Training a Gaussian Process model

Part I. Creating speed profile.

1. Run pure-pursuit based on a default track file. (Using Shanghai_centerline.csv)

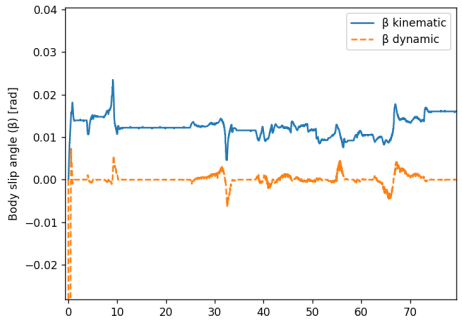
	A	B	C	D	E
1	# x_m	y_m	w_tr_right_m	w_tr_left_m	
2	0	0	1.1	1.1	
3	-0.447089597885696	-0.09416882273686	1.1	1.1	
4	-0.894165762181885	-0.188412946682923	1.1	1.1	
5	-1.34121798361788	-0.282776876314988	1.1	1.1	
6	-1.78823584430791	-0.377305024724873	1.1	1.1	
7	-2.23520892636627	-0.472041896389386	1.1	1.1	
8	-2.68212662913728	-0.567031904400359	1.1	1.1	
9	-3.12897871750513	-0.662319644619549	1.1	1.1	
10	-3.57575459081415	-0.757949438753833	1.1	1.1	
11	-4.0224462985726	-0.853954916469443	1.1	1.1	
12	-4.46905630817447	-0.950325202955837	1.1	1.1	
13	-4.91558946302277	-1.04703818305206	1.1	1.1	
14	-5.36205088067541	-1.14407146744227	1.1	1.1	
15	-5.80844531315045	-1.2414029409655	1.1	1.1	
16	-6.25477796939077	-1.33901048846081	1.1	1.1	
17	-6.7010536014144	-1.43687172061236	1.1	1.1	
18	-7.14727723539431	-1.53496461364415	1.1	1.1	

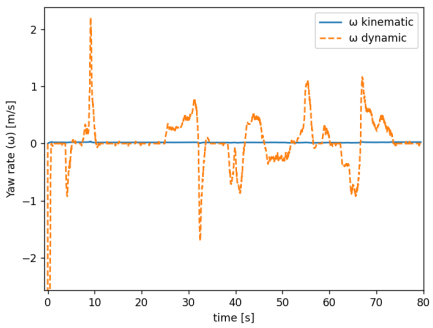
Vehicle speed = 7.0 m/s

2. Save data to `'/home/ning/dev_ws/src/f1tenth_ros2/f1tenth_ros2/data/f1tenth-DYN-{}-{}.npz'`

Dynamic model: states [x, y, δ, v, φ, ω, β],
inputs [acc, Δδ],
dstates d([x, y, δ, v, φ, ω, β])/dt,

