Reinforcement Learning for Robotic Control

Xuezhi Niu

CPS Lab, IT Department, Uppsala University

xuezhi.niu@it.uu.se

March 26, 2025

SysCon Seminar

UPPSALA
UNIVERSITET

Agenda

1. Introduction
2. Reinforcement Learning
3. Results
4. Discussion

1. Introduction
2. Reinforcement Learning
3. Results
4. Discussion

Robotics

What is a Robot?

A robot is a physical system that **perceives** its environment, **decides** how to act, and **executes** actions through mechanical components.

A robot is a controlled system.

- Input: control commands (forces, torques, velocities)
- System: robot dynamics (often nonlinear, uncertain)
- Output: measurable states (positions, velocities, forces)

Example of Robots in Our Lab

- Locomotion
- Navigation
- Manipulation
- Many More..

Figure 1: Drone Tello from DJI.

Figure 2: Go2 Quadruped Robot from Unitree (coming soon).

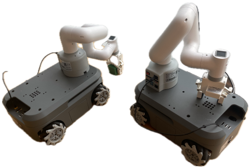


Figure 3: myAGVs and myCobots from Elephant Robotics.

Figure 4: Nova5 from Dobot Robotics with Robotiq 2F85 Gripper.

Robot Controls

Control design answers the question

"Given what the robot knows, how do we choose actuator signals to achieve desired behavior?"

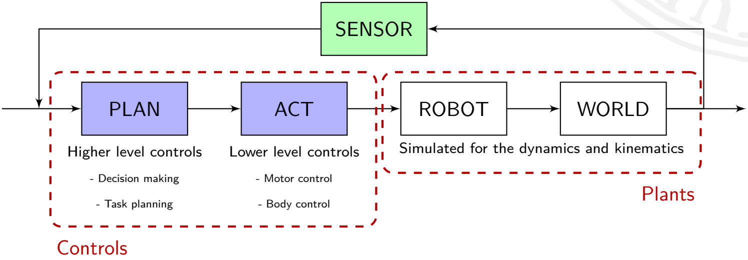


Figure 5: Typical robot control architecture with sensing feedback.

Why RL?

Reinforcement Learning (RL) comes into play when

- The model is unknown or complex
  - nonlinear dynamics
  - unmodeled effects
- The interaction to world changes
  - different terrains
  - varying payloads
  - environmental disturbances
- The task requires too many manual loops to tune
  - complicated objectives
  - multi-step decision making
  - competing constraints

These are often summarized under the sim2real gap [1]

RL Commons

RL

- Solves MDPs without known models (although model-based RL)
- Learns through trial and error using rewards and penalties
- Data is not independent and identically distributed (i.i.d.); past outputs affect future inputs

Optimal Control

- Solves control problems with known models
- Minimizes a cost function derived from system physics

Supervised Learning

- Given i.i.d. data $\mathcal{D} = \mathbf{x}_i, y_i$, learn to predict y from $\mathbf{x}$
- Assumes known ground truth in training

Dynamic Programming

- A framework used by both RL and Optimal Control
- Needs known dynamics for direct application

How RL? In a quadruped robot example

Rule-based Controls [2]

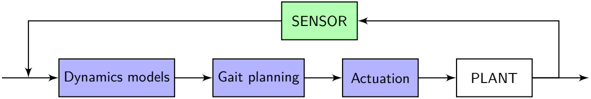


Figure 6: Typical robot control architecture with sensing feedback.

RL-based Controls

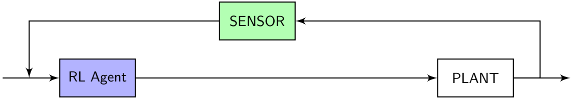


Figure 7: Typical robot control architecture with sensing feedback.

