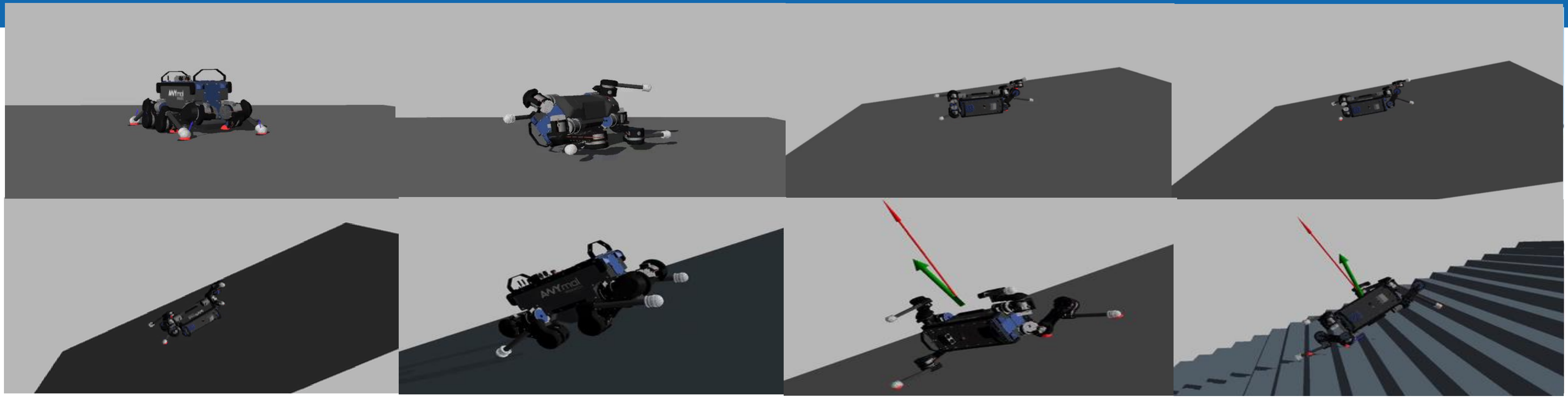


Recovery controller for ANYmal on various terrains

Kyoungyeon Choi

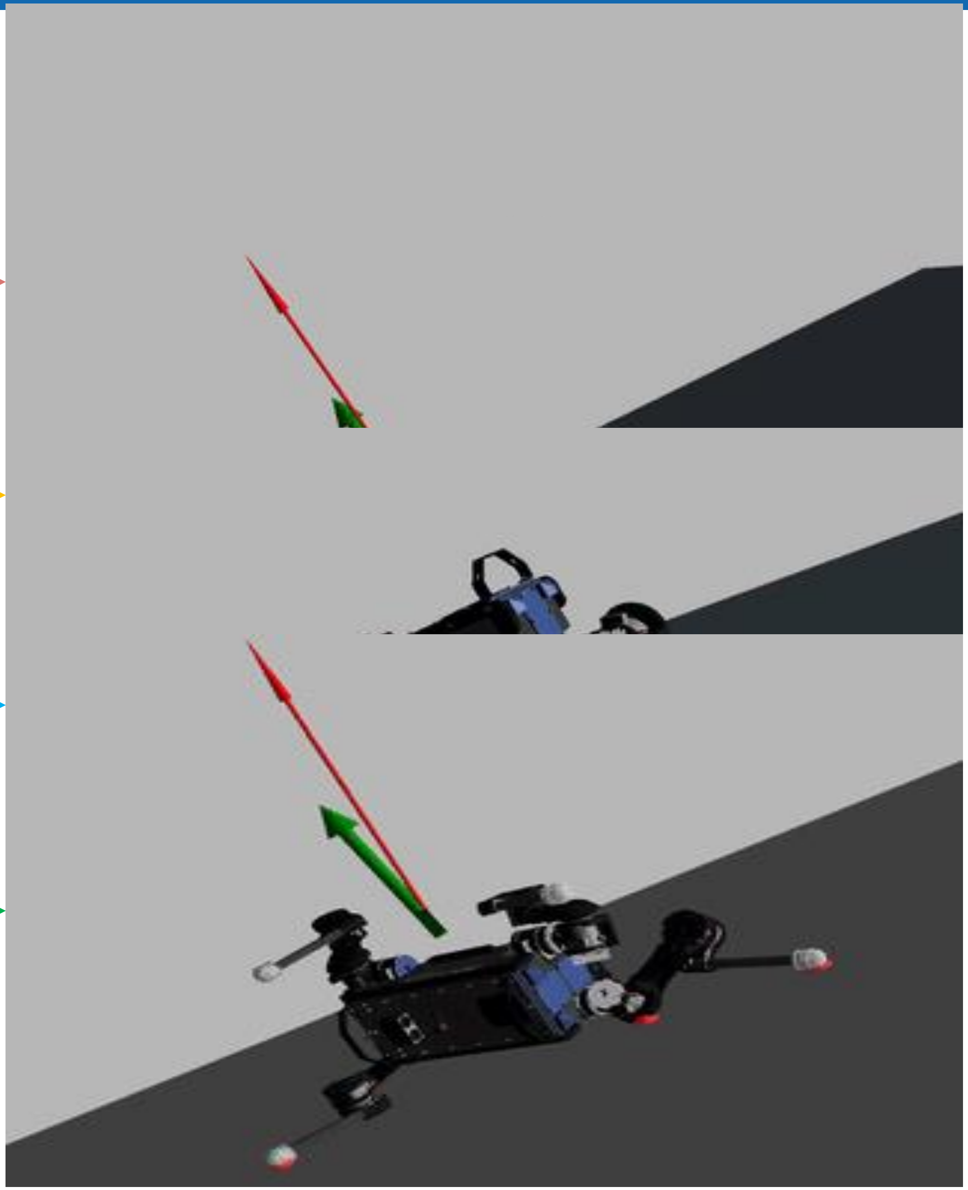
15 credits project
Supervisor: Joonho Lee, Takahiro Miki

Intro: Motivation

Backflipping MIT Mini Cheetah
https://www.youtube.com/watch?v=xNeZWP5Mx9s

Intro: Approach & Contributions

1. Recovery controller
2. Terrain normal estimation
3. Standing up controller
4. Finite state machine

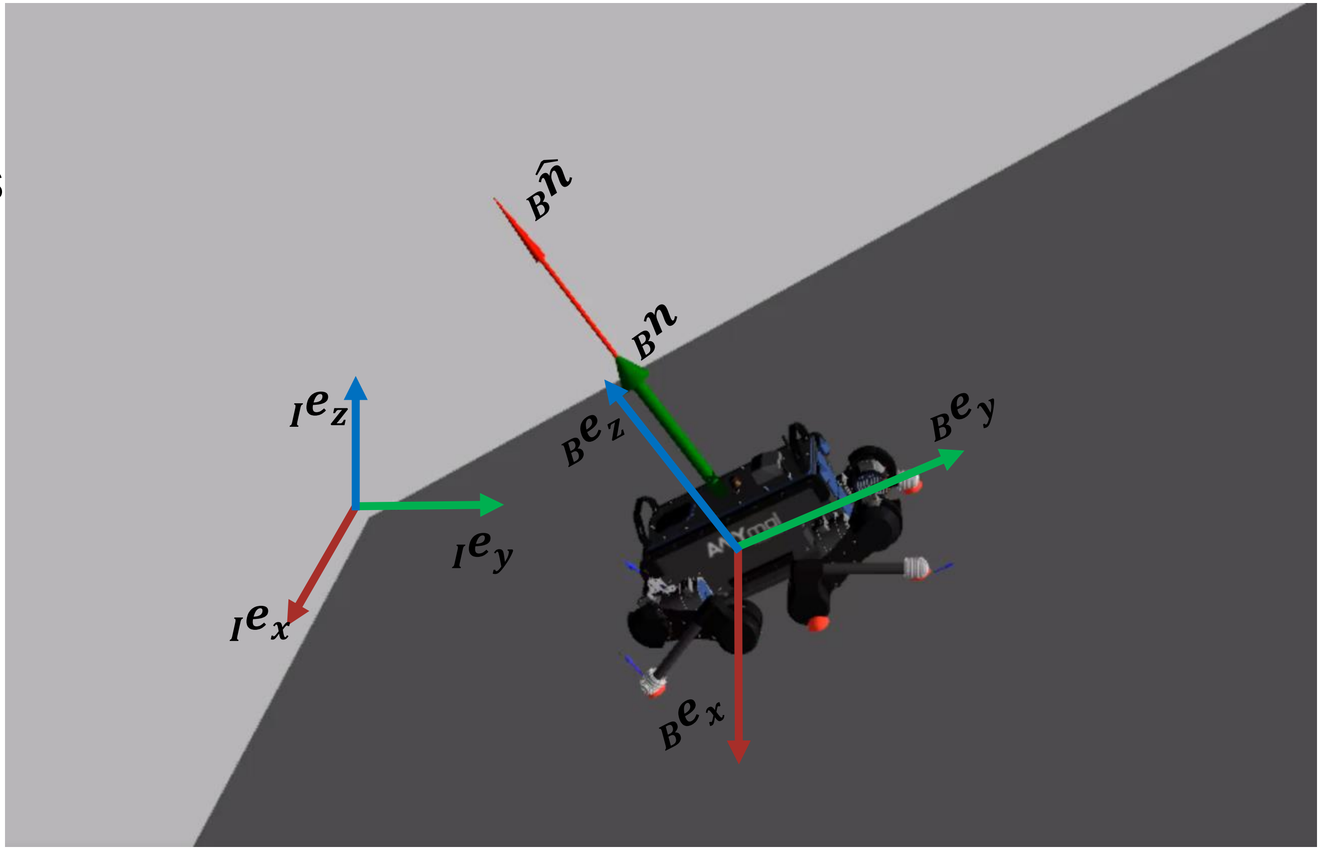


Method: Recovery Controller

$$I : inertial\ frame$$
$$B : Body\ frame$$
$${}_B n : terrain\ normal\ vector\ expressed\ in\ B$$
$${}_B \hat{n} : normal\ vector\ observation\ expressed\ in\ B$$

Task Objective :

Recover from arbitrary falling configurations and sit with a stable position.
(on flat and slope(0~36deg) terrains)



Method: Recovery Controller

MDP observation space O_t :

$<e_g,\ {}_B v_{IB},\ {}_B\omega_{IB}, \psi, \dot{\psi}_{t_k}, \dot{\psi}_{t_k - 0.01}, \dot{\psi}_{t_k - 0.02}, a_{k-1} > \in R^{69}$

[1,2]

Gravity vector : $e_g \in R^3$

Linear velocity : ${}_B v_{IB} \in R^3$

Angular velocity : ${}_B\omega_{IB} \in R^3$

Joint positions : $\psi \in R^{12}$

Joint velocity history : $\dot{\psi}_{t_k}, \dot{\psi}_{t_k - 0.01}, \dot{\psi}_{t_k - 0.02} \in R^{36}$

Previous action: $a_{k-1} \in R^{12}$

MDP action space a_t :

$\psi_d = \psi_t + \mathrm{k}\, a_t$ (k = 0.1), $a_t \in R^{12}$

Current joint positions : $\psi_t \in R^{12}$

Desired joint positions : $\psi_d \in R^{12}$

Cost terms :

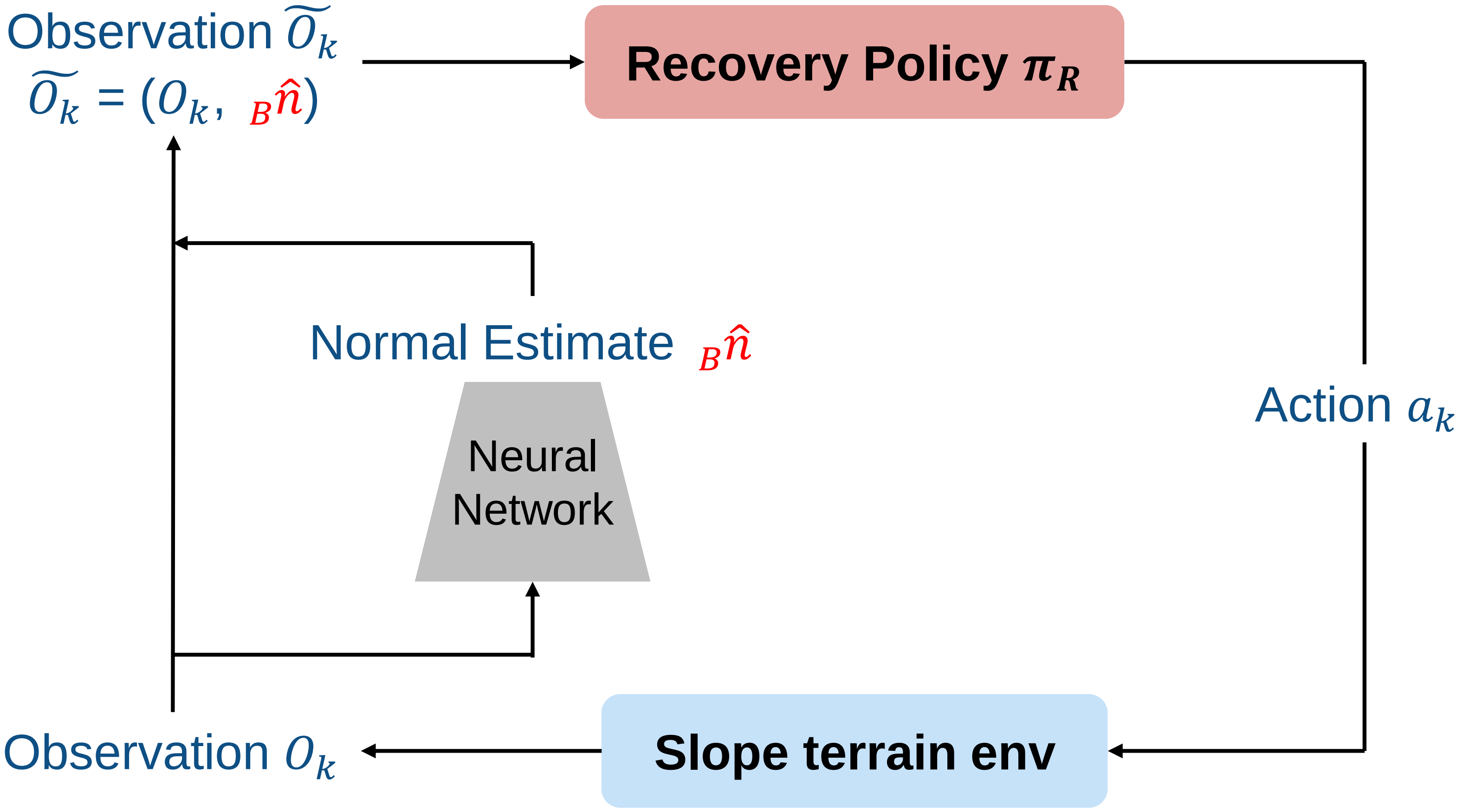
Orientation, Joint position, Self collision, Joint acceleration, Joint velocity, Torque

