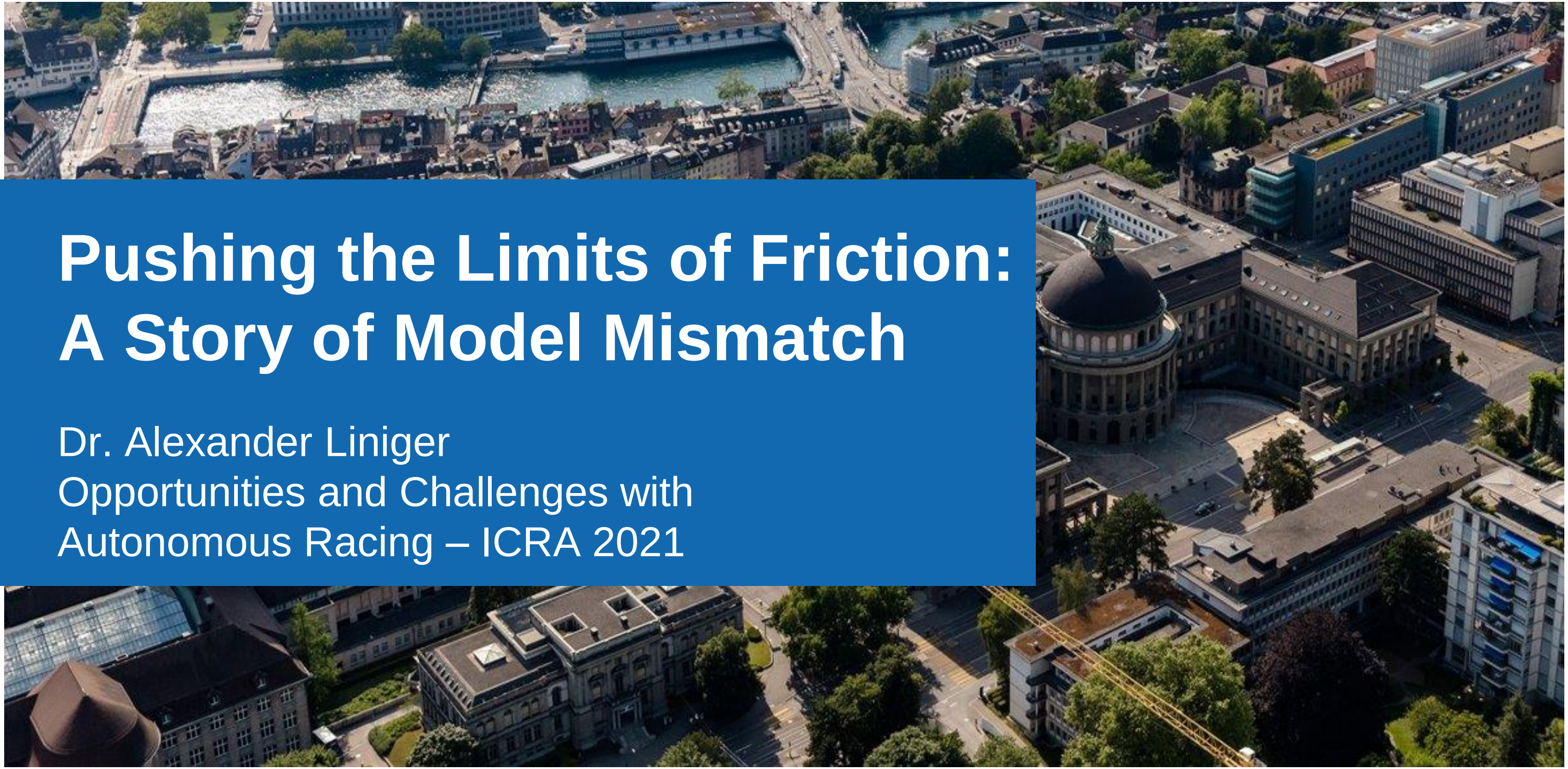


Pushing the Limits of Friction: A Story of Model Mismatch

Dr. Alexander Liniger

Opportunities and Challenges with Autonomous Racing – ICRA 2021

Motivation

- How can we design a **motion planning and control** strategy that pushes an autonomous racecar to the limit
- Model Predictive Control is a flexible and efficient method for this task
- Accurate but simple models are fundamental
    - Accurate to plan precise motion
    - Simple to keep computation in check

Motivation

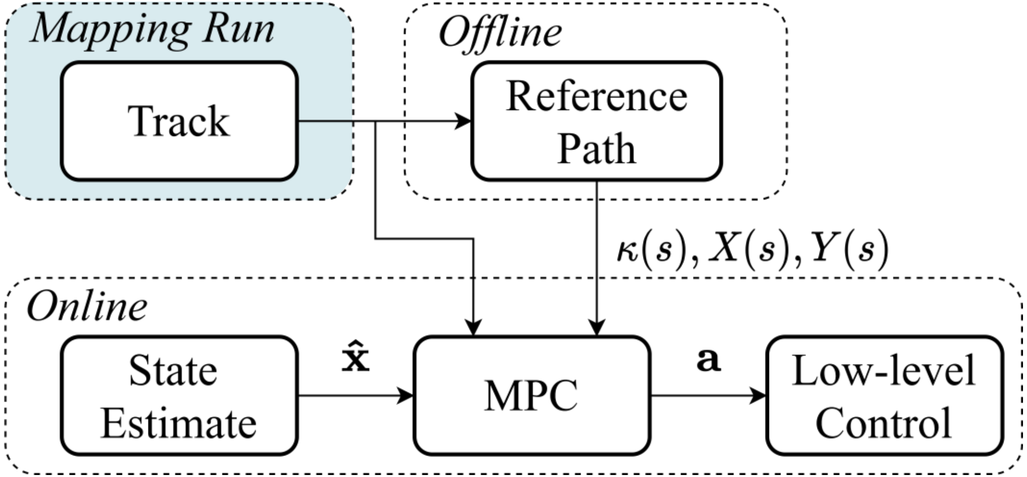
- Model accuracy is fundamental
  - Same controller, same model structure, **different model parameters**

