CS686: Motion Planning and Applications
Paper Presentation - II

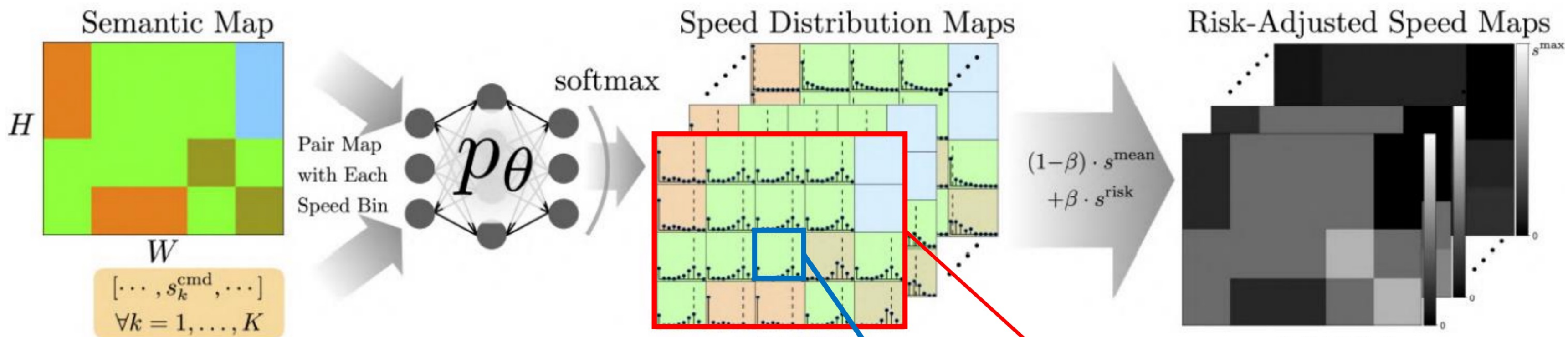
Learning Terrain-Aware Kino-dynamic Model for Autonomous Off-Road Rally Driving With Model Predictive Path Integral Control

RA-L 2023

Jeil Jeong
(정제일)