Orientation Cost : $({}_B\hat{n} \cdot {}_B e_z - 1)^2$

Terrain normal vector exp in body frame: ${}_B\hat{n} \in R^3$

Body z axis exp in body frame: ${}_B e_z \in R^3$

Method: Normal Estimator

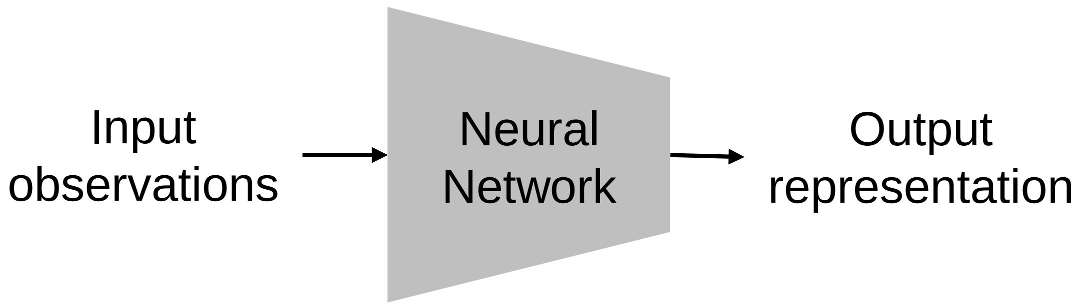


Method: Normal Estimator

Input observations:

$< e_g \, , \psi > \in \; R^{15}$

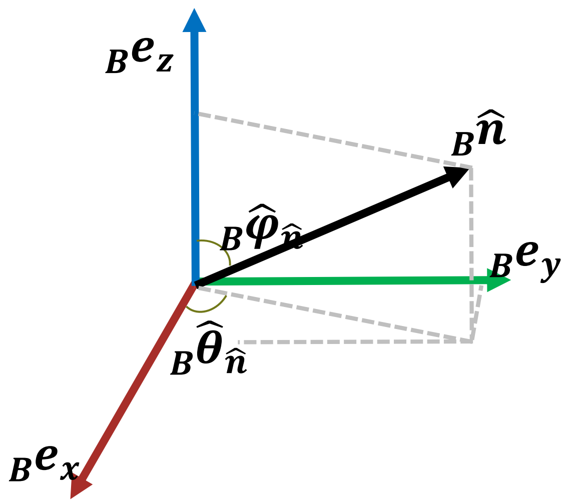
Gravity vector : $e_g \in \; R^3$
Joint positions : $\psi \in \; R^{12}$



Output representation :

$< \cos \; {}_B\theta_n \, , \, \sin \; {}_B\theta_n \; , \, \cos \; {}_B\varphi_n \, , \, \sin \; {}_B\varphi_n \; > \in \; R^4$ [3]

Spherical representation of ${}_B\hat{n}$: ${}_B\hat{\varphi}_{\hat{n}}$, ${}_B\hat{\theta}_{\hat{n}}$



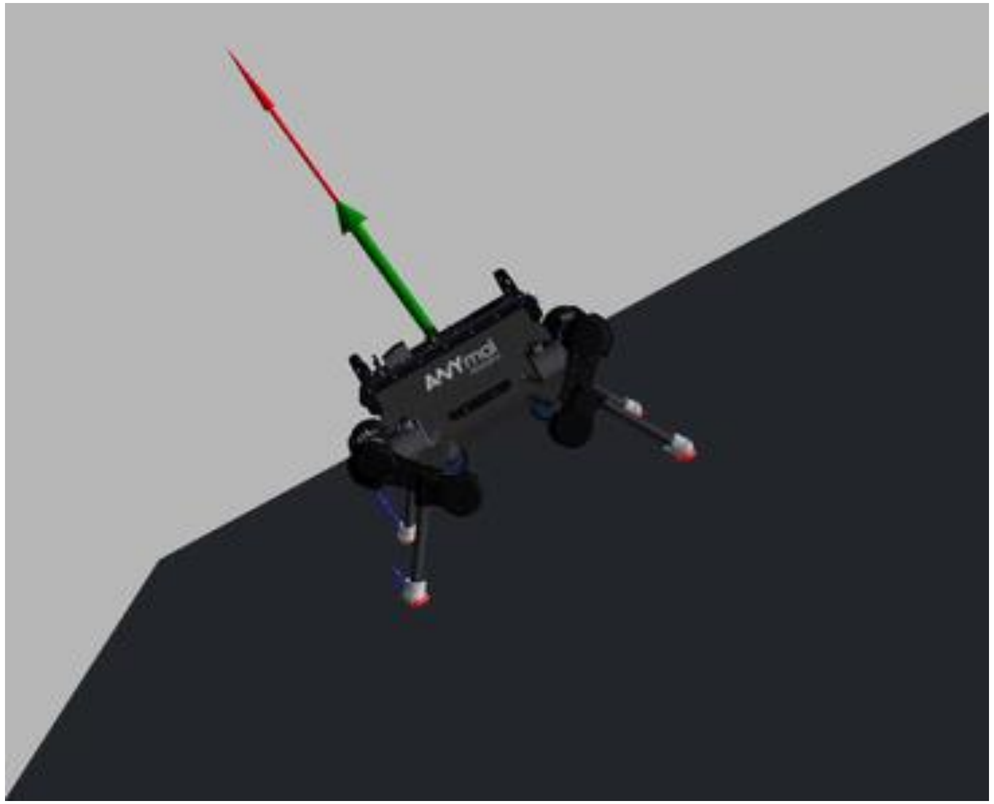
Network architecture:

Size : 128×128×128
Activation : ReLu

Method: Standing up Controller

Task Objective :

Stand up from near sitting configuration
and balance with a stable position.
(on flat and slope(0~36deg) terrains)



Method: Standing up Controller

MDP observation space O_t :

$< e_g,\ {}_B v_{IB},\ {}_B\omega_{IB}, \psi, \dot{\psi}_{t_k}, \dot{\psi}_{t_k - 0.01}, \dot{\psi}_{t_k - 0.02}, a_{k-1} >$ [1,2]

Gravity vector : $e_g \in R^3$

Linear velocity : ${}_B v_{IB} \in R^3$

Angular velocity : ${}_B\omega_{IB} \in R^3$

Joint positions : $\psi \in R^{12}$

Joint velocity history : $\dot{\psi}_{t_k}, \dot{\psi}_{t_k - 0.01}, \dot{\psi}_{t_k - 0.02} \in R^{36}$

Previous action: $a_{k-1} \in R^{12}$

MDP action space a_t :

$\psi_d = \psi_t + k\, a_t$ (k = 0.1), $a_t \in R^{12}$

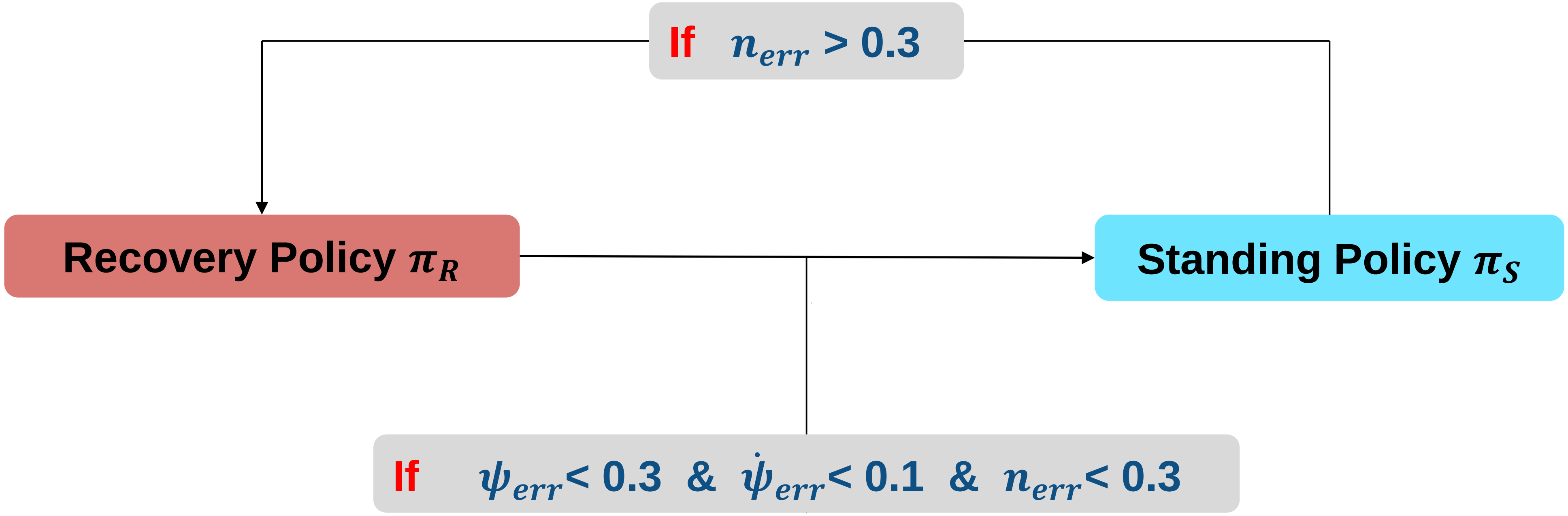
Current joint positions : $\psi_t \in R^{12}$

Desired joint positions : $\psi_d \in R^{12}$

Cost terms :

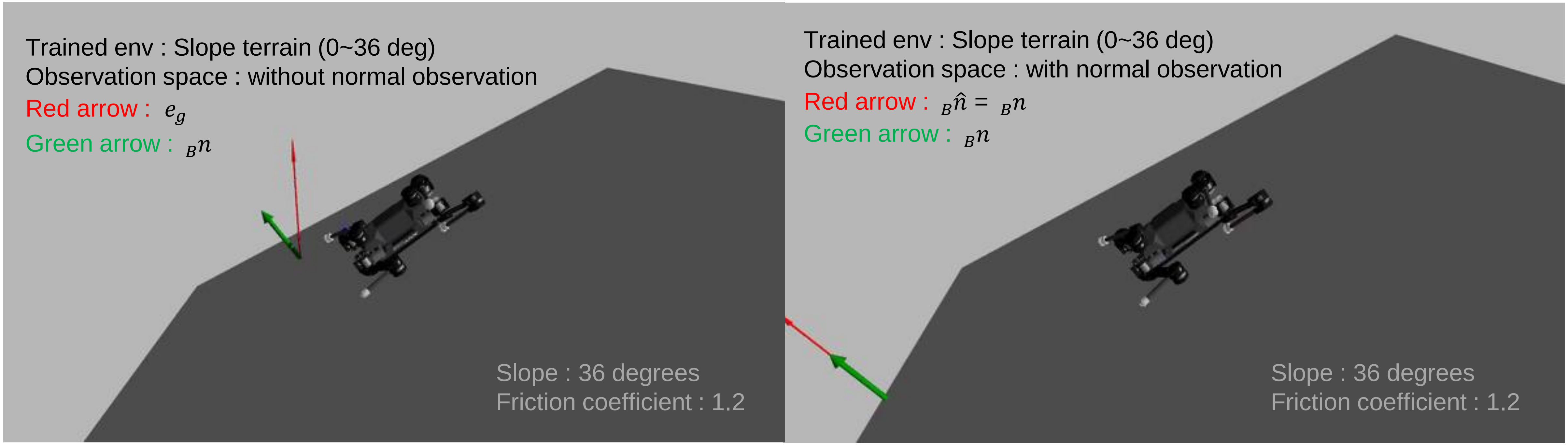
Joint position, Height, Slip, Orientation, Self collision, Joint acceleration, Joint velocity, Torque

Method: Finite state machine



Results: Recovery Controller simulation results (flat terrain)

Results: Recovery Controller simulation results (slope terrain)



Results: Normal estimator error

1) No history

Input : Gravity vector $e_g \in R^3$, Joint position $\psi \in R^{12}$

Output : $\cos {}_B\theta_n$, $sin\ {}_B\theta_n$, $\cos {}_B\varphi_n$, $\sin {}_B\varphi_n$

Hidden layer size : $128 \times 128 \times 128$

Activation : ReLu

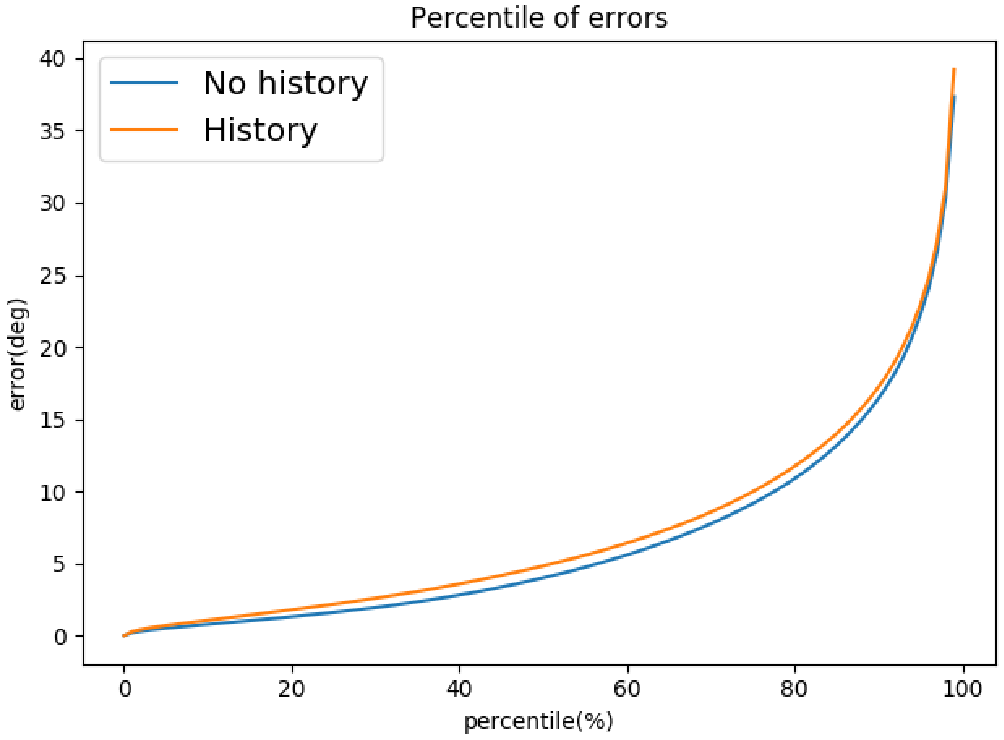
2) History input

Input : $(o_{k-4}, a_{k-4}, o_{k-3}, a_{k-3} \cdots , o_k , a_k)$, $o_k = (e_g , \psi)$

Output : $\cos {}_B\theta_n$, $sin\ {}_B\theta_n$, $\cos {}_B\varphi_n$, $\sin {}_B\varphi_n$

Hidden layer size : $128 \times 128 \times 128$

Activation : ReLu



	Mean(deg)	Std(deg)
No history	6.806	8.17
History input	7.544	8.69

Table 1. Model validation L1 loss (1600000 time steps)

Results: Policy comparison

		Observation		
		No normal	Normal(truth)	Normal(estimate)
Env	flat_without_normal (fwn)	Train(flat)/Test(slope)		
	slope_without_normal (swn)	Train(slope)/Test(slope)		
	slope_normal(Explicit) (snE)		Train(slope)/Test(slope)	
	slope_normal_Predicted (snP)		Train(slope)	Test(slope)
	slope_normal(Implicit) (snI)	Test(slope)	Train(slope)	

