Inductive biases in robot learning for contact detection and obstacle avoidance

Stéphane Caron
Works with **Ü. Bora Gökbakan** and **Valentin Tordjman--Levavasseur**

November 4, 2025

Département informatique de l'École normale supérieure (CNRS, ENS–PSL, INRIA)

Before we start

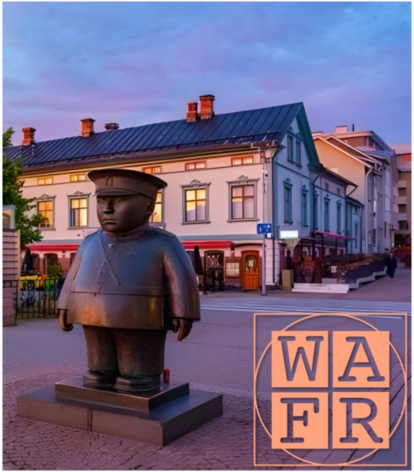
Topics:

- Design and analysis of robotic algorithms
- Mathematical foundations of robotics
- Cross-disciplinarity most welcome!

Timeline:

- Paper submission: January 15, 2026
- Notification of acceptance: March 15, 2026
- Conference: June 15-17, 2026

Publication: Springer Proceedings in Advanced Robotics (SPAR) series. Selected papers invited for publication in special issues.



Inductive biases

MACHINE LEARNING HAS COME TO ROBOTICS

Figure 1: 1998

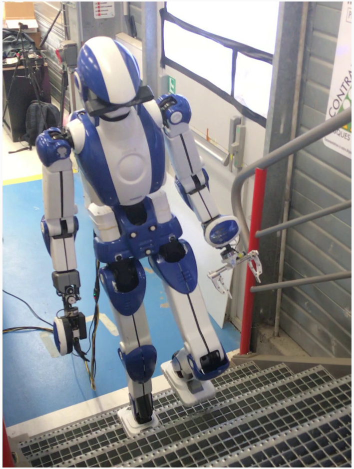


Figure 2: 2019

Figure 3: 2025

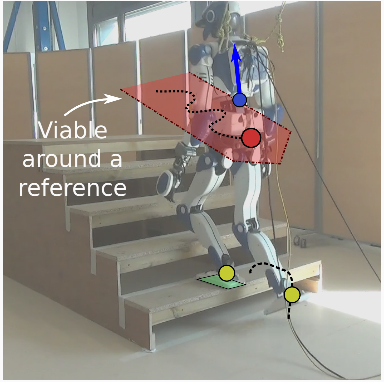


Figure 4: Model-based, ~ 50 planning and control parameters. Tuned by hand.

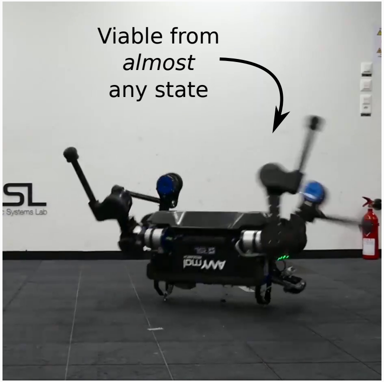
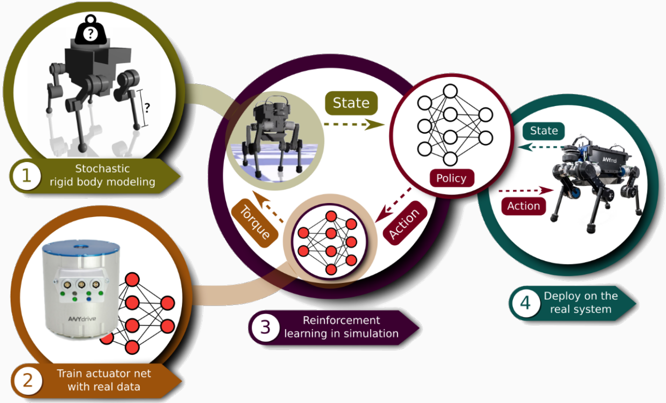


Figure 5: Model-free, ~ 500 parameters. Tuned by reinforcement learning.