4. Run **simulate_ekm_f110.py** to generate data of E-kinematic vehicle model
Only different in [ω, β]

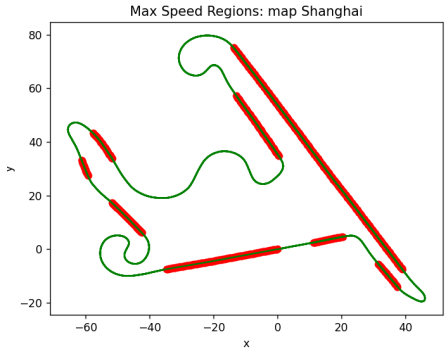




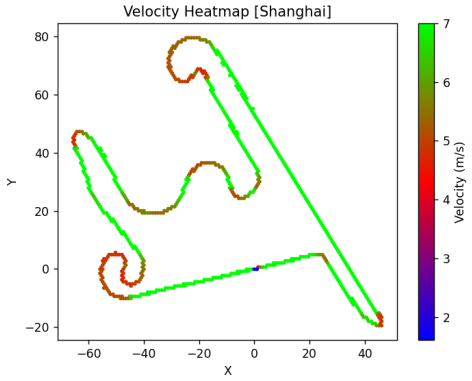
3. Data processing

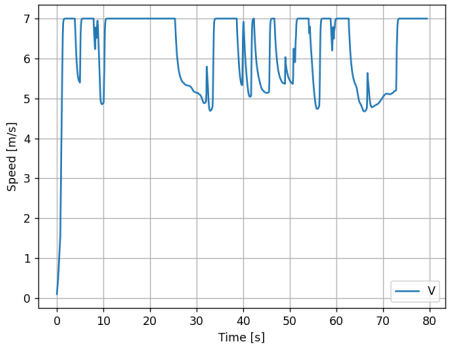
3.1 Downsample data (Dyn & Kin) [3583 → 1195]

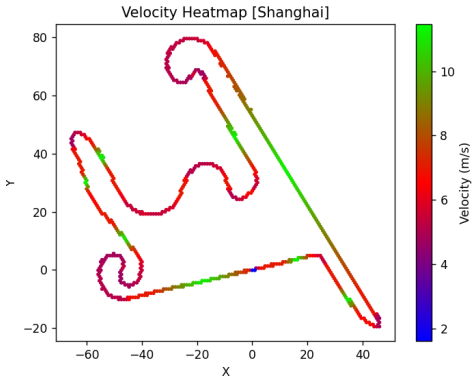
3.2 Find max speed regions on track

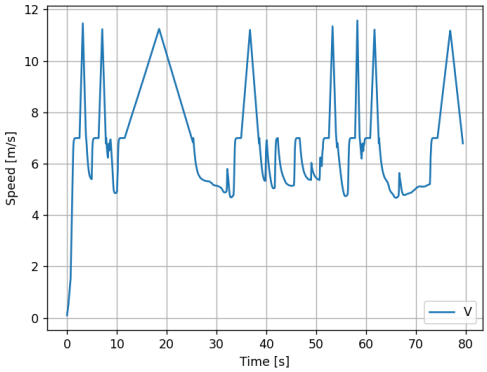


3.3. Change speed with constant acceleration ($x = x_0 + v_0*t + 1/2*a*t^2$)





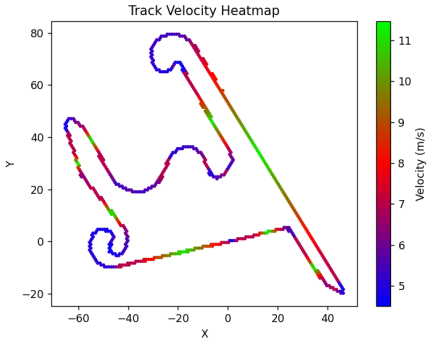


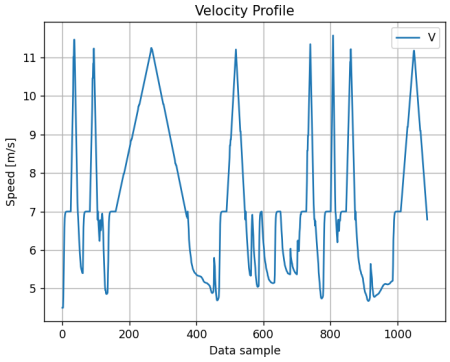


3.4. Assign speed to track file

At each timestep, we know vehicle position (x,y) and speed (v), locate positions in raceline using nearest waypoint method, and assign corresponding speed to that waypoint.

- We only change speed profile of the regions keep max constant velocity (7 m/s), the others are all same as before.
- We set the lower bound of ref speed to 4.5 m/s

