Fakultät für Informatik
Lehrstuhl für Echtzeitsysteme und Robotik

Report Group 9: Adaptive Reachability Analysis to Safe Driving of Autonomous Vehicles

Maximilian Stefan Schreber

Practical Course *Motion Planning for Autonomous Vehicles* WS 2023/2024

Advisor: Di Liu, Simone Baldi

Supervisor: Prof. Dr.-Ing. Matthias Althoff

Submission: 02. February 2024

Practical Course: Reachability Analysis to Safe Driving of Autonomous Vehicles

Maximilian Stefan Schreber
Technische Universität München
Email: max.schreber@tum.de

***Abstract*—This Report summarizes my efforts for this course during the semester. Starting with the Problem Description, followed by a concept overview and the actual implementation.**

I. Problem Statement

This Problem statement focuses itself to a Situation of a so called *mixed platoon*. Which describes a sequence of vehicles driving closely behind each other on the same lane. While all preceding vehicles are human-driven, the last vehicle is automated and has the ability to collect location data and telemetry for example acceleration.

A. Description of the mixed platoon

Our sequence of following cars is indexed by i , starting with the preceding vehicle $i = 1$ and optionally the following vehicle $i - 1$. See Figure 1 for a general overview.

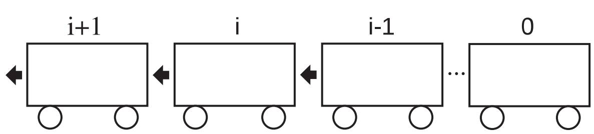

Fig. 1. Pictograph of the platoon

In this Practical Course, we concentrate ourselves on the effect of the preceeding vehicle $i + 1$ on vehicle i. In further studies, (handled by Di Liu,... **BRO BITTE HIER REFERENCE VERMERK**) This is used to approach a control structure that minimizes potential human caused errors.

B. System Description

For this Course, we consider the following quantities:

- acceleration of vehicle i: a_i
- velocity of vehicle i: v_i

In addition we introduce two errors:

- spacing error of vehicle i: e_i
- relative velocity of vehicle i in respect to vehicle $i - 1$: ν_i

The spacing error e_i is the difference between the actual distance and the desired distance between two vehicles, considering the *standstill distance* r and the *time headway* h.

$$e_i = s_{i-1} - s_i - hv_i - r$$

The relative velocity ν_i is the velocity difference between vehicle i and $i - 1$. $\nu_i = v_{i-1} - v_i$. The vehicle acceleration a_i is described by the Optimal Velocity Model (OVM):

$$a_i = \alpha \ (V(d_i) - v_i) + \beta \ (v_{i-1} - v_i)$$

$$V(d_i) = \frac{v_{max}}{2} \left(1 + erf \left(10 \frac{d_i - \frac{h_{go}+h_{st}}{2}}{\pi(h_{go} - h_{st})} \right) \right)$$

$V(d_i)$ denotes the Optimal Velocity Function which approximates the optimal velocity of the vehicle with a given distance d_i to the preceeding vehicle. h_{st}, h_{go} denote the time headway at standstill and while in motion. v_{max} is the current maximum velocity. These quantities can now be defined as a nonlinear system with input $a_i - 1$ and output e_i, ν_i and a_{i-1}: (*)

$$\begin{bmatrix} \dot{e_i} \\ \dot{\nu_i} \\ \dot{v_{i-1}} \end{bmatrix} = \begin{bmatrix} \nu_i - h(\alpha(V(e_i + h(v_{i-1} - \nu_i) + r) - v_{i-1} + \nu_i + \beta\nu_i) \\ i - 1 - \alpha(V(e_i + h(v_{i-1} - \nu_i) + r) - v_{i-1} + \nu_i) - \beta\nu_i \\ a_{i-1} \end{bmatrix}$$

C. Description of the Reachable Set

Since the disturbances i.e the acceleration a_{i-1} from the preceding vehicle are bounded, the output is also limited. These limits can be represented by the *Reachable Set*. When considered only e_i and a_i as output variables, the Reachable Set can be shown as an ellipsoid.
My task is to determine this Reachable Set for simulated disturbances using Reachability Analysis. Furthermore, the Reachable Set of every Vehicle in the mixed Platoon can be calculated, in order to create a new control structure that compensates the disturbances of all human driven vehicles. Due to the extensive networking of the cars, the plan is to develop an appropriate control strategy, in order to minimize the Reachable Set of the automated vehicle:

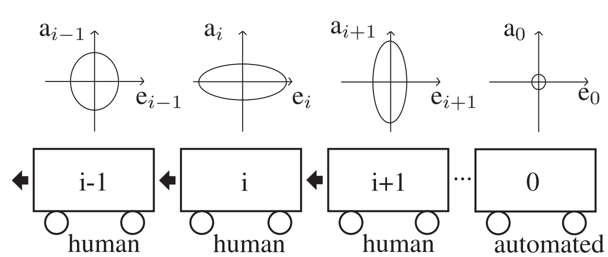

Fig. 2. Example mixed platoon with example reachable sets

D. Example values for this project

In order to clarify possible results or resolve issues during the project, the system values are fixed at $\alpha = 0.6; \beta = 0.9; v_{max} = 30\frac{m}{s}; h_{st} = 5; h_{go} = 35; v^* = 15\frac{m}{s}; d^* = 20m; h = 0.6283; r = 10.5752m.$

II. Proposed methods and implementation: Reachable Set of one vehicle

The Reachability Analysis can be acheived by using MATLAB calculations. In this particular case I will use the CORA(**Reference maybe)** Toolbox provided by the TUM Chair of Cyber Physical Systems.

A. CORA Toolbox

The CORA Toolbox, where CORA stands for **CO**ntinuous **R**eachability **A**nalyzer, provides a range of methods for calculating reachable sets, simulating disturbance trajectories, and presenting results in an academic fashion. Details about each used method are presented in the following chapters.

B. defining the parameters

To perform the reachability analysis, one must define all necessary parameters and options. The initial state is defined through a *CORA-Set-Representation*. I opted for a three-dimensional zonotope, containing $[e_i \approx 0m \; ; \nu_i \approx 0m \; ; v_{i-1} = 8\frac{m}{s}]$ as the centre point and an appropriate generator matrix. Additionally I had to limit the input disturbance to $[-5\frac{m}{s^2}, 5\frac{m}{s^2}]$. This is done by using the *params.R0* method for the initial state and the *params.U* method for the input set. *params.tFinal* limits the amout of time steps in this calculation.

C. defining the nonlinear system

To define the dynamic system, I defined in Section I of this report, I used the method

```
sys = nonlinearSys(fun, n, m, outFun, o)
```

where `fun` denotes the state function, `n, m, o` is the number of states, inputs and outputs and `out` is the output function. the state function and the output function are defined, using the MATLAB function notation. The state function is represented by the nonlinear system defined in section I.
Since we are only interested in the acceleration a_i and spacing error e_i. I needed to define an output function $g(x, u)$:

g(x,u)= $\begin{bmatrix} x_1 \\ \alpha(V(x_1 + h(x_3 - x_2) + r) - x_3 + x_2) + \beta x_2 \end{bmatrix} = \begin{bmatrix} e_i \\ a_i \end{bmatrix}; x = \begin{bmatrix} e_i \\ \nu_i \\ v_{i-1} \end{bmatrix}$

D. Performing reach()

After adding some additional options (e.g. timestep size, maxError, zonotopeOrder or the used algorithm), the reachability analysis method can be defined:

```
    R = reach(sys, params, options)
```

this method takes the previously defined dynamic system (`sys`), the parameter handle (`params`) and the options handle (`options`) as arguments, and returns the Reachable Set as an object of class `reachSet`.

E. Performing simulations

After calculating the reachable set, one can simulate random disturbances and calculate trajectories. These trajectories start from a randomly generated initial state and follow the input signal, in this case the accelleration of the preceeding vehicle a_{i+1}. In order to do so, the CORA Toolbox provides suitable methods, such as `simulateRandom()`:

```
simRes = simulateRandom(sys,params,simOpt)
```

Similar to the `reach`-command, this method takes the nonlinear system (`sys`), the `param` struct, and the `options` struct (here defined as `simOpt`). My implementation stores the simulation in the object `simRes`. This method generates random initial states which are located inside the predefined initial set (`params.R0`). Additionally, I opted for the standard sampling method, resulting in initial states having an undefined distribution compared to a potential Gaussian normal distribution.
To limit the amount of simulations, I used the `.points` setting. This setting fixates the number of random generated initial states to a specific integer value.

F. Visualization of single vehicle calculations

In order to illustrate the reachable Set, the simulations and the initial set in the same graph, I created `plot()`-commands for every object and staggered them in one `figure`. Due to the reason that the system output only considers two outputs and only a_i or e_i are important for our reachability analysis, two graph-dimensions are sufficient enough to continue. Together with the CORA-typical colorset `CORA:contDynamics` I am able to visualize my first calculations.