[1] Joonho Lee, Jemin Hwangbo, and Marco Hutter. **"Robust recovery controller for a quadrupedal robot using deep reinforcement learning".** In: *arXiv preprint arXiv:1901.07517* (2019).

Simulation and actuator nets are models:



[2] Jemin Hwangbo et al. **“Learning agile and dynamic motor skills for legged robots”.** In: *Science Robotics* 4.26 (2019).

INDUCTIVE BIASES

Model-based vs model-free is not the most meaningful distinction.

How about inductive biases?

Definition

An inductive bias is an assumption the learner uses to generalize to OOD inputs.

Machine learning examples:

- 2D convolutional layers: process local patterns, assume spatial hierarchy
- Maximum margin in support vector machines

Robotics examples:

- Rigid bodies: prioritize explanations where solid objects don't deform
- Gauss's principle of least action: prioritize proximity to free motions

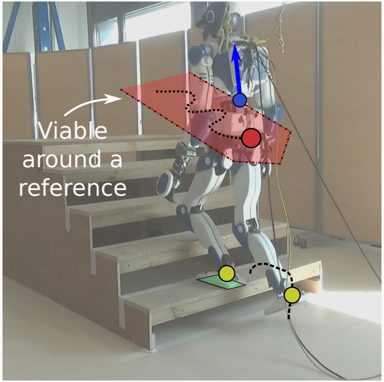


Figure 6: More IB.

Figure 7: *Hic sunt dracones.*

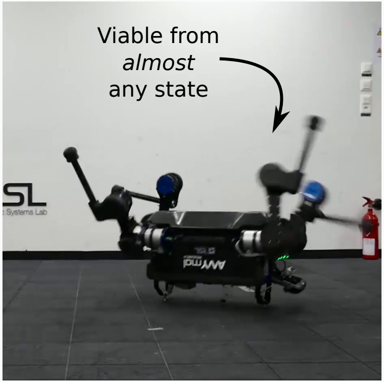


Figure 8: Fewer IB.

THIS PRESENTATION

In the following two works, we apply machine learning with inductive biases to two robotics problems:

- **Contact detection [GDC25]:** is a leg in contact with the ground?
- **Collision avoidance:** don't bump into obstacles.

Contact estimation

Problem: Detect when the robot makes and breaks contact with the ground.

This work:

- No contact sensor
- Leveraging data (model-free).

(a) Solo-12 [Gri+20] (b) Upkie [Car+25]

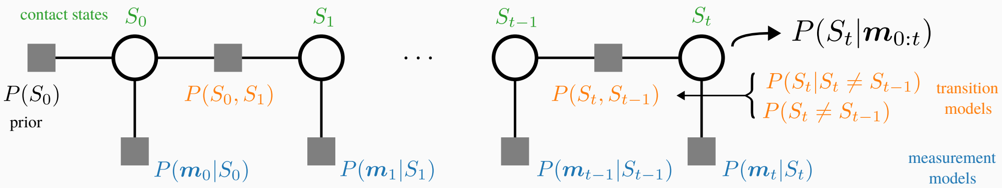
Figure 9: Open-source robots not equipped with contact sensors.

Prior works relied on gait priors, kinodynamic models or collocated sensors:

Method	IMU	Proprio.	F/T sensors[3]	Data-driven	Morphology
[Hwa+16; Jen+19]	+	+	−	−	Quadruped
[RSR18]	+	+	+	−	Biped
[You+24]	+	+	−	+	Quadruped
[GDC25] (This work)	+	+	−	+	Wheeled-biped

[3] Dedicated force-torque sensors collocated with contacts.