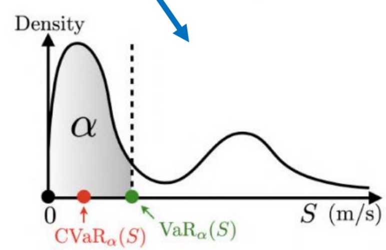
Review

Risk-Aware Off-Road Navigation via Learned Speed Distribution Map



- **Speed Distribution Maps**
  - **Divide v_{min} and v_{max} by K**

Speed Distribution Map for v_{max}



s^{mean}: *expected value*

s^{risk}: *CVaR value*

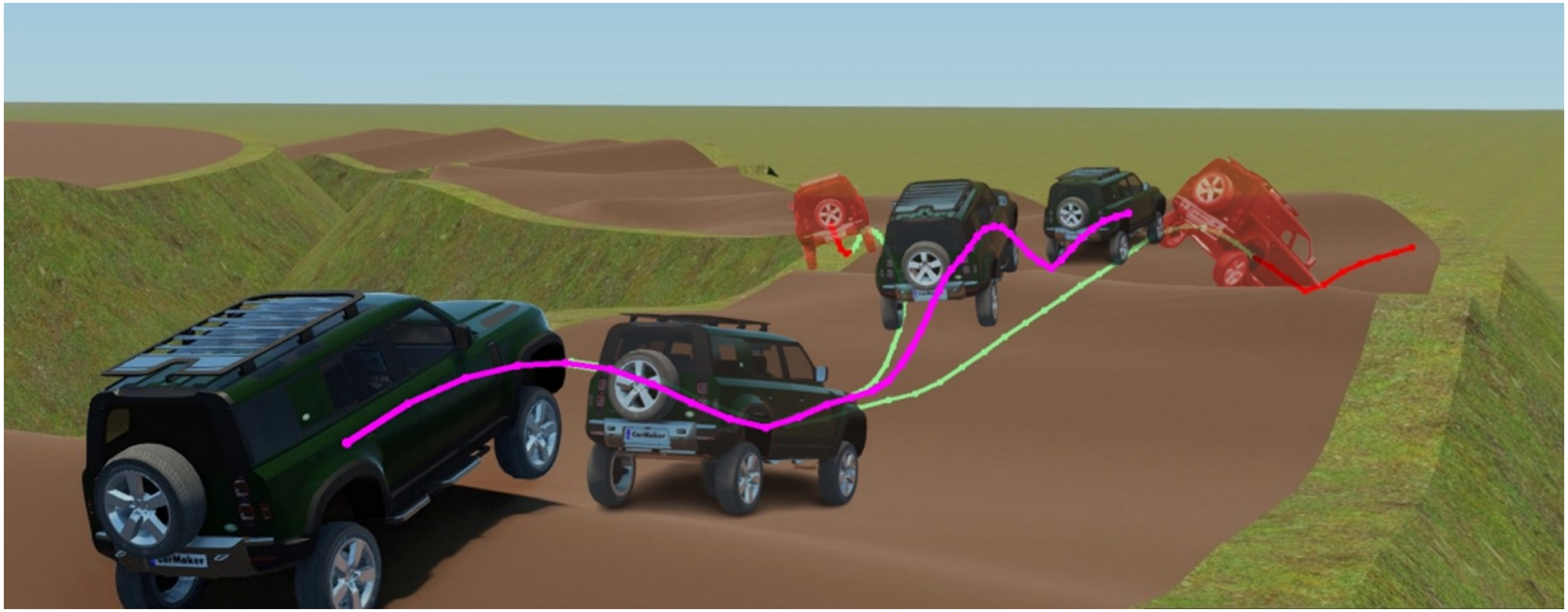
Table of Contents

1. Motivations

Motivations

Off-road environment and High-speed autonomous driving

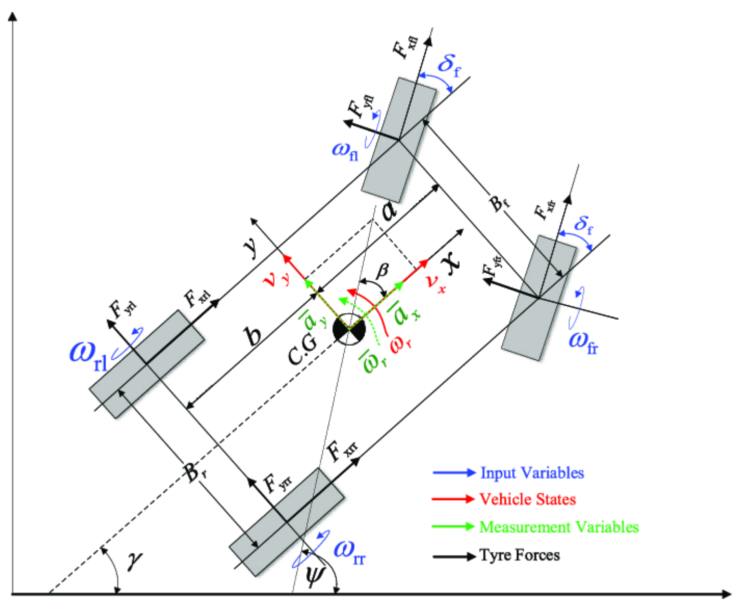
It requires accurate modeling the interaction between the vehicle and the terrain



Motivations

Off-road environment and High-speed autonomous driving

It has been addressed by analytical modeling with simplification[1]



Simplification

- Planar model (3-DoF)
- Constant vehicle-terrain contact
- Single point contact

Inaccurate dynamics modeling

[1] Liu, Wei, Hongwen He, and Fengchun Sun. "Vehicle state estimation based on minimum model error criterion combining with extended Kalman filter." *Journal of the Franklin Institute* 353.4 (2016): 834-856.

