DEPARTMENT OF MECHANICAL AND ENERGY ENGINEERING
机械与能源工程系

CLEAR
Control & Learning for Robotics and Autonomy Lab

Deep RL for MPC of Quadruped Locomotion

By Yulun Zhuang
SUSTech
Professor Wei Zhang

Content

01

Thesis Motivation and Demo

Quadruped Locomotion
Control and Learning Framework
Compatible with Multi-Type Robots

02

Model Predictive Control FWK

MIT Cheetah Software in Python
Easy to Use MPC Controller
Mobility across Slopes and Steps

03

RL Training FWK

Deep RL through Parallelism Training
Combine RL Policy and MPC Controller

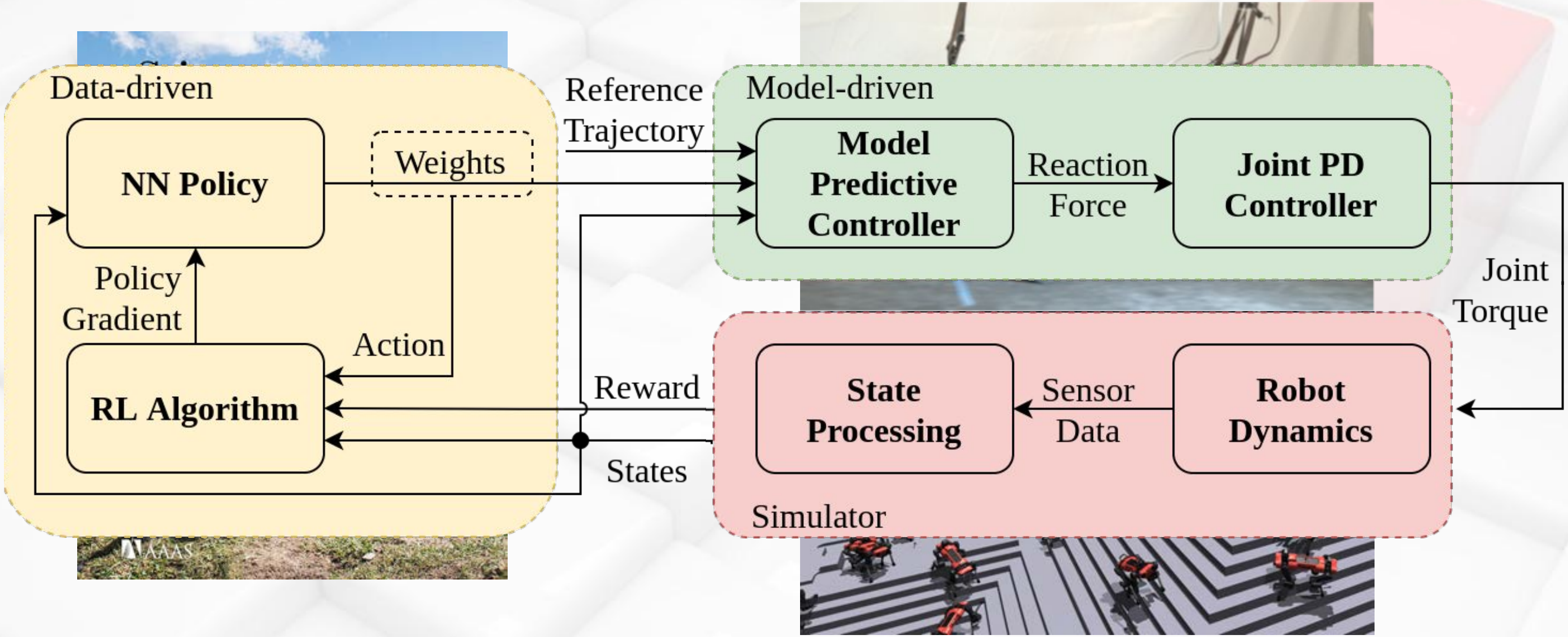
04

Experiments and Results

Inconsistent update timesteps
Inconsistent loss functions
Unavailable sim-to-real transfer