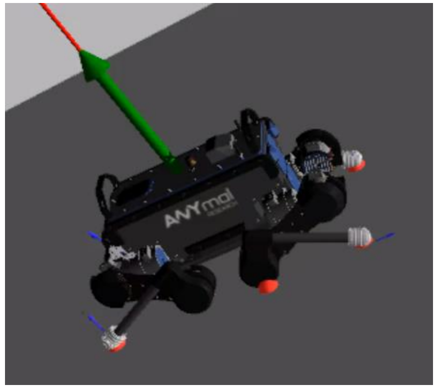
Table 2. Policies trained with different observations

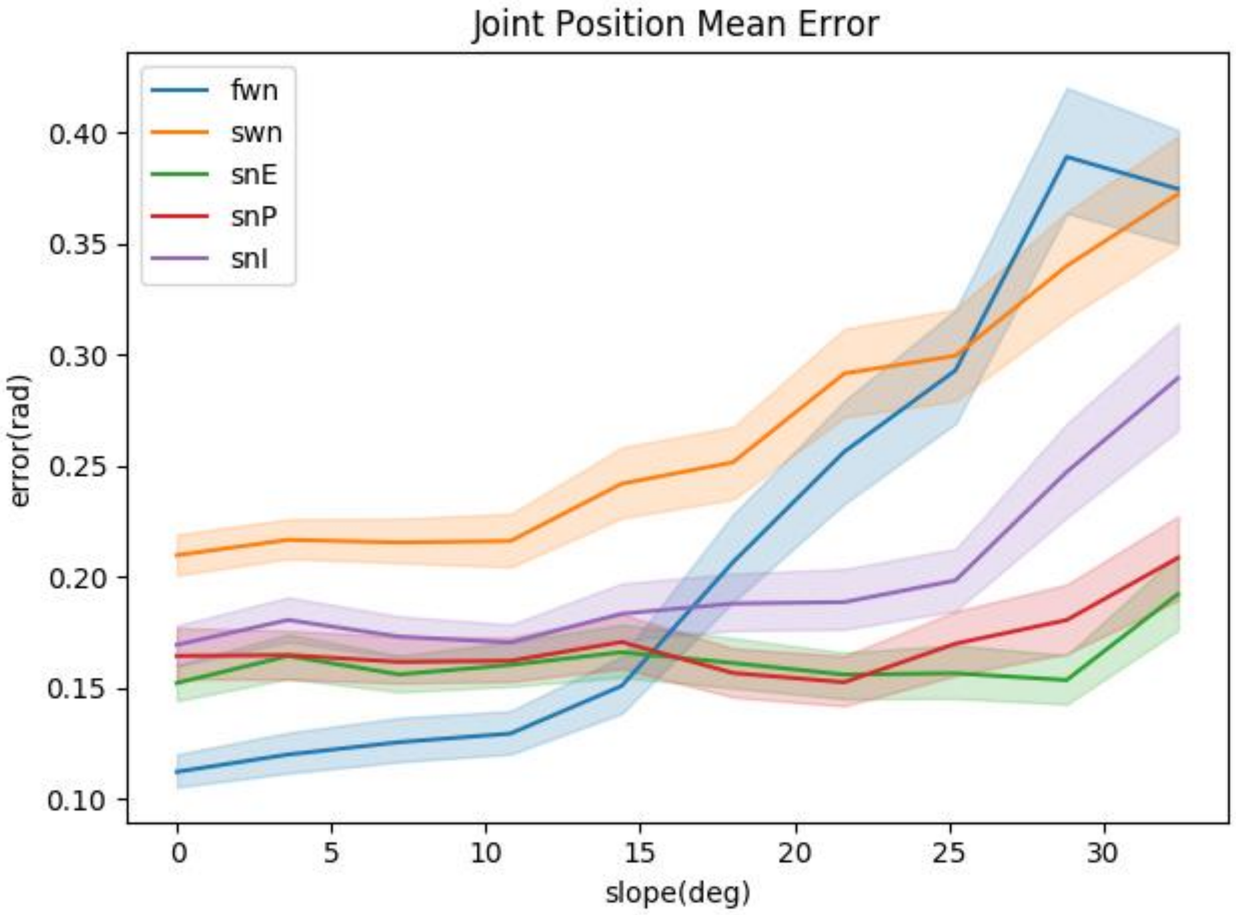
Results: Joint position error

1000 test episodes per slope (0, 6, 12, …, 36deg)

Last time step observation : Joint positions $\psi \in R^{12}$
Desired position (sit) : $\psi_{des} \in R^{12}$

Joint position error : $|\psi_{des} - \psi|$

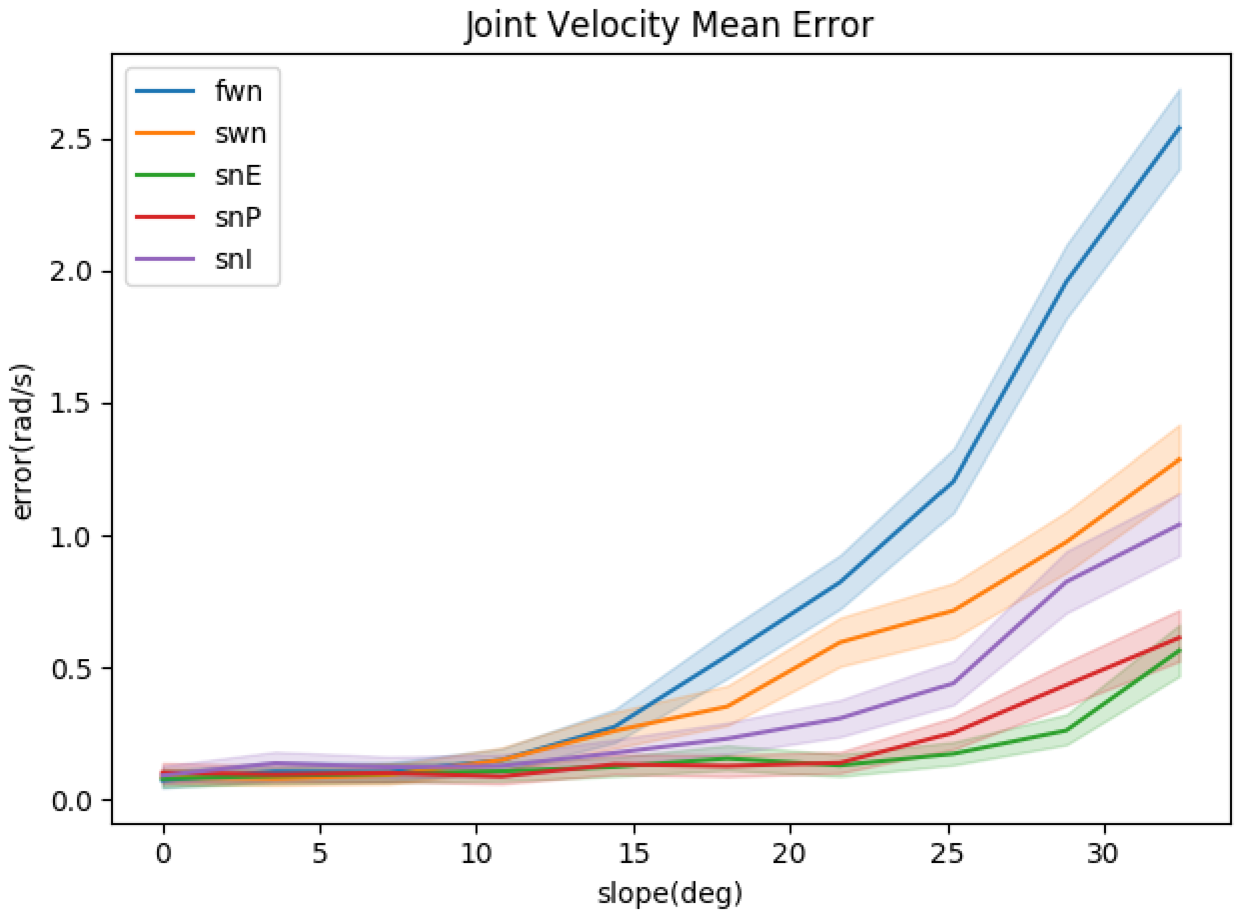




Results: Joint velocity error

Last time step observation : Joint velocities $\dot{\psi} \in R^{1}$
Desired velocity : $\psi_{des} \in R^{12}$, $\dot{\psi}_{des} = 0$

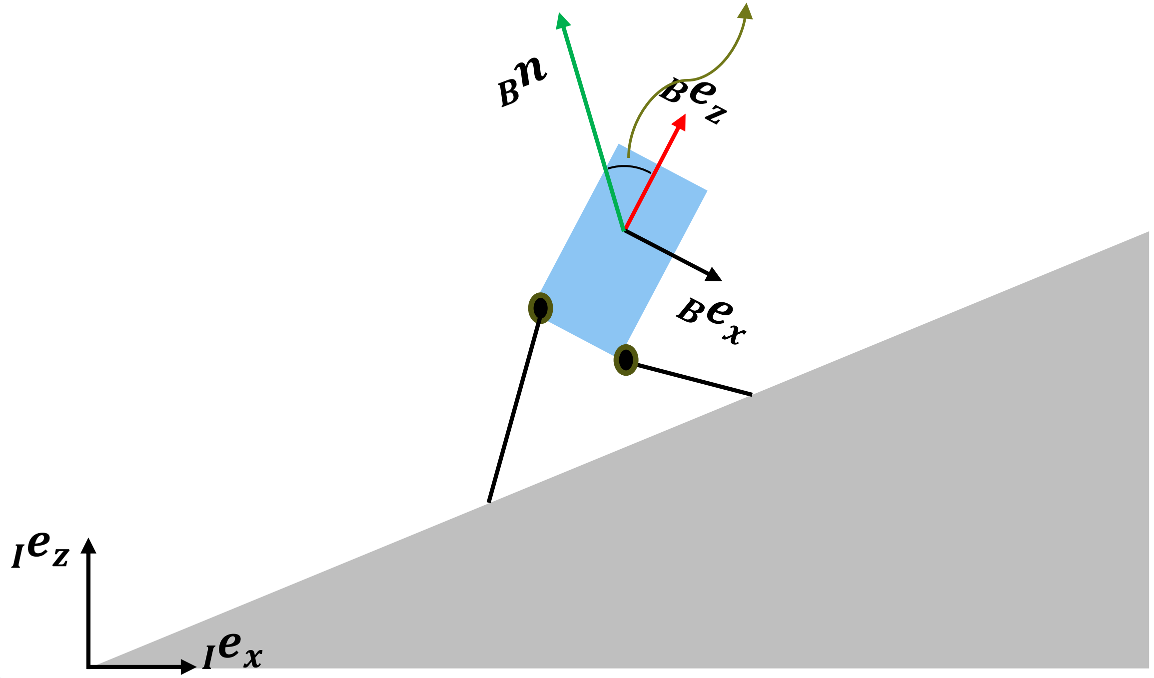
Joint velocity error : $|\dot{\psi}_{des} - \dot{\psi}|$

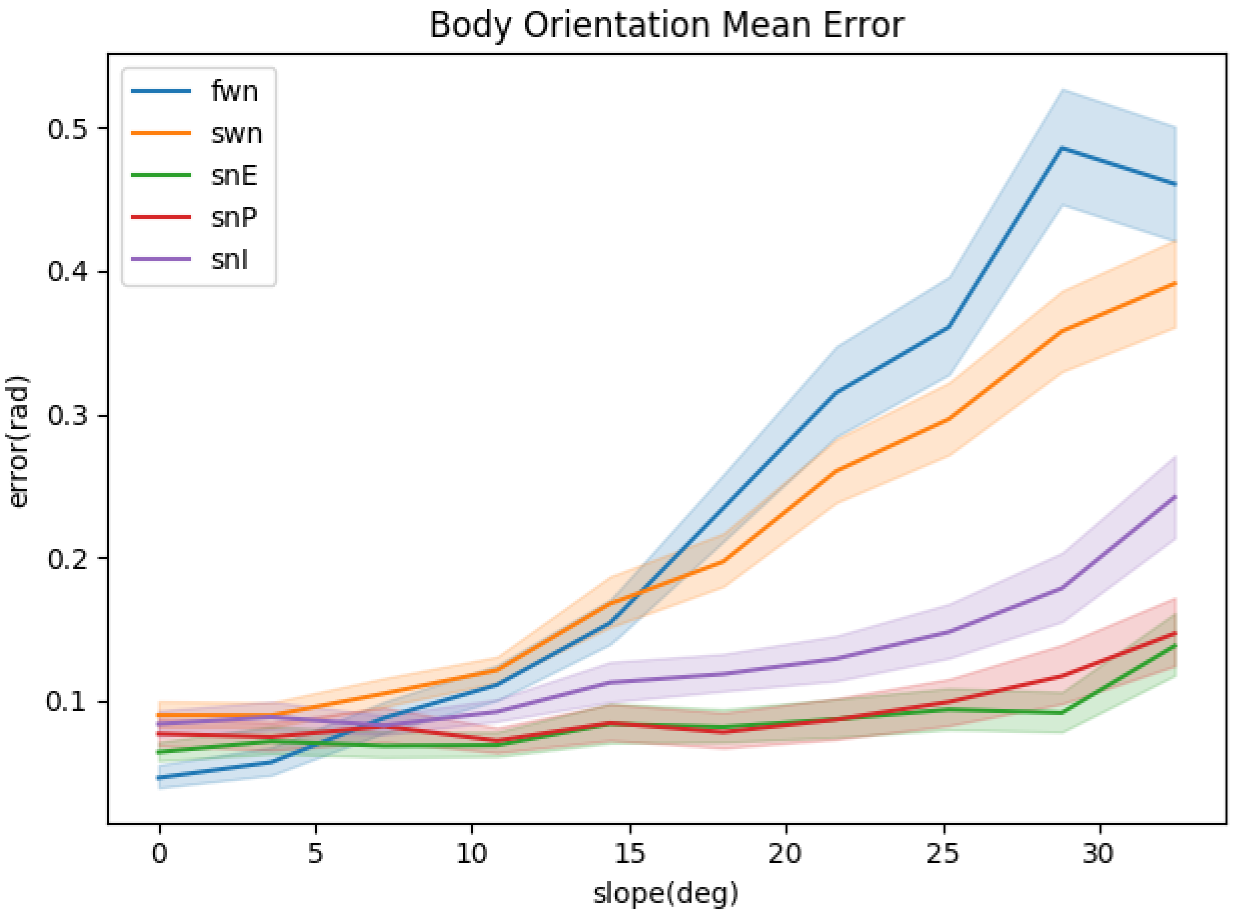


Results: Body orientation error

Last time step observation : Base z axis ${}_B e_z \in R^3$
Desired orientation : ${}_B e_{des} = {}_B n$