Motivations

Off-road environment and High-speed autonomous driving

Data-driven methods, has been solely considered proprioceptive information

Need an ability to encode environmental context using exteroceptive information(외부수용 정보)

Motivations

This problem has been approached by

- **Analytical modeling with simplification**
- **Solely relying on proprioceptive information**

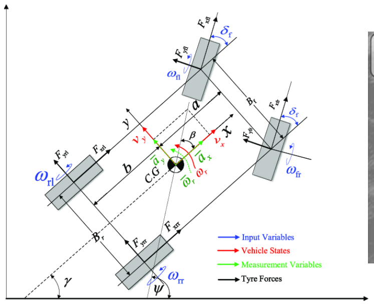
The main idea of this paper is that using

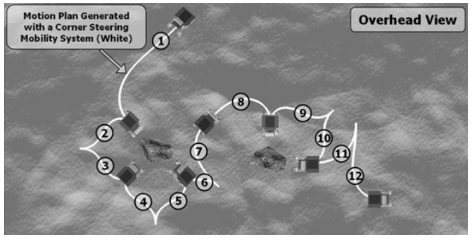
- **Neural Network based dynamics modeling**
- **Not only proprioceptive but also exteroceptive**

2. Related Work

Related Work

➢ Model-Based method

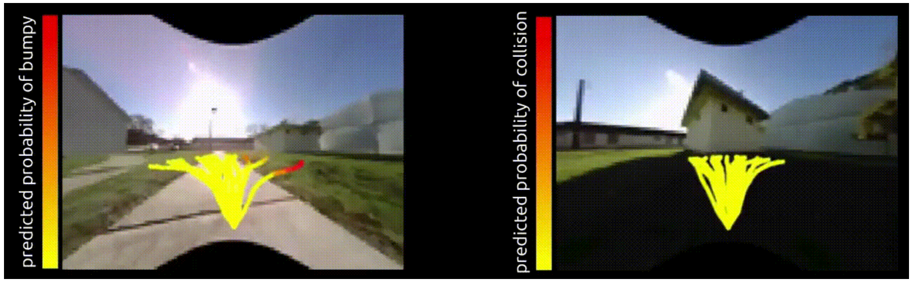




- Simple and Easy to interpret
- Accurate in planar environment
- Cannot represent full – dynamics (Inaccurate dynamics modeling)
- Simplification leads unstable control tracking performance

Howard, Thomas M., and Alonzo Kelly. "Optimal rough terrain trajectory generation for wheeled mobile robots." *The International Journal of Robotics Research* 26.2 (2007): 141-166.

➢ Learning-Based method



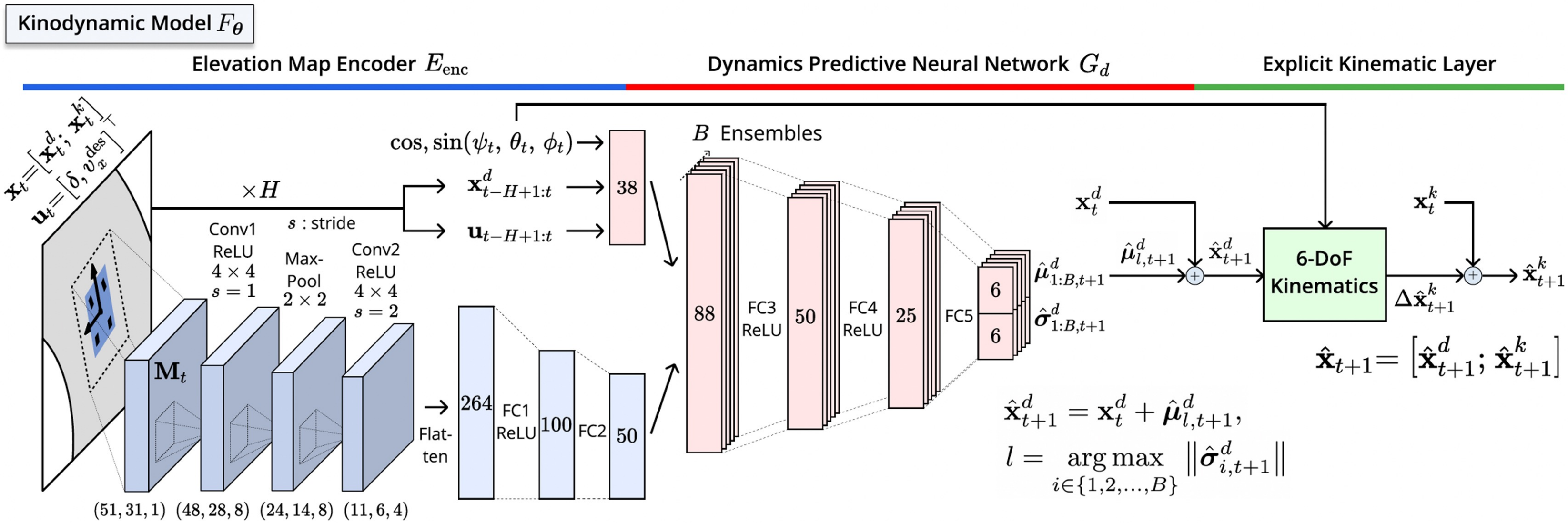
- Can represent full-dynamics based on vehicle state's history data
- Can integrate exteroceptive information
- Hard to analyze networks process
- Sim - Real Gap exists

