An Inverse Optimal Control Approach for Trajectory Prediction of Autonomous Race Cars

Rudolf Reiter, Florian Messerer, Markus Schratter, Daniel Watzenig and Moritz Diehl

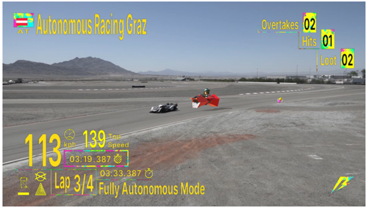
Systems Control and Optimization Laboratory
Autonomous Racing Graz
ELO-X

ECC 2022
July 12-15, 2022

Overview

What is Roborace? "The world's first extreme competition of teams developing self-driving AI." (roborace.com)
What is the relation of Roborace to this work? It is the setting where the necessity of these developed algorithms emerged and where the proposed algorithm was tested.
Key facts:

- Pre-season "Beta" until summer 2022
- 6 teams (4 university teams, 2 commercial teams)
- Mainly PhDs and postdocs involved



Introduction

General Goal: Prediction of Opponents

Within autonomous driving

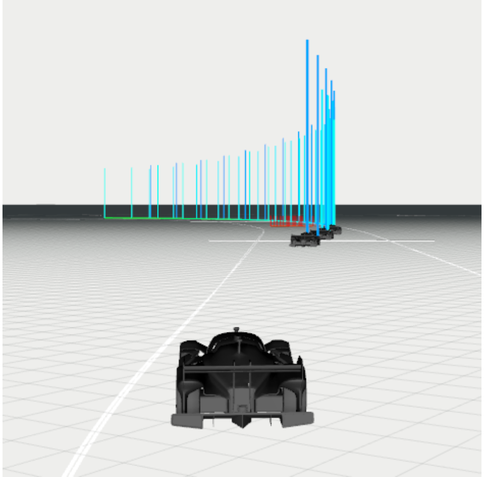
- In real world autonomous driving scenarios, a core challenge is the prediction of other agents in the environment.
- The prediction algorithms differ related to the scenario and to the availability of data

Within autonomous racing

- No a priori knowledge available about the opponents, except of their racing intention
- Impossible to use an a priori fully parameterized vehicle model as a basis for prediction.
- An extensive system and behavior identification is impossible due to the short time the vehicle can be observed before it needs to be overtaken.

Introduction

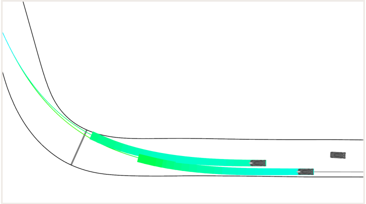
General Goal: Prediction of Opponents



Introduction

Our Approach

- Including a physics-based parametric model of the opponent inducing a *racing intention*
- The racing intention is modeled by means of a parametric nonlinear low-level program (LLNLP) for progress maximization
- The estimation of the parameters is performed by solving an inverse optimal control (IOC) problem, which enforces the optimality conditions for the LLNLP as constraints
- Output: Predicted trajectories

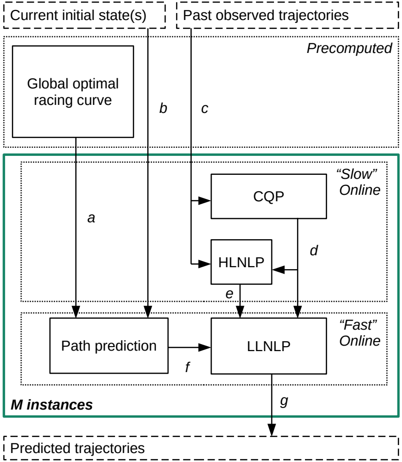


Advantages:

- A physically explainable prediction
- A good prediction even without any data
- Adaptive algorithm that improves with amount of data
- Fast improvement

Disadvantages:

- We ignore interactive behavior of any kind
- Structural bias even with an infinite amount of data



a: global racing path

b: initial state $\bar{x}_0$

c: trajectory data samples

d: constraints $a_{\max}$

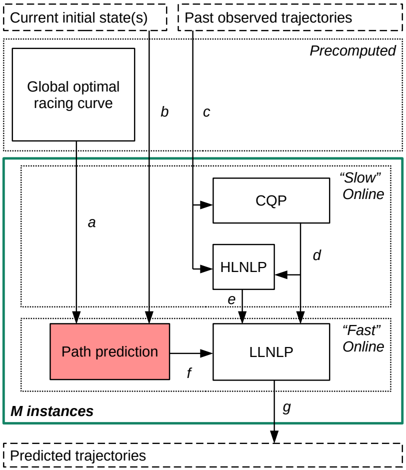
e: weights w

f: Cartesian coordinates and curvature parameters of blended path segment $\bar{\kappa}$