Figure 8: Unitree Go2 locomotion

Deployment on Real Robot

Deployed on Raspberry pi 4B, trained on a normal PC [3]

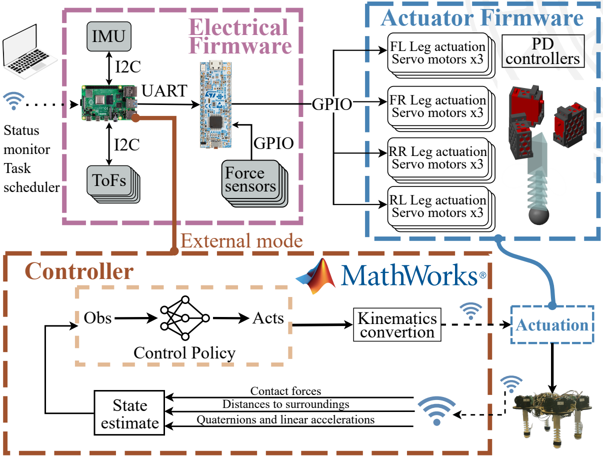


Figure 9: Real deployment control architecture.

1. Introduction
2. Reinforcement Learning
3. Results
4. Discussion

What is RL?

RL Goal

Learn a policy that maximizes the expected cumulative reward

- Agent: follows a policy/function to act based on the state and received rewards
- Policy (π): maps states to actions
- Action (a_t): decision or control input
- State (s_t): current situation

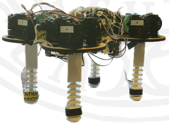
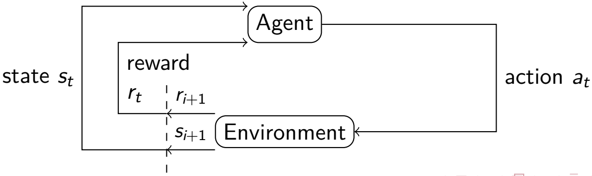


Figure 10: A tendon-driven Soft Quadruped robot (SoftQ).



RL Taxonomy

Robot-environment interaction process is modeled as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, P, r \rangle$, and RL is to optimize

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_t \gamma^t r_t \right]$$

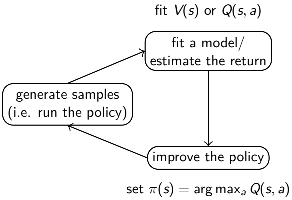


Figure 12: Value based RL

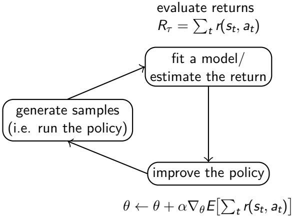


Figure 13: Policy based RL

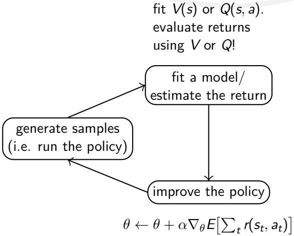


Figure 14: Actor-critic (AC) RL

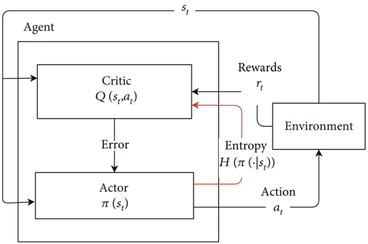
SAC

RL problem can be defined to find the optimal policy π^*

$$\pi^* = \arg\max_{\pi} V_\pi(s_t) = \arg\max_{\pi} \mathbb{E}_{a_t \sim \pi}\left[Q_\pi(s_t, a_t) - \alpha \ln \pi(a_t \mid s_t)\right]$$

SAC maximizes both expected return and policy entropy

$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi \left[\sum_t r(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t))\right]$$



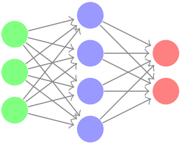
Model-Free Gait Learning

Trot gait actor network (SAC) $\rightarrow$ SoftQ [4]