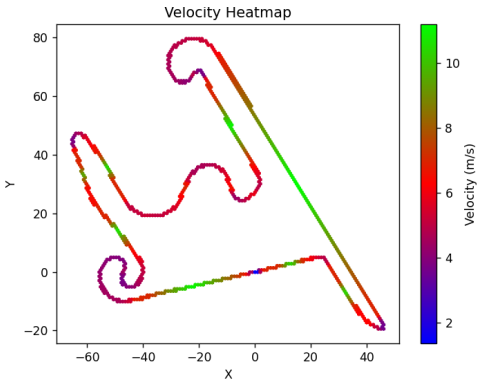


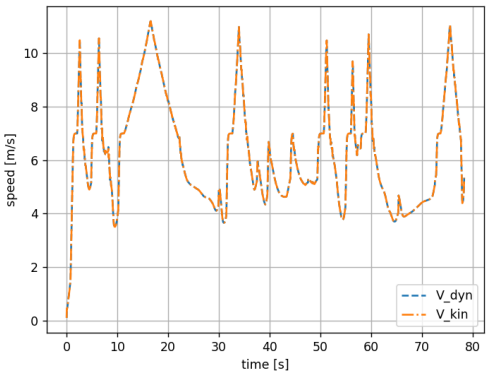


	A	B	C
1	X	Y	velocity
2	0	0	6.79108484572802
3	-0.447089597885696	-0.09416882273686	6.90653790943426
4	-0.894165762181885	-0.188412946682923	7.02199097314051
5	-1.34121798361788	-0.282776876314988	7.13744403684675
6	-1.78823584430791	-0.377305024724873	7.25289710055299
7	-2.23520892636627	-0.472041896389386	7.36835016425924
8	-2.68212662913728	-0.567031904400359	7.48380322796548
9	-3.12897871750513	-0.662319644619549	7.59925629167172
10	-3.57575459081415	-0.757949438753833	7.71470935537797

4. Run pure pursuit with new track file

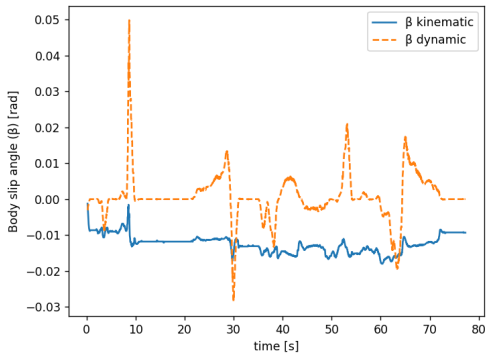
In **pp_controller.py** & **pure_puresuit.py**, change speed_profile to **True**