@IfA ETH Zurich 2013

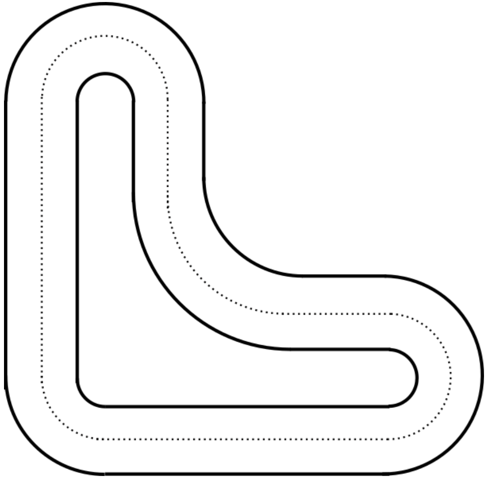
@IfA ETH Zurich 2015

An accurate vehicle model is crucial for high-performance control

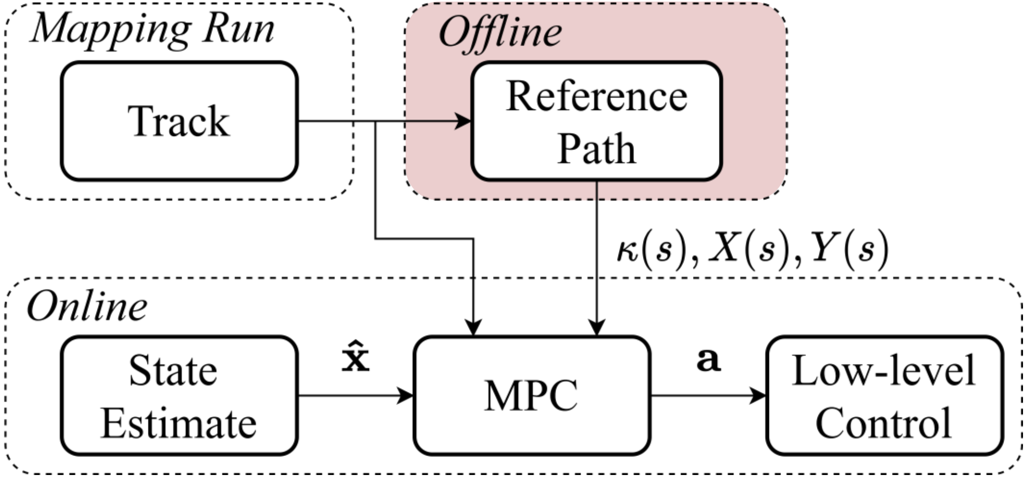
Hierarchical Planning & Controls Framework



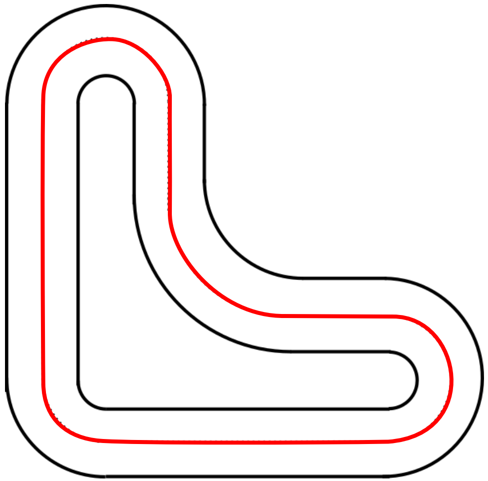
Before race:
Map race track



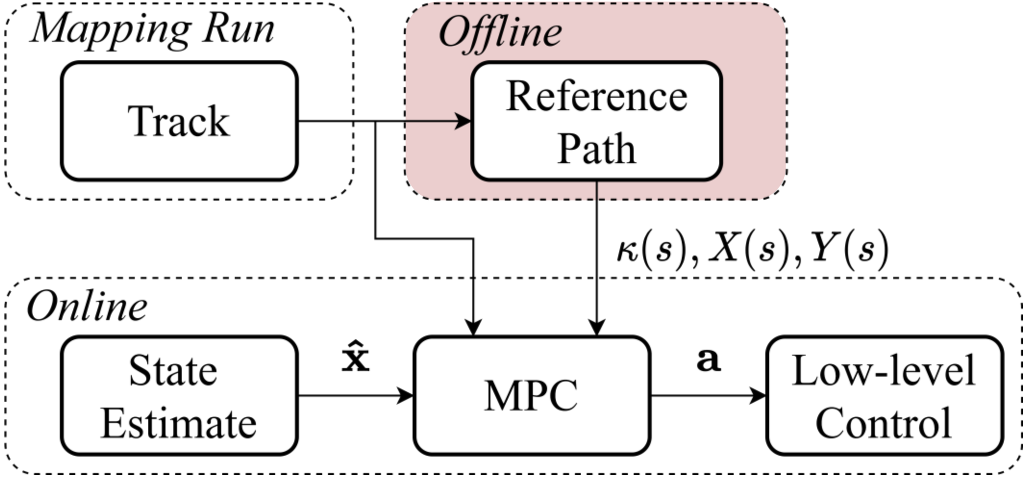
Hierarchical Planning & Controls Framework