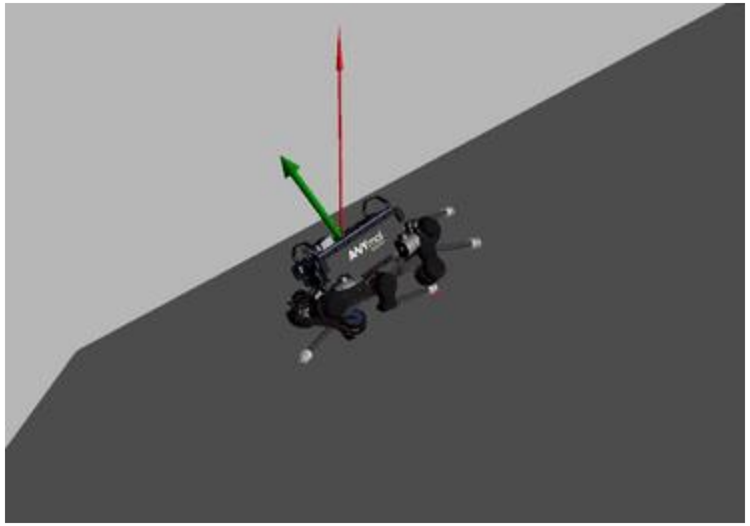
Body orientation error : $\cos^{-1}({}_B n \cdot {}_B e_z)$

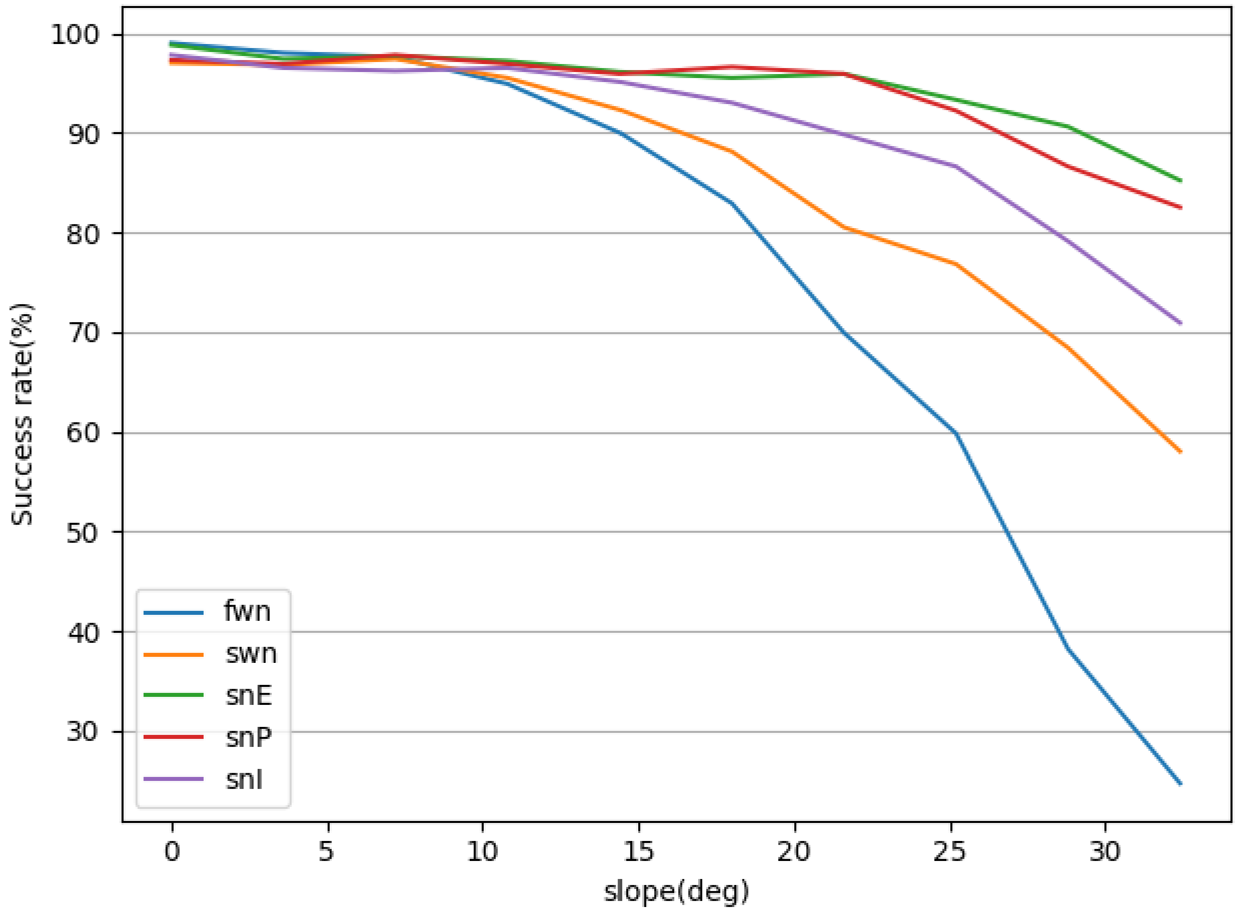




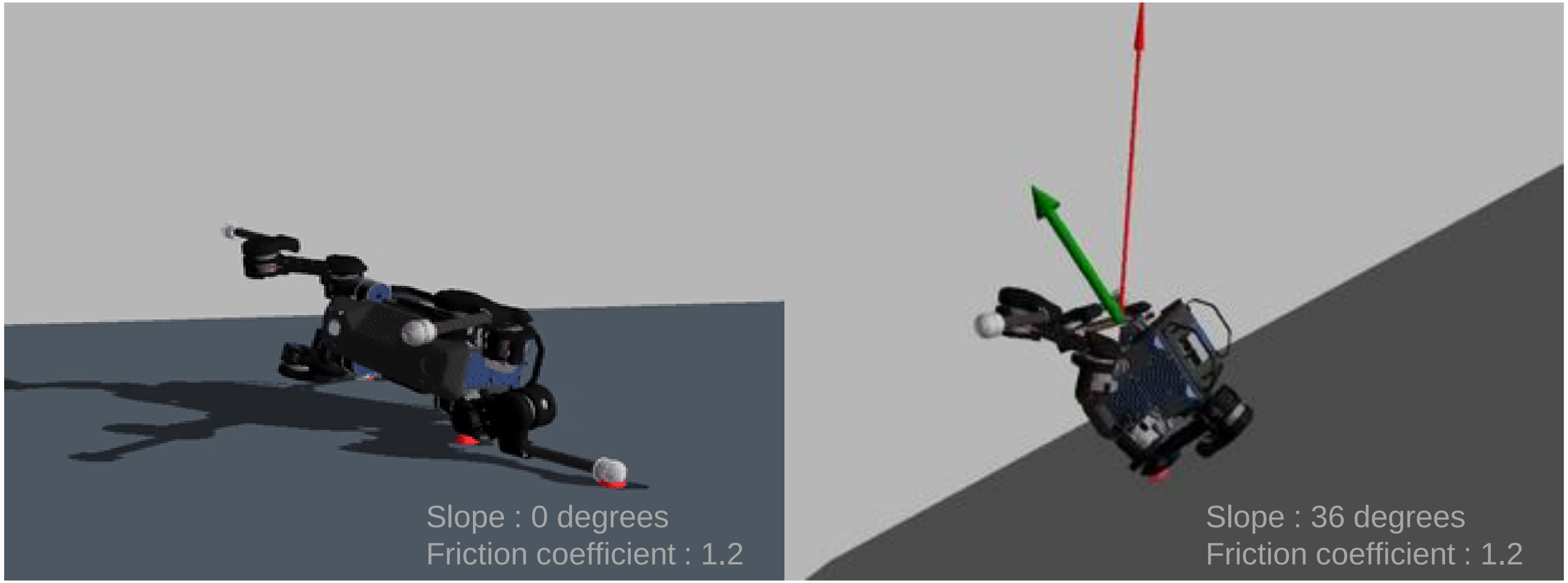
Results: Success rate

Success definition :
Joint position error < 0.5(rad)
Joint velocity error < 0.1(rad/s)
Body orientation error < 1.0(rad)

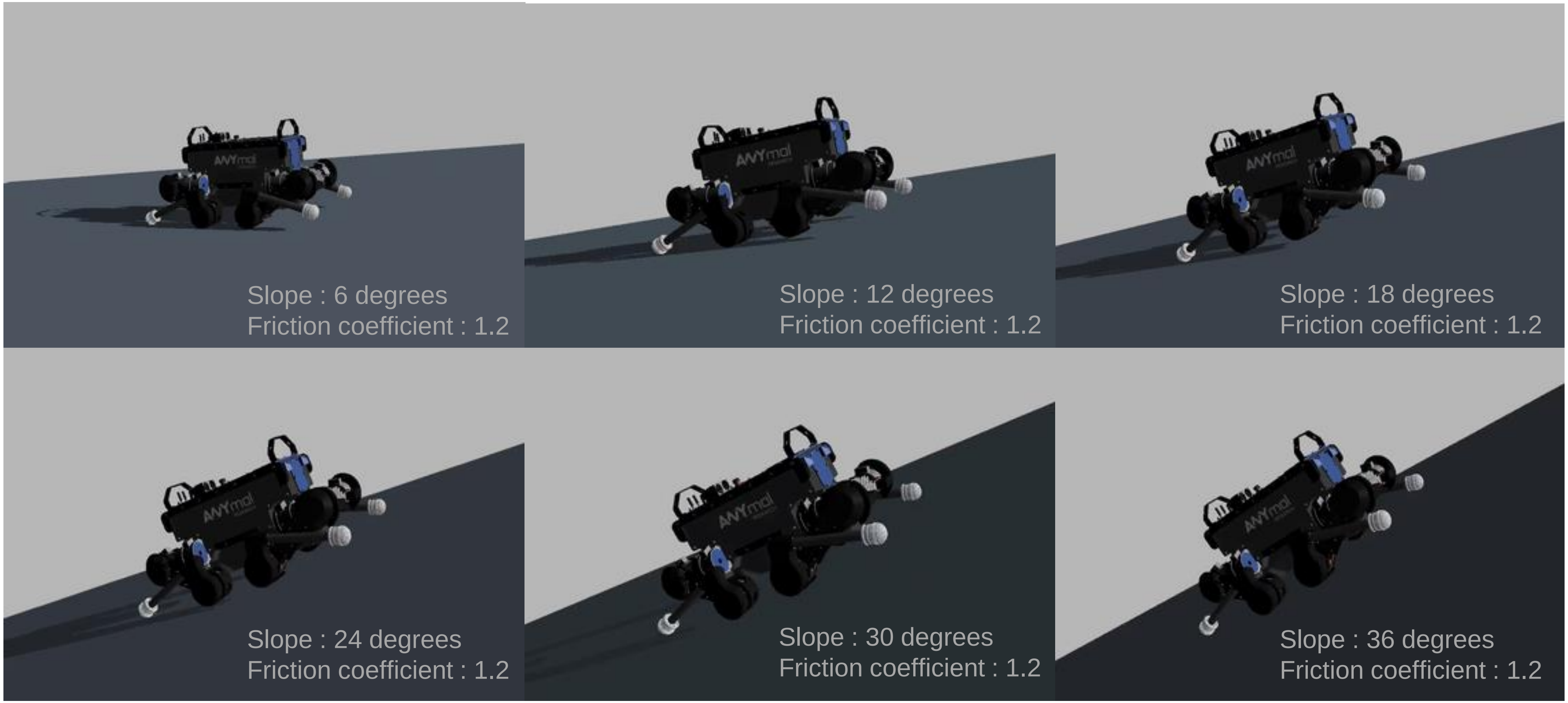




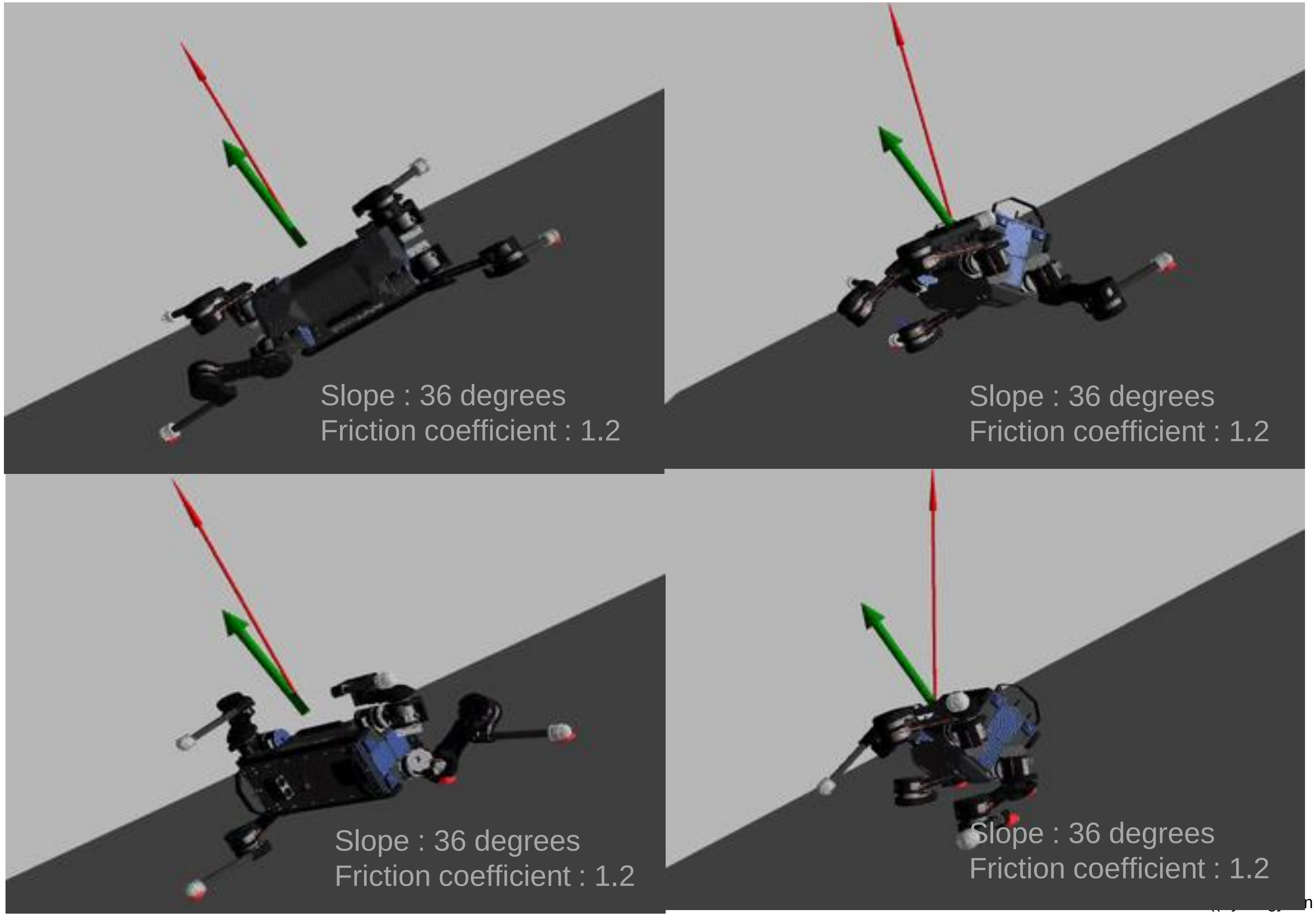
Results: Different behaviors on slope



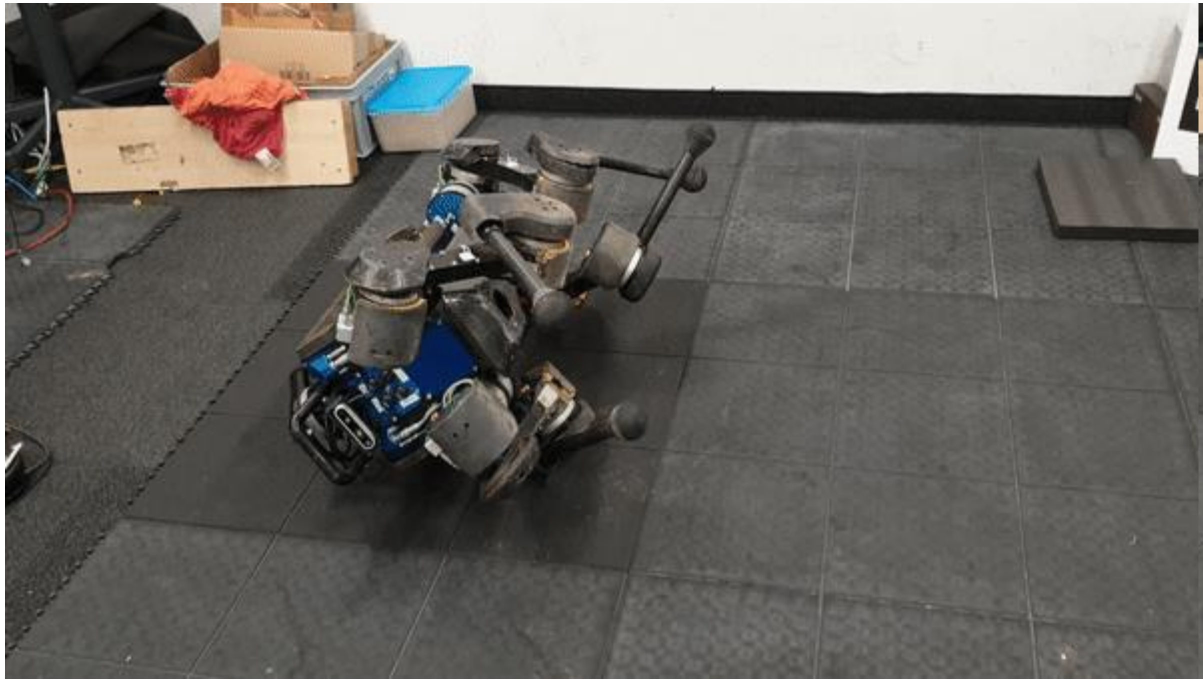
Results: Standing up Controller simulation results