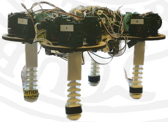
Motions & Contacts

$\theta_{x,y,z}$, $v_{x,y,z}$, $F_{\text{nFL,FR,RR,RL}}$

$u(\text{PWMs} \times 12)$



PWMs $\times 12$

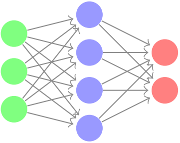


Trot gait policy network (PPO) $\rightarrow$ Go2 [5]

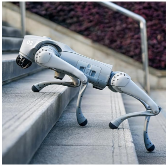
Motions & Joints

$v_{x,y,z}$, $\omega_{x,y,z}$, g, $v^{\text{cmd}}_{x,y,z}$, q, $\dot{q}$,

h, $u(q \times 12)$



Joint actions $\times 12$



Model-Based Gait Learning

Learn a system model from data, improving training efficiency and minimizing the exploration space [3]

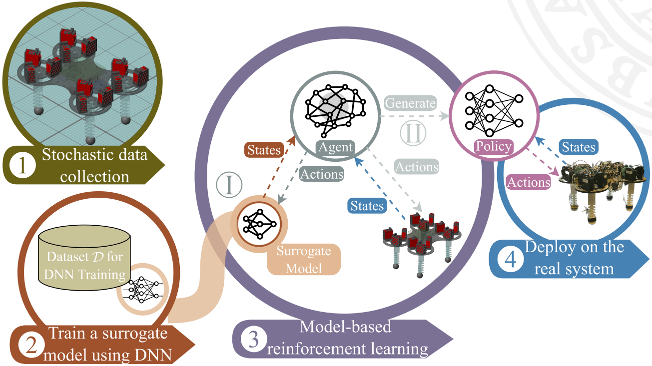
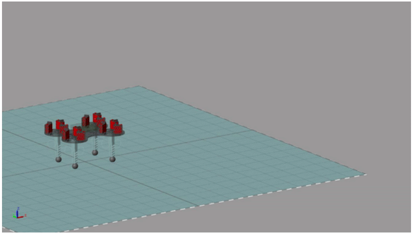


Figure 15: Training process diagram for model-based RL

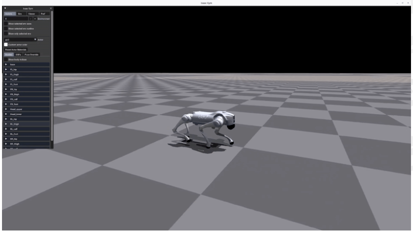
Walking Gait (Trot)

Training results in simulations and reality

SoftQ [3]



Go2 [5]



Reward Functions

Reward shaping is an art, often requiring intuition

Reward for SoftQ [3]: $r = \epsilon_1 \frac{T_s}{T_f} + (1 - \epsilon_2 |v_x(t) - v_{\text{ref}}|) - \epsilon_3 \|\ddot{a}_t\| - \epsilon_4 \|\mathbf{a}_t - \sigma_{\text{threshold}}\| - \epsilon_5 \left(\mathbf{a}_t - \frac{\sum_{i=1}^{T} \mathbf{a}_i}{T}\right)^2.$

Reward for Go2 [5]:

Term	Equation	Weight
$r_{v^{\text{cmd}}_{x,y}}$: xy velocity tracking	$\exp\{-\lvert\mathbf{v}_{xy} - \mathbf{v}^{\text{cmd}}_{xy}\rvert^2/\sigma_{vxy}\}$	0.02
$r_{\omega^{\text{cmd}}_z}$: yaw velocity tracking	$\exp\{-(\boldsymbol{\omega}_z - \boldsymbol{\omega}^{\text{cmd}}_z)^2/\sigma_{\omega z}\}$	0.01
$r_{c^{\text{cmd}}_f}$: swing phase tracking (force)	$\sum_{\text{foot}}[1 - C^{\text{cmd}}_{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert\mathbf{f}^{\text{foot}}\rvert^2/\sigma_{cf}\}$	-0.08
$r_{c^{\text{cmd}}_v}$: stance phase tracking (velocity)	$\sum_{\text{foot}}[C^{\text{cmd}}_{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert\mathbf{v}^{\text{foot}}_{xy}\rvert^2/\sigma_{cv}\}$	-0.08
$r_{\boldsymbol{h}^{\text{cmd}}_z}$: body height tracking	$(\boldsymbol{h}_{\boldsymbol{z}} - \boldsymbol{h}^{\text{cmd}}_z)^2$	-0.2
$r_{\boldsymbol{\phi}^{\text{cmd}}}$: body pitch tracking	$(\boldsymbol{\phi} - \boldsymbol{\phi}^{\text{cmd}})^2$	-0.1
$r_{\boldsymbol{s}^{\text{cmd}}_y}$: raibert heuristic footswing tracking	$(\mathbf{p}^{f}_{x,y,\text{foot}} - \mathbf{p}^{f,\text{cmd}}_{x,y,\text{foot}}(\boldsymbol{s}^{\text{cmd}}_y))^2$	-0.2
$r_{\boldsymbol{h}^{f,\text{cmd}}_z}$: footswing height tracking	$\sum_{\text{foot}}(\boldsymbol{h}^{f}_{z,\text{foot}} - \boldsymbol{h}^{f,\text{cmd}}_z)^2 C^{\text{cmd}}_{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)$	-0.6
z velocity	$\mathbf{v}^2_z$	$-4e{-4}$
roll-pitch velocity	$\lvert\boldsymbol{\omega}_{xy}\rvert^2$	$-2e{-5}$
foot slip	$\lvert\mathbf{v}^{\text{foot}}_{xy}\rvert^2$	$-8e{-4}$
thigh/calf collision	$\mathbb{1}_{\text{collision}}$	-0.02
joint limit violation	$\mathbb{1}_{q_i > q_{max} \lvert\rvert q_i < q_{min}}$	-0.2
joint torques	$\lvert\boldsymbol{\tau}\rvert^2$	$-2e{-5}$
joint velocities	$\lvert\dot{\mathbf{q}}\rvert^2$	$-2e{-5}$
joint accelerations	$\lvert\ddot{\mathbf{q}}\rvert^2$	$-5e{-9}$
action smoothing	$\lvert\mathbf{a}_{t-1} - \mathbf{a}_t\rvert^2$	$-2e{-3}$
action smoothing, 2nd order	$\lvert\mathbf{a}_{t-2} - 2\mathbf{a}_{t-1} + \mathbf{a}_t\rvert^2$	$-2e{-3}$

Task

Augmented Auxiliary

Fixed Auxiliary

Table 1: Reward structure: task rewards, augmented auxiliary rewards, and fixed auxiliary rewards.

What if Multi-Robots?

Many real-world robotic tasks involve multiple robots working together:

- Cooperative transport
- Multi-robot exploration
- Multi-arm manipulation

Key challenges and needs:

- Building **decentralized** control systems with **heterogeneity**.
- Agents must learn policies not only from the environment but also by anticipating and responding to other agents' actions
- The environment becomes **non-stationary** for each agent (others are constantly learning too)
- Requires learning **coordination**, **adaptability**, and often **negotiation**

Symbiosis!

What is Symbiosis?

A biological relationship where two or more organisms interact for continuous existance, including mutualism, commensalism, and parasitism.

- Mycorrhizal networks between trees and fungi — sharing resources and information to support collective survival

Figure 16: Mycorrhizal networks between trees and fungi.

Focus on **mutualism**: Agents shares critical information to support collective behaviors [6].