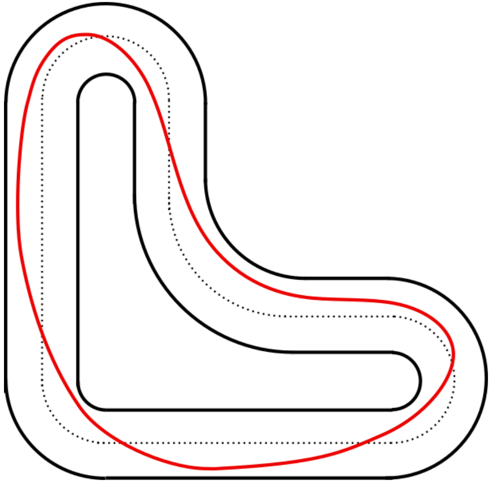
Before race:
Compute reference line



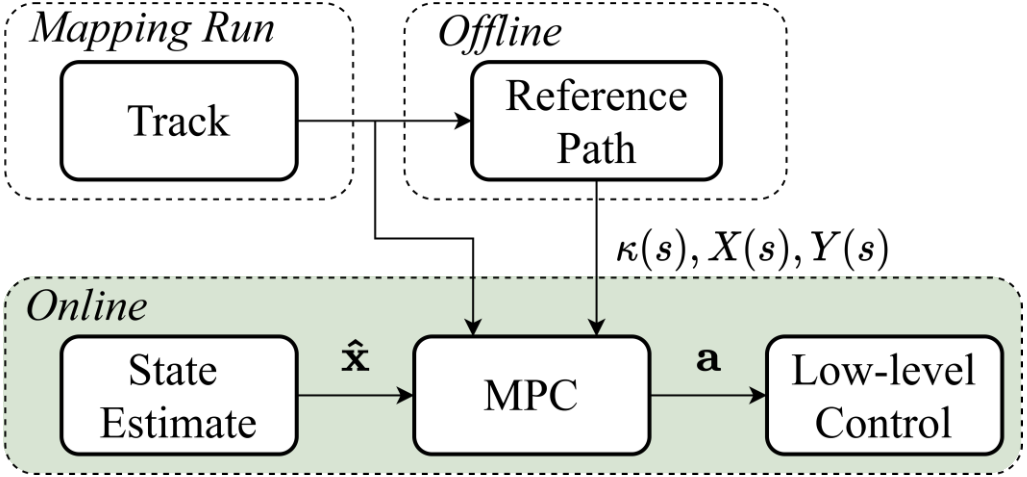
Hierarchical Planning & Controls Framework



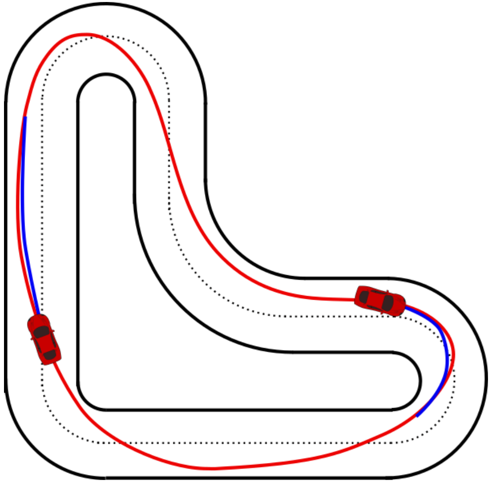
Before race:
Compute optimal raceline



Hierarchical Planning & Controls Framework

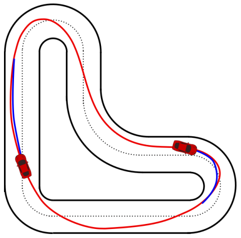


During race:
Race using online MPC



MPC Real-time Controller

- Progress maximizing path following MPC (MPCC or MPC-Curv)
- Follow the reference line while maximizing the progress
    - NMPC with 4th order Runge Kutta discretized model
    - Prediction horizon of ~2s



$$\begin{aligned} \min_{\mathbf{X},\mathbf{U}} \quad & \sum_{t=0}^{T} j_{\mathrm{MPC}}(\mathbf{x}_t, \mathbf{u}_t) \\ \text{s.t.} \quad & \mathbf{x}_0 = \hat{\mathbf{x}} \\ & \mathbf{x}_{t+1} = f_t^d(\mathbf{x}_t, \mathbf{u}_t) \\ & \mathbf{x}_t \in \mathcal{X}_{\mathrm{Track}}, \quad \mathbf{x}_t \in \mathcal{X}_{\mathrm{FE}} \\ & \mathbf{a}_t \in \mathcal{A}, \quad t = 0, ..., T \end{aligned}$$

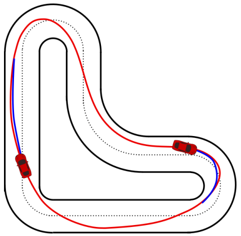
max progress rate + min regularizers

discretized curvilinear bicycle model

track, friction ellipse and action constraints

MPC Real-time Controller

- Progress maximizing path following MPC (MPCC or MPC-Curv)
- Follow the reference line while maximizing the progress
  - NMPC with 4th order Runge Kutta discretized model
  - Prediction horizon of ~2s



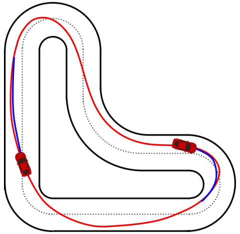
$$\begin{aligned} \min_{\mathbf{X},\mathbf{U}} \quad & \sum_{t=0}^{T} j_{\text{MPC}}(\mathbf{x}_t, \mathbf{u}_t) \\ \text{s.t.} \quad & \mathbf{x}_0 = \hat{\mathbf{x}} \\ & \mathbf{x}_{t+1} = f_t^d(\mathbf{x}_t, \mathbf{u}_t) \\ & \mathbf{x}_t \in \mathcal{X}_{\text{Track}}, \quad \mathbf{x}_t \in \mathcal{X}_{\text{FE}} \\ & \mathbf{a}_t \in \mathcal{A}, \quad t = 0, \ldots, T \end{aligned}$$

$$j_{\text{MPC}}(\mathbf{x}, \mathbf{u}) = \underbrace{\boxed{-\text{progress rate}}}_{\text{drive as fast as possible}} + \underbrace{\boxed{\text{regularizers}}}_{\text{follow path, input + slip angle reg}}$$