Motivation and Demonstration

Thesis Motivation

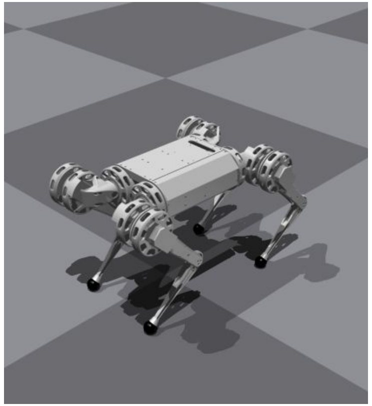


★ A Model-Data Hybrid-Driven Hierarchical Control Method for Quadruped

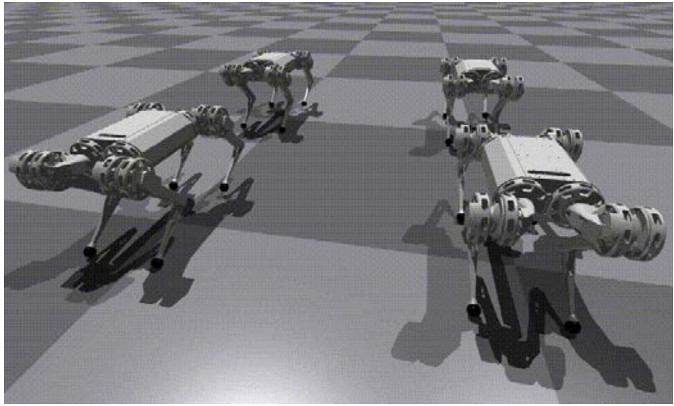
A Quadruped Control & Learning FWK

Unitree Aliengo

Unitree A1

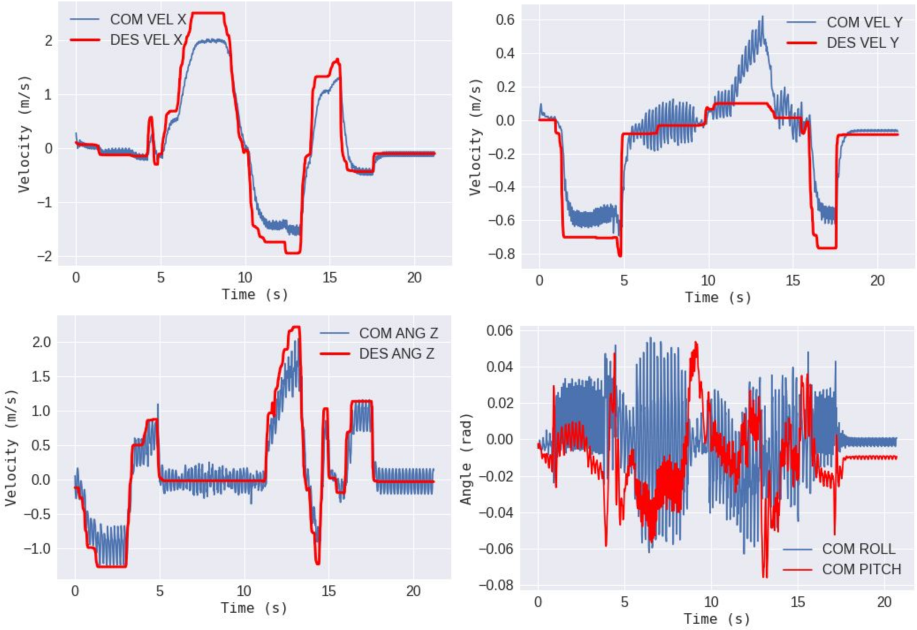


Mini Cheetah



Parallel Simulation Demo

MPC Controller Performance



RL Training Results - Part 1

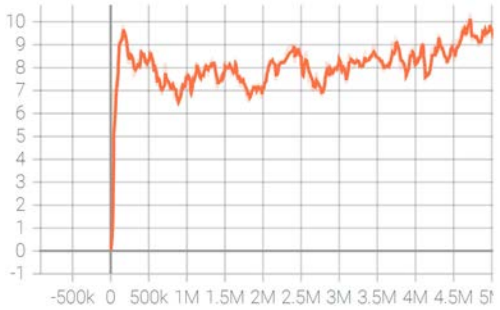
❖ Computing Device

RL Training: NVIDIA GTX 1060

MPC Controller: Intel Core i7-7850H

RAM: 16 G

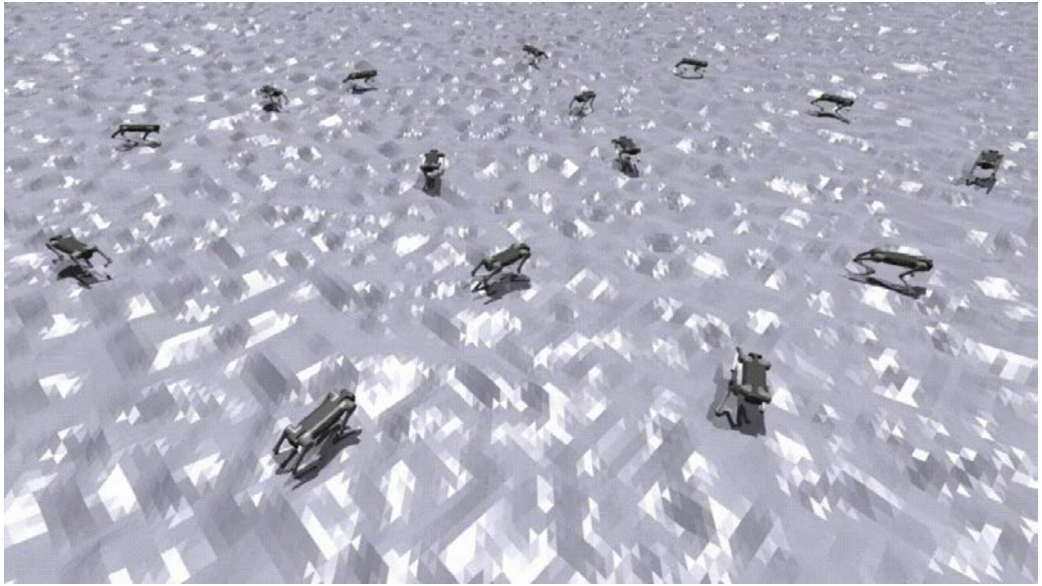
Envs	Batch Size	GPU Occupancy
1	24	35%
4	96	28%
8	192	20%
16	384	**15%**
32	768	8%
64	1536	6%
512	12288	1%



Reward

Entropy Loss

RL Training Results - Part 2

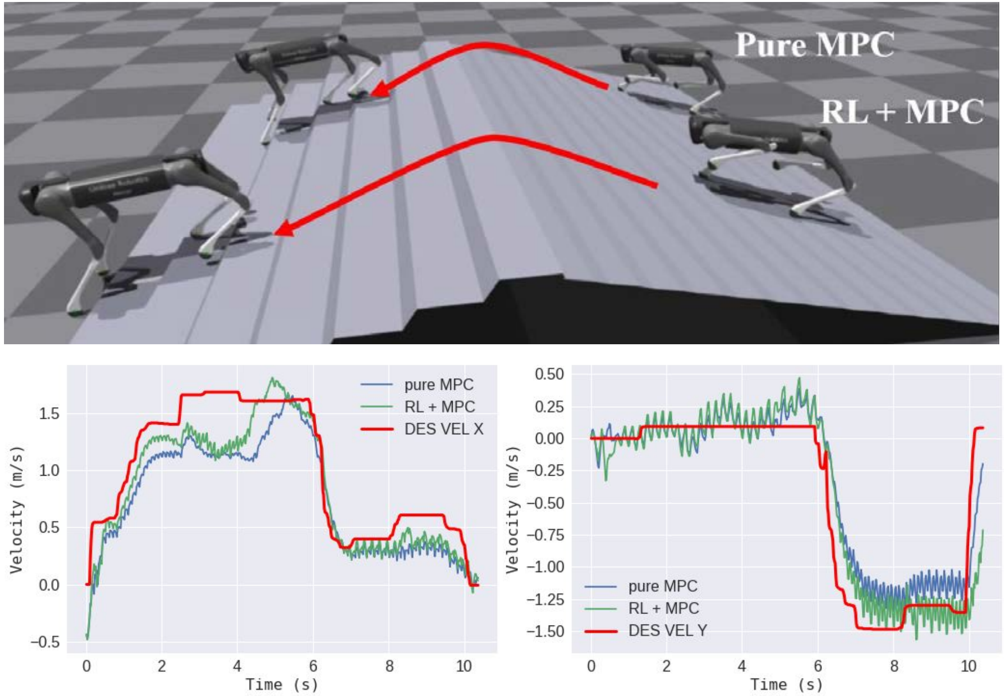


Weight Policy Training

Linear velocity:	[-2, 2]
Angular velocity:	[-1.5, 1.5]
Simulation:	500 Hz
MPC:	50 Hz

Actors:	16
Algorithm:	PPO
Policy:	3 layer MLP
Action clip:	[-1, 1]

Performance Comparison - Part 1



Linear velocity: [-2, 2]

Angular velocity: [-2.5, 2.5]

