Challenges for optimal resource utilization in autonomous systems

PhD Student

Aloysio GALVÃO LOPES

Advisors

Eric GOUBAULT
Laurent PAUTET
Sylvie PUTOT

LIX - École Polytechnique, LTCI - Télécom Paris

{galvaolopes,goubault,putot}@lix.polytechnique.fr
laurent.pautet@telecom-paris.fr

November 22, 2022

Outline

Outline

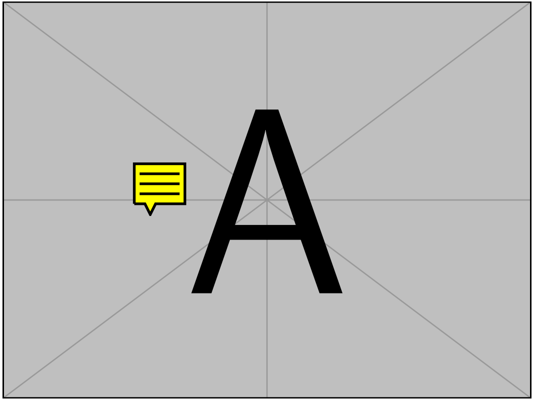
- Motivation
- Internship
- Thesis
- Conclusion

Motivation

What kinds of tasks an UAV might need to do?

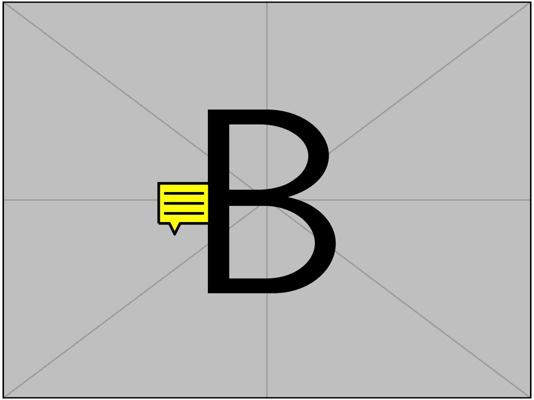
Motivation

What kinds of tasks an UAV might need to do?



Motivation

What kinds of tasks an UAV might need to do?



Outline

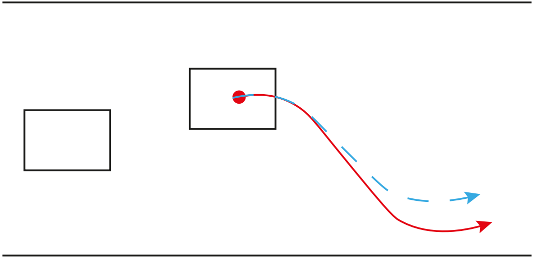
Context

Predictions without formal guarantees

Predictions alongside reachability analysis

Context

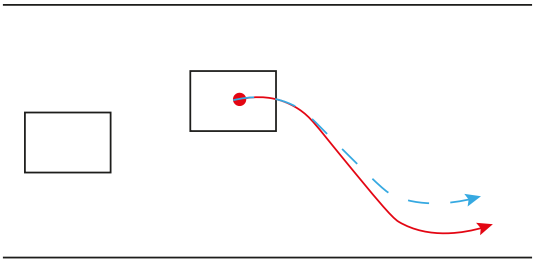
Predictions without formal guarantees



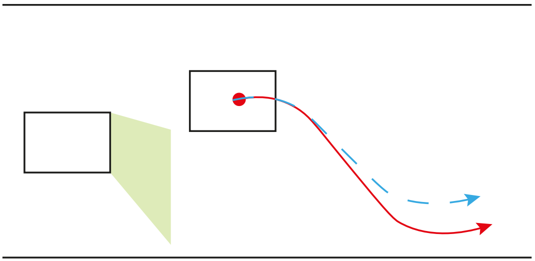
Predictions alongside reachability analysis

