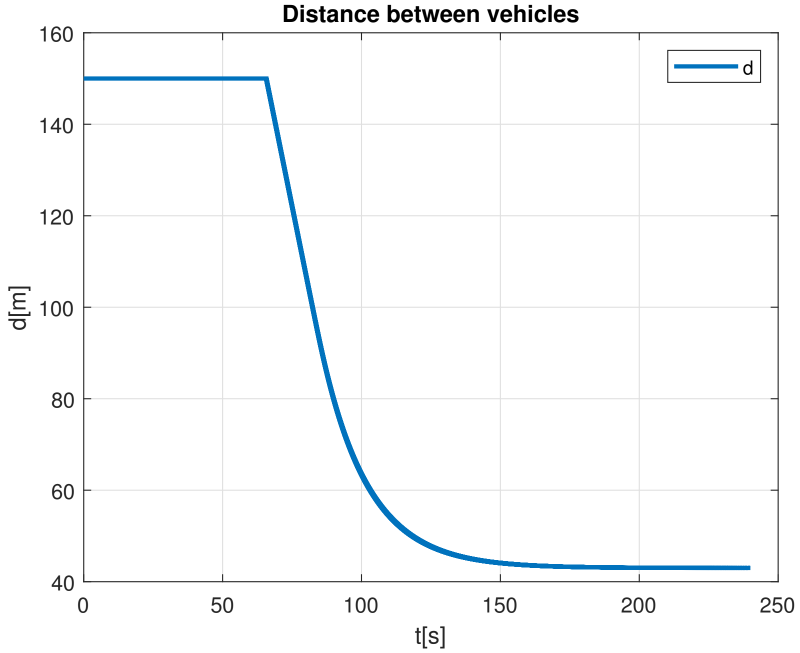

Figure 5.3: Distance between vehicles Capture and Follow

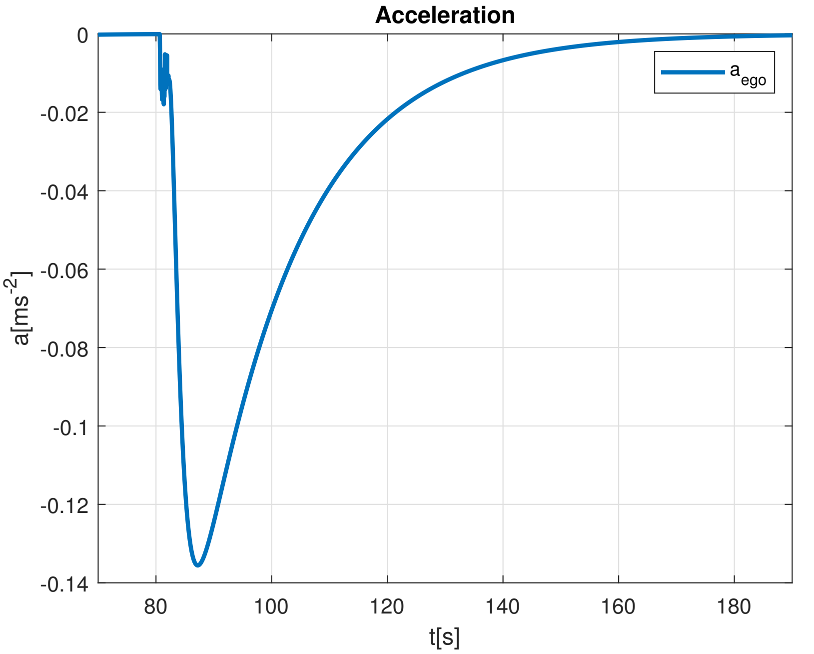

Figure 5.4: Ego absolute Ego acceleration in Capture and Follow

Fig. 5.4 and 5.5 are graphs of Ego acceleration a_{ego} and Ego jerk j_{ego}. From this graph ca be seen level of comfort of ACC developed ACC system. Values of jerk in this driving simulation are very low. It keeps comfort of passengers.