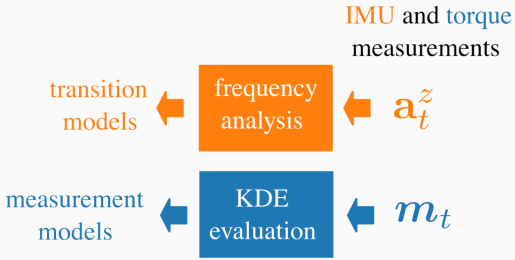
Contact estimation can be cast as a probabilistic state machine:

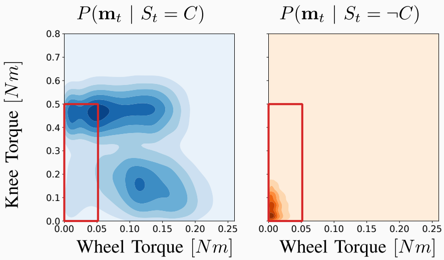


[4] Jemin Hwangbo et al. **"Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics".** In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, Oct. 2016.

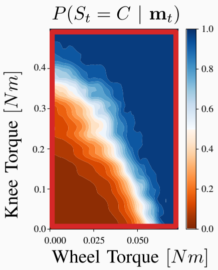
In order to estimate the contact state, we decompose the problem into two parts:

- **Transition probabilities:** based on IMU readings.
- **Measurement likelihoods:** based on joint torques.





(a) Likelihoods for *contact* (left) and *no contact* (right).



(b) Normalized KDEs.

We use **Gaussian Kernel Density Estimation** (KDE) to estimate measurement likelihoods (10a), and for reference a “measurement-only” contact probability (10b).

KDEs are trained from **7 min of real-robot data**.

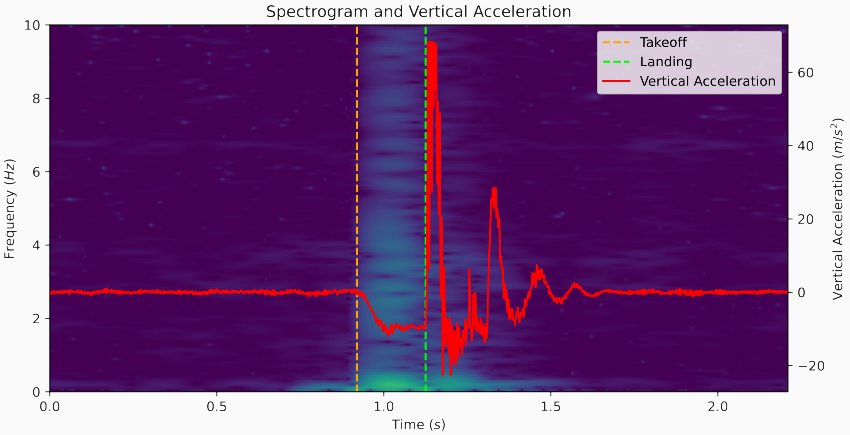
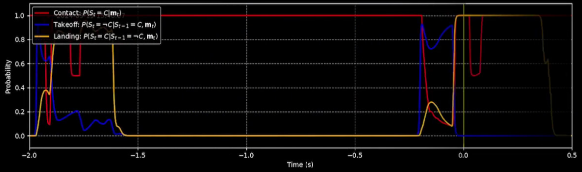


Figure 11: Vertical acceleration $\mathbf{a}_t^z$ (red) and power spectral density around takeoff (orange) and landing (green) events on a real robot.

REAL-WORLD EXPERIMENTS



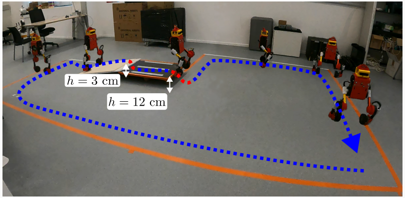


Figure 12: During evaluation, the robot was driven 5 times around a track, where it climbed a 15 cm high ramp, followed by two 3 cm and 12 cm drops (video).