Context

Predictions without formal guarantees

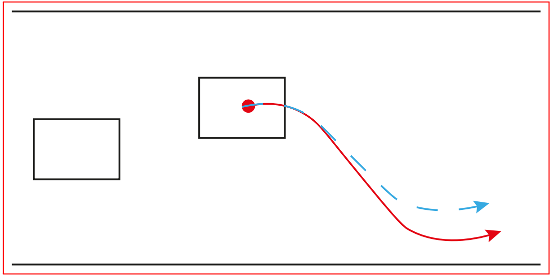


Predictions alongside reachability analysis

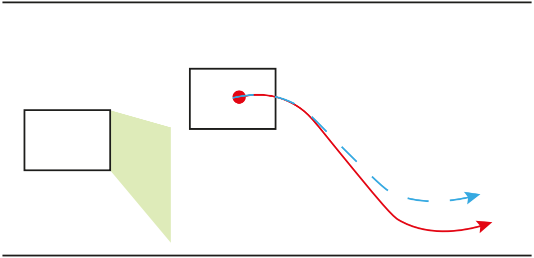


Context

Predictions without formal guarantees



Predictions alongside reachability analysis



Overview

- First, we need to be able to have training data in a suitable form to our problem.
- As this data is not readily available, especially for the real world scenario, we need to generate ourselves diverse data.
- We generate this data by varying parameters of a controller in different maps.
- Then, training takes place considering data correctly converted to the Frenet-Serret reference frame.
- Finally, we evaluate our performance considering different horizons of prediction (short and large).

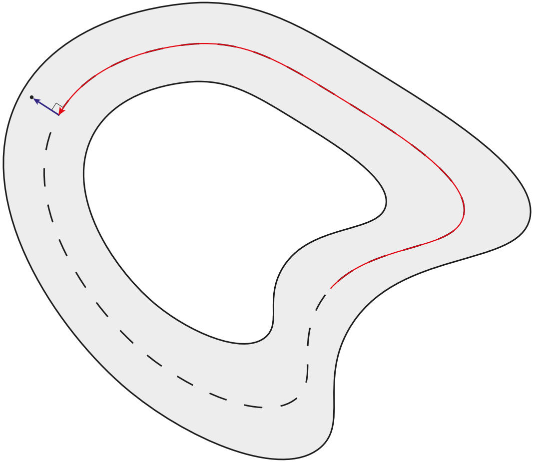
Mapping and filtering

Mapping is done with Gmapping[1], then, the map is filtered with a custom algorithm.

[1] *Gmapping - ROS Wiki*. URL: http://wiki.ros.org/gmapping (visited on 07/21/2022).

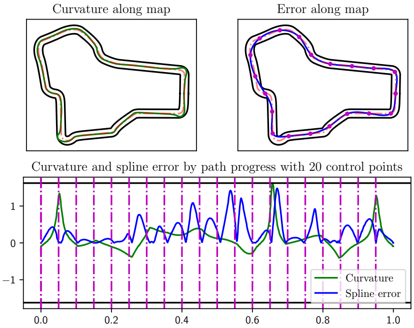
Conversion to spline representation

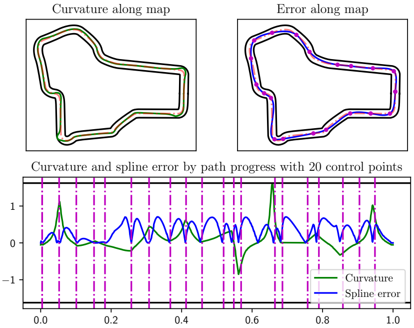
The points are represented in our reference frame using the notation (s, δ):



Centerline spline optimization

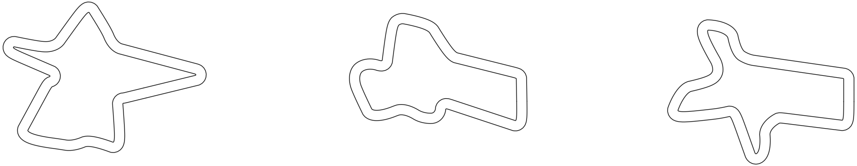
Centerline points are fitted by a cubic spline (in the case below 0.71m vs 1.48m error).