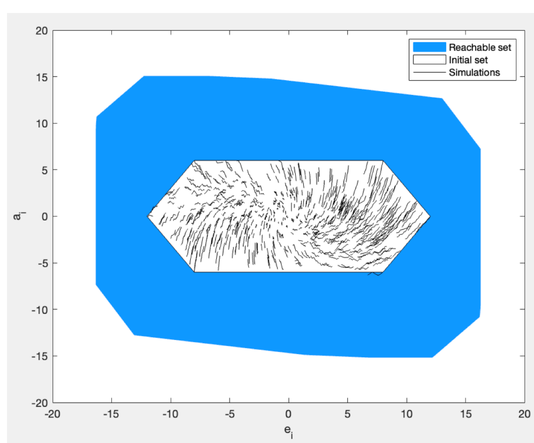

Fig. 3. plot of reachable set with final time of 0.2, time step size of 0.1, 600 simulations, a *adaptive linear* algorithm and a hexagonal zonotope as initial set.

III. Reachable sets for multiple vehicles

After calculating the reachable set for one individual vehicle, one can start combining the dynamic system of several other vehicles. Therefore I need to adjust and automate different processes in our system.

A. *Adding a second vehicle to the system dynamics*

To integrate the dynamics of the second vehicle, I added $\dot{e}_{i-1}$ and $\dot{\nu}_{i-1}$ of the following vehicle $i-1$. The third Variable $\dot{v}_i$ can be omitted. To clear the dependencies from system variable $\dot{v}_i$, the v_i on the right hand side can be substituted by $v_{i+1} - \nu_i$ through $\nu_i = v_{i-1} - v_i$. This leads us to the following system: $\left[\dot{e}_i;\ \dot{\nu}_i;\ \dot{v}_{i+1};\ \dot{e}_{i-1};\ \dot{\nu}_{i-1};\right]$ =

$$\begin{bmatrix} \nu_i - h(\alpha(V(e_i + h(v_{i+1} - \nu_i) + r)) - v_{i+1} + \nu_i) + \beta\nu_i \\ a_{i+1} - \alpha(V(e_i + h(v_{i+1} + \nu_i) - \beta\nu_i \\ a_{i+1} \\ \nu_{i-1} - h(\alpha(V(e_{i-1} + h((\underline{v_{i+1} - \nu_i}) - \nu_{i-1}) + r)) - (\underline{v_{i+1} - \nu_i}) + \nu_{i-1}) + \beta\nu_{i-1} \\ a_i - \alpha(V(e_{i-1} + h((\underline{v_{i+1} - \nu_i}) + \nu_{i-1}) + r) - (\underline{v_{i+1} - \nu_i}) + \nu_{i-1} - \beta\nu_{i-1} \end{bmatrix}$$

B. *Adding arbitrarily many vehicles*

Since every vehicle adds two variables to the system, and only accesses values from the preceding vehicles, the system can be expanded to $2n + 1$ rows with n vehicles. I modified the two-vehicle calculations, with reasonable changes to the state function, output function and the initial set. To model the system adaptively based on the number of vehicles, I used for-loops that always refer to the preceding vehicle for ν_{i+1}, e_{i+1}, and v_i. This results in a progressively built structure.

C. *Visualization*

It is worth noting that the remodeled output function only considers the acceleration and spacing error of the last following vehicle. As a result, it is only feasible to plot the reachable set of this vehicle and not those of the previous vehicles. That means that one has to calculate the reachable set for every vehicle individually to visualize all reachable sets. To do that, I use the `reach()` method for every vehicle in the platoon. Utilizing the previously defined reachability analysis for n vehicles, I can iterate through every vehicle by using a `for`-loop, and the loop index as parameter for every `reach()` iteration. This program then results in a sequence of graphs that allow us to inspect the process of change for every single vehicle.

IV. Conclusion

A. *Summary of my work*

What have I accomplished?

B. *weak points of this project*

what Improvements can be made in future projects

C. *Advantages to previous work*

In what case is it better than existing work? Does it have a special feature?

D. *Relevance to practical applications*

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

V. TODO:

- explain the given literature a bit more, include references to the slides, paper and CORA manual which gaps in here am I trying to fill? Include work that already has been done (Paper by Di Liu and S.Baldi) and how i anknÃ¼f on it :D How does it go beyond previous research insert an overview of the structure Predefine what my thesis assumes, so that nothing appears to be missing