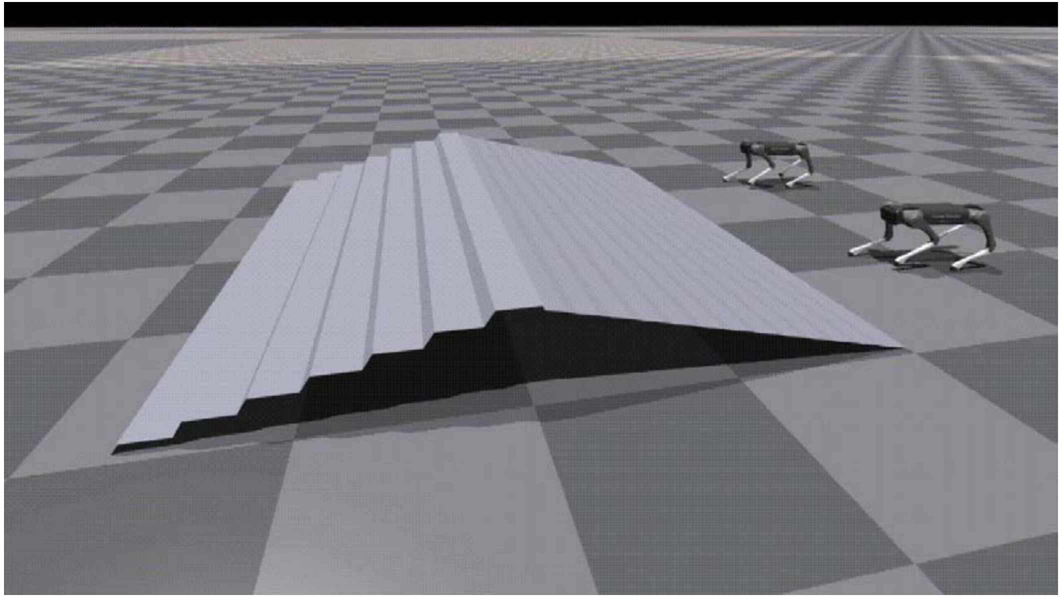
Simulation: 500 Hz

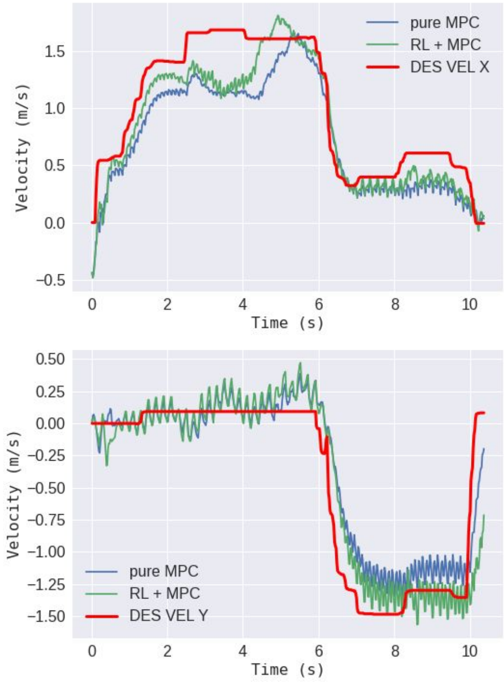
MPC: 50 Hz

Model inference time: ~0.001 s

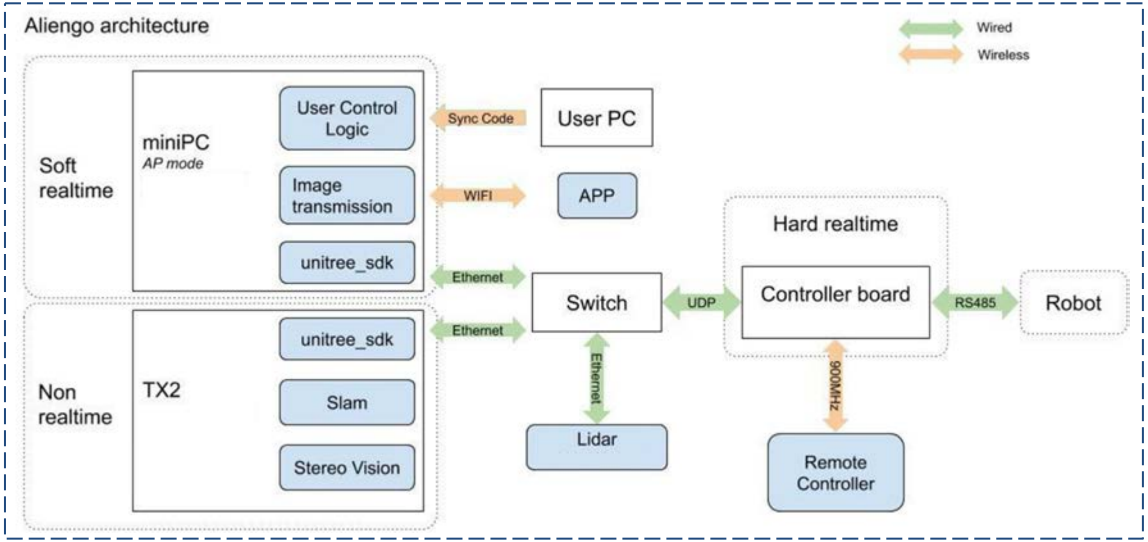
Policy update time: ~0.0015 s

Performance Comparison - Part 2

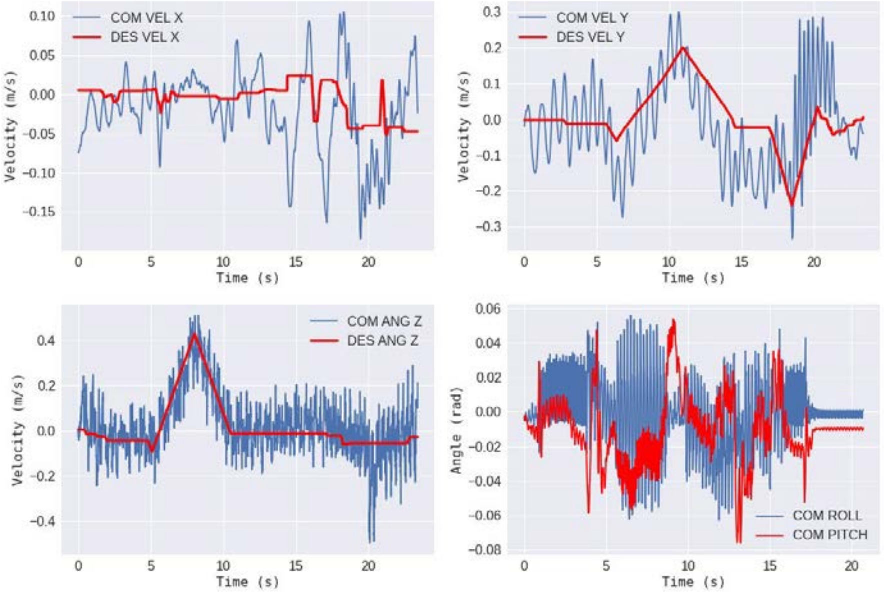




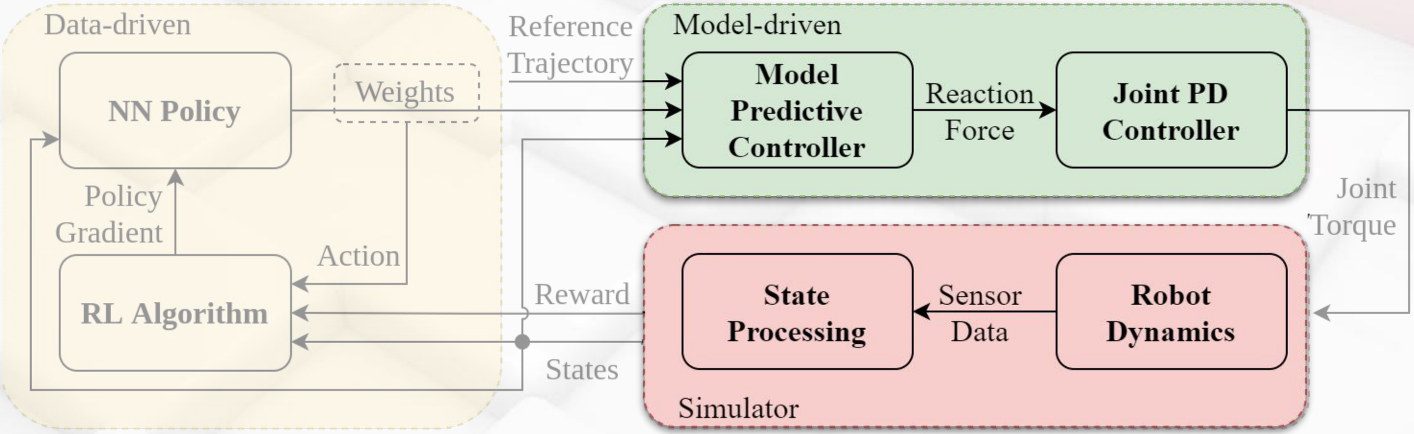
Sim2Real Transfer - Part 1



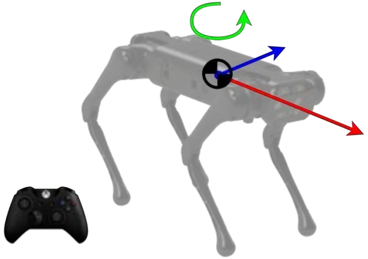
Sim2Real Transfer - Part 2

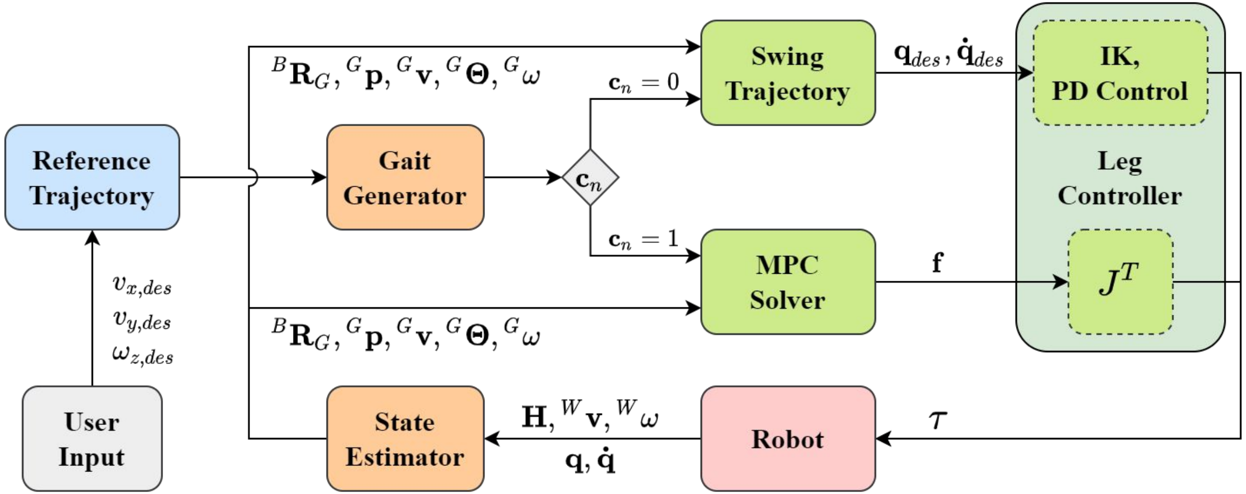


Model Predictive Control Framework



Controller Overview





- ❖ Operator Input:
  - ➢ velocity in xy plane and yaw turn rate.