- max progress rate + min regularizers
- discretized dynamic bicycle model
- track, friction ellipse and action constraints

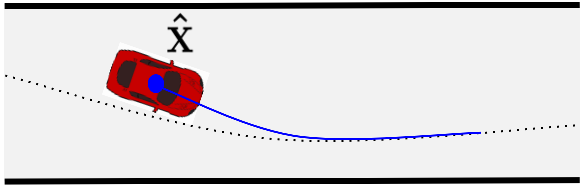
MPC Real-time Controller

- Progress maximizing path following MPC (MPCC or MPC-Curv)
- Follow the reference line while maximizing the progress
  - NMPC with 4th order Runge Kutta discretized model
  - Prediction horizon of ~2s



$$\min_{\mathbf{X},\mathbf{U}} \sum_{t=0}^{T} j_{\mathrm{MPC}}(\mathbf{x}_t, \mathbf{u}_t)$$

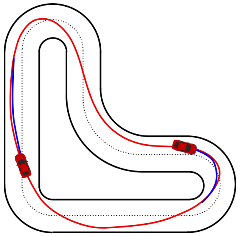
$$\begin{aligned} \text{s.t.} \quad & \mathbf{x}_0 = \hat{\mathbf{x}} \\ & \mathbf{x}_{t+1} = f_t^d(\mathbf{x}_t, \mathbf{u}_t) \\ & \mathbf{x}_t \in \mathcal{X}_{\mathrm{Track}}, \quad \mathbf{x}_t \in \mathcal{X}_{\mathrm{FE}} \\ & \mathbf{a}_t \in \mathcal{A}, \quad t = 0, \dots, T \end{aligned}$$



- max progress rate + min regularizers
- discretized dynamic bicycle model
- track, friction ellipse and action constraints

MPC Real-time Controller

- Progress maximizing path following MPC (MPCC or MPC-Curv)
- Follow the reference line while maximizing the progress
  - NMPC with 4th order Runge Kutta discretized model
  - Prediction horizon of ~2s



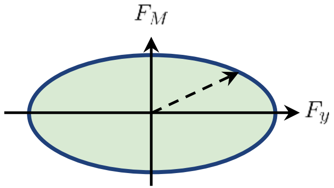
$$\min_{\mathbf{X},\mathbf{U}} \sum_{t=0}^{T} j_{\text{MPC}}(\mathbf{x}_t, \mathbf{u}_t)$$

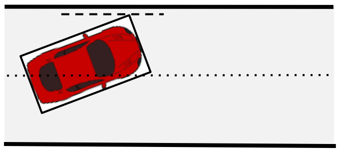
$$\text{s.t.} \quad \mathbf{x}_0 = \hat{\mathbf{x}}$$

$$\mathbf{x}_{t+1} = f_t^d(\mathbf{x}_t, \mathbf{u}_t)$$

$$\mathbf{x}_t \in \mathcal{X}_{\text{Track}}, \quad \mathbf{x}_t \in \mathcal{X}_{\text{FE}}$$

$$\mathbf{a}_t \in \mathcal{A}, \quad t = 0, \ldots, T$$





max progress rate + min regularizers

discretized dynamic bicycle model

track, friction ellipse and action constraints

Dynamic Bicyle Model - Curvilinear

$$\dot{s} = \frac{v_x \cos\mu - v_y \sin\mu}{1 - n\kappa(s)}$$

$$\dot{n} = v_x \sin\mu + v_y \cos\mu$$

$$\dot{\mu} = r - \kappa(s)\dot{s}$$

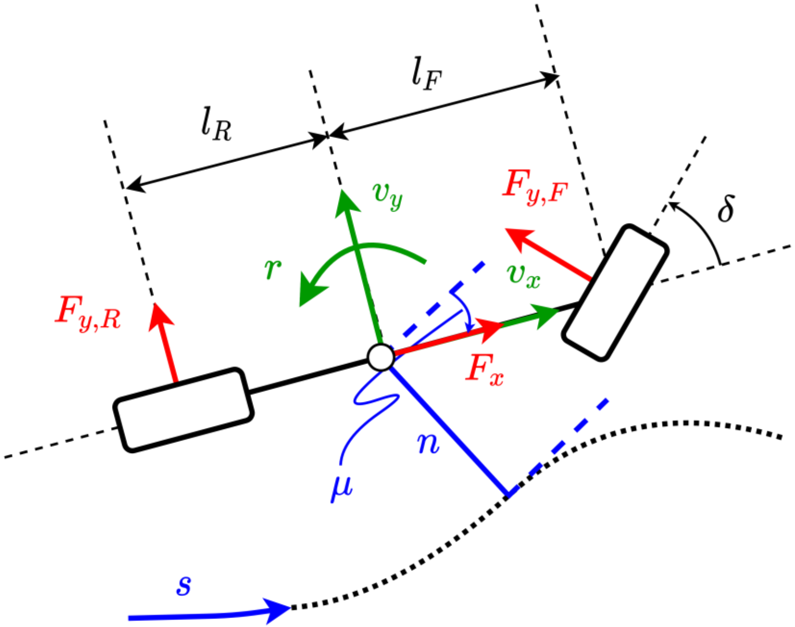
$$\dot{v}_x = \frac{1}{m}(F_x - F_{y,F}\sin\delta + m v_y r)$$