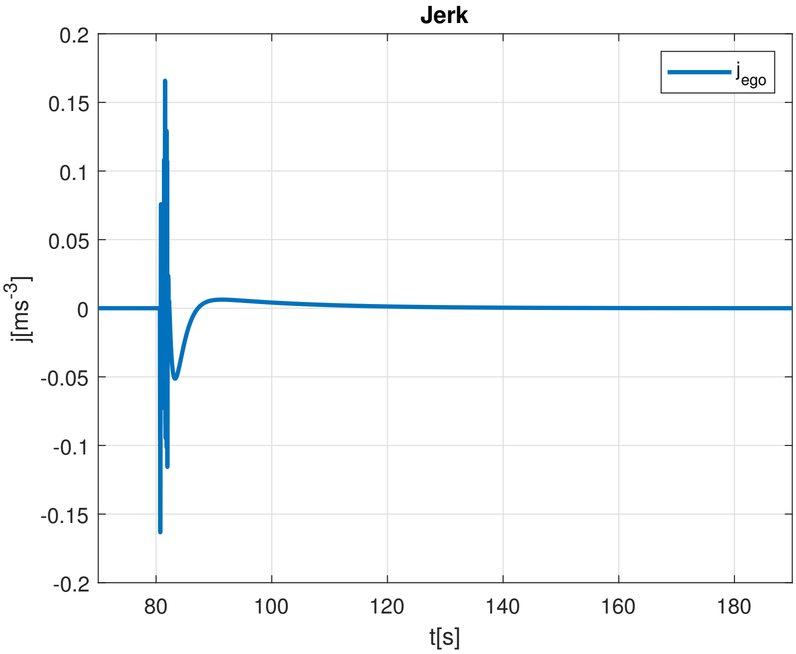

Figure 5.5: Ego jerk in Capture and Follow

On these figures (5.4 and 5.5) can be also seen oscilation, which probably arise from discretization of signals or maybe some false detection of radar.

5.2 Stop and Go Result

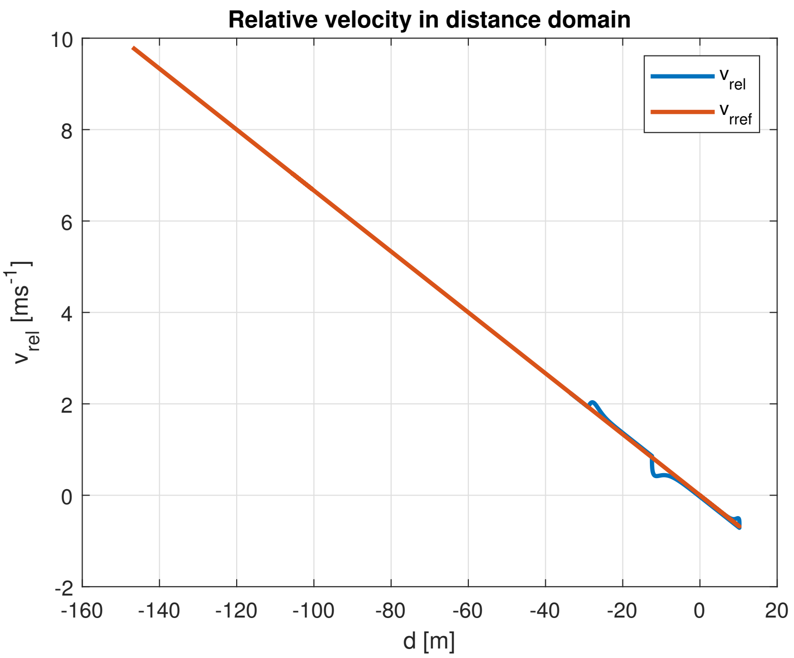

Figure 5.6: Relative Ego velocity in distance domain

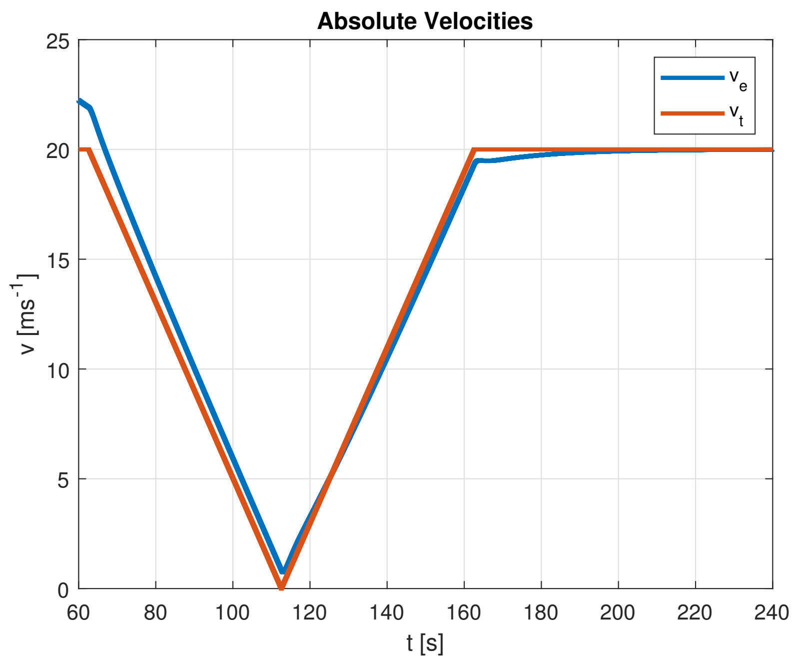

Figure 5.7: Absolute velocities in Stop and Go

In Fig. 5.6 can be seen some overshoots. First overshoot is caused by higher Ego velocity v_e than Target velocity v_t in moment, where Target vehicle starts to decrease its velocity (Fig. 5.7). Second overshoot is caused by change of monotony of Target velocity form decreasing to increasing (Fig. 5.7). The last overshoot is cuased by change of monotony of Target velocity from increasing to steady.