g: predicted trajectory

The Prediction Algorithm

Path Prediction

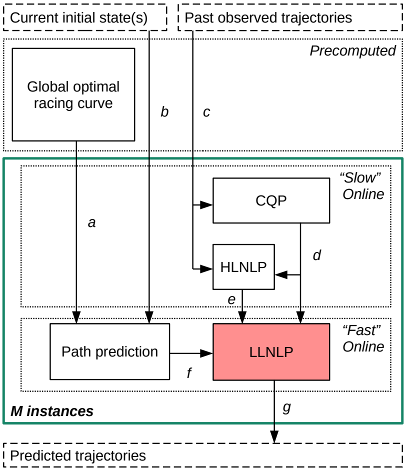


- Pre-path $p_{\text{topt}}(s)$: We compute a optimal racing path for the whole race track in advance
- Linear prediction based on current position $p_{\text{c}}(s)$
- Blend linear prediction in Frenet-frame towards pre-path until final prediction position s_f

$$p_{\text{p}}(s) = \frac{s}{s_f} p_{\text{topt}}(s) + \frac{s - s_f}{s_f} p_{\text{c}}(s). \tag{1}$$

The Prediction Algorithm

Low-Level Program for Trajectory Prediction (LLNLP)

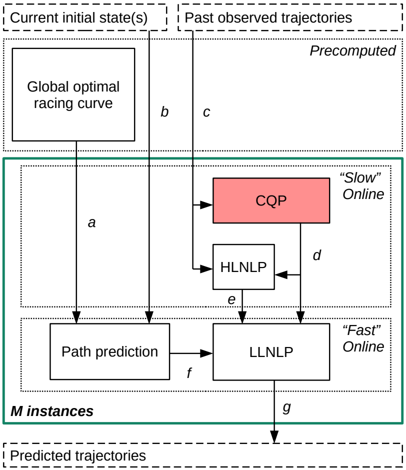


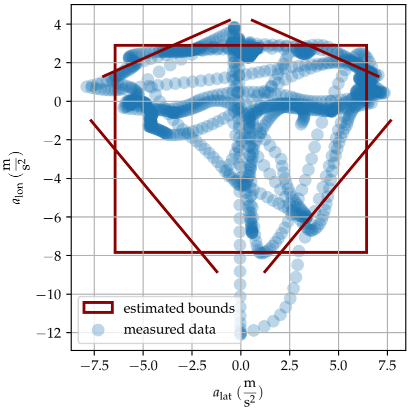
- Nonlinear program to maximize progress (x_N) along given path
- Weights R for trade-off w.r.t. inputs u (*To be estimated*)
- Acceleration constraints $h_a(x_k, \bar{\kappa}, a_{\max})$ (*To be estimated*)

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}\\ s_0,\ldots,s_N}} \quad & \sum_{k=0}^{N-1} \|x_k - x_k^{\mathrm{r}}\|_{2,W}^2 + \|u_k - u_k^{\mathrm{r}}\|_{2,R}^2 + q_N^\top x_N + \sum_{k=0}^{N} \alpha_1 \mathbf{1}^\top s_{\mathrm{LL},k} + \alpha_2 \|s_{\mathrm{LL},k}\|_2^2 \\
\text{s.t.} \quad & x_0 = \bar{x}_0, \\
& x_{k+1} = F(x_k, u_k, \Delta t), \qquad k = 0, \ldots, N-1, \\
& \underline{x} \preccurlyeq x_k \preccurlyeq \overline{x}, \\
& 0 \preccurlyeq h_a(x_k, \bar{\kappa}, a_{\max}) + s_{\mathrm{LL},k}, \\
& 0 \preccurlyeq s_{\mathrm{LL},k}, \qquad k = 0, \ldots, N,
\end{aligned}
\tag{2}
$$

The Prediction Algorithm

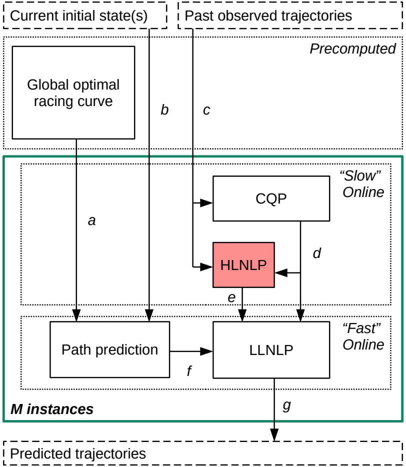
Quadratic Program for Constraint Estimation