$$\dot{v}_y = \frac{1}{m}(F_{y,R} + F_{y,F}\cos\delta - m v_x r)$$

$$\dot{r} = \frac{1}{I_z}(F_{y,F} l_F \cos\delta - F_{y,R} l_R + M_{tv})$$

$$\dot{\delta} = \Delta\delta$$

$$\dot{T} = \Delta T$$



Dynamic Bicyle Model - Curvilinear

$$\dot{s} = \frac{v_x \cos\mu - v_y \sin\mu}{1 - n\kappa(s)}$$

$$\dot{n} = v_x \sin\mu + v_y \cos\mu$$

$$\dot{\mu} = r - \kappa(s)\dot{s}$$

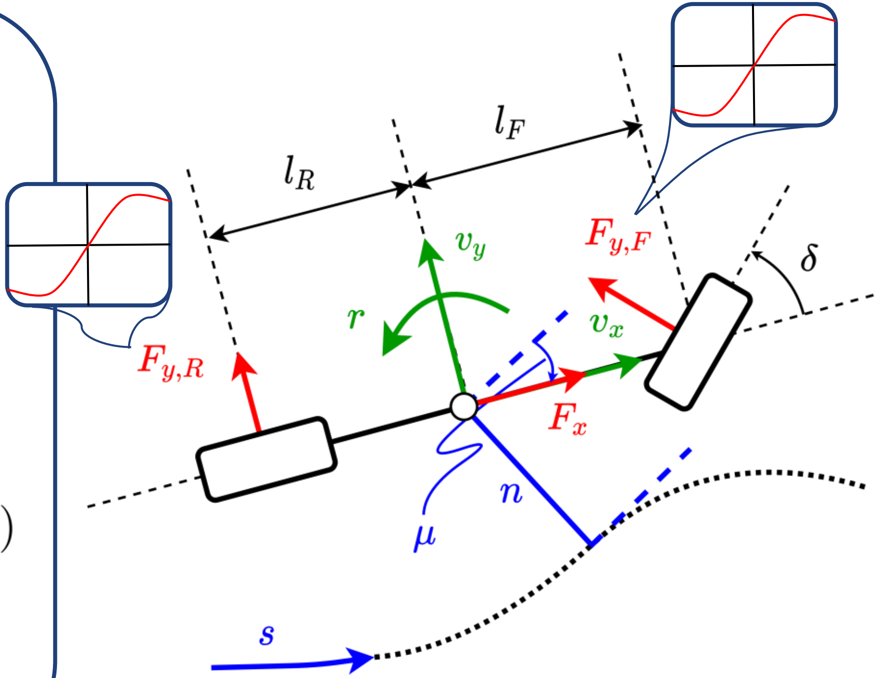
$$\dot{v}_x = \frac{1}{m}(F_x - F_{y,F}\sin\delta + m v_y r)$$

$$\dot{v}_y = \frac{1}{m}(F_{y,R} + F_{y,F}\cos\delta - m v_x r)$$

$$\dot{r} = \frac{1}{I_z}(F_{y,F} l_F \cos\delta - F_{y,R} l_R + M_{tv})$$

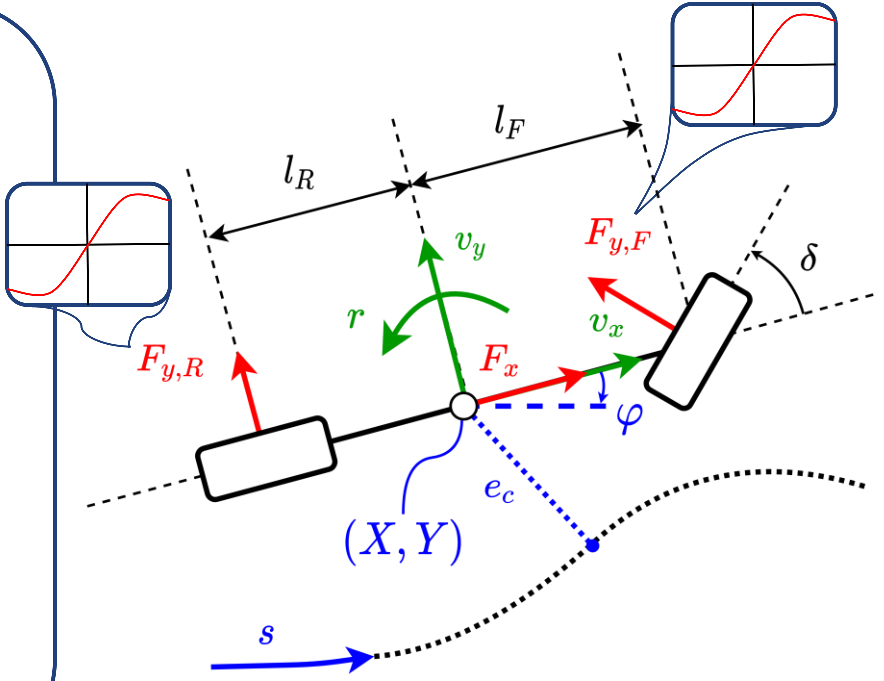
$$\dot{\delta} = \Delta\delta$$

$$\dot{T} = \Delta T$$



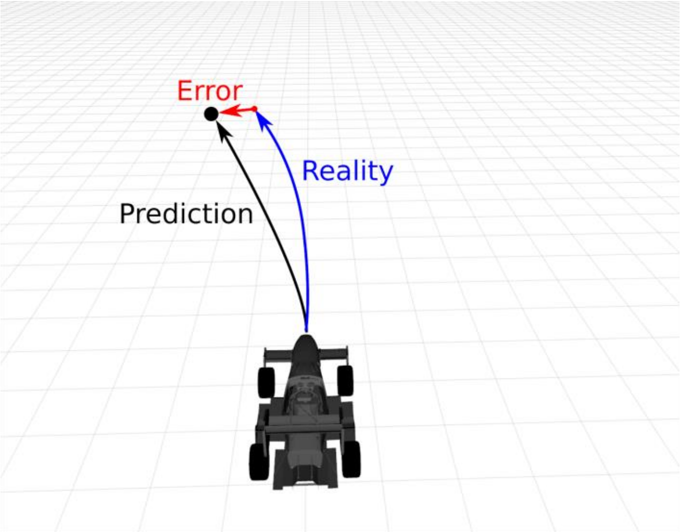
Dynamic Bicyle Model - Contouring

$$\dot{X} = v_x \cos\varphi - v_y \sin\varphi$$
$$\dot{Y} = v_x \sin\varphi + v_y \cos\varphi$$
$$\dot{\varphi} = r$$
$$\dot{v}_x = \frac{1}{m}(F_x - F_{y,F}\sin\delta + m v_y r)$$
$$\dot{v}_y = \frac{1}{m}(F_{y,R} + F_{y,F}\cos\delta - m v_x r)$$
$$\dot{r} = \frac{1}{I_z}(F_{y,F} l_F \cos\delta - F_{y,R} l_R + M_{tv})$$
$$\dot{s} = v_s$$
$$\dot{\delta} = \Delta\delta$$
$$\dot{T} = \Delta T$$
$$\dot{v}_s = \Delta v_s$$