Kahn, Gregory, Pieter Abbeel, and Sergey Levine. "Badgr: An autonomous self-supervised learning-based navigation system." *IEEE Robotics and Automation Letters* 6.2 (2021): 1312-1319.

3. Method

Method

Terrain Aware Kino-dynamic Model



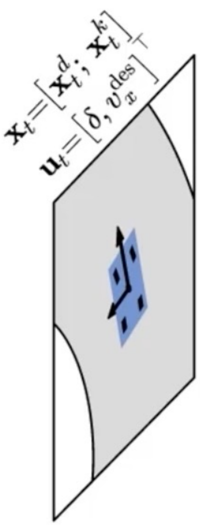
1. **It consists of 3 steps**
2. **This model predicts the change in the vehicle's state induced by control inputs and contact interactions**

$$\hat{X}_{t+1} = F_\theta(X_t, u_t, M_t)$$

Method

Terrain Aware Kino-dynamic Model

$$\hat{X}_{t+1} = F_\theta(X_t, u_t, M_t)$$

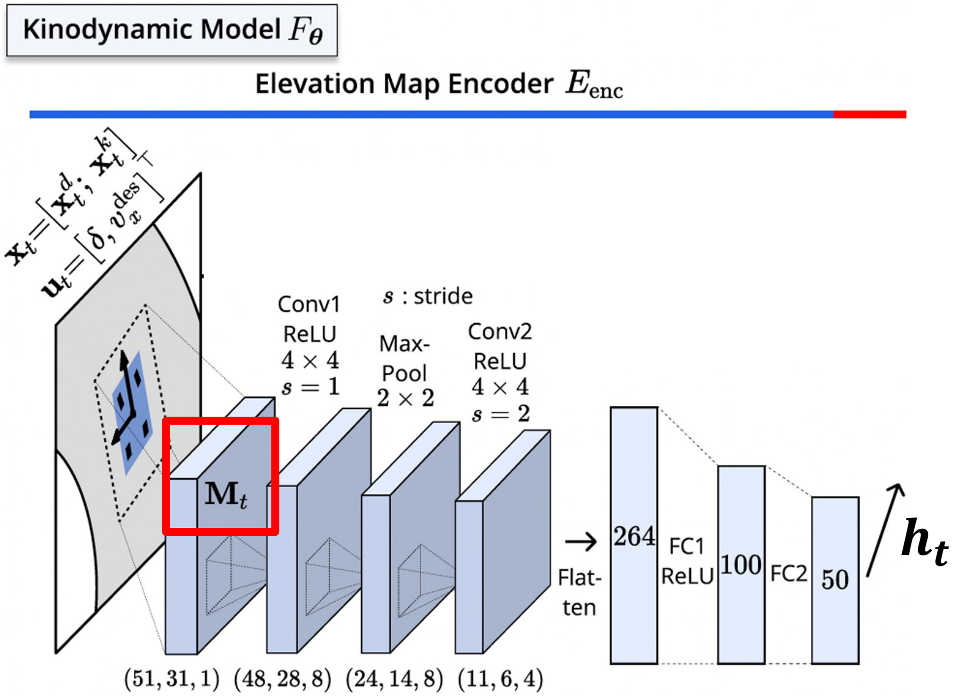


- **6-DoF vehicle's proprioceptive state X_t**
  - X_t^d : **Linear and Angular Velocity**
  - X_t^k : **Position and Orientation**
- **Control Input u_t**
  - δ : **Steering angle**
  - v_x^{des} : **Desired longitudinal speed**
- **Local Elevation Map M_t**