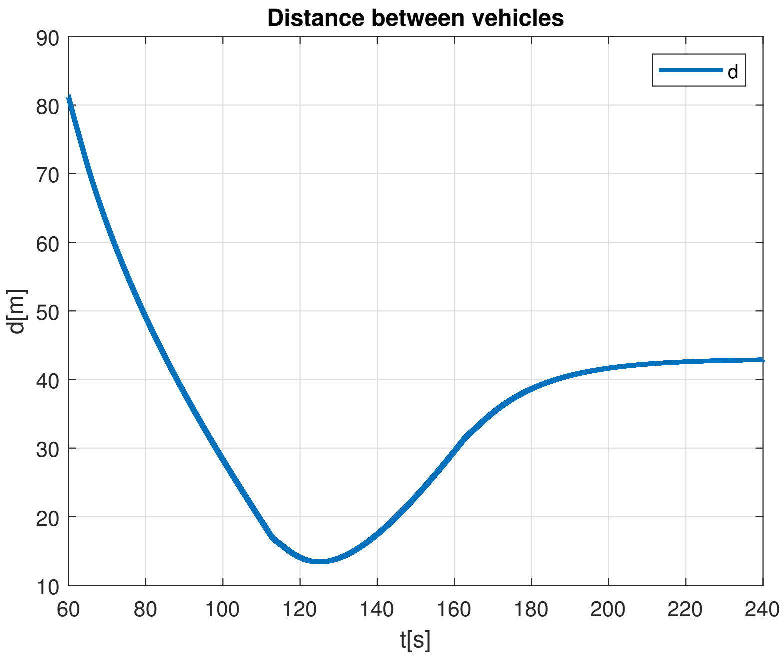

Figure 5.8: Distance between vehicles Stop and Go

ACC system in Ego vehicle tries to approach desired spacing d_0, which is changing with changing Ego velocity (Fig. 5.8).