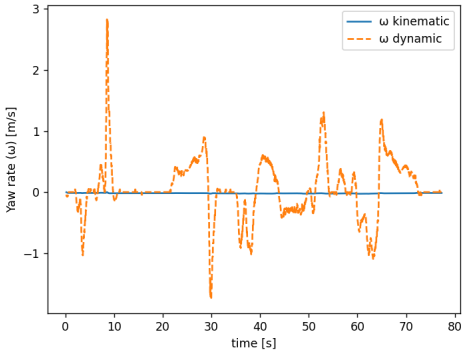


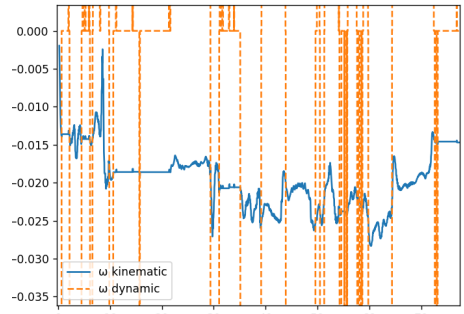


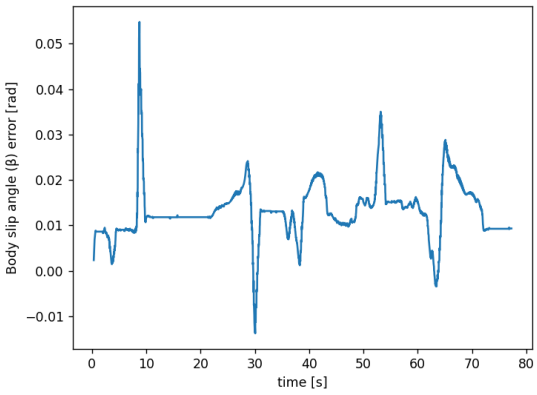
4.1 Generate ekm data

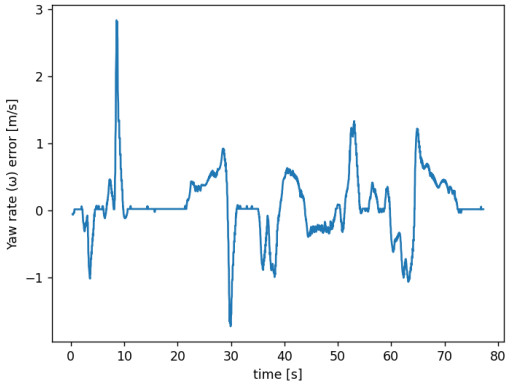
4.2 plots of vehicle states

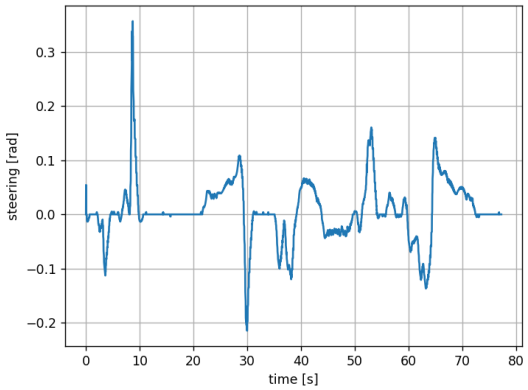


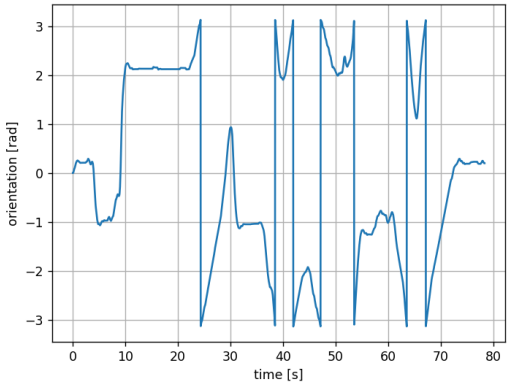


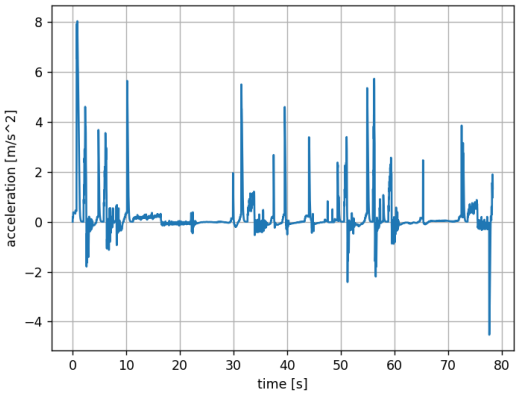


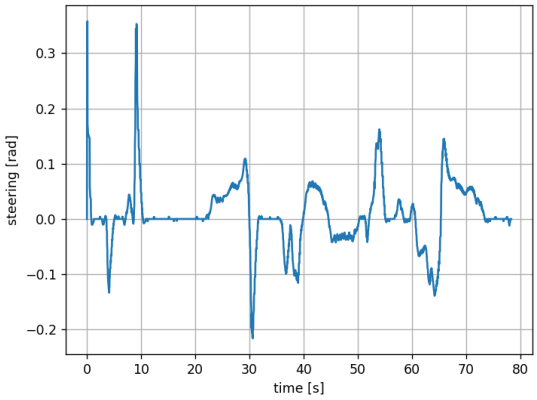




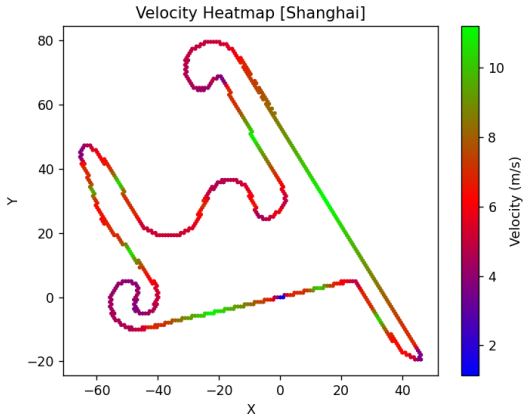


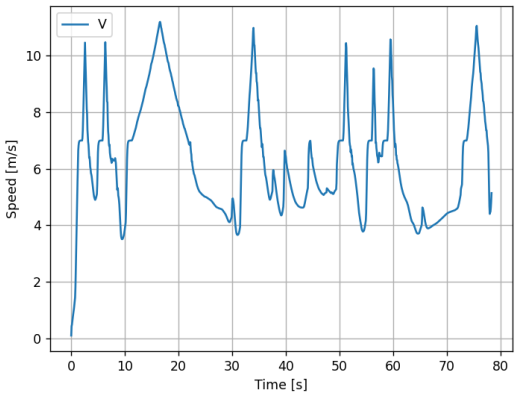






6. Downsample data (Dyn & Kin) [3552 → 1184]



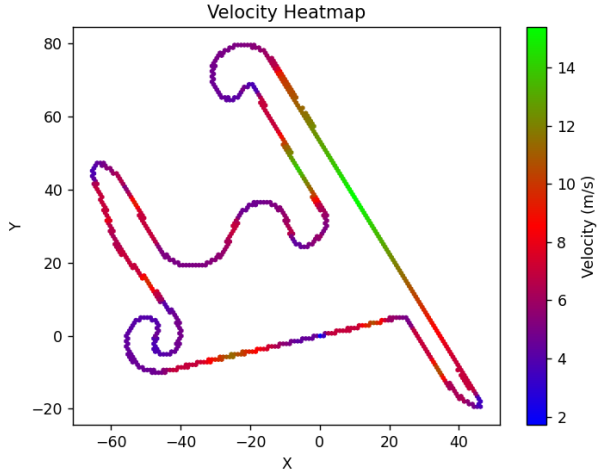


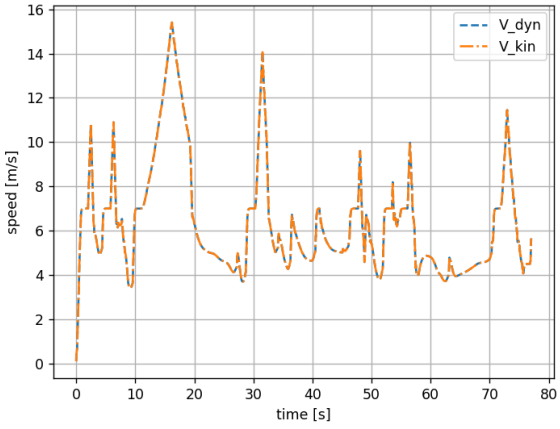
Part II. Training GP model

1. Creating testing data set (*setting test_mode = True*):

Similar to the previous process of creating speed profile, but:

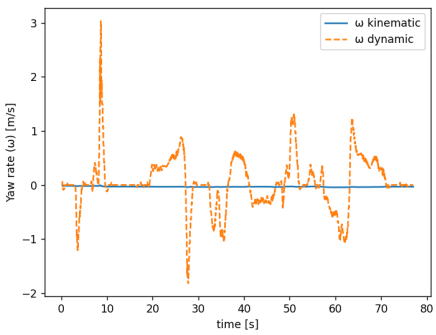
- Increase value of constant acceleration
- Increase distance of acceleration
- Increase value of deceleration

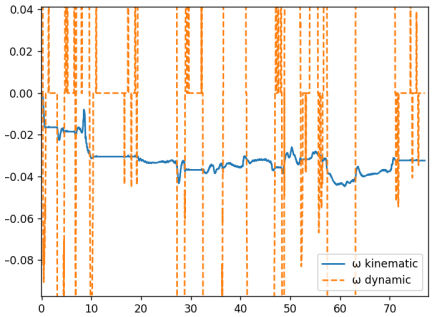


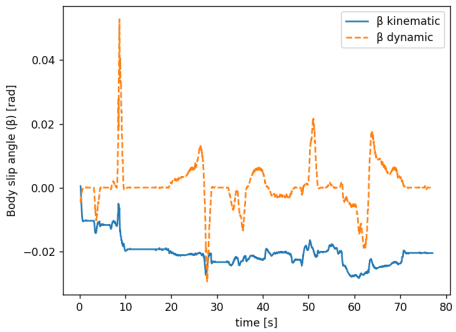


1.1 Plots of testing set

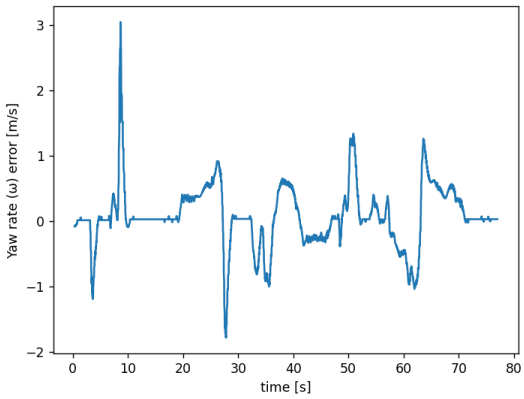
- ω_{dyn}, β_{dyn} vs ω_{kin}, β_{kin}

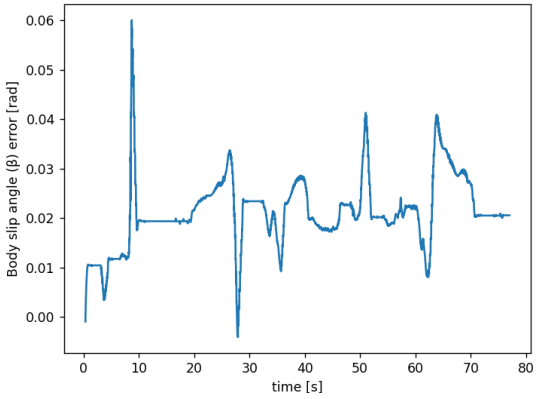






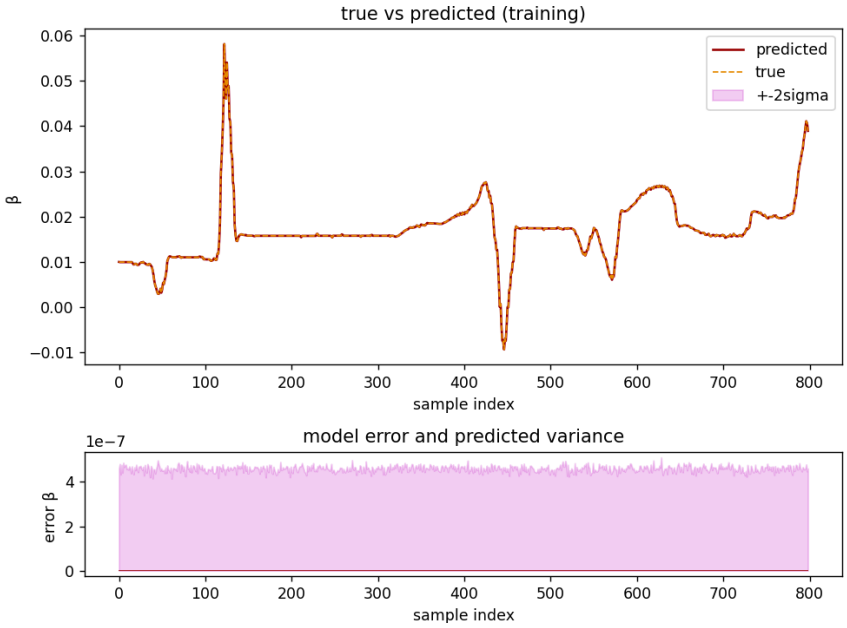
- Errors between ω_{dyn}, β_{dyn} vs ω_{kin}, β_{kin}