Di Carlo, Jared, et al. "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control." *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018.

Rigid-body Dynamics

❖ Rigid body dynamics:

$$\ddot{\mathbf{p}} = \frac{\sum_{i=1}^{n} \mathbf{f}_i}{m} - \mathbf{g}$$

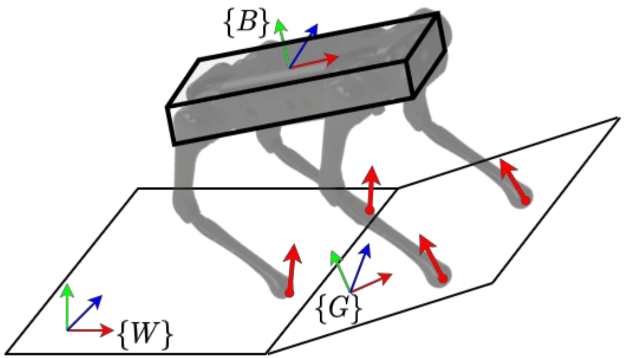
$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{I}\boldsymbol{\omega}) = \sum_{i=1}^{n} \mathbf{r}_i \times \mathbf{f}_i$$

$$\dot{\mathbf{R}} = [\boldsymbol{\omega}]_\times \mathbf{R}$$

where

$$\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{I}\boldsymbol{\omega}) = \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \approx \mathbf{I}\dot{\boldsymbol{\omega}}$$



❖ Approximations:

1. Small value of roll and pitch
2. Robot is not pointed vertically ($\cos(\theta) \neq 0$)
3. The $\omega \times (\mathbf{I}\omega)$ term is small for bodies with small angular velocities

State Space Model

- Calculate state derivatives:

$$\frac{\mathrm{d}}{\mathrm{dt}}\hat{\mathbf{\Theta}} = R_z(\psi)\hat{\mathbf{\Theta}}$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\hat{\mathbf{p}} = \dot{\hat{\mathbf{p}}}$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\hat{\omega} = \hat{\mathbf{I}}^{-1}\sum_{i=1}^{n}\mathbf{r}_i \times \mathbf{f}_i = \hat{\mathbf{I}}^{-1}([\mathbf{r}_1]_\times \mathbf{f}_1 + \cdots + [\mathbf{r}_n]_\times \mathbf{f}_n)$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\dot{\hat{\mathbf{p}}} = \frac{\sum_{i=1}^{n}\mathbf{f}_i}{m} - \mathbf{g} = \frac{\mathbf{f}_1 + \cdots + \mathbf{f}_n}{m} - \mathbf{g}$$

where $n = 4$

- Add additional gravity state:

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix}\hat{\mathbf{\Theta}}\\ \hat{\mathbf{p}}\\ \hat{\omega}\\ \dot{\hat{\mathbf{p}}}\\ \mathbf{g}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 & \mathbf{0}_{3\times1}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times1}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times1}\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & [0\ 0\ 1]^T\\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0\end{bmatrix}\begin{bmatrix}\hat{\mathbf{\Theta}}\\ \hat{\mathbf{p}}\\ \hat{\omega}\\ \dot{\hat{\mathbf{p}}}\\ \mathbf{g}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_3 & \cdots & \mathbf{0}_3\\ \mathbf{0}_3 & \cdots & \mathbf{0}_3\\ \hat{\mathbf{I}}^{-1}[\mathbf{r}_1]_\times & \cdots & \hat{\mathbf{I}}^{-1}[\mathbf{r}_4]_\times\\ \mathbf{I}_3/m & \cdots & \mathbf{I}_3/m\\ \mathbf{0}_{1\times3} & \cdots & \mathbf{0}_{1\times3}\end{bmatrix}\begin{bmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_4\end{bmatrix}$$

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c(\psi)\mathbf{x}(t) + \mathbf{B}_c(\mathbf{r}_1, \ldots, \mathbf{r}_n, \psi)\mathbf{u}(t)$$

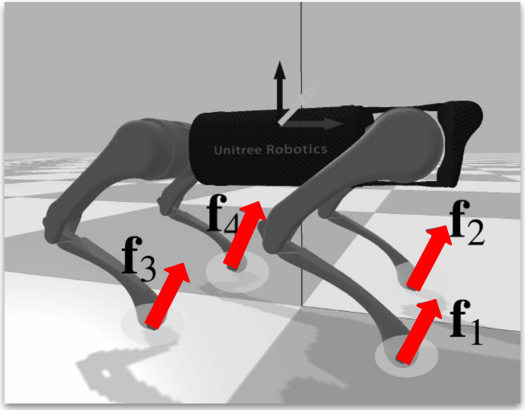
where $\mathbf{A}_c \in \mathbb{R}^{13\times13}$ and $\mathbf{B}_c \in \mathbb{R}^{13\times3n}$.

Stance Leg Control

- Stance Leg Force Control:

$$\boldsymbol{\tau}_i = \mathbf{J}_i^\top \mathbf{R}_i^\top \mathbf{f}_i$$

f is the vector of forces calculated from the **MPC** controller in the world coordinate frame.



- Zero Order Hold Discretization

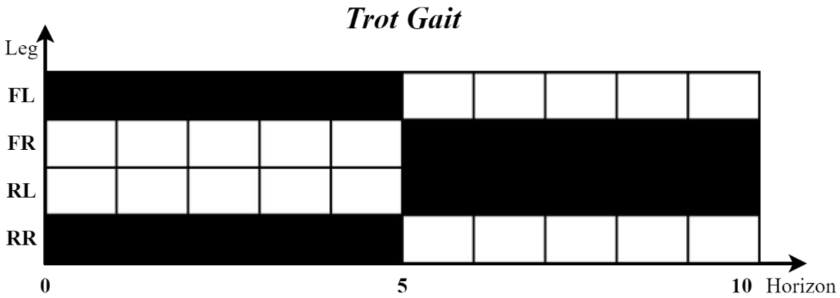
$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}_d \mathbf{x}_k + \mathbf{B}_d \mathbf{u}_k \\
\mathbf{A}_d &= e^{\mathbf{A}T} \\
\mathbf{B}_d &= \int_0^T e^{\mathbf{A}\tau}\, d\tau \mathbf{B}
\end{aligned}$$

MPC Formulation

❖ Convex Model Predictive Control Problem

$$\min_{\mathbf{x},\mathbf{u}} \quad \sum_{i=0}^{k-1} ||\mathbf{x}_{i+1} - \mathbf{x}_{i+1,\text{ref}}||_{\mathbf{Q}_i} + ||\mathbf{u}_i||_{\mathbf{R}_i}$$

$$\begin{aligned} \text{subject to} \quad \mathbf{x}_{i+1} &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i, i = 0 \ldots k-1 \\ \underline{\mathbf{c}}_i \leq \mathbf{C}_i\mathbf{u}_i &\leq \overline{\mathbf{c}}_i, i = 0 \ldots k-1 \\ \mathbf{D}_i\mathbf{u}_i &= 0, i = 0 \ldots k-1 \end{aligned}$$

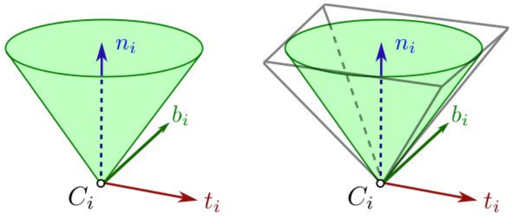


❖ Constraints

➢ Equality constraints: set all forces from feet off the ground to zero, enforcing the desired gait.

➢ Inequality constraints: $f_{\min} \leq f_z \leq f_{\max}$

- max z-force $-\mu f_z \leq \pm f_x \leq \mu f_z$
- friction cone $-\mu f_z \leq \pm f_y \leq \mu f_z$



Di Carlo, Jared, et al. "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control." *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018.