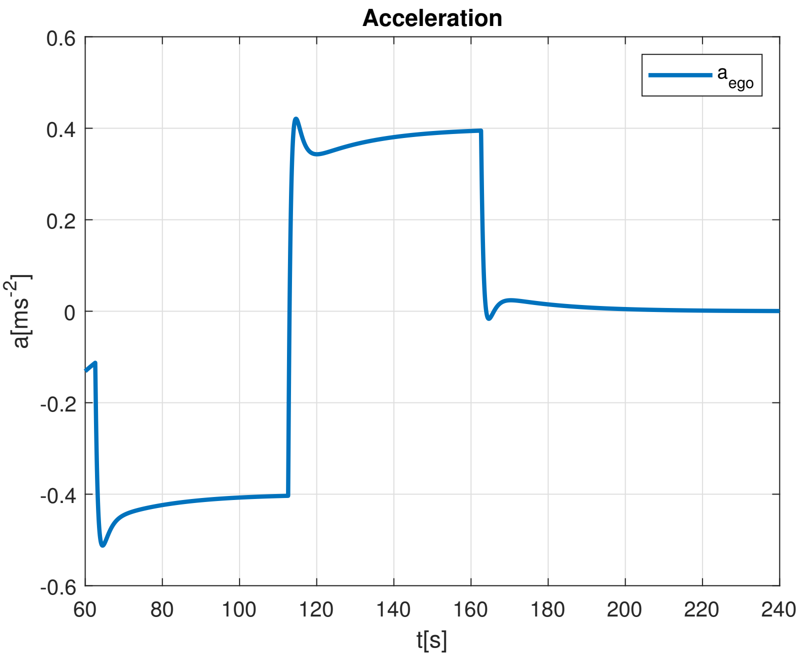

Figure 5.9: Ego absolute acceleration in Stop and Go

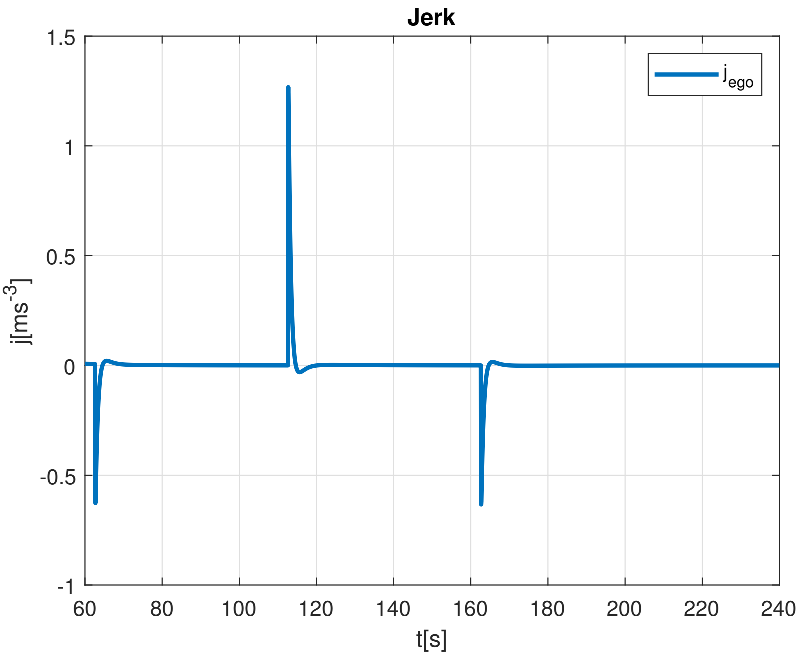

Figure 5.10: Ego jerk in Stop and Go

In Figs. 5.9 and 5.10 can be seen peaks in time, when Target vehicle changing monotony of Target velocity. However, peaks in Ego jerk j_{ego} are small and then subside to zero.

5.3 Cut-in Maneuver Result

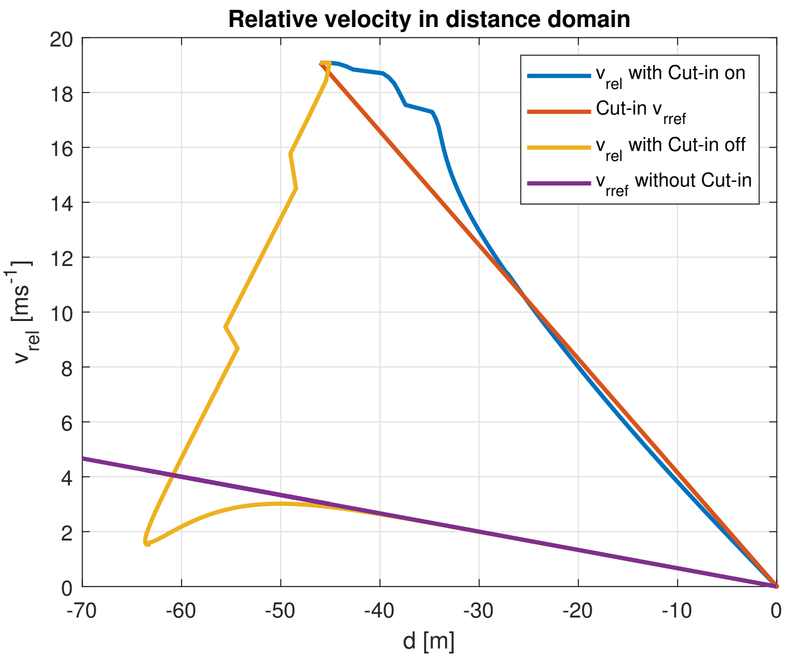

Figure 5.11: Relative Ego velocity in distance domain in Cut-in maneuver

Without Cut-in functionality ACC system has to regulate large velocity error and it will result in large overshoots (Fig. 5.11). ACC system with Cut-in create new reference by interpolation detection point with linear function (Fig. 5.11). This behaviour will result in much smaller overshoots, possibly no overshoots (Fig. 5.12).

As can be seen in Figs. 5.12 and 5.13 system with Cut-in has also faster approach to $v_{rel} = 0\text{ms}^{-1}$ and desired spacing $d_0 = 23$ m.