Symbiosis into MARL (1)

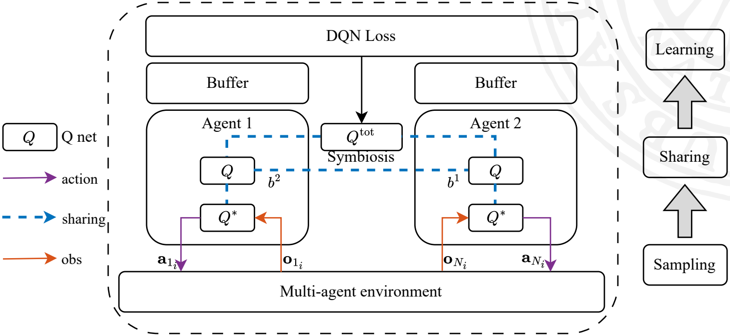


Figure 17: Agents share battery information through symbiosis connections (blue dashed lines) while maintaining individual Q-networks for local decision making. The framework integrates sampling from the environment (orange arrows), sharing of symbiotic information, and learning through DQN loss computation. Q and Q* represent online and target networks respectively, with individual buffers for experience replay. [6]

Preliminary results

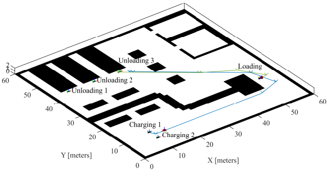


Figure 18: Layout of the simulated warehouse environment (60m × 60m). [6]

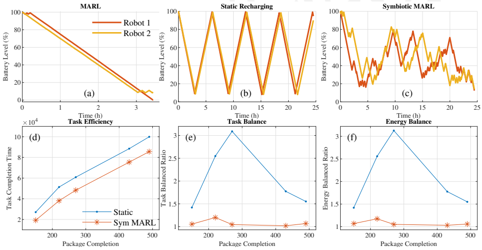


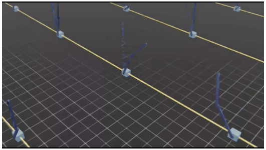
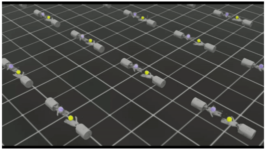
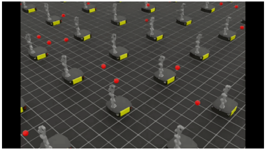
Figure 19: Evaluation of static recharging and MARL with and without symbiosis. [6]

10.7% system performance improvement and 13.81% resource utilization efficiency

Symbiosis into MARL (2)

Symbiosis into **reward shaping**. [7]

$$R_i = \alpha P_i + \beta \sum_{j \neq i} \Delta P(a_i, a_j),$$

Thank you for your attention!

I look forward to your questions and discussions.

Feel free to reach out:

`xuezhi.niu@it.uu.se`

References

Back to start

[1] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard *et al.*, "Sim2real in robotics and automation: Applications and challenges," *IEEE transactions on automation science and engineering*, vol. 18, no. 2, pp. 398–400, 2021.

[2] "Rule-based control," https://www.sciencedirect.com/topics/engineering/rule-based-control, accessed: 2025-03-19.

[3] N. Xuezhi, T. Kaige, G. B. Didem, and F. Lei, "Optimal gait control for a tendon-driven soft quadruped robot by model-based reinforcement learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[4] Q. Ji, S. Fu, K. Tan, S. T. Muralidharan, K. Lagrelius, D. Danelia, G. Andrikopoulos, X. V. Wang, L. Wang, and L. Feng, "Synthesizing the optimal gait of a quadruped robot with soft actuators using deep reinforcement learning," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102382, 2022.

[5] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," *Conference on Robot Learning*, 2022.

[6] N. Xuezhi, C. B. Natalia, and G. B. Didem, "Enabling symbiosis in multi-robot systems through multi-agent reinforcement learning," in *2025 IEEE 8th International Conference on Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2025.

[7] N. Xuezhi and G. B. Didem, "Investigating symbiosis in robotic ecosystems: A case study for multi-robot reinforcement learning reward shaping," in *2025 9th International Conference on Robotics and Automation Sciences (ICRAS) (under review)*. IEEE, 2025.