QP Formulation

❖ Batch Formulation

$$\mathbf{x_k} = \mathbf{A}^k\mathbf{x}_0 + \sum_{i=0}^{k-1} \mathbf{A}^{k-1-i}\mathbf{B}\mathbf{u}_i$$

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}^1 \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^k \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{k-1}\mathbf{B} & \mathbf{A}^{k-1}\mathbf{B} & \mathbf{0} & \cdots & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{k-1} \end{bmatrix}$$

Therefore,

$$\mathrm{X} = \mathrm{A}_{\mathrm{qp}}\mathrm{x}_0 + \mathrm{B}_{\mathrm{qp}}\mathrm{U}$$

where $\mathrm{X} \in \mathbb{R}^{13k}$ and $\mathrm{U} \in \mathbb{R}^{3nk}$

❖ Rewrite the objective function

$$J(\mathrm{U}) = ||\mathrm{A}_{\mathrm{qp}}\mathrm{x}_0 + \mathrm{B}_{\mathrm{qp}}\mathrm{U} - \mathrm{x}_{\mathrm{ref}}||_{\mathrm{L}} + ||\mathrm{U}||_{\mathrm{K}}$$

❖ Convert to convex QP

$$\begin{array}{ll} \min_{\mathrm{U}} & \frac{1}{2}\mathrm{U}^\top \mathbf{H}\mathrm{U} + \mathrm{U}^\top \mathbf{g} \\ \text{s. t.} & \underline{\mathbf{c}} \leq \mathbf{C}\mathrm{U} \leq \overline{\mathbf{c}} \end{array}$$

where

$$\mathbf{H} = 2(\mathrm{B}_{\mathrm{qp}}^\top \mathrm{L}\mathrm{B}_{\mathrm{qp}} + \mathrm{K})$$

$$\mathbf{g} = 2\mathrm{B}_{\mathrm{qp}}^\top \mathrm{L}(\mathrm{A}_{\mathrm{qp}}\mathrm{x}_0 - \mathbf{y})$$

MPC Solver Implementation

QP Solver	Language	Binding	Run Time	Total Time	Choice
CVXOPT	Python	N/A	100 ms	N/A	❌
OSQP	Python	N/A	10 ms	15 ms	❌
OSQP-Eigen	C++	pybind11	1.33 ms	28 ms	❌
qpOASES	C	pybind11	< 1 ms	< 2 ms	✔

Refer to MIT's Implmnt (CVXOPT, OSQP, OSQP-Eigen)

Refer to Yuxiang's Implmnt → qpOASES

Di Carlo, Jared, et al. "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control." *2018 IROS*. IEEE, 2018.
Yang, Yuxiang, et al. "Fast and efficient locomotion via learned gait transitions." Conference on Robot Learning. PMLR, 2022.

Swing Leg Control

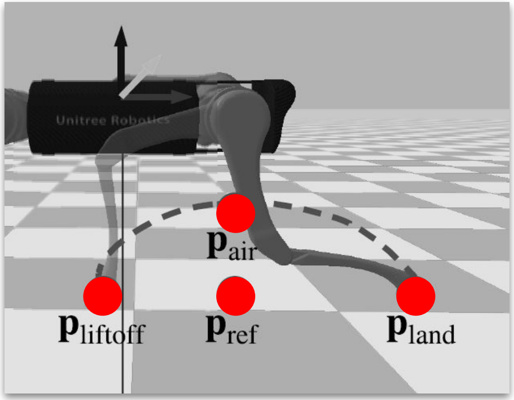
❖ PD Feedback Control:

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T \left[\mathbf{K}_p \left({}^H\mathbf{p}_{i,ref} - {}^H\mathbf{p}_i\right) + \mathbf{K}_d \left({}^H\mathbf{v}_{i,ref} - {}^H\mathbf{v}_i\right)\right]$$

It plans a trajectory using a **cubic Bezier spline**, then uses position feedback control for tracking.

❖ Foot Placement – Raibert heuristic

$$p_{\text{land}} = p_{\text{ref}} + v_{\text{CoM}} T_{\text{stance}} / 2$$

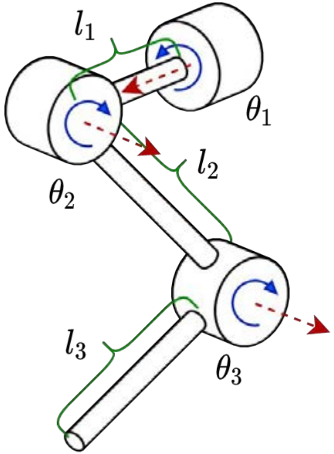


Single Leg Kinematics

- Homogeneous Transformation Matrix

$$\mathbf{H} = Rot_{x,\theta_1} Trans_{y,l_1} Rot_{y,\theta_2} Trans_{z,l_2} Rot_{y,\theta_3} Trans_{z,l_3}$$

$$= \begin{bmatrix} c_{23} & 0 & s_{23} & l_3 s_{23} + l_2 s_2 \\ -s_{23}s_1 & -c_1 & c_{23}s_1 & l_1 c_1 + l_3 s_1 c_{23} + l_2 c_2 s_1 \\ s_{23}c_1 & -s_1 & -c_{23}c_1 & l_1 s_1 - l_3 c_1 c_{23} - l_2 c_2 c_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- Forward Kinematics

$$\mathbf{p} = \begin{bmatrix} l_3 s_{23} + l_2 s_2 \\ l_1 c_1 + l_3 s_1 c_{23} + l_2 c_2 s_1 \\ l_1 s_1 - l_3 c_1 c_{23} - l_2 c_2 c_1 \end{bmatrix}$$

- Velocity Jacobian Matrix

$$\mathbf{v} = \dot{\mathbf{p}} = J\dot{\boldsymbol{\theta}}$$

$$J = \begin{bmatrix} 0 & l_3 c_{23} + l_2 c_2 & l_3 c_{23} \\ -l_1 s_1 + l_3 c_1 c_{23} + l_2 c_2 c_1 & -l_3 s_1 s_{23} - l_2 s_2 s_1 & -l_3 s_1 s_{23} \\ l_1 c_1 + l_3 s_1 c_{23} l_2 c_2 s_1 & l_3 c_1 s_{23} + l_2 s_2 c_1 & l_3 c_1 s_{23} \end{bmatrix}$$

Ground Slope Estimation

- Use measurements of each footstep location:

$$\boldsymbol{p}_i = (p_i^x, p_i^y, p_i^z)$$

- To approximate the local slope of the walking surface:

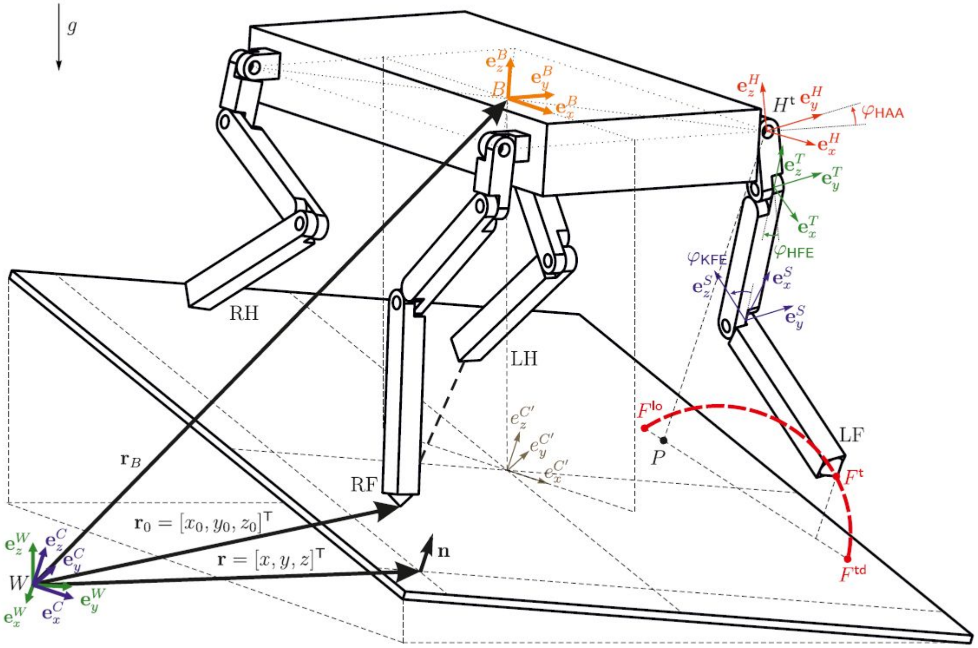
$$z(x, y) = a_0 + a_1 x + a_2 y$$

- Coefficients $\boldsymbol{a}$ are obtained through least squares:

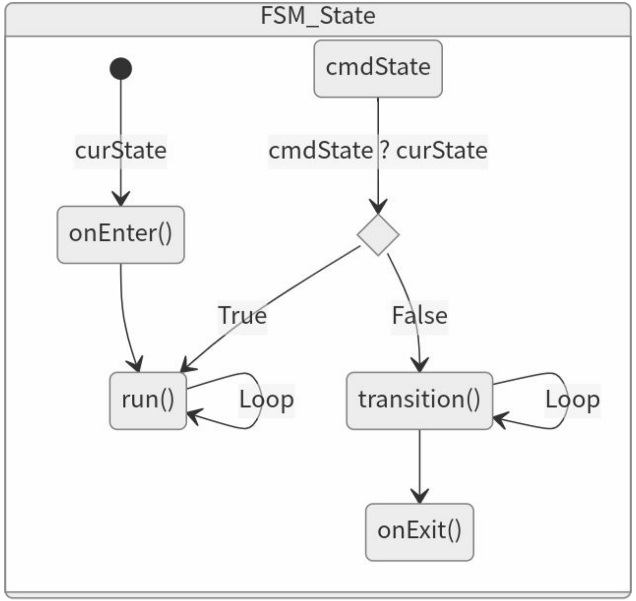
$$\boldsymbol{a} = \left(\boldsymbol{W}^T \boldsymbol{W}\right)^{\dagger} \boldsymbol{W}^T \boldsymbol{p}^z$$

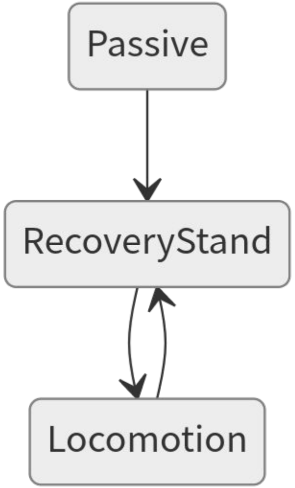
$$\boldsymbol{W} = \begin{bmatrix} \mathbf{1} & \boldsymbol{p}^x & \boldsymbol{p}^y \end{bmatrix}_{4\times 3}$$

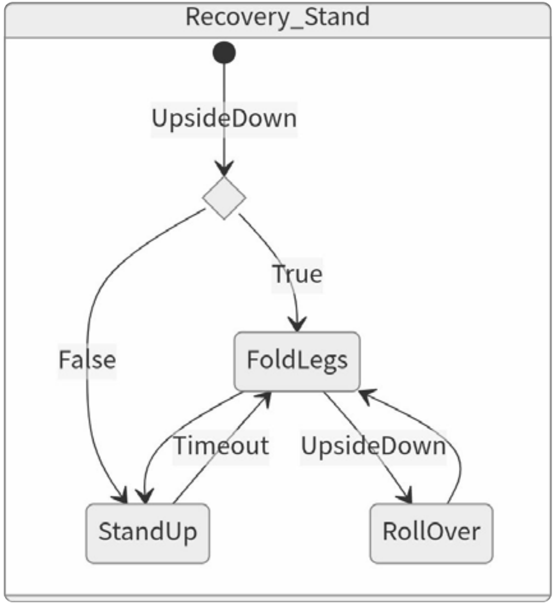
where $\boldsymbol{p}^x = (p_1^x, p_2^x, p_3^x, p_4^x)$



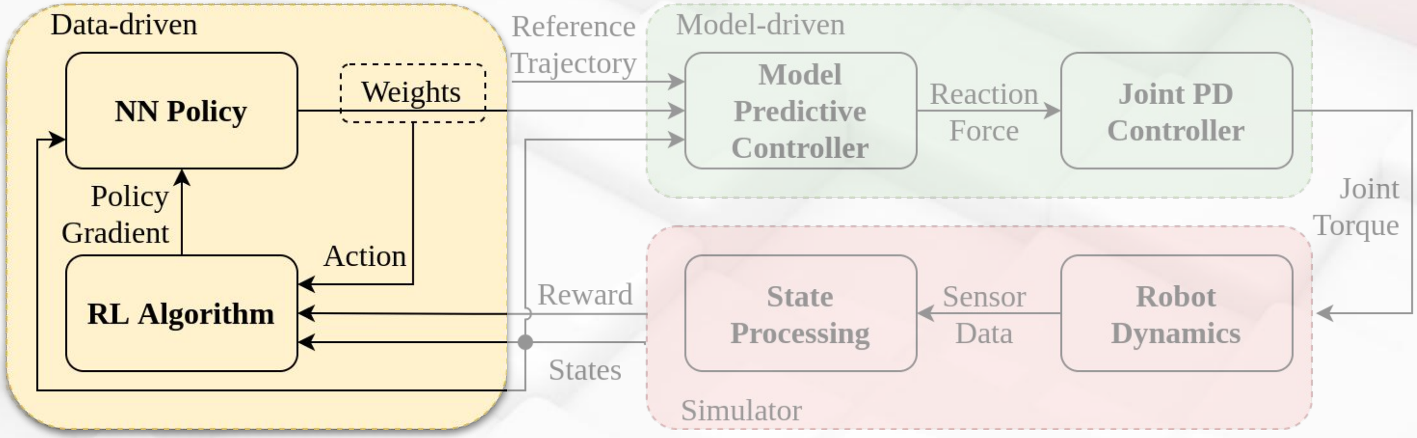
Finite State Machine







Deep RL Training Framework



RL Framework

Simulator: NVIDIA Isaac Gym

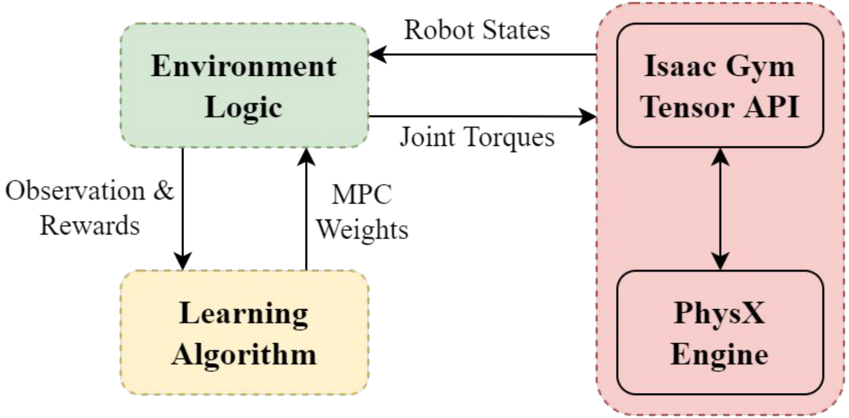
RL Algorithm: PPO (A-C Style)

Policy Network: 3-Layer MLP

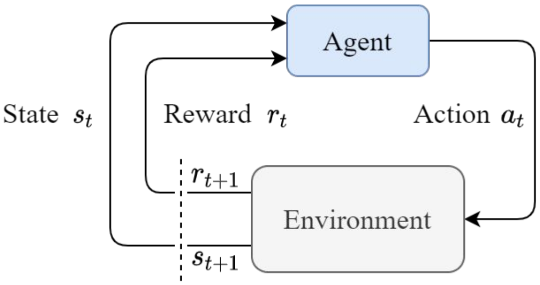
Device: NVIDIA GTX 1060

Parallelism:

- ➢ $B = n_{\text{robots}} n_{\text{steps}}$ (trade off between efficiency and performance)
- ➢ Reset based on a time out breaks the infinite horizon assumption of PPO