Method	Takeoff		Landing	
	Precision	Recall	Precision	Recall
Bayes filter	**0.91**	**1.0**	**0.71**	**1.0**
Meas. only	0.59	**1.0**	0.55	**1.0**
NMN [You+24]	0.53	0.90	0.56	0.90

Table 1: Detection of transition events over 10 drops during a session of 4 minutes.

Method	Takeoff	Landing
Bayes filter	$\mathbf{77.1 \pm 31.8}$	17.9 ± 9.1
Meas. only	83.1 ± 31.9	$\mathbf{10.1 \pm 7.37}$
NMN [You+24]	125.0 ± 59.6	22.4 ± 7.4

Table 2: Transition latencies (ms). Only correctly identified transitions were considered.

- Data-driven contact estimation from IMU and proprioception.
- **Sample-efficiency:** trained with only a few minutes of real-robot data.
- Better recall and precision than the neural network baseline.
- Open source code and data.[5]

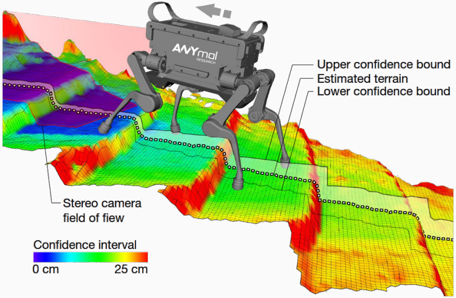
[5] `https://github.com/ubgk/contact_agent`

Collision avoidance

One approach to perceptive locomotion:

- **Mapping:** build an elevation map.
- **Localization:** estimate where the robot is in that map.
- **Control:** step over flat surfaces, avoid obstacles.

Plus, elevation maps are interpretable.



[6] Peter Fankhauser, Michael Bloesch, and Marco Hutter. **"Probabilistic Terrain Mapping for Mobile Robots With Uncertain Localization".** In: *IEEE Robotics and Automation Letters* 3.4 (Oct. 2018).

Solving too hard a problem?

- Time complexity of mapping and localization in an elevation map.
- Representation is not suited to unstructured environments.

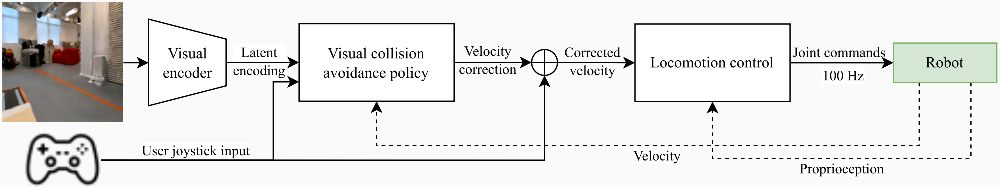
We consider an alternative with less computations from sensors to locomotion.

[7] Takahiro Miki et al. **"Learning robust perceptive locomotion for quadrupedal robots in the wild".** In: *Science Robotics* 7.62 (Jan. 2022).

Approach

Replace elevation map by a **trained latent space**.

- **Visual encoder:** trained on monocular depth prediction
- **Collision avoidance:** trained by reinforcement learning
- **Locomotion:** velocity tracking and balancing by model predictive control



AUGMENTING SIMULATION WITH NOVEL VIEW SYNTHESIS

Generate RGB and depth first-person views in simulation.

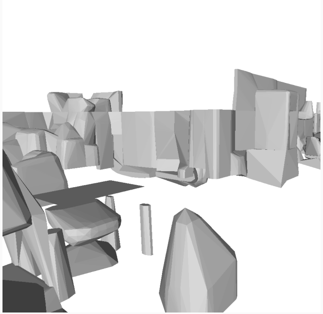
- Capture a video of a room and extract about 500 images
- COLMAP [SF16] to extract the image extrinsics and camera intrinsics
- Align and scale the image reference frame of the images
- Train a **2D Gaussian splatting** model
- Query images at simulated camera poses

[8] Binbin Huang et al. **"2D Gaussian Splatting for Geometrically Accurate Radiance Fields".** In: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*. SIGGRAPH '24. ACM, July 2024.