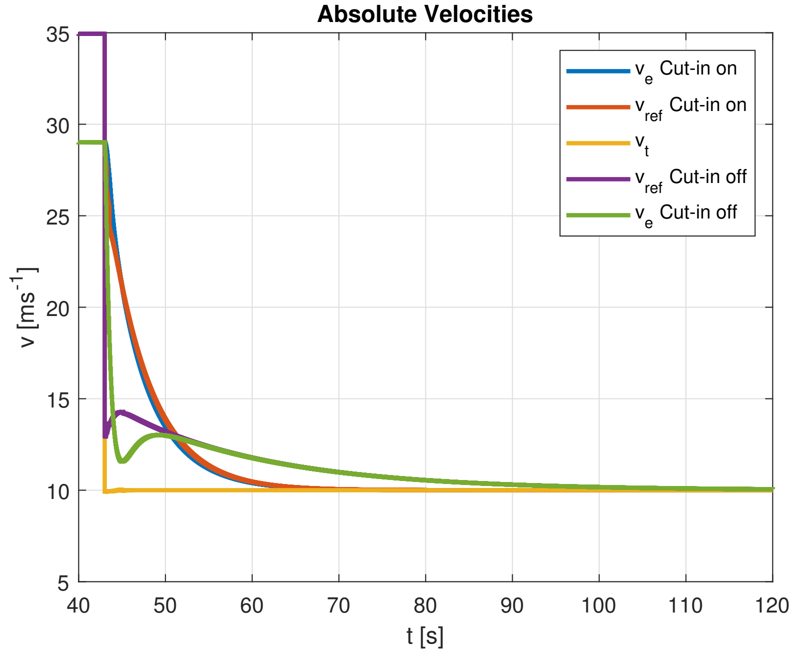

Figure 5.12: Absolute velocities in Cut-in maneuver

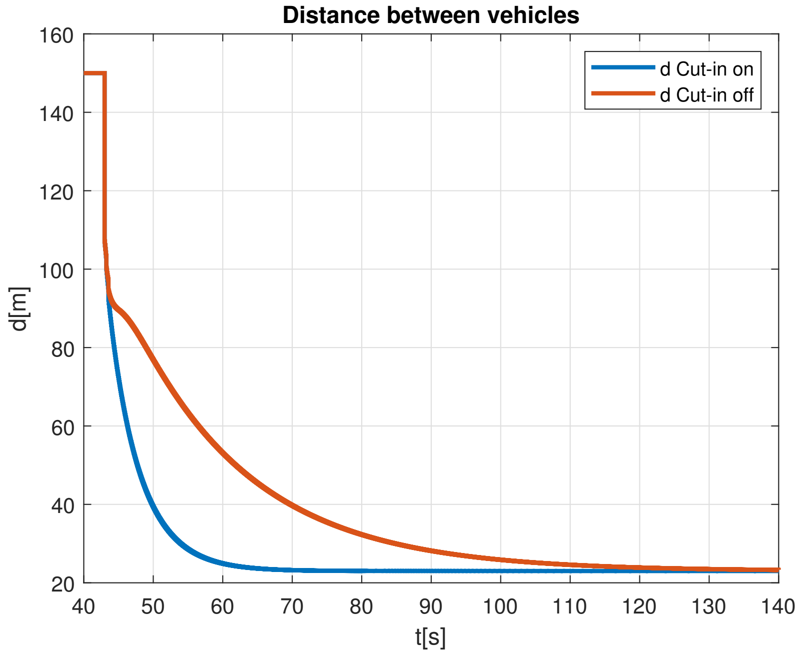

Figure 5.13: Distance between vehicles in Cut-in maneuver

In matters of passengers comfort system with Cut-in is also more comfort as can be seen form acceleration and jerk grapghs 5.14 and 5.15, where we can see much milder peaks and more smooth approach to desired spacing d_0.