RL Algorithm



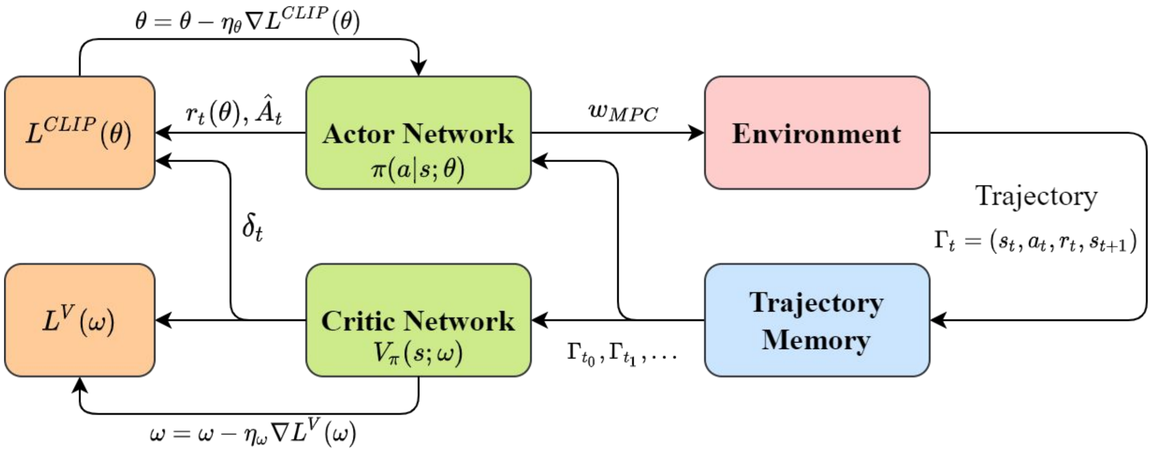
$$\pi(a|s) = \mathbf{P}(a_t = a | s_t = s)$$

$$R_t = \Sigma_{k=0}^{\infty} \gamma^k r_{t+k}, \ \gamma \in (0, 1]$$

$$V_\pi(s) = \mathbf{E}\left[R_t | s_t = s\right]$$

❖ Proximal Policy Optimization

$$L^{CLIP}(\theta) = \hat{\mathbf{E}}\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right]$$

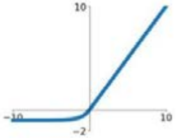


Policy & Observation & Action

- MLP (Multilayer Perceptron)
  - Input Dim: 48
  - Output Dim: 12
  - Hidden Unit: [256, 128, 64]
  - Activation:

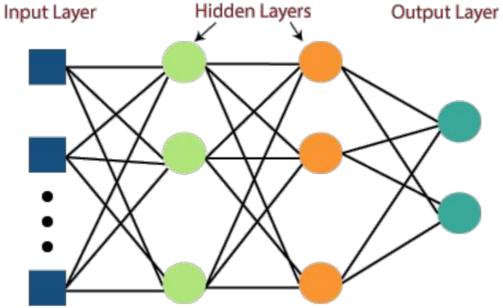
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Observation Space	Dimensions
COM position	3
COM velocity	3
COM Euler angle	3
COM angular velocity	3
DOF position	12
DOF velocity	12
Actions	12
Total	**48**

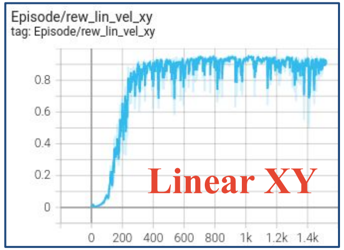
Action Space	Dimensions
Position weights	3
Velocity weights	3
Euler Angle weights	3
Angular Velocity weights	3
Total	**12**

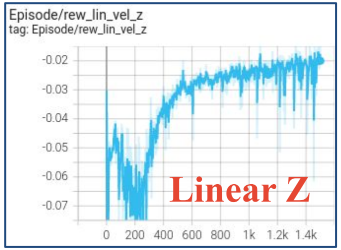


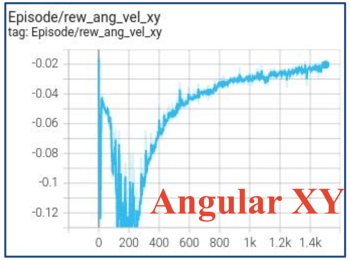
Observation and Reward Design

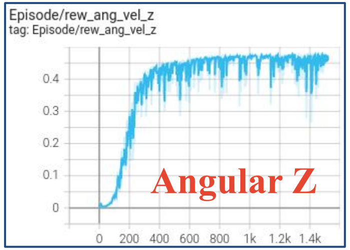
Observation space		Degrees of freedom
Base velocity	positional	3
	angular	3
Body-relative gravity		3
Target X, Y, yaw velocities		3
DOF states	position	12
	velocity	12
Actions		12
Total number of observations		48

Reward	Definition	Weight
Linear velocity tracking	$\phi(\mathbf{v}^*_{b,xy} - \mathbf{v}_{b,xy})$	$1dt$
Angular velocity tracking	$\phi(\boldsymbol{\omega}^*_{b,z} - \boldsymbol{\omega}_{b,z})$	$0.5dt$
Linear velocity penalty	$-\mathbf{v}^2_{b,z}$	$4dt$
Angular velocity penalty	$-\|\|\boldsymbol{\omega}_{b,xy}\|\|^2$	$0.05dt$
Joint motion	$-\|\|\ddot{\mathbf{q}}_j\|\|^2 - \|\|\dot{\mathbf{q}}_j\|\|^2$	$0.001dt$
Joint torques	$-\|\|\boldsymbol{\tau}_j\|\|^2$	$0.00002dt$
Action rate	$-\|\|\dot{\mathbf{q}}^*_j\|\|^2$	$0.25dt$
Collisions	$-n_{collision}$	$0.001dt$







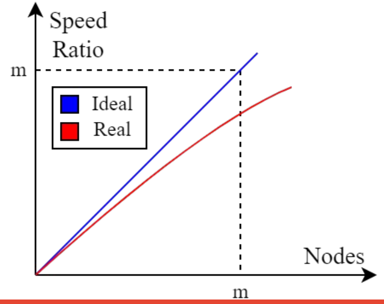


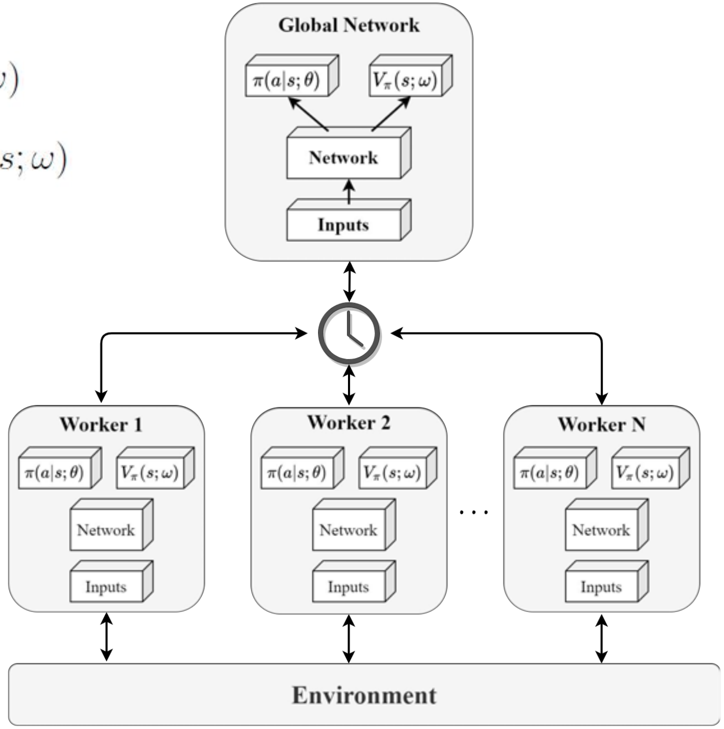
...

Makoviychuk, Viktor, et al. "Isaac gym: High performance gpu-based physics simulation for robot learning."