Method

(1) Elevation Map Encoder

It processes the local elevation map(M_t) and outputs the latent terrain feature vector (h_t)



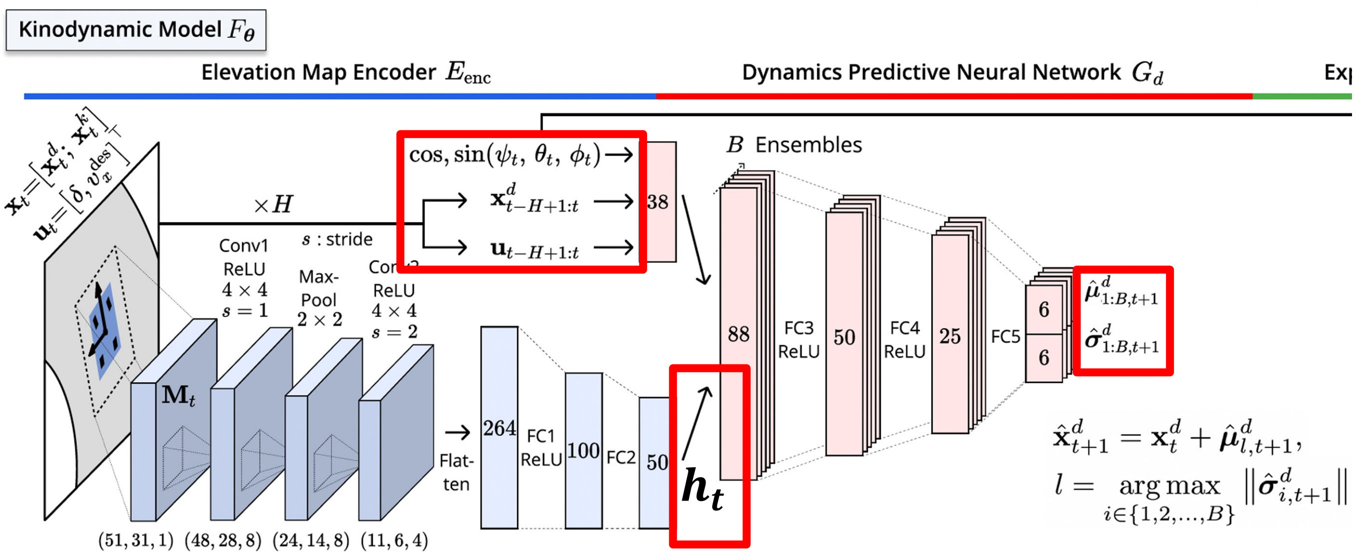
$$\mathbf{h}_t = E_{\mathrm{enc}}(\mathbf{M}_t)$$

Method

(2) Dynamics Predictive Neural Network

It predicts the change in velocities($\hat{\mu}^{d}_{1:B,t+1}$) using proprioceptive information($x^{d}_{t-H+1:t}, u_{t-H+1:t}$) and a terrain feature vector(h_t)

$$\hat{\boldsymbol{\mu}}^{d}_{i,t+1},\ \hat{\boldsymbol{\sigma}}^{d}_{i,t+1} = G_d\left(\mathbf{x}^{d}_{t-H+1:t},\ \mathbf{u}_{t-H+1:t},\ h_t,\ \mathrm{c}_{\psi},\ \mathrm{s}_{\psi},\ \mathrm{c}_{\theta},\ \mathrm{s}_{\theta},\ \mathrm{c}_{\phi},\ \mathrm{s}_{\phi}\right)$$