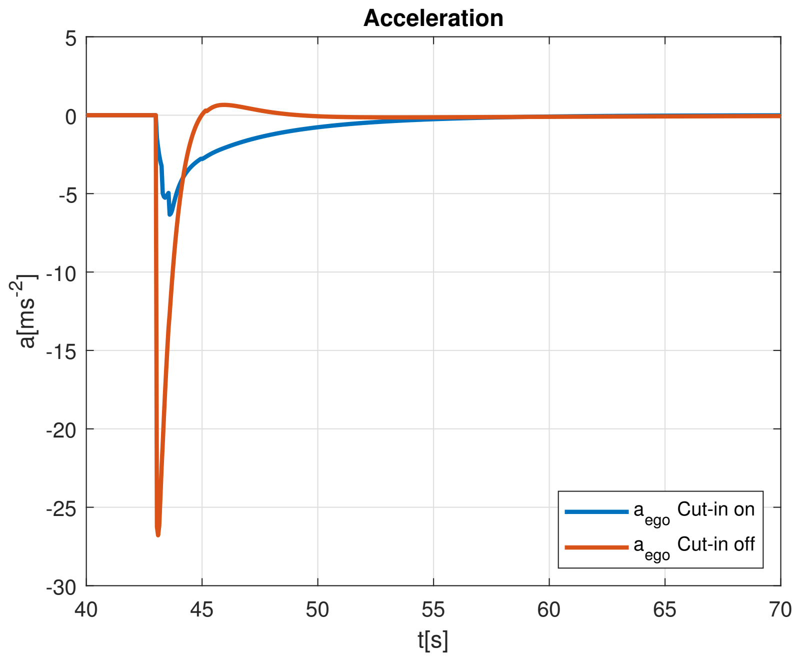

Figure 5.14: Eho absolute acceleration in Cut-in maneuver

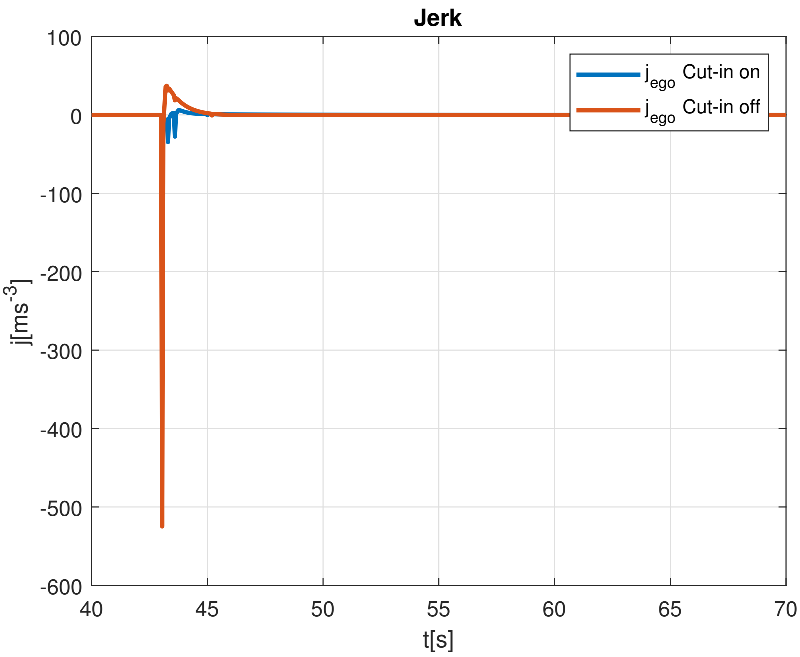

Figure 5.15: Ego jerk in Cut-in maneuver

5.4 Emergency Breaking Maneuver Result

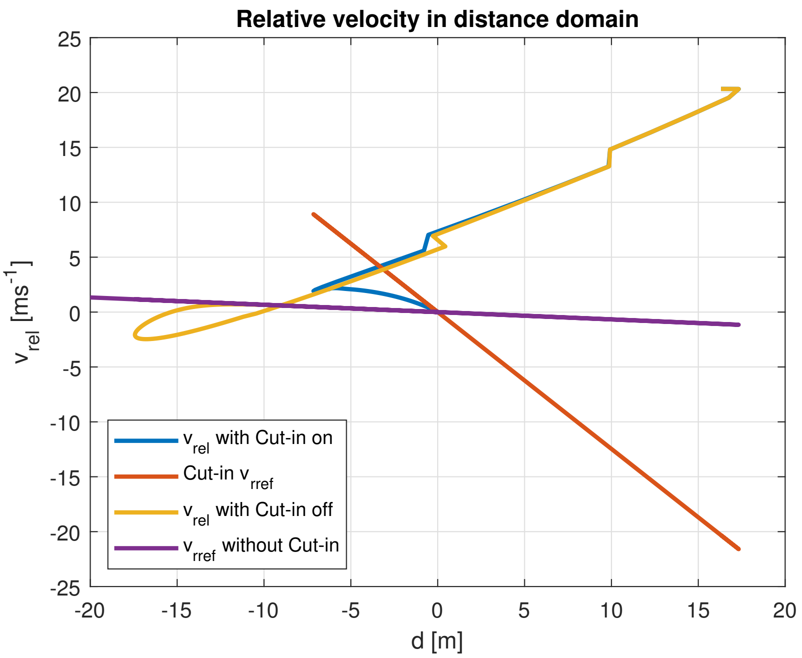

Figure 5.16: Relative Ego velocity in distance domain in Emergency breaking maneuver

Emergency breaking utility force Ego vehicle to quickly change its velocity v_e to prevent crashing into Target vehicle (Figs. 5.16 and 5.17). Purpose of this utility is mainly to keep safety of passengers. It means that comfort is no more important in these maneuvers. We could see some overshoots and mild oscillations even in version with Cut-in system implemented.