- Constraints are estimated separatly from the weights
- Symmetric polytope with 8 bounds (5 independent) fitted to data
- Iterative QP, with previously estimated value as "arrival term" (moving horizon estimation)

The Prediction Algorithm
High Level Program for Weight Estimation (HLNLP)



The Prediction Algorithm

High Level Program for Weight Estimation (HLNLP)

- We optimize for the weights w of the LLNLP
- L2 loss on observed trajectories and predicted trajectories
- We use only states x and controls u that are solutions of the LLNLP $P_{\mathrm{LL}}(w, \bar{x}_0, \bar{\kappa}, a_{\max})$
- $\rightarrow$ bi-level optimization problem

$$\begin{aligned} \min_{x,u,w} \quad & \sum_{k=1}^{N_T-1} \|x_k - \bar{x}_k\|^2_{2,Q_k} + \|w - \hat{w}\|^2_{2,P^{-1}} \\ \text{s.t.} \quad & x, u \in \operatorname{argmin} P_{\mathrm{LL}}(w, \bar{x}_0, \bar{\kappa}, a_{\max}), \\ & w \succcurlyeq 0 \end{aligned} \tag{3}$$

- We use the the KKT conditions of the LLNLP as constraints in the HLNLP
- Needs penalized relaxation to avoid complementarity minima
- Arrival cost with weights P^{-1}

The simulation:

- Simulation framework with dynamic vehicle model
- Comparisons with Notebook
- Hardware-in-the-loop for competitions
- Las Vegas race track
- 1k randomly parameterized test runs
- (Due Covid currently only simulated races)

The setup:

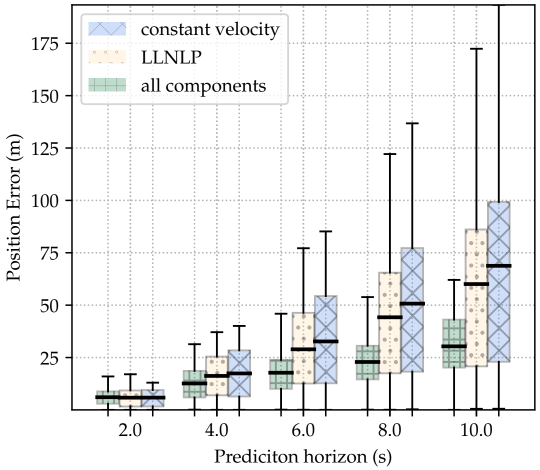
- Hardware: HP Elitebook, Intel Core i7-8550 CPU (1.8 GHz) and Nvidia Drive PX2
- The used frequency for the synchronous LLNLP was 10 Hz
- The HLNLP and CQP ran asynchronously
- 200 seconds until HLNLP converged

Table: Solver and timing statistics

Component	Solver	t_{max} (ms)	t_{ave} (ms)	fail rate (%)
PP	none	< 1	< 1	0
CQP	OSQP	15.5	8.1	0
HLNLP	IPOPT	6237	520	5
LLNLP	acados hpipm(QP)	2748	91	0.2

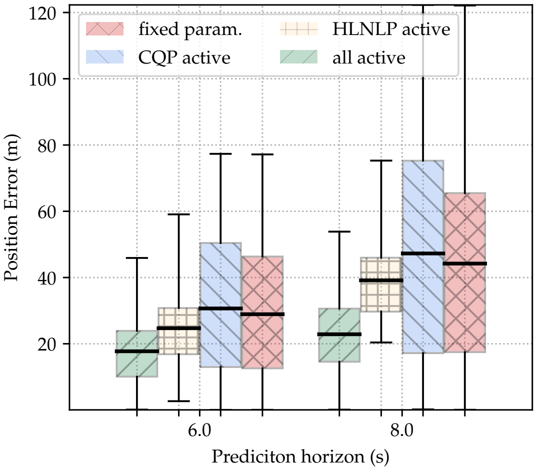
Results

Final Prediction Errors by Prediction Horizon (converged)



Results

Final Prediction Errors by Used Components (converged)



Conclusions

- We presented a physics based approach for predicting non-interactive race car trajectories with few data
- Solving nonlinear programs online for prediction
- Bi-Level optimization for weight estimation
- Quadratic programming for constraint estimation
- No guarantees of finding the best solution (heavily nonconvex NLP)
- Works well in practice

Outlook

- Find linear/quadratic/convex formulation of the LLNLP in order to get convergence guarantees
- Include interactive behavior through game-theoretic reasoning

Thank you for your attention!