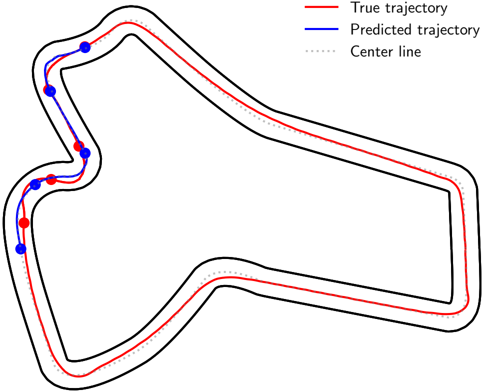
Diverse map generation

Diverse maps are generated programmatically as shown below:



Trajectory generation

For each map, trajectories are generated with a controller which tries to keep the car in the centerline at constant speed. Evaluation considers the maximum error and the mean error.



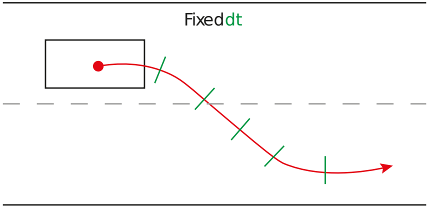
Trajectory prediction

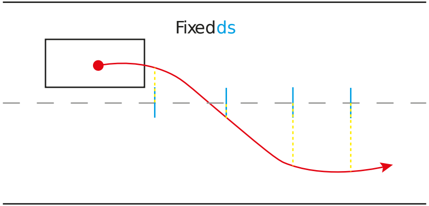
Inputs for fixed dt:

- Curvatures considering fixed speed.
- ds_i, the past variations in progress for fixed dt.
- δ_i, the past deviations to the centerline.

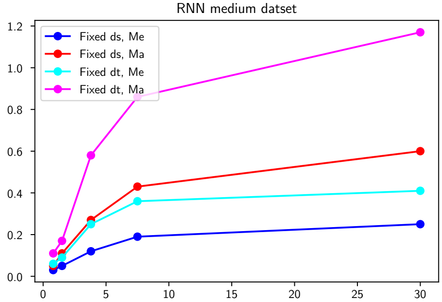
Inputs for fixed ds:

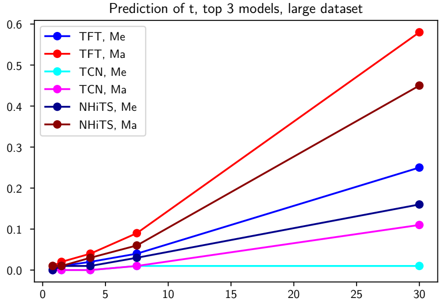
- Curvatures.
- dt_i, the past variations in time for fixed ds.
- δ_i, the past deviations to the centerline.





Some highlights





In the left we compare in a single model (RNN) using fixed *ds* and *dt* in two different metrics. In the right we compare the performance of different models in predicting the speed. The x-axis is the time horizon and the y-axis is the error.

Observations and conclusions

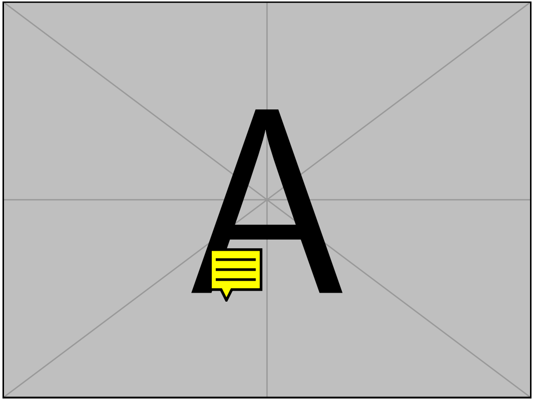
- Some architectures proved to benefit from inversion: fixed ds to fixed dt, notably RNN and TCN. We highlight that those architectures give lots of importance to local information, therefore accurate curvatures might have had a bigger impact because of that.