Model Mismatch

- Error between predicted next state and actual realization
- Easy to compute based on historical data
- How can we reduce the model mismatch



Learn a correction model using Gaussian Processes
"Learning-based Model Predictive Control for Autonomous Racing", Kabzan et. Al.

Make the car behave like the prediction model
"A Holistic Motion Planning and Control Solution to Challenge a Professional Racecar Driver", Srinivasan et. Al.

Model Correction

- Improve model using **sparse** Gaussian process

$$\mathbf{x}_{t+1} = \underbrace{f_t^d(\mathbf{x}_t, \mathbf{u}_t)}_{\text{physical model}} + \underbrace{\mathbf{B}_d}_{\text{selection matrix}} \underbrace{(d(\mathbf{z}_t)}_{\text{GP}} + \underbrace{\mathbf{w}_t}_{\text{noise}})$$

- Five features for error prediction

$$z = [v_x, v_y, r, \delta, T] \qquad z = [v_x, v_y, r, \delta + 0.5T_s\Delta\delta, T + 0.5T_s\Delta T]$$

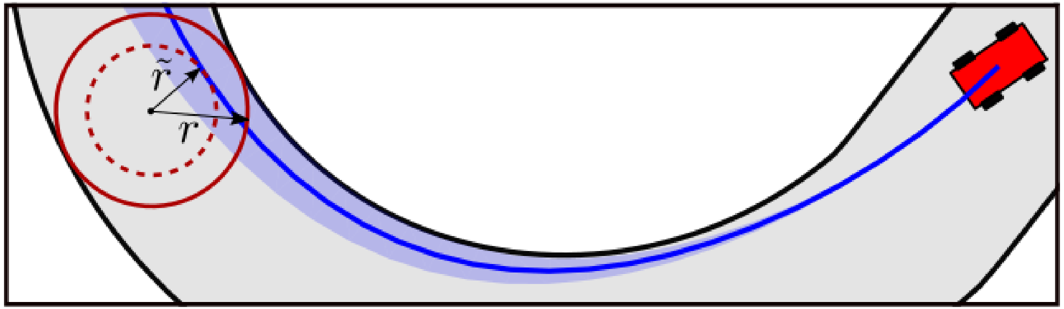
- GP estimates errors in the three velocities v_x, v_y, and r
- Set selection matrix correct only velocity states $\mathbf{B}_d = [\mathbf{0}, \mathbf{I}, \mathbf{0}]^T$

Gaussian Process MPC

- How to use the sparse GP corrected model?
- Use mean prediction in the model constraint (sparse GP for tractability)

$$\mathbf{x}_{t+1} = f_t^d(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{B}_d \mathbb{E}[d(\mathbf{z}_t)]$$

- Use variance prediction and linearized dynamics for constraint tightening

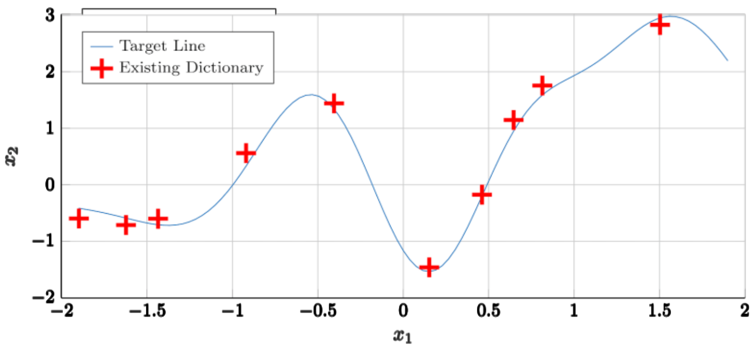


Gaussian Process Online Learning

- GP is a non-parametric technique:
  Explicitly requires data points to make prediction
- Data selection to limit computation effort
- Variance as a measure of informativeness

$$\gamma_i = k^a_{\mathbf{z}_i \mathbf{z}_i} - \mathbf{k}^a_{\mathbf{z}_i \mathbf{Z}_{\setminus i}} (\mathbf{K}^a_{\mathbf{Z}_{\setminus i} \mathbf{Z}_{\setminus i}} + \mathbf{I}\lambda)^{-1} \mathbf{k}^a_{\mathbf{Z}_{\setminus i} \mathbf{z}_i}$$

- Adaptive outlier removal
  - Prevents sudden changes of GP
  - Improve robustness of adaptive controller



1st Lap: Data collection with **nominal controller**

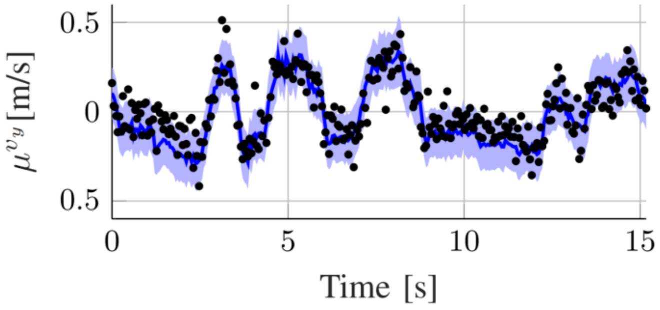
Lap:	
1	20.2
2	20.3
3	19.2
4	18.8
5	18.5
6	18.4
7	17.9
8	
9	