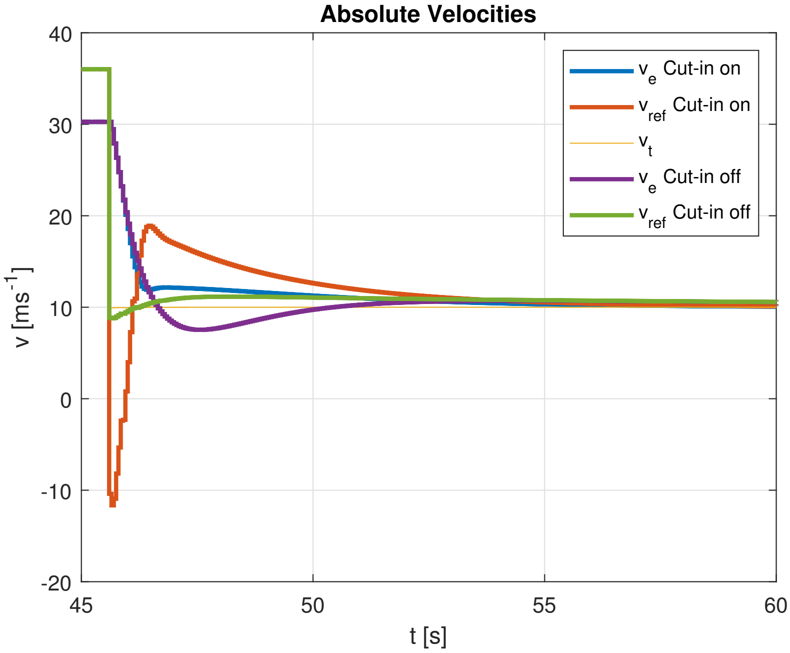

Figure 5.17: Absolute velocities in Emergency breaking maneuver

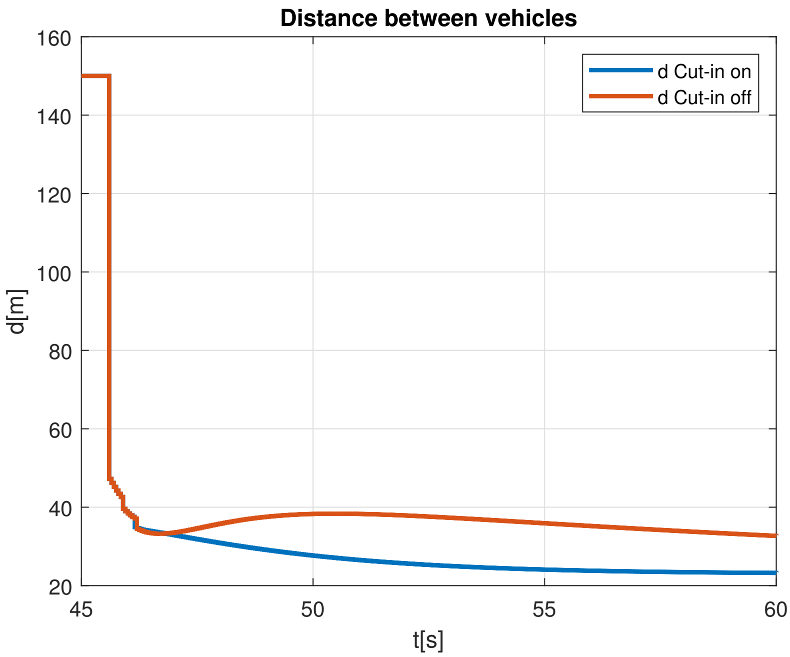

Figure 5.18: Distance between vehicles in Emergency breaking maneuver

However, as can be seen in Figs. 5.19 and 5.20 some level of comfort is still kept in comparison with system without Cut-in implemented.