- TCN had impressive results for the prediction of the time t. However, we need to verify if that still holds for vehicles without constant speed.

- Increasing the data improved mostly the bigger models, such as TFT.

Outline

What kinds of tasks an UAV might need to do?



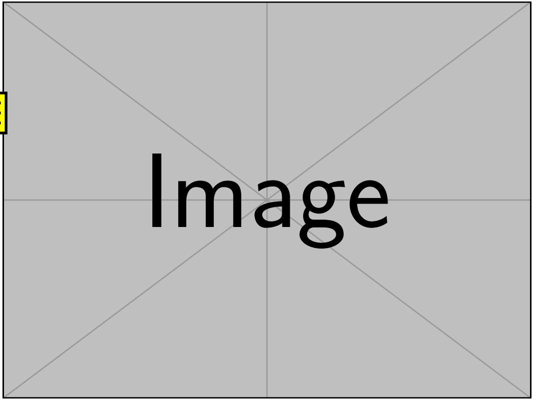
Core problems

- How to ensure safety in the context of trajectory planning?
- How to maximize the resource usage and, at the same time, ensure a tolerable level of safety?

Core problems

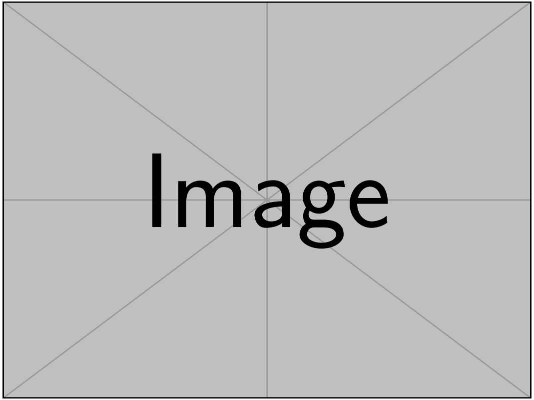
- How to ensure safety in the context of trajectory planning?
- How to maximize the resource usage and, at the same time, ensure a tolerable level of safety?
- Some methods that can help to answer these questions and model these problems are mixed criticality systems and probabilistic real time systems.
- The focus now is to provide probabilistic predictions and use those predictions to optimize resource usage.
- The study of probabilities in real time systems focuses on the analysis of the uncertain behavior of the processor not the environment.

How can mixed criticality be useful



Probabilistic collision detection

Works such as[2] propose probabilistic and safe collision detection methods, but they are not learning based.



[2] Yi Chou, Hansol Yoon, and Sriram Sankaranarayanan. "Predictive Runtime Monitoring of Vehicle Models Using Bayesian Estimation and Reachability Analysis". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV, USA: IEEE, Oct. 24, 2020, pp. 2111–2118.

Outline

Next steps

- Adapt our method to provide probabilistic predictions and safety guarantees.
- Train our model with more complex adversary behavior.
- Study applicability of the concepts of probabilistic real time systems in our setting.

Summary

Summary:

- We developed a framework which allows the use of time series prediction models for trajectory prediction.
- The focus of the thesis will be on the optimization of the usage of computational resources in the context of UAV tasks.
- The main next areas of research will be mixed criticality systems, probabilistic trajectory prediction and probabilistic real time systems.

Takeaways:

- In the Frenet-Serret reference system, discretizing on space rather than time could be very beneficial.
- Probabilistic methods will be the next topic of research.

References I

Gmapping - ROS Wiki. URL: http://wiki.ros.org/gmapping (visited on 07/21/2022).

Chou, Yi, Hansol Yoon, and Sriram Sankaranarayanan. "Predictive Runtime Monitoring of Vehicle Models Using Bayesian Estimation and Reachability Analysis". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV, USA: IEEE, Oct. 24, 2020, pp. 2111–2118. ISBN: 978-1-72816-212-6. DOI: 10.1109/IROS45743.2020.9340755. URL: https://ieeexplore.ieee.org/document/9340755/ (visited on 04/10/2022).