Results: Finite state machine simulation results



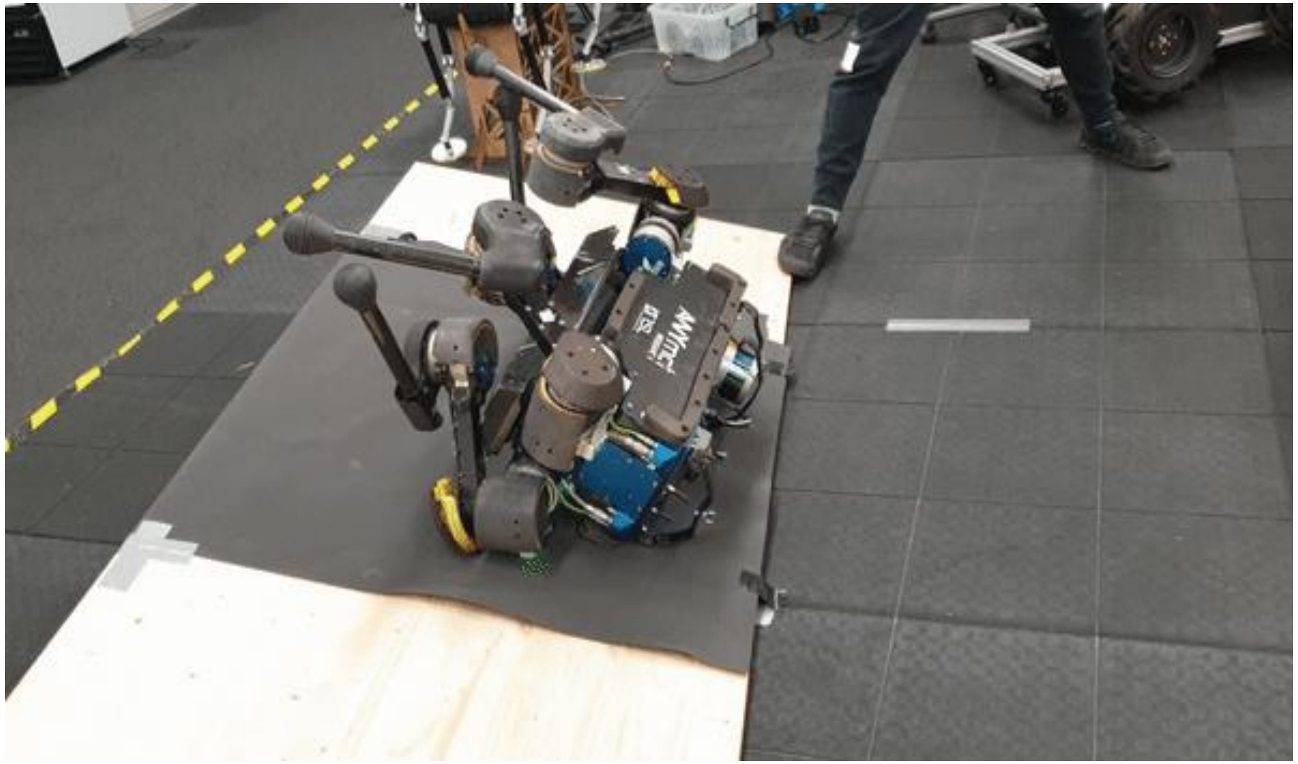
Results: Real-world experiments(flat terrain)



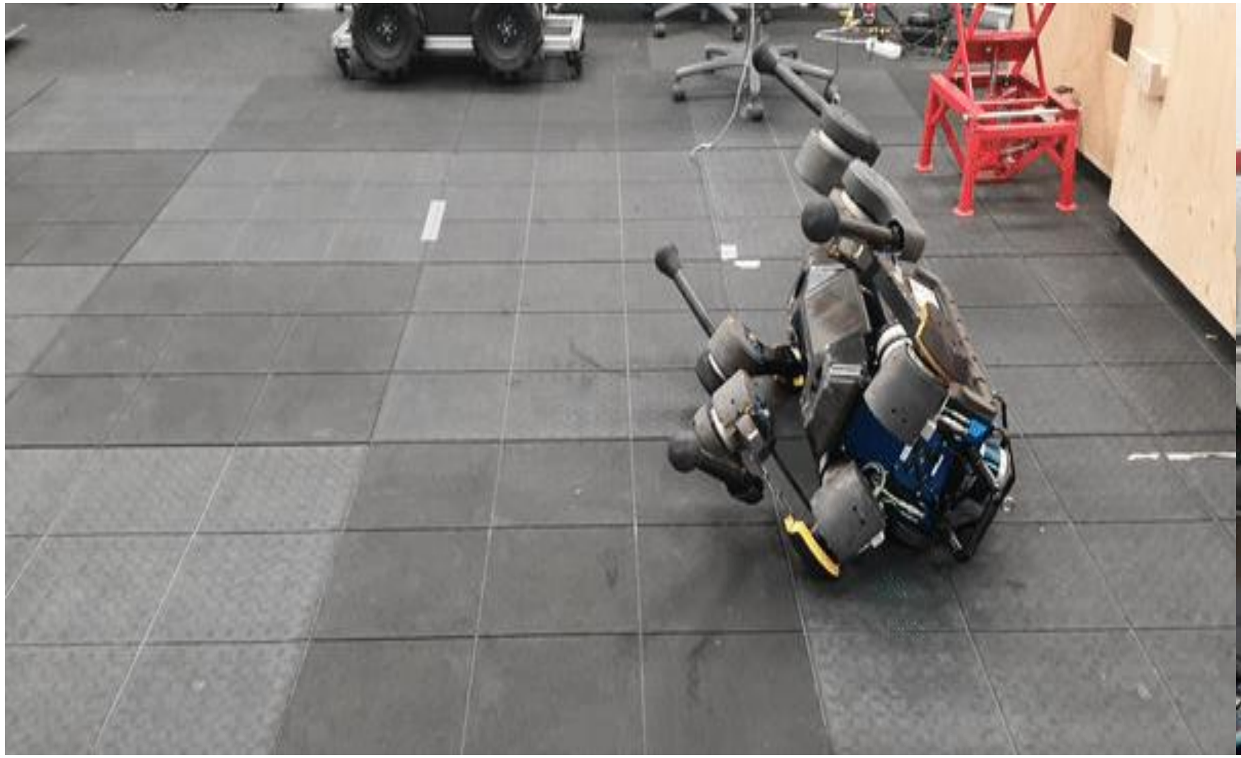
Results: Real-world experiments(slope terrain)

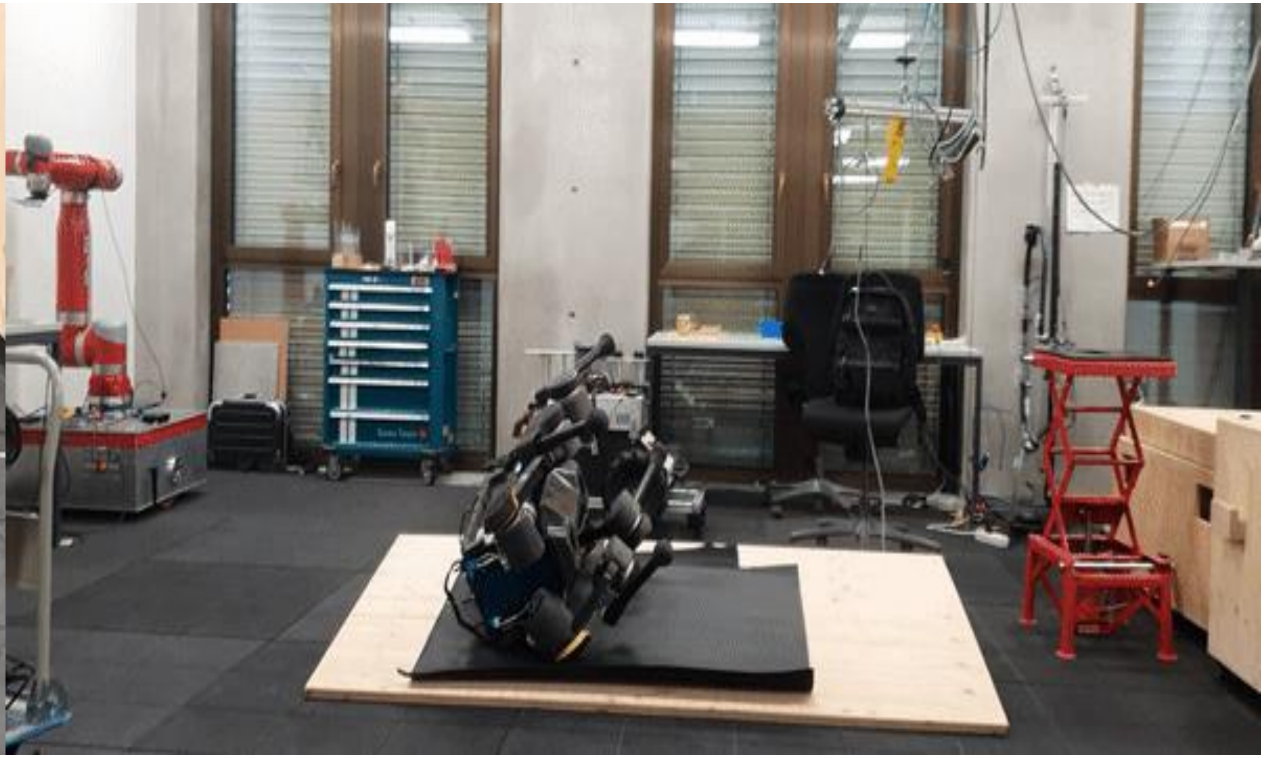
Problem : Linear velocity, Height drift
State drift leads to abnormal behaviors

Solution : Exclude states that have drift issues



Results: Real-world experiments(finite state machine)





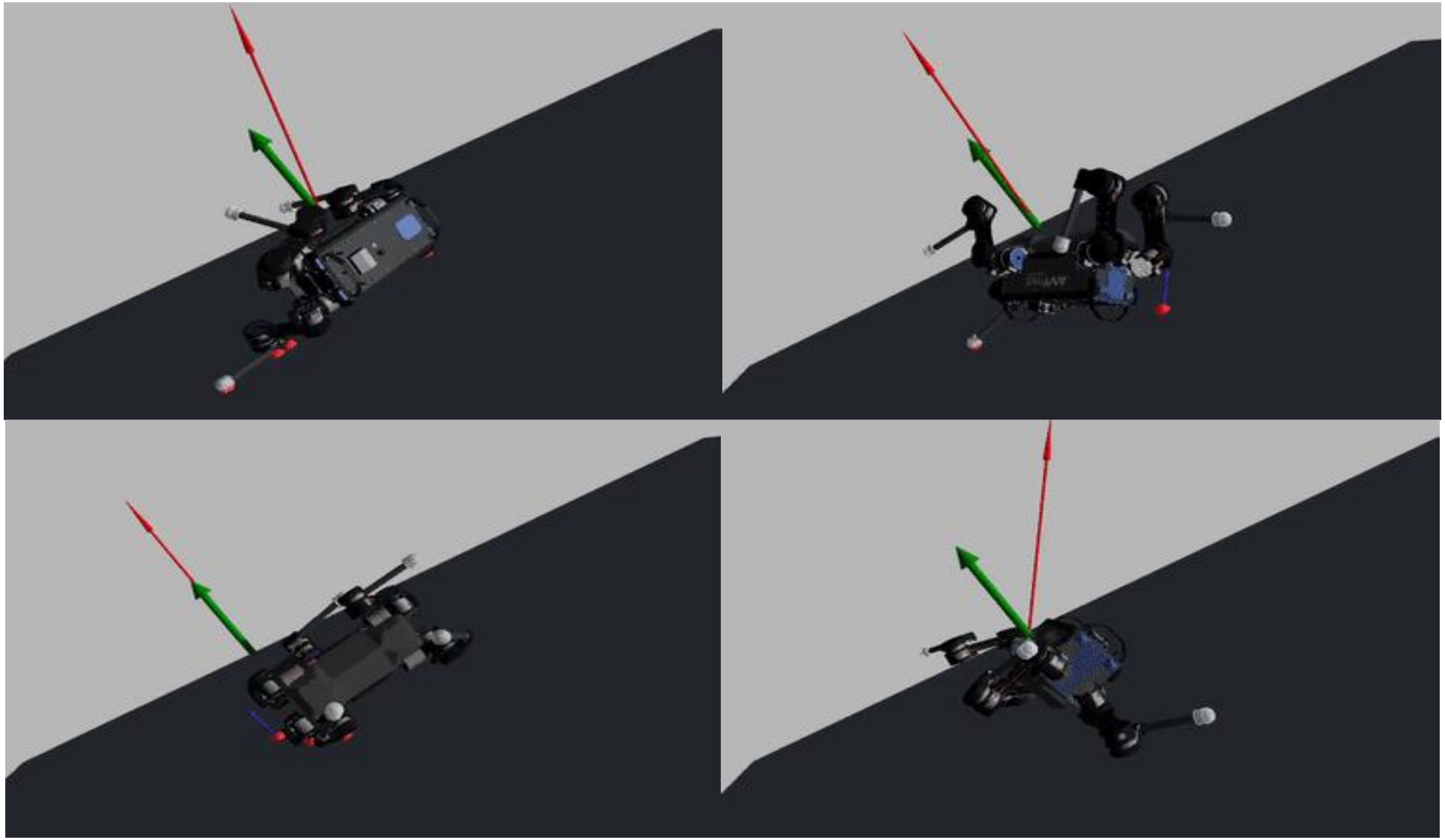
Results: Real-world experiments(finite state machine)