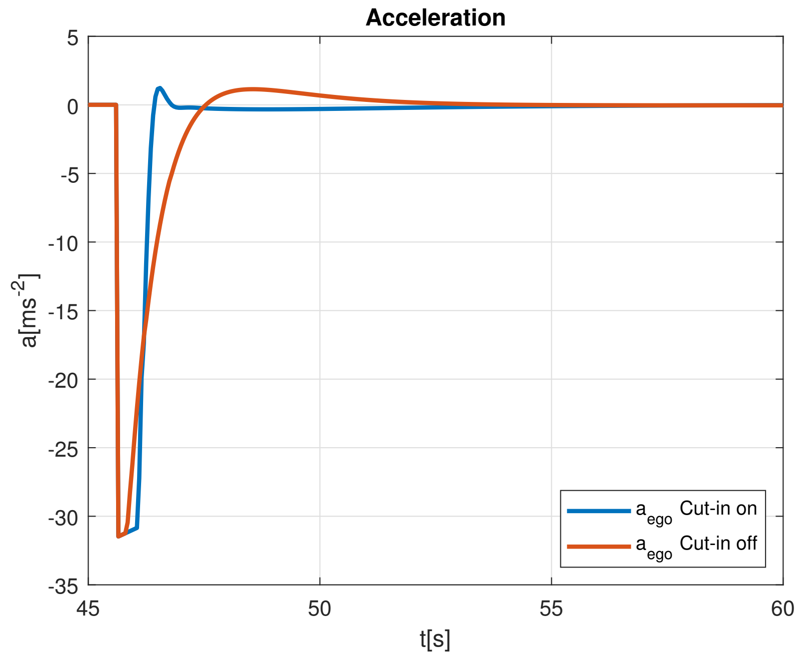

Figure 5.19: Ego absolute acceleration in Emergency breaking maneuver

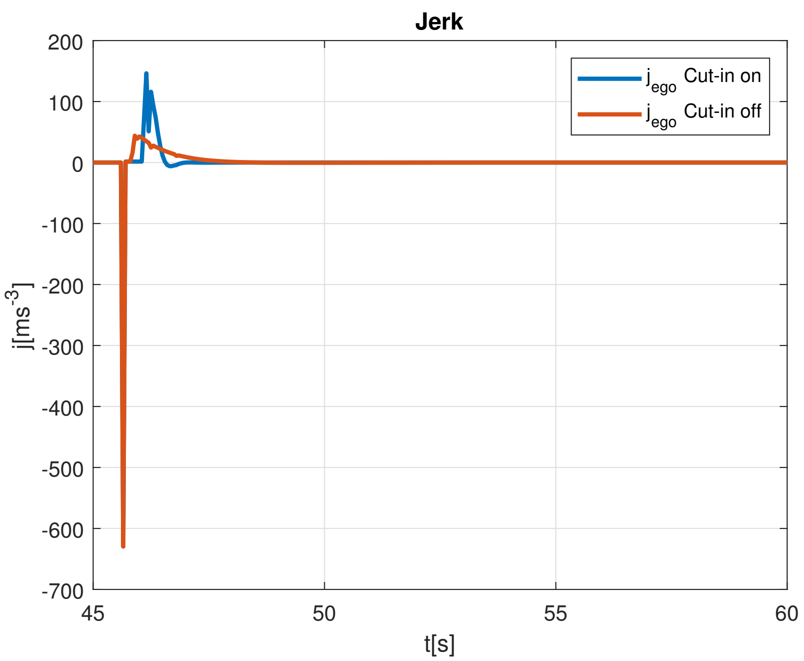

Figure 5.20: Ego jerk in Emnergency breaking maneuver

Chapter 6

Conclusion

As was mentioned in the Introduction, motivation for this work was to develop and implementnt testing framewotk for purpose of ACC system testing in virtual environment and to design ACC with best possible safety and comfort of passengers.
Testing framework was designed with usage of ROS2 network and with virtual vehicle environment. The framework can be used even in local environment or in distributional system. In future could be also deployed on vehicle simulators.
It was succesfully designed ACC system with its three parts, Target Vehicle Follow system, which approaches desrired spacing between vehicles and keeps it, Cruise Control system, which keeps predefined velocity and Cut-in system, which can handle potentionally dangerous situation in traffic. All these systems were succesfully designed with certain level of safety and comfort of passangers

Appendix A

Bibliography

[Cas20] S. Castro, *Getting Started with MATLAB, Simulink, and ROS*, November 8, 2017 (accesed 2020), https://blogs.mathworks.com/racing-lounge/2017/11/08/matlab-simulink-ros/.

[FPEN10] G.F. Franklin, J.D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamics Systems*, Sixth Edition **6** (2010), 39–41.

[Inc20a] The MathWorks, Inc., *Automated Driving Toolbox Documentation*, accesed 2020, https://www.mathworks.com/help/driving/.

[Inc20b] ———, *Generate Sensor Detection Blocks Using Driving Scenario Designer*, accesed 2020, https://www.mathworks.com/help/driving/ug/generate-sensor-detection-blocks-using-driving-scenario-designer.html.

[Inc20c] ———, *ROS Toolbox Documentation*, accesed 2020, https://www.mathworks.com/help/ros/.

[Raj11a] R. Rajamani, *Vehicle Dynamics and Control*, Second Edition **2** (2011), 150–151.

[Raj11b] ———, *Vehicle Dynamics and Control*, Second Edition **2** (2011), 158–160.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Švancar Jan** Personal ID number: **474467**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Adaptive Cruise Control algorithm development

Bachelor's thesis title in Czech:

Vývoj algoritmů adaptivního tempomatu

Guidelines:

The longitudinal Advance Driver Assistance Systems are critical building blocks for autonomous driving technology. This thesis will implement Adaptive Cruise Control algorithms for purpose of autonomous vehicle testing. The thesis will address following points:
1. Implement testing framework for ACC functionality development and testing
2. Implement ACC algorithm
3. Design testing scenarios using virtual environment
4. Validate ACC functionality

Bibliography / sources:

[1] Dieter Schramm, Manfred Hiller, Roberto Bardini – Vehicle Dynamics – Duisburg 2014
[2] Hans B. Pacejka - Tire and Vehicle Dynamics – The Netherlands 2012
[3] Franklin, Powell, Emami-Naeini: Feedback Control of Dynamics Systems. Prentice Hall, USA
[4] Robert Bosch GmbH - Bosch automotive handbook - Plochingen, Germany : Robet Bosch GmbH ; Cambridge, Mass. :Bentley Publishers

Name and workplace of bachelor's thesis supervisor:

Ing. Tomáš Haniš, Ph.D., Department of Control Engineering, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **04.02.2020** Deadline for bachelor thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

Ing. Tomáš Haniš, Ph.D. Supervisor's signature	prof. Ing. Michael Šebek, DrSc. Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student's signature