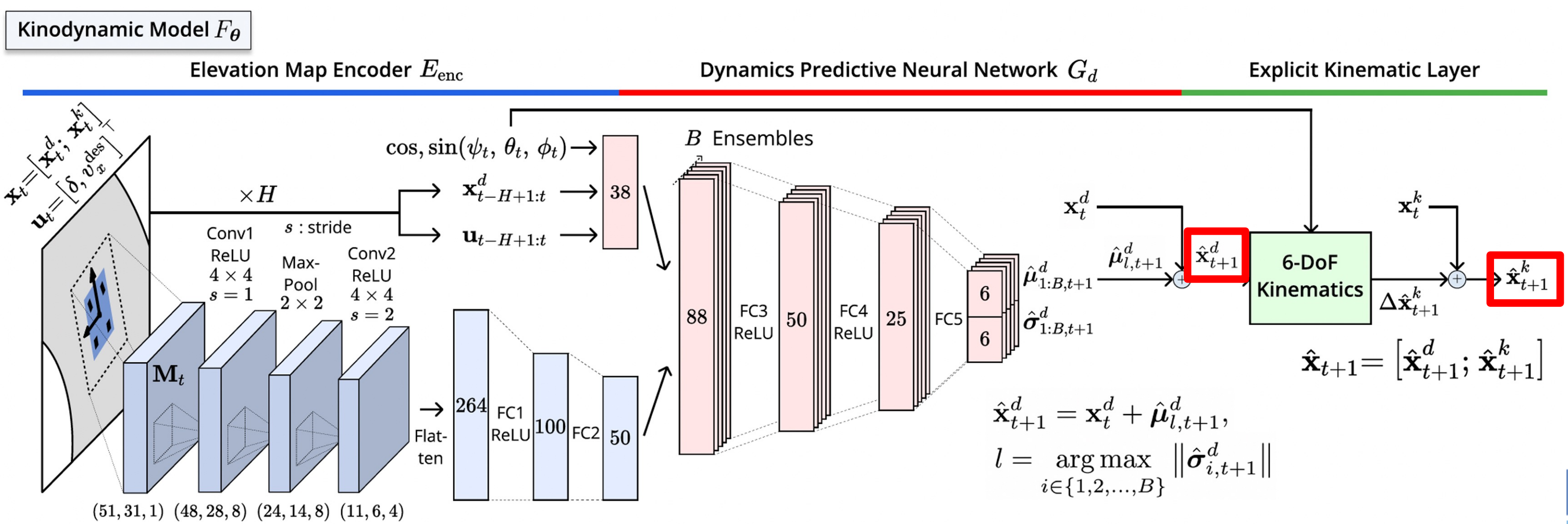


Method

(3) Explicit Kinematic Layer

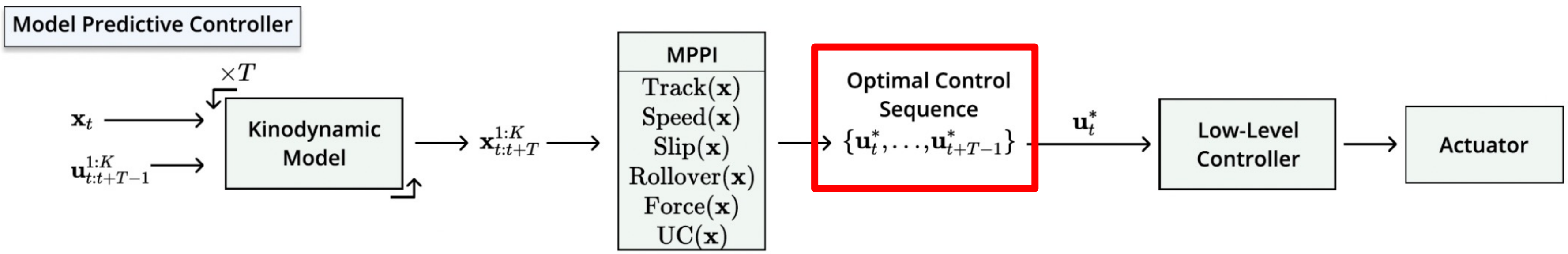
It analytically calculates changes in position and orientation($\widehat{X}^k_{t+1}$) using predicted change in velocity($\widehat{X}^d_{t+1}$)

$$\hat{\mathbf{x}}^k_{t+1} = \mathbf{x}^k_t + \begin{bmatrix} \Delta\hat{\mathbf{p}}_s \\ \Delta\hat{\mathbf{e}}_s \end{bmatrix}_{t+1}$$