Problem : Normal estimation noise
Policy trained only with ground truth normal

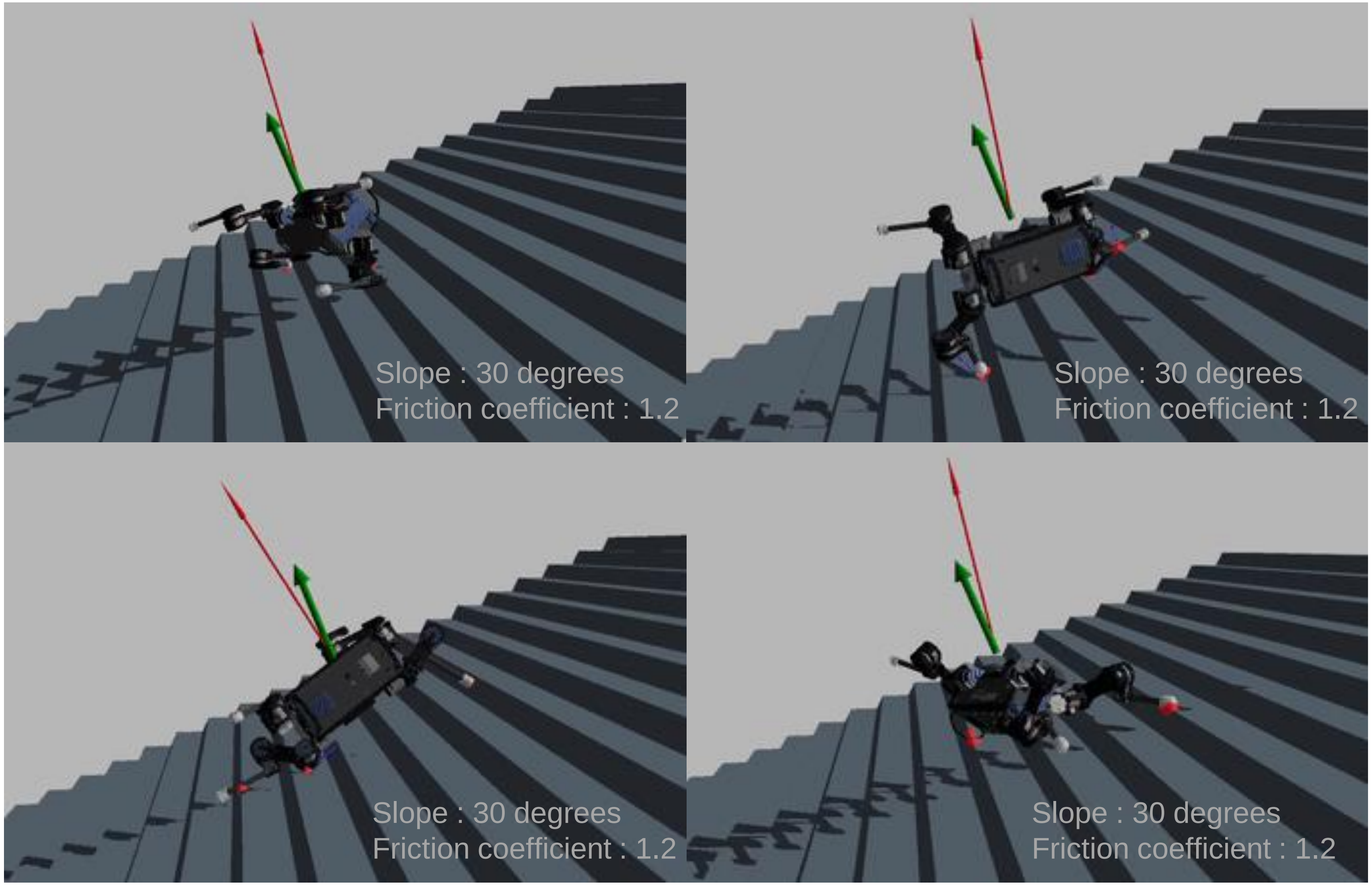
Solution : Train policy with normal estimator

		Observation		
		No normal	Normal(truth)	Normal(estimate)
Env	Train(slope)			✓
	Test(slope)			✓

Results: Simulation results(trained with estimator)



Thank you


Slope : 30 degrees
Friction coefficient : 1.2
Slope : 30 degrees
Friction coefficient : 1.2
Slope : 30 degrees
Friction coefficient : 1.2
Slope : 30 degrees
Friction coefficient : 1.2

References

1. J. Lee, J. Hwangbo, and M. Hutter, “Robust recovery controller for a quadrupedal robot using deep reinforcement learning,” arXiv preprint arXiv:1901.07517, 2019.

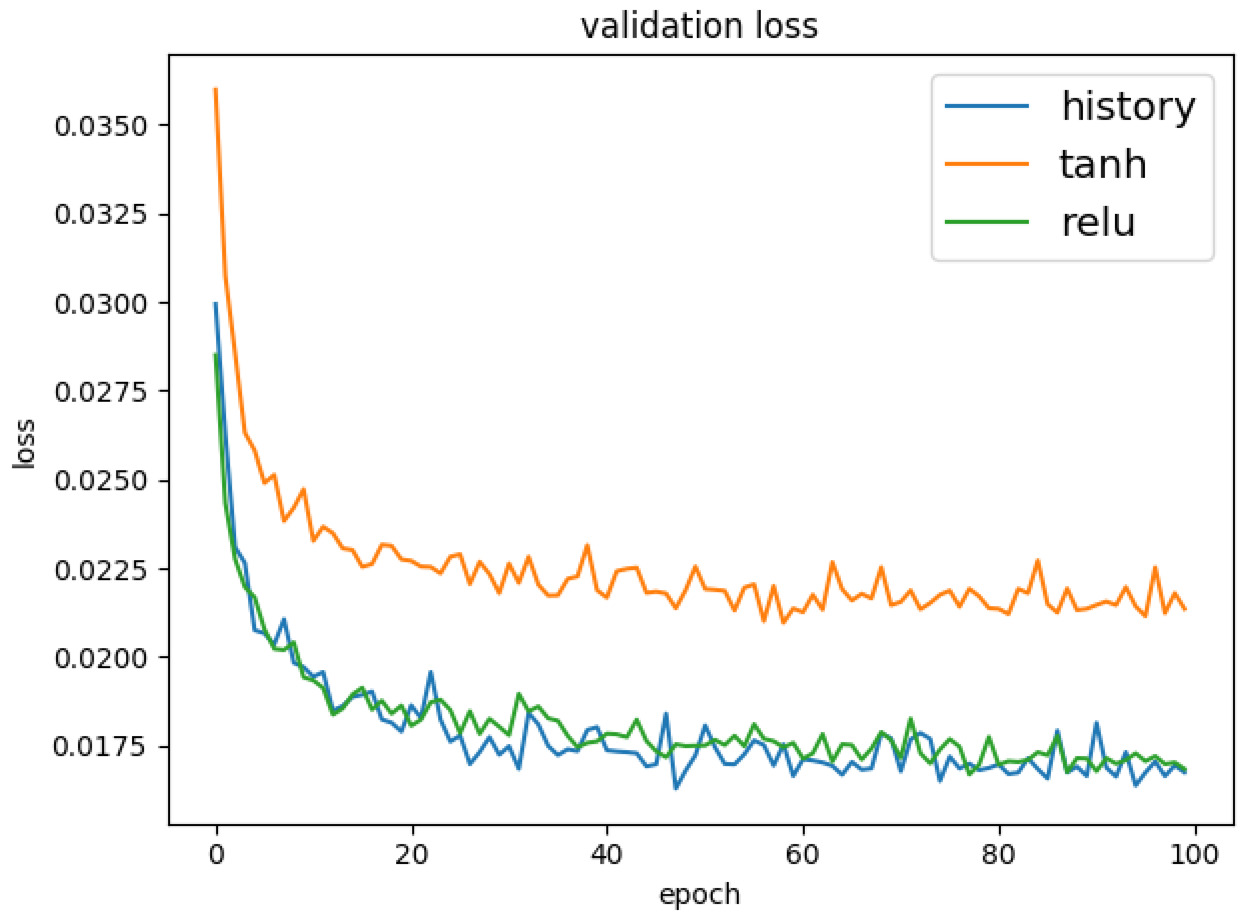
2. Hwangbo, Jemin et al. “Learning Agile and Dynamic Motor Skills for Legged Robots.” Science Robotics 4.26 (2019)

3. Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. arXiv preprint arXiv:1812.07035, 2018.

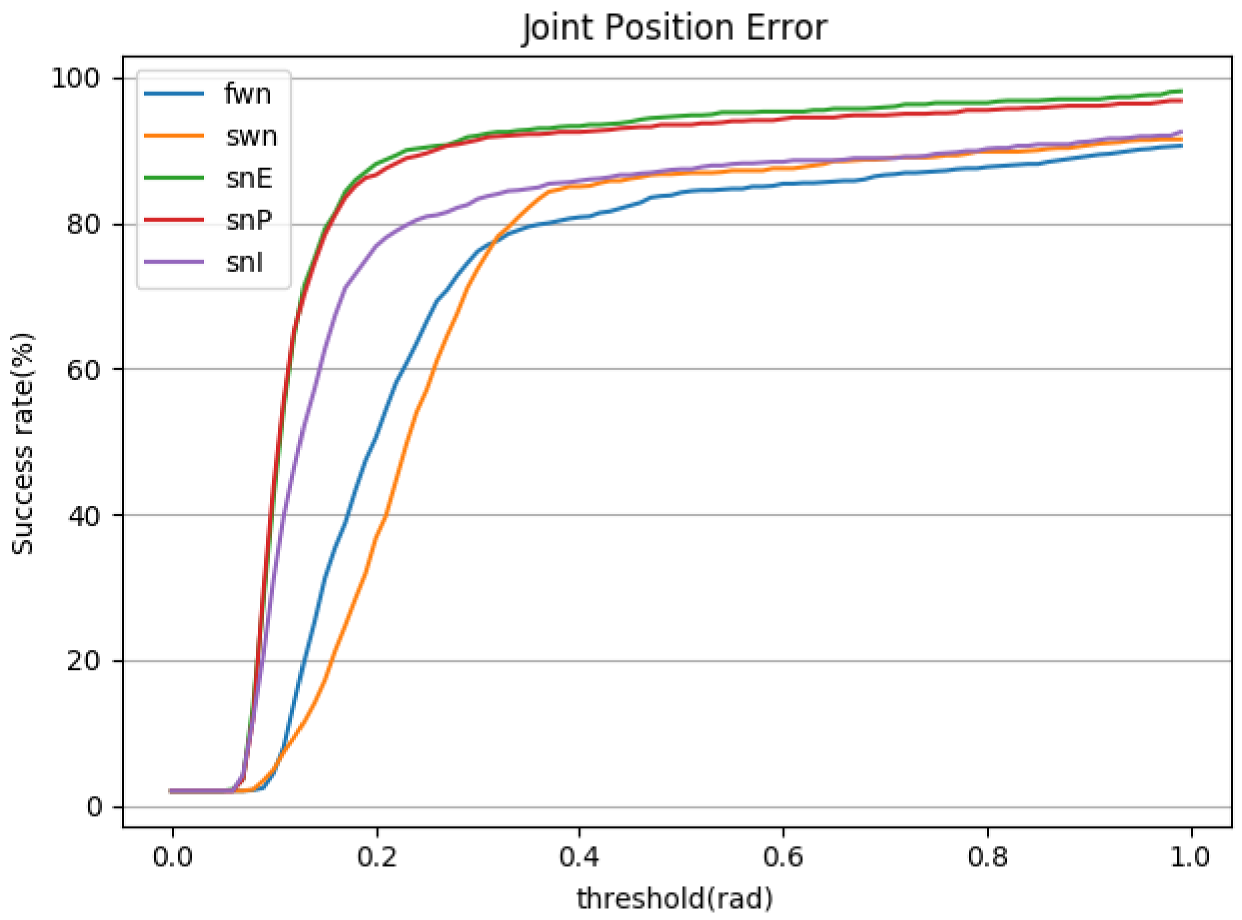
ETH zürich

Q&A

appendix

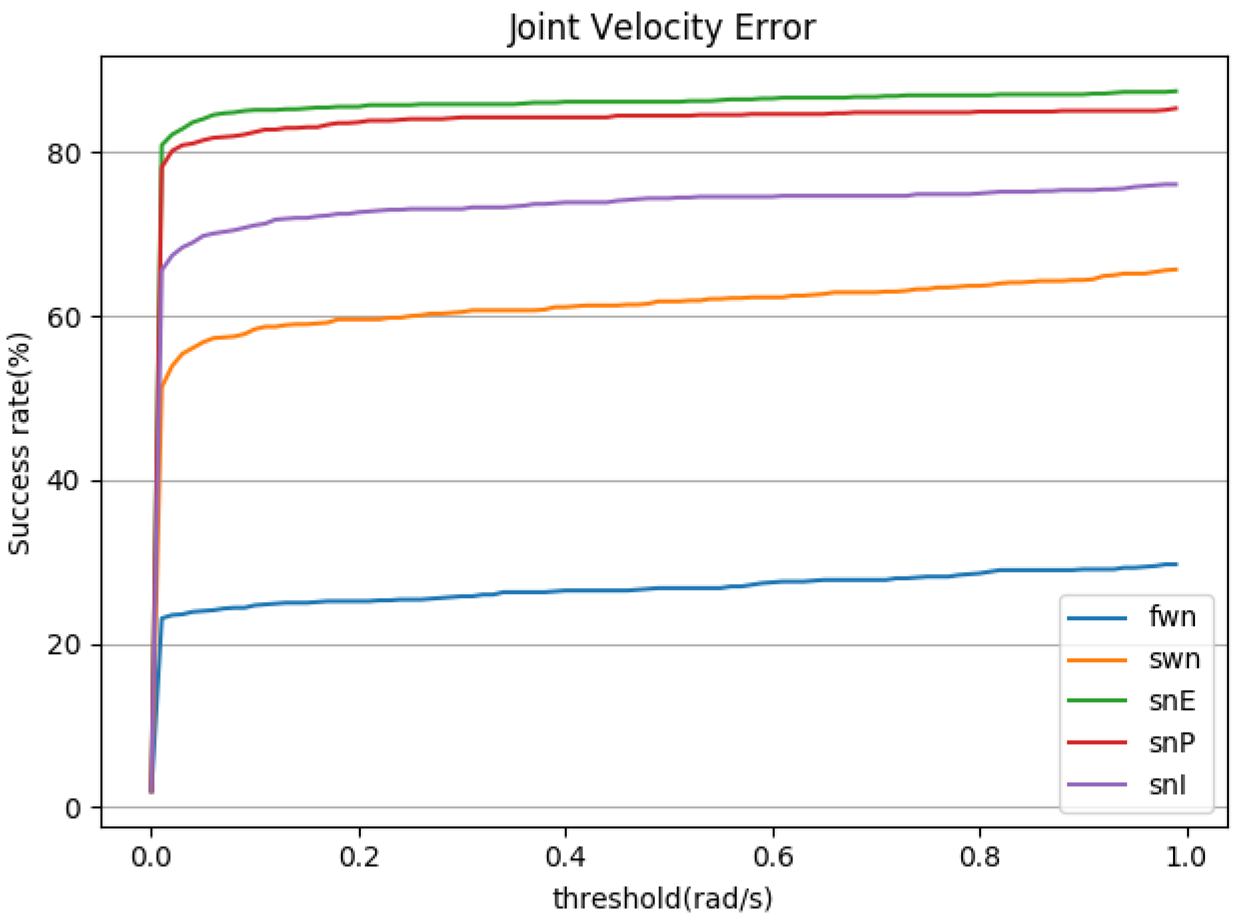

validation loss
history
tanh
relu
loss
0.0350
0.0325
0.0300
0.0275
0.0250
0.0225
0.0200
0.0175
0
20
40
60
80
100
epoch