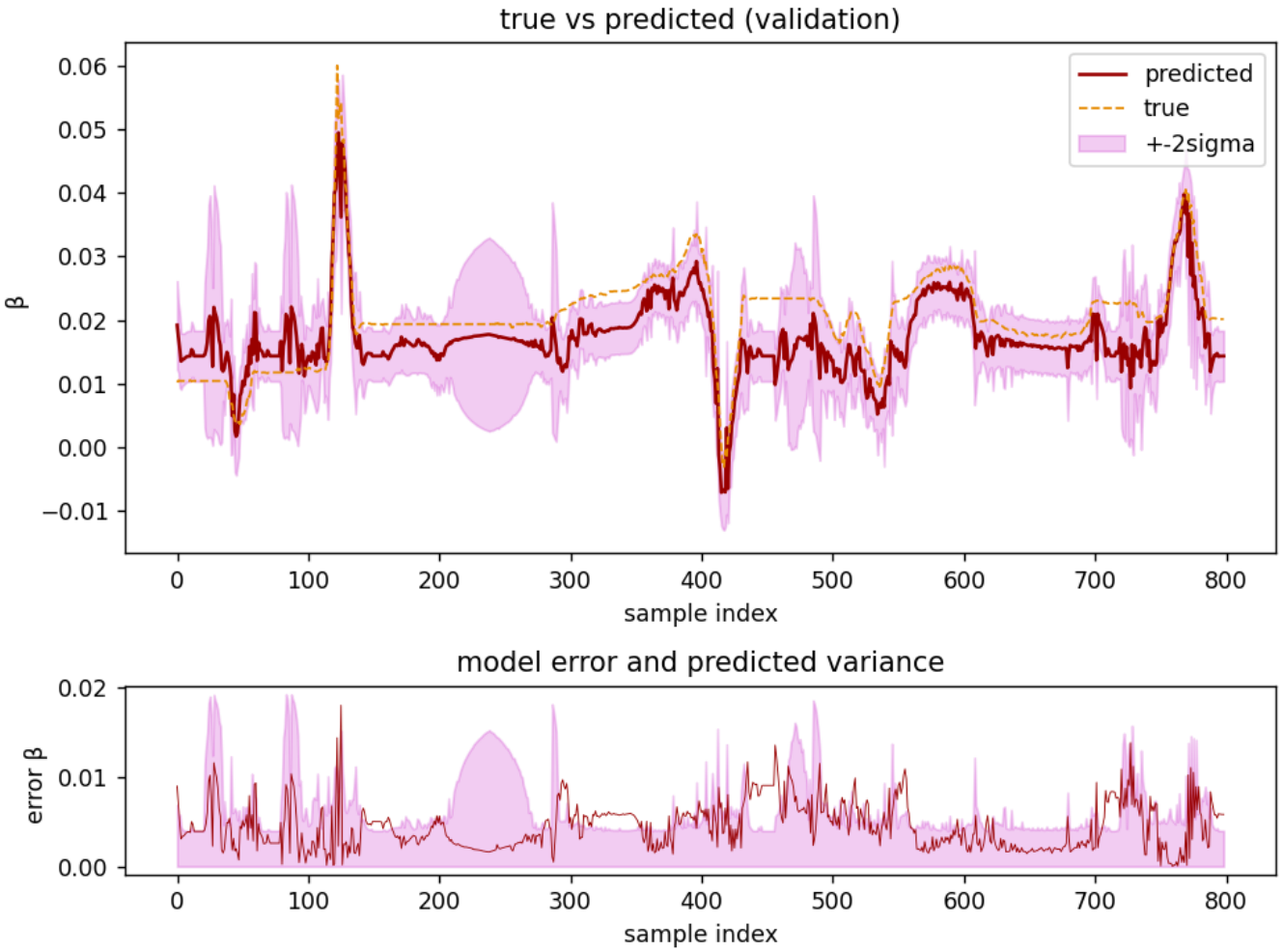




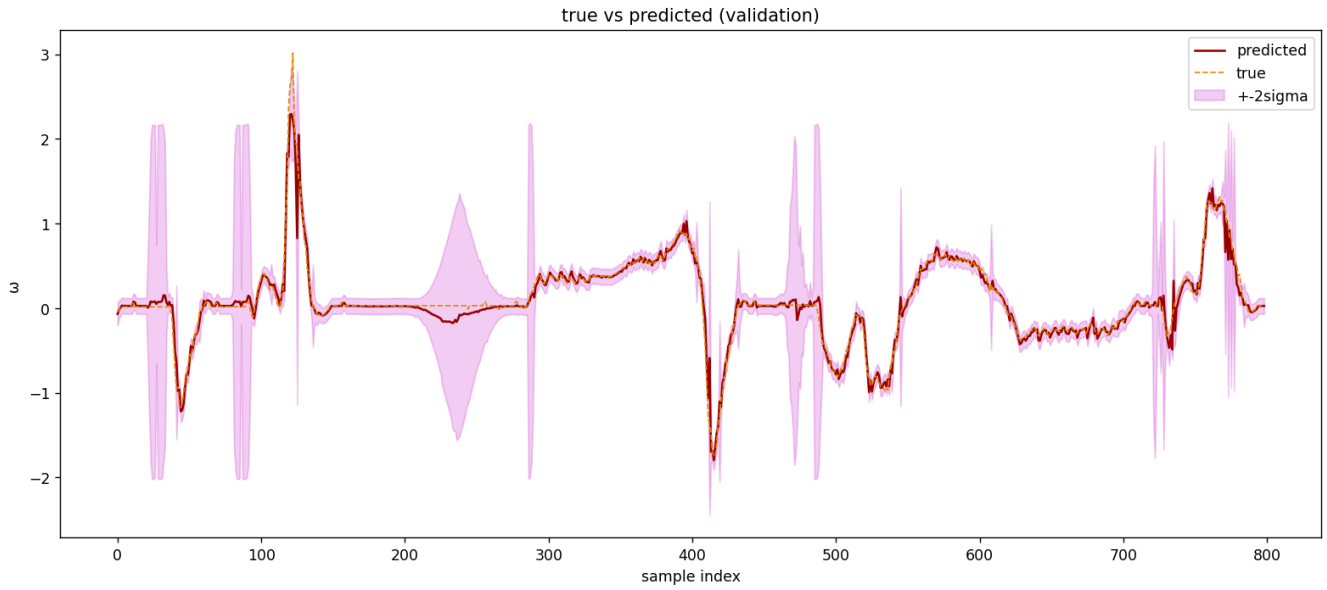
Training set (800/1184)
Testing set (800/1182)

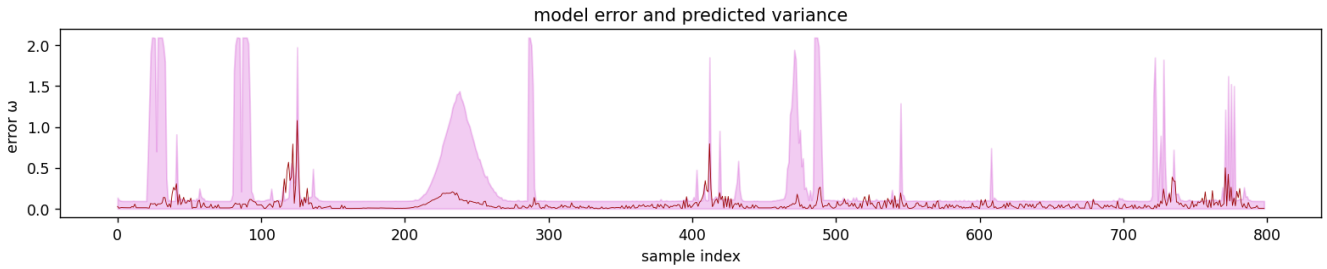
Input [acc, Δδ, δ, v],
Output [e_ω, e_β]





root mean square error: [0.00515288]
R2 score: [0.47196359]
explained variance: [0.67408147]





root mean square error: [0.10025354]
R2 score: [0.95792342]
explained variance: [0.95831086]