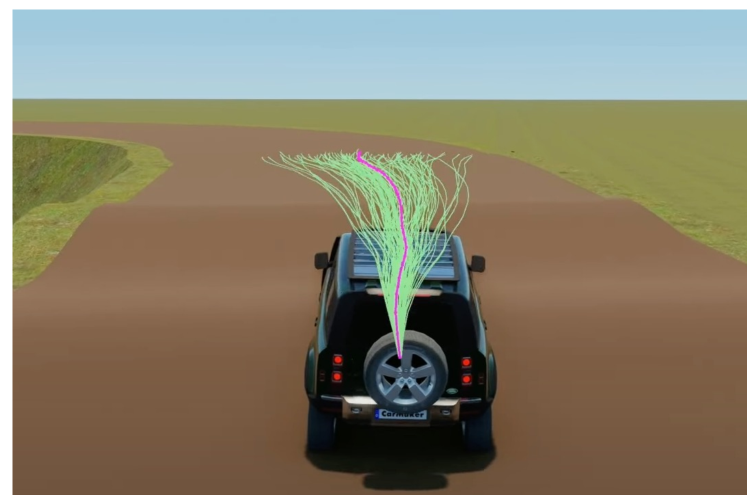
Method

Model Predictive Path Integral (MPPI)



Using trained kino-dynamic model, it finds optimal control sequence that minimize MPPI cost ($c(x)$)

$$c(\mathbf{x}) = w_1\mathrm{Track}(\mathbf{x}) + w_2\mathrm{Speed}(\mathbf{x}) + w_3\mathrm{Slip}(\mathbf{x}) + w_4\mathrm{Rollover}(\mathbf{x}) + w_5\mathrm{Force}(\mathbf{x}) + w_6\mathrm{UC}(\mathbf{x})$$



4. Experimental Results

Experimental Results

Task : Autonomous Rally Driving Task

“2D baseline model : A 3-DoF plane model”

Experimental Results

Task : Autonomous Rally Driving Task

TABLE II

EXPERIMENTAL RESULTS ON THE RACE TRACK. TO ASSESS THE DIFFICULTY OF THE DRIVING TASK, WE CONDUCTED EXPERIMENTS USING A BASELINE MODEL AT BOTH $v_{\text{REF}} = 30$ KM/H AND $v_{\text{REF}} = 40$ KM/H. WE SET K AND T AS 2000 AND 20, RESPECTIVELY, WHICH ENABLES OUR ALGORITHM TO OPERATE AT 10HZ ON THE NVIDIA RTX 3090 GPU WITH CUDA AND PYTORCH. WE DISPLAYED THE MEAN AND STANDARD DEVIATION FOR LAP TIME AND F_z^{PEAK}.

#	Model	v_{ref} (km/h)	Off-Road Cost Functions			Lap Time (s)	# of Failure	$\lvert v\rvert_{\text{avg}}$ (km/h)	$\lvert v\rvert_{\text{max}}$ (km/h)	$\lvert\phi\rvert_{\text{max}}$ (deg)	$\lvert\theta\rvert_{\text{max}}$ (deg)	F_z^{peak} (kN)
			Rollover($\mathbf{x}$)	Force($\mathbf{x}$)	UC($\mathbf{x}$)							
1	*2D*	30	✗	✗	✗	127.62 ± 2.21	5	28.79	39.16	79.21	31.53	50.11 ± 20.17
2	*2D*	40	✗	✗	✗	116.59 ± 4.76	19	32.56	42.22	79.41	36.90	53.23 ± 24.27
3	*Ours*	40	✗	✗	✗	117.45 ± 3.69	17	32.09	41.85	78.09	33.91	53.86 ± 21.96
4	*Ours*	40	✓	✗	✗	120.86 ± 8.55	23	32.49	41.29	79.25	38.74	53.70 ± 26.44
5	*Ours*	40	✓	✓	✗	136.32 ± 2.07	3	28.04	41.37	78.71	76.42	44.50 ± 19.97
6	*Ours*	40	✓	✓	✓	134.69 ± 1.64	0	28.11	40.48	28.50	28.87	43.17 ± 16.22

Thank you

Problems

1) How many steps are required in Terrain Aware Kino-dynamic model ?

a) 2
b) 3
c) 4

2) What controller is used to find optimal control sequence ?

a) Model Predictive Path Integral
b) Model Predictive Action Integral
c) Dynamic Window Approach