appendix

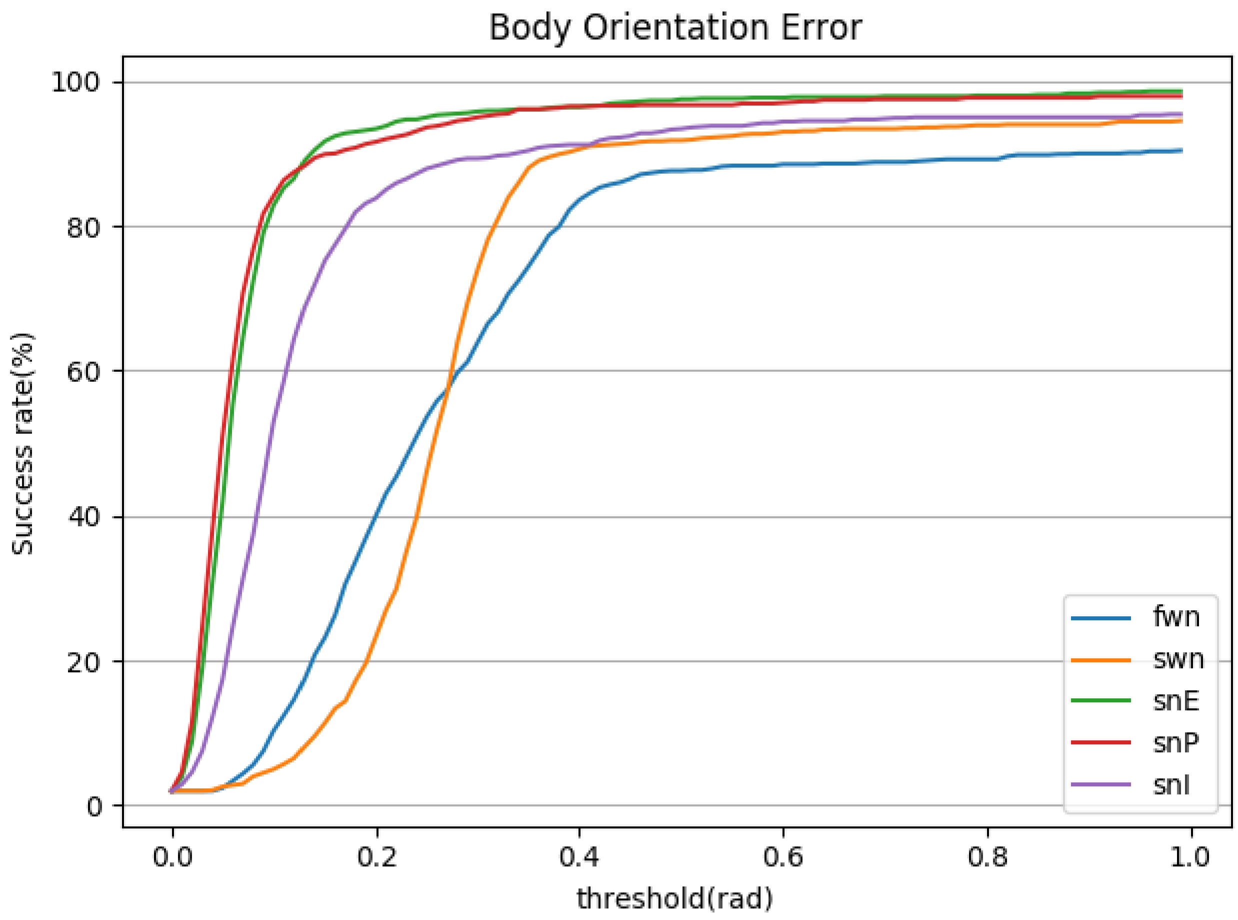


Slope : 36 degrees

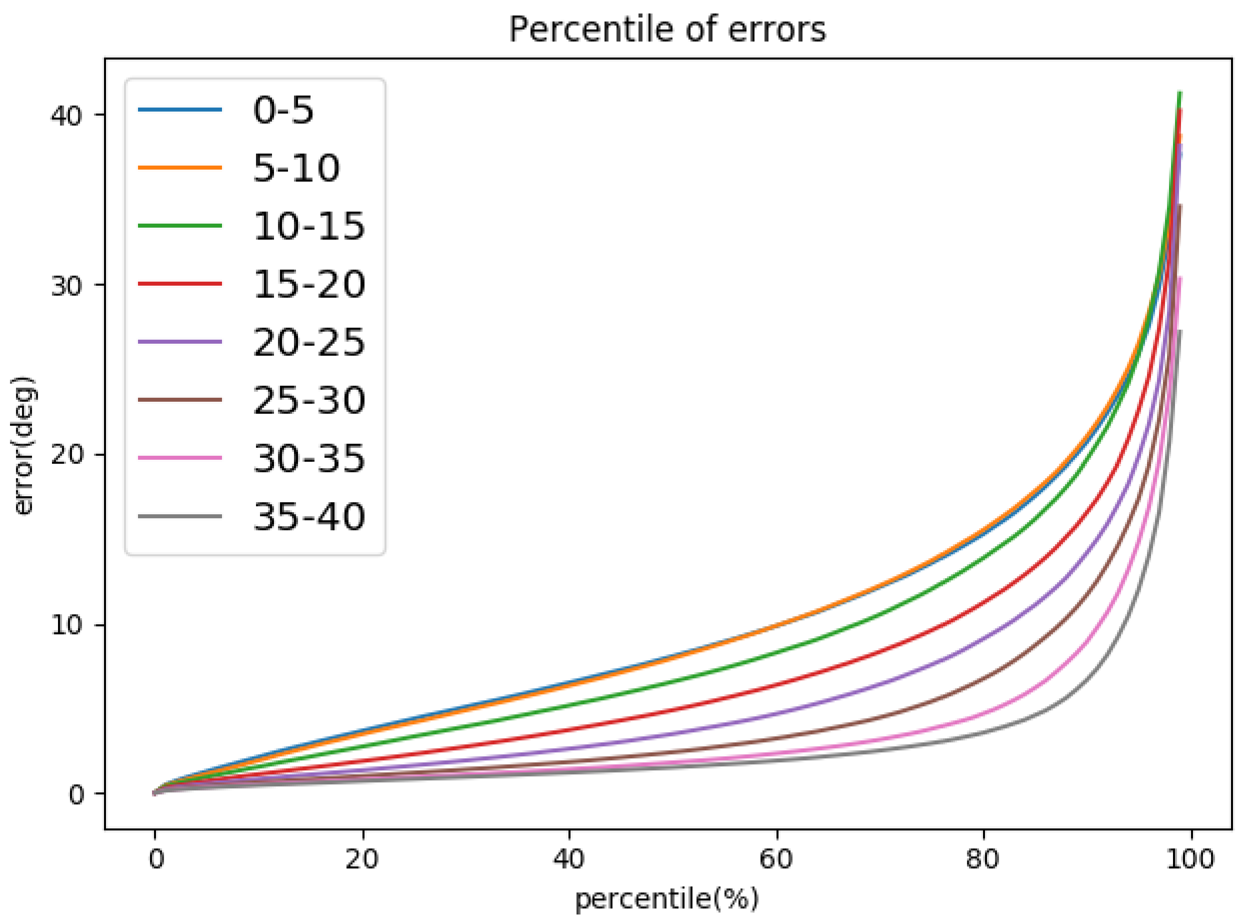
appendix


Joint Velocity Error
Success rate(%)
threshold(rad/s)
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
fwn
swn
snE
snP
snI

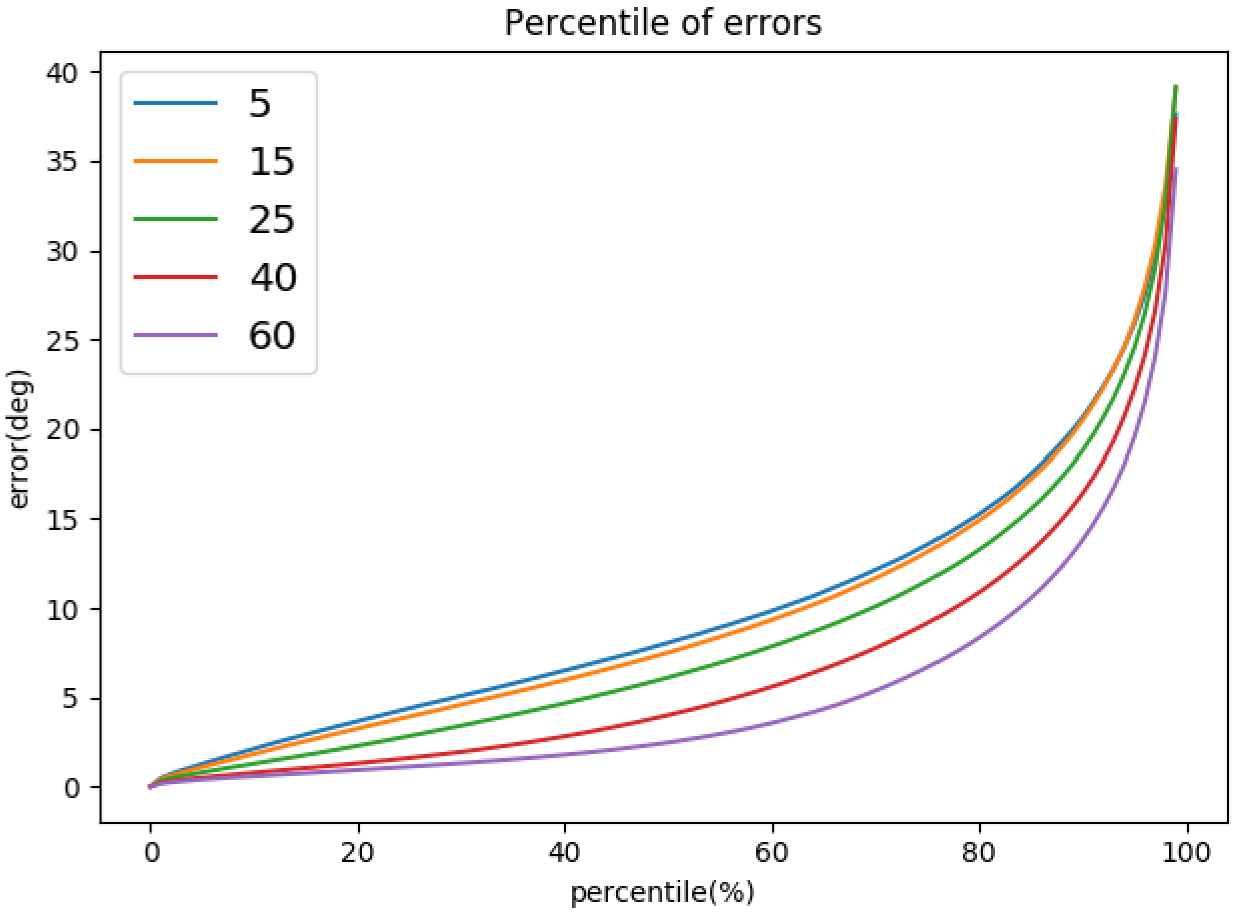
appendix


Body Orientation Error
Success rate(%)
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0
threshold(rad)
fwn
swn
snE
snP
snI

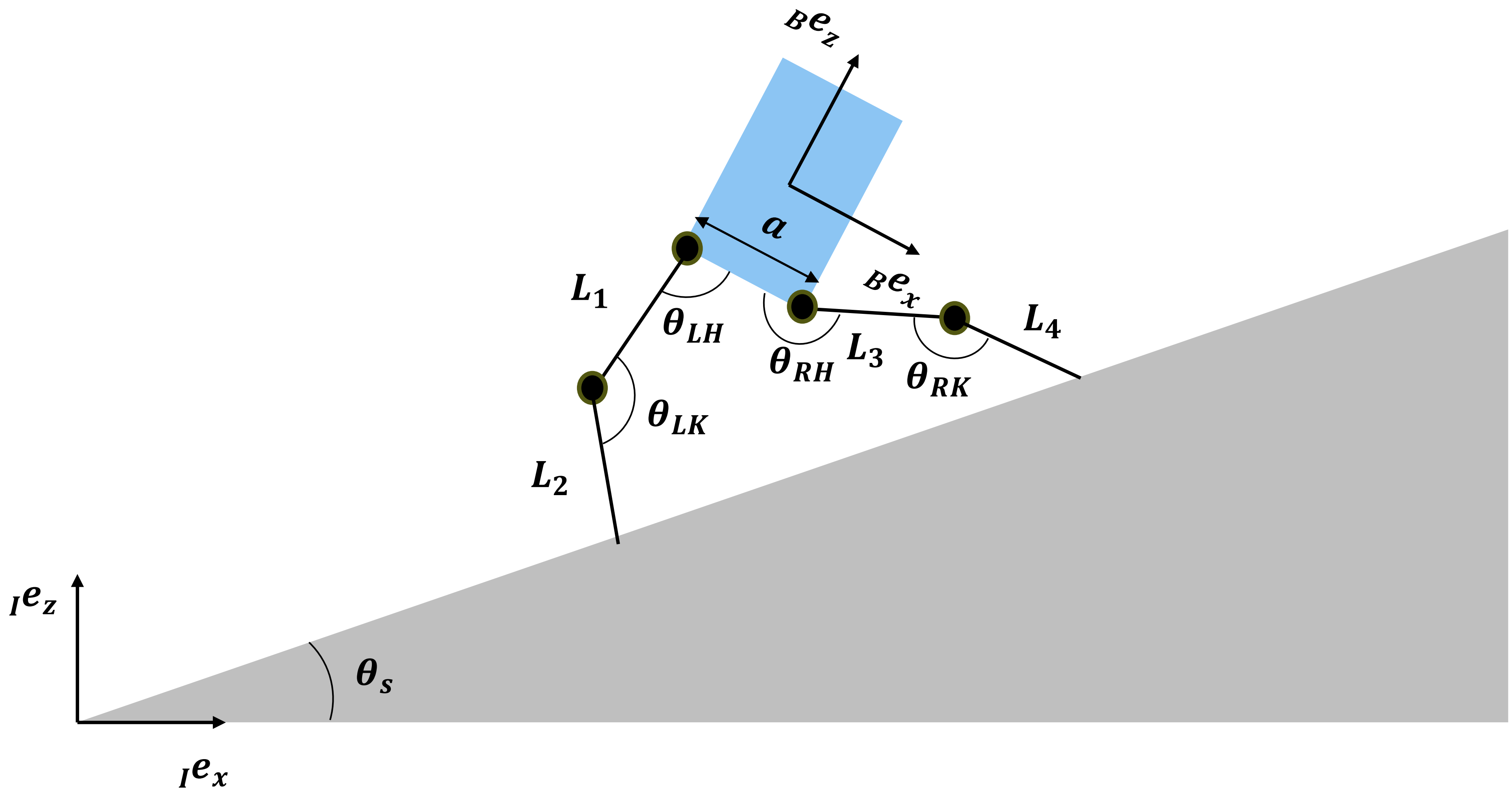
appendix


Percentile of errors
error(deg)
percentile(%)
0
10
20
30
40
0
20
40
60
80
100
0-5
5-10
10-15
15-20
20-25
25-30
30-35
35-40

appendix


Percentile of errors
error(deg)
percentile(%)
5
15
25
40
60
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100

Normal estimation



Normal estimation

$$\theta_{slope} = \alpha + \theta_{RH} - \theta_g - \frac{\pi}{4}$$

$${}_B n = C_{BI}\ {}_I n = \begin{pmatrix} cos\theta_g & -sin\theta_g \\ sin\theta_g & cos\theta_g \end{pmatrix}\ {}_I n$$

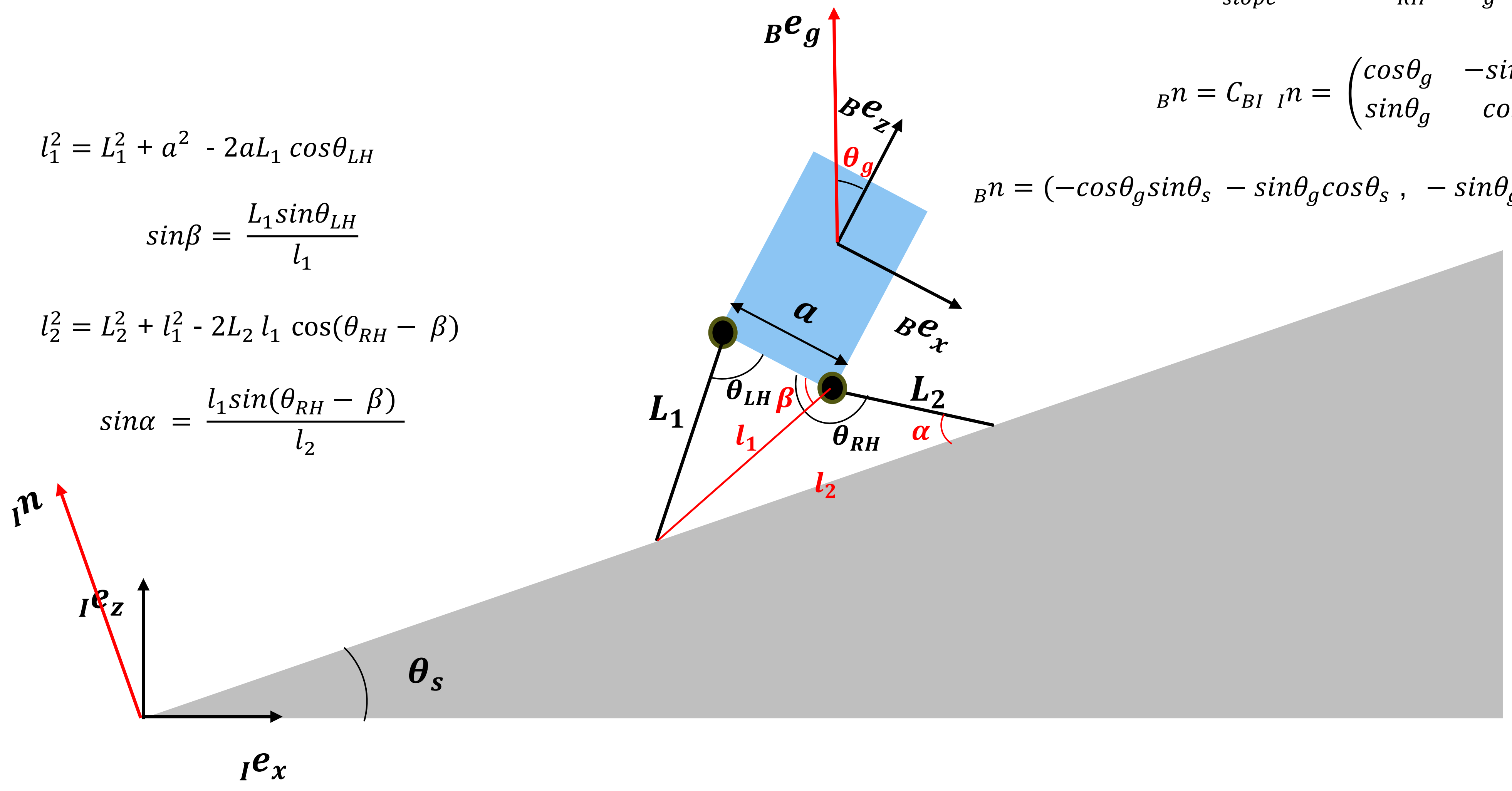
$${}_B n = (-cos\theta_g sin\theta_s\ - sin\theta_g cos\theta_s\ ,\ - sin\theta_g sin\theta_s + cos\theta_g cos\theta_s)^T$$

$$l_1^2 = L_1^2 + a^2\ \text{-}\ 2aL_1\ cos\theta_{LH}$$

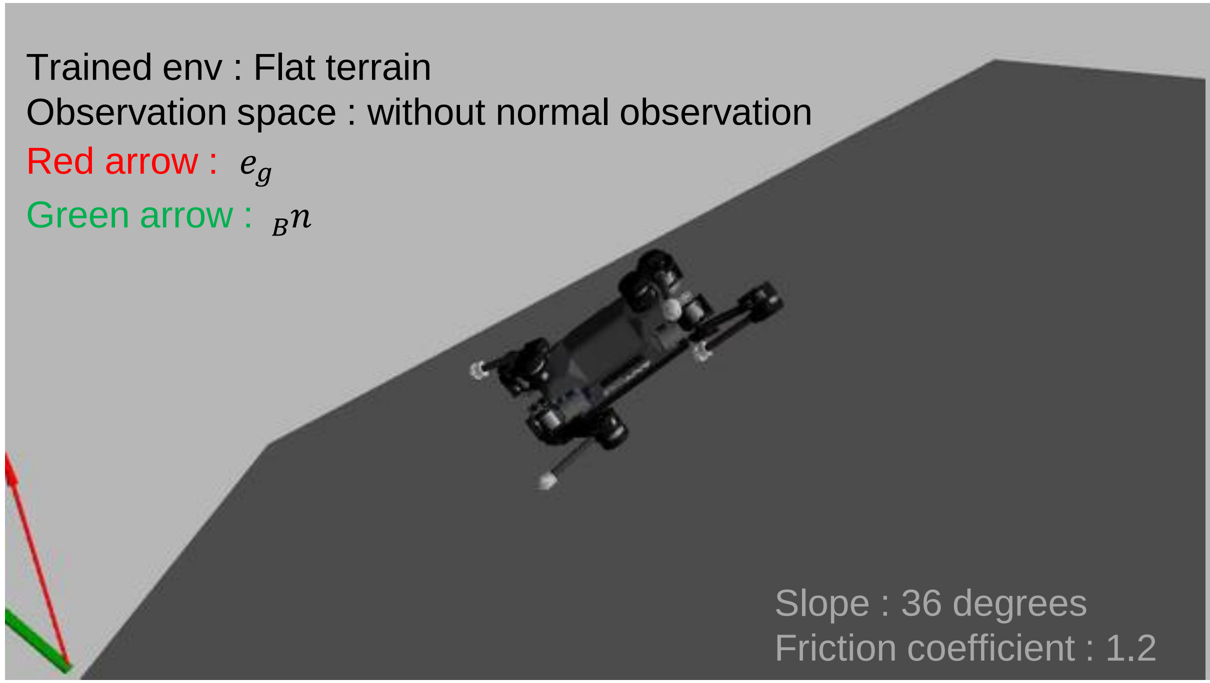
$$sin\beta = \frac{L_1 sin\theta_{LH}}{l_1}$$

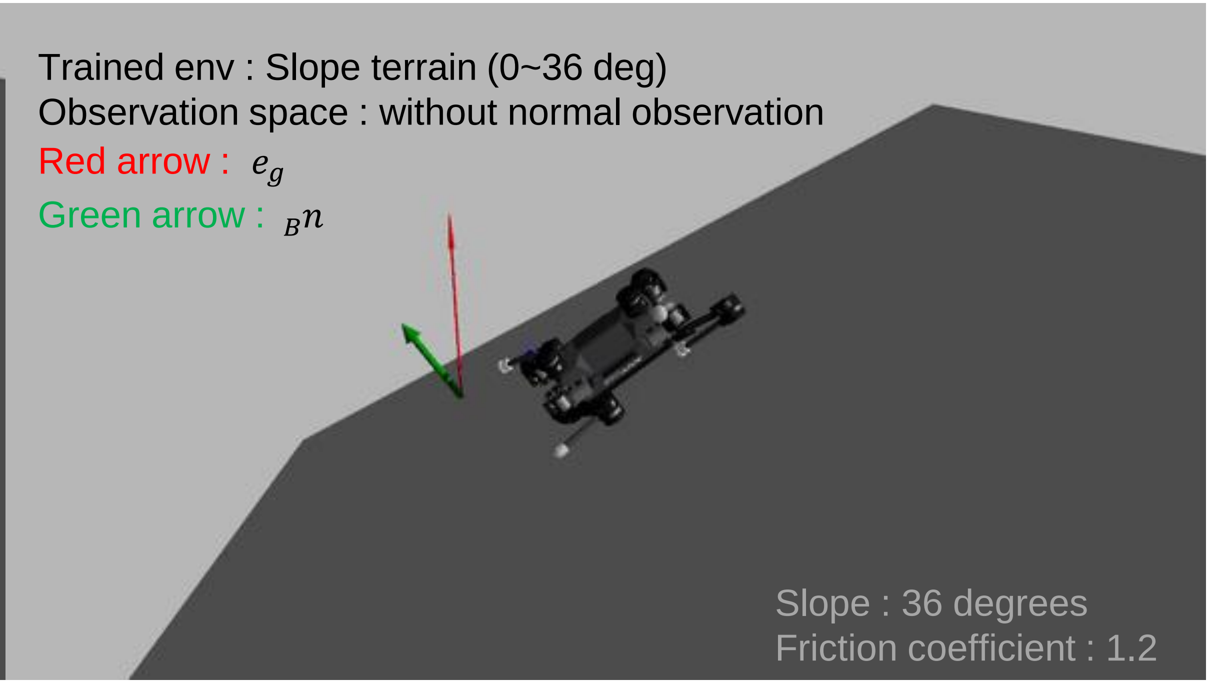
$$l_2^2 = L_2^2 + l_1^2\ \text{-}\ 2L_2\ l_1\ \cos(\theta_{RH} - \beta)$$

$$sin\alpha\ = \frac{l_1 sin(\theta_{RH} - \beta)}{l_2}$$



Policy without normal observation

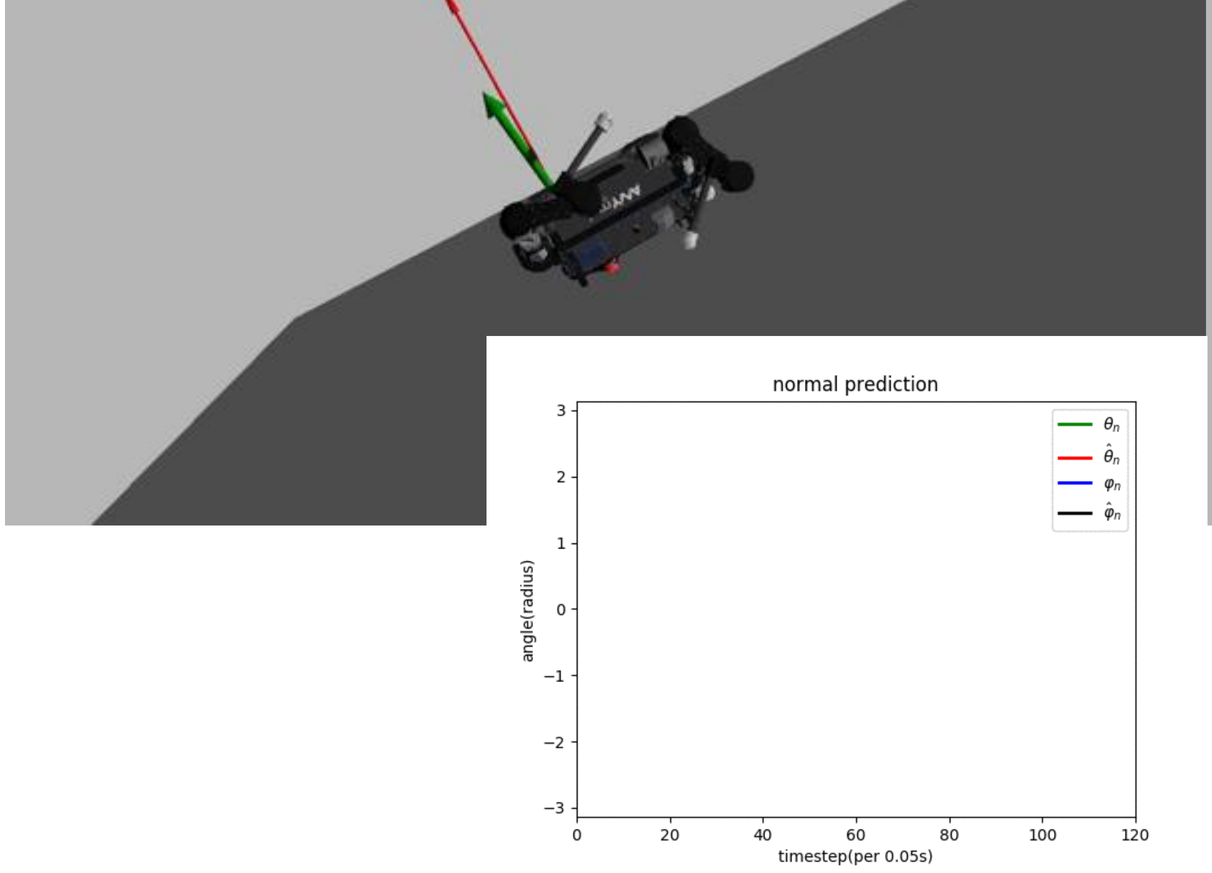




Estimation error comparison

Estimator Output : ${}_B\hat{\varphi}_{\hat{n}}$, ${}_B\hat{\theta}_{\hat{n}}$

Slope : 36 degrees



Estimator Output : $\cos {}_B\hat{\varphi}_{\hat{n}}$, $\sin {}_B\hat{\varphi}_{\hat{n}}$, $\cos {}_B\hat{\theta}_{\hat{n}}$, $\sin {}_B\hat{\theta}_{\hat{n}}$

Slope : 36 degrees