Parallel RL Training - Part 1

- Global network maintain $\pi(a|s;\theta)$ and $V_\pi(s;\omega)$
- Each worker have a copy of $\pi(a|s;\theta)$ and $V_\pi(s;\omega)$
- Each worker interact the environment ***n*** steps to gain experience and calc gradients
- Global network update $\pi(a|s;\theta)$ and $V_\pi(s;\omega)$ after receiving **all** gradients from workers

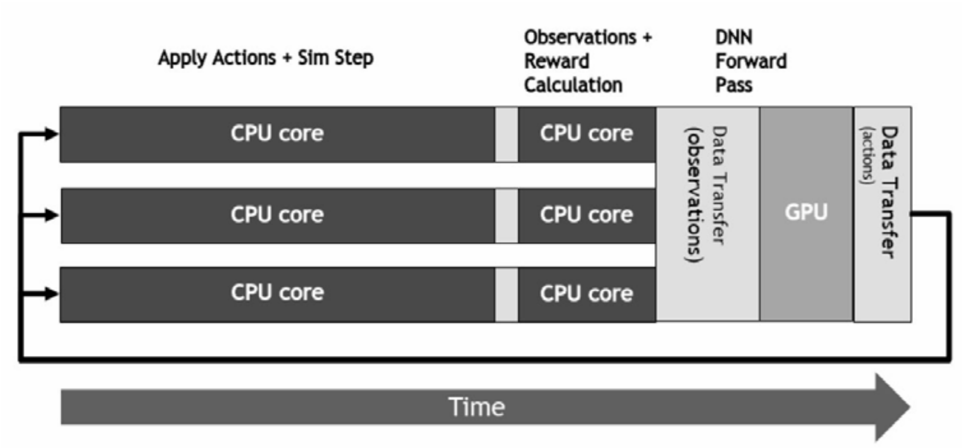




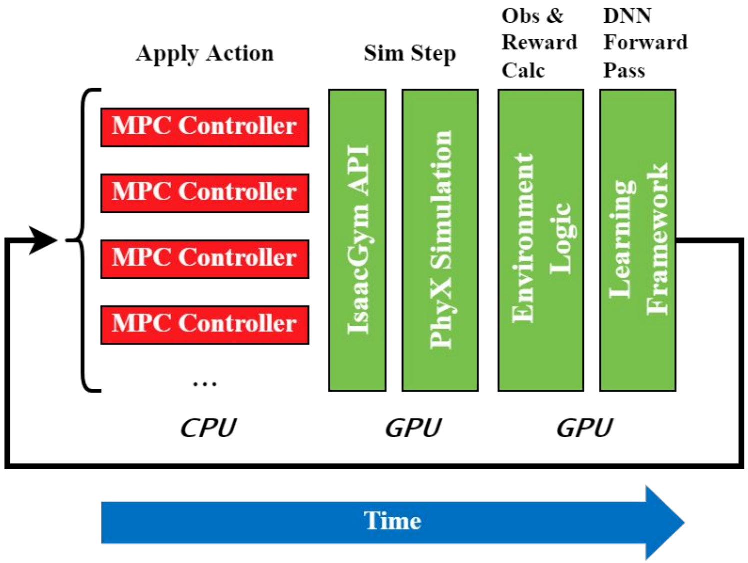
Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016: 1928-1937.

Parallel RL Training - Part 2

- Traditional RL Experience Collection



★ Isaac Gym + MPC Experience Collection



Makoviychuk, Viktor, et al. "Isaac gym: High performance gpu-based physics simulation for robot learning."

Conclusions and Future Work

★ Proposed and implemented a hierarchical control architecture for quadruped

- ◆ Periodic gait such as walking and trotting can be done with simple tuning
- ◆ Sim2Real transfer on real Aliengo robot

➔ Controller Performance :

- ◆ 3 m/s Vx
- ◆ 2 m/s Vy
- ◆ 5 rad/s ω
- ◆ 0.04 rad max orientation deviation
- ◆ 1 ms policy inference

➔ Cons :

- ◆ Insufficient network training steps
- ◆ Only open loop gaits are using
- ◆ No external sensors like camera

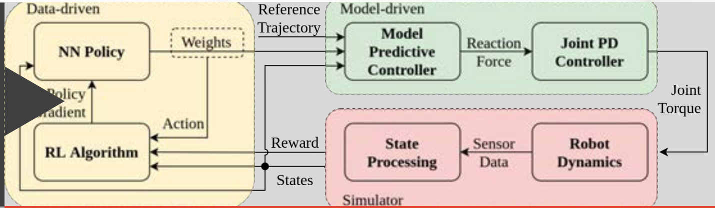
DEPARTMENT OF
MECHANICAL AND
ENERGY ENGINEERING
机械与能源工程系

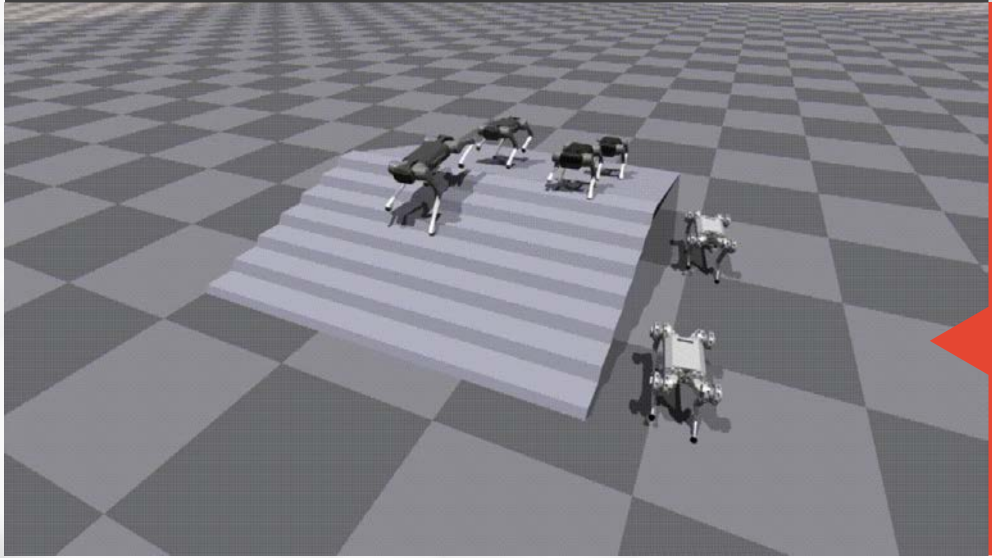
CLEAR
Control & Learning for Robotics and Autonomy Lab

Thanks for Listening

Yulun Zhuang

Quadruped Ctrl and Learn FWK based on IsaacGym

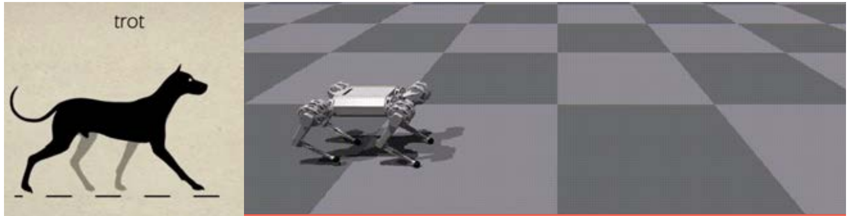




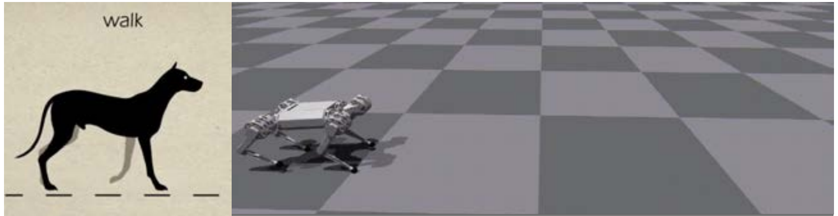
Parallel MPC Control Demo

Simulation:	1000 Hz
Control:	500 Hz
MPC:	50 Hz

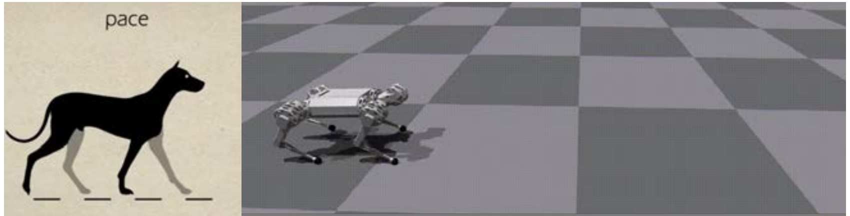
Locomotion Gaits



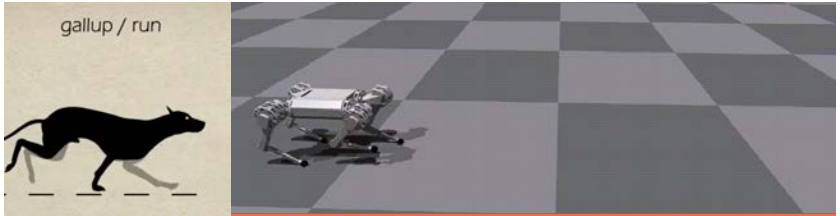
Trot



Walk



Pace



Gallop