The learning-based controller is activated after the 2^{nd} lap

Successive improvement of lap times is observed

Learning MPC - Results

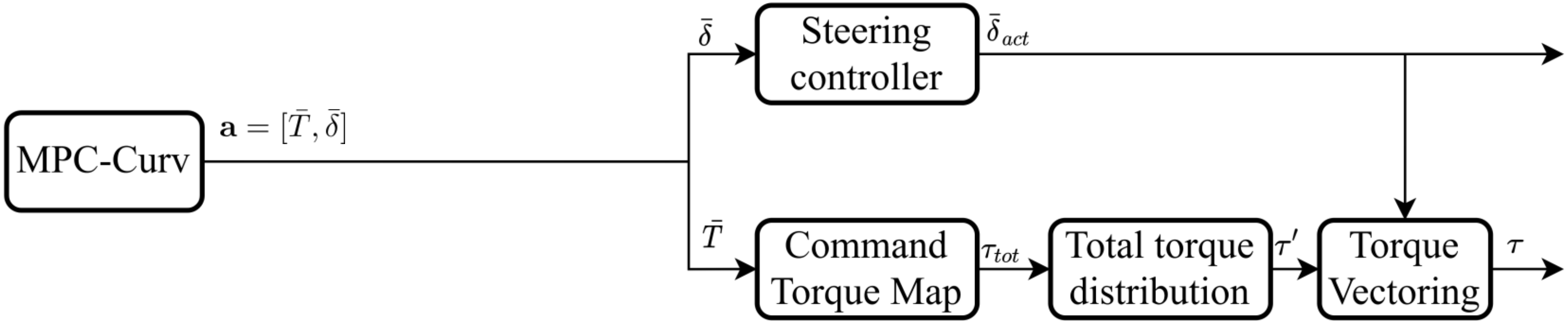
- Model learning reduces lap time by 12%
- Error with nominal model increases with increasing performance/speed
- Error with corrected model remains approximately constant



Lap	Time [s]	$\overline{\lVert \mathbf{e}_{\text{nom}} \rVert}$	$\overline{\lVert \mathbf{e}_{\text{GP}} \rVert}$	$\lVert a \rVert_{\max}$ [g]
1	20.19	0.16	-	1.52
2	20.29	0.18	-	1.49
3	19.16	0.19	0.15	2.05
4	18.80	0.23	0.15	2.01
5	18.47	0.23	0.16	2.02
6	18.44	0.24	0.15	2.00
7	17.98	0.23	0.15	1.82
8	18.47	0.24	0.17	2.15
9	18.25	0.24	0.16	2.10

Model Matching using Low-Level Control

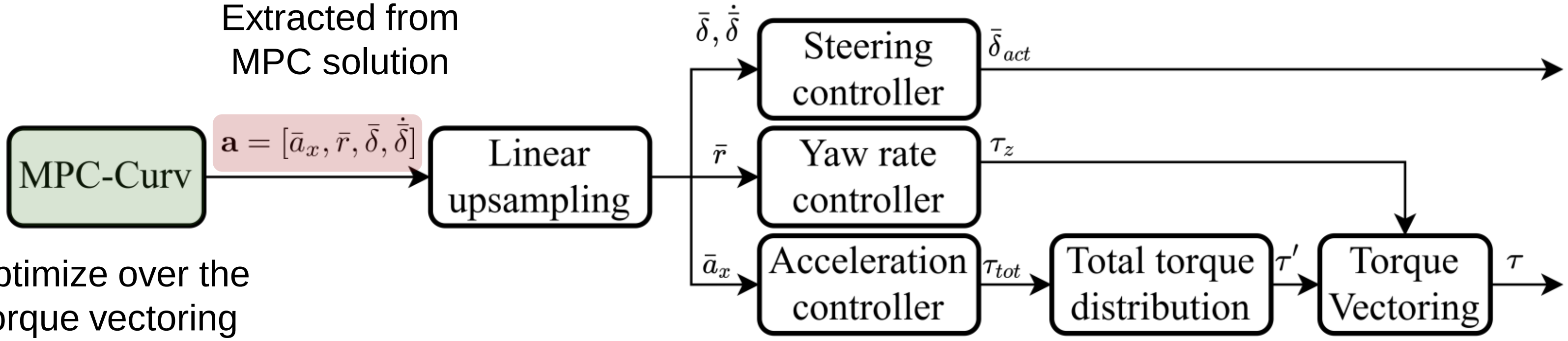
- Make the MPC model more accurate using model learning
- Make the real car behave more like the MPC model using low-level control
- Use the 4 independent wheel hub motors to reduce model mismatch



Model Matching using Low-Level Control

- Make the MPC model more accurate using ML
- Make the real car behave like the MPC model using low-level control
- Use the 4 independent wheel hub motors to reduce model mismatch