Gaussian splatting also allows rendering of depth images:

- TSDF fusion to get a mesh of the rendered room.
- Decompose nonconvex mesh into to convex subparts using COACD [Wei+22].
- Collision detection using the Coal[9] library.

(a) Nonconvex raw mesh



(b) Convex collision meshes

[9] https://github.com/coal-library/coal

Train the vision encoder separately on monocular depth prediction:

- Collect RGB–log(Depth) pairs using Gaussian splatting.
- Encoder-decoder convolutional architecture with a latent of dimension 32.
- PSNR at 0.5m on test set: 16.8.

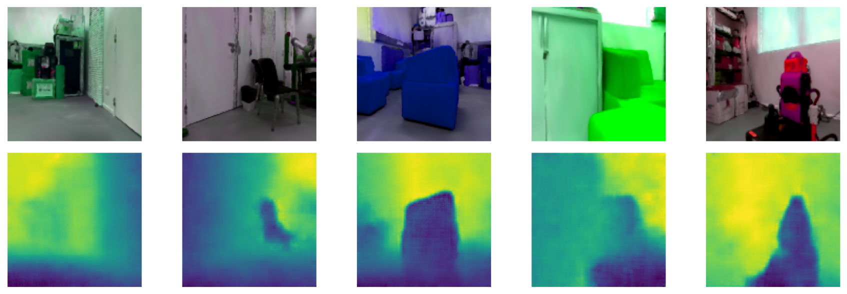


Figure 14: Samples of generated RGB images and predicted depth reconstructions

Train a policy by reinforcement learning in a navigation environment:

- **Observation:** position, velocity, user input, previous action, latent encoding.
- **Action:** velocity correction in $\mathfrak{se}(2)$.
- **Reward:** survival bonus + correction penalty + distance penalty.

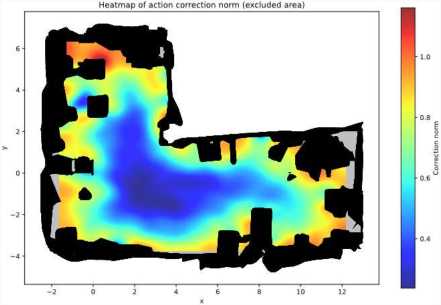


Figure 15: Norm of velocity corrections applied by the agent in the training room.

- Input velocities and collision-avoidance compensation combined:

$$\zeta_t^* = (v_t^*, 0, \omega_t^*) = j_t + u_t$$

- Angular velocity ω_t^* mapped to wheel velocities via differential-drive model
- Sagittal velocity v_t^* tracked by MPC over wheeled-inverted-pendulum model
- Linearized optimal-control problem formulated as a quadratic program (QP)
- Solved in real-time using ProxQP[10] with hot-starting

[10] Antoine Bambade et al. **"ProxQP: an Efficient and Versatile Quadratic Programming Solver for Real-Time Robotics Applications and Beyond".** In: *IEEE Transactions on Robotics* (2025).

Experiment over eight scenarios with comparison to FOA [HSB22]:

- Higher success rate on average.
- Completes the obstacle course faster.
- Lower correction of user inputs.

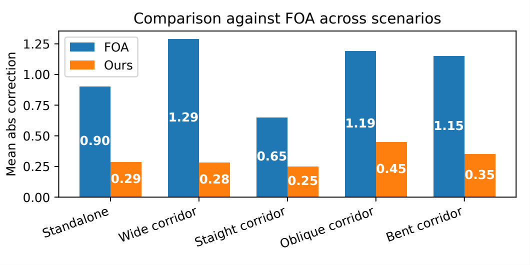