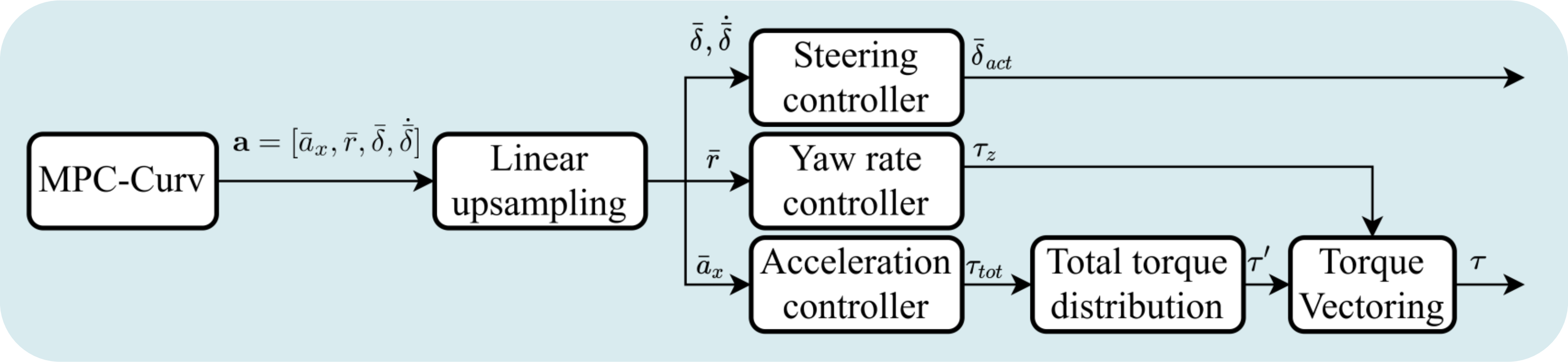
Extracted from MPC solution



Optimize over the torque vectoring

Model Matching using Low-Level Control

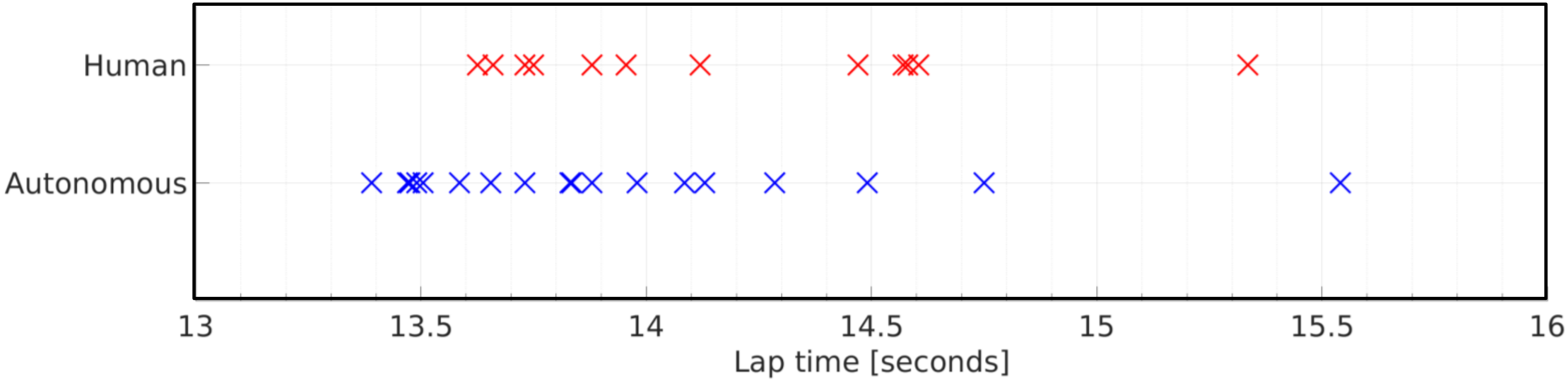
- Make the MPC model more accurate using ML
- Make the real car behave like the MPC model using low-level control
- Use the 4 independent wheel hub motors to reduce model mismatch
- Holistic design from high level trajectory planning to low level control




Garage
Bucher AG

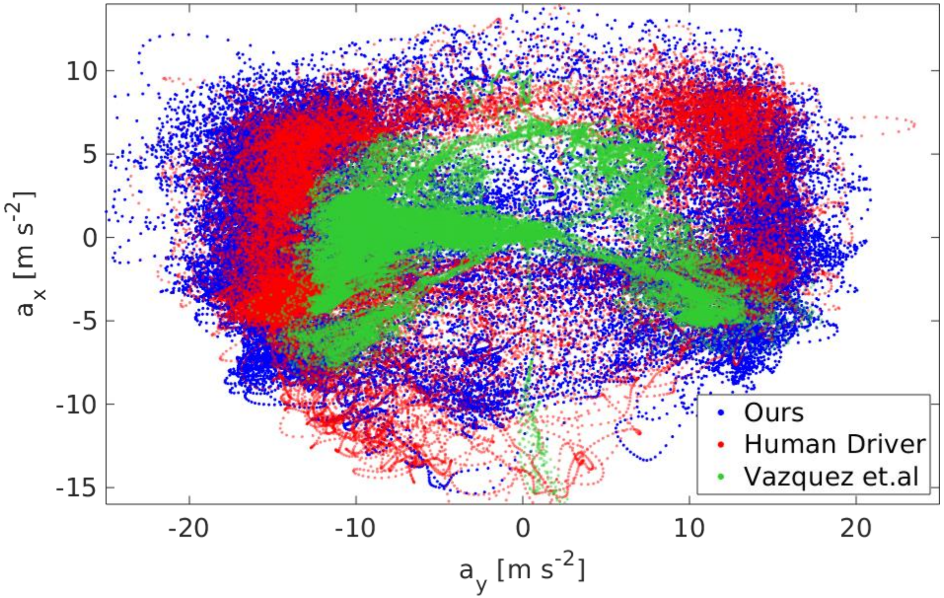
Comparison to Human Driver

- Not 100% fair, top speed limit (17m/s) for MPC, but 70kg lighter car
- Experienced Formula Student driver



Comparison to Human Driver

- Not 100% fair, top speed limit (17m/s) for MPC, but 70kg lighter car
- Experienced Formula Student driver



Conclusion

- Final Remarks
  - Highlighted importance of model mismatch for autonomous racing
  - Proposed two dual approaches to solve the model mismatch problem
  - Both methods achieve lateral accelerations of 2g
  - Matching the performance of a top human driver
- Future Work
  - Can we combine the two methods to get the best of both worlds

Thanks

- Collaborators
  - Juraj Kabzan
  - José L Vázquez
  - Marius Brühlmeier
  - Sirish Srinivasan
  - Sebastian Nicolas Giles
  - Dr. Lukas Hewing
  - Dr. Alisa Rupenyan
  - Prof. Melanie Zeilinger
  - Prof. John Lygeros
- AMZ
  - Team 2018
  - Team 2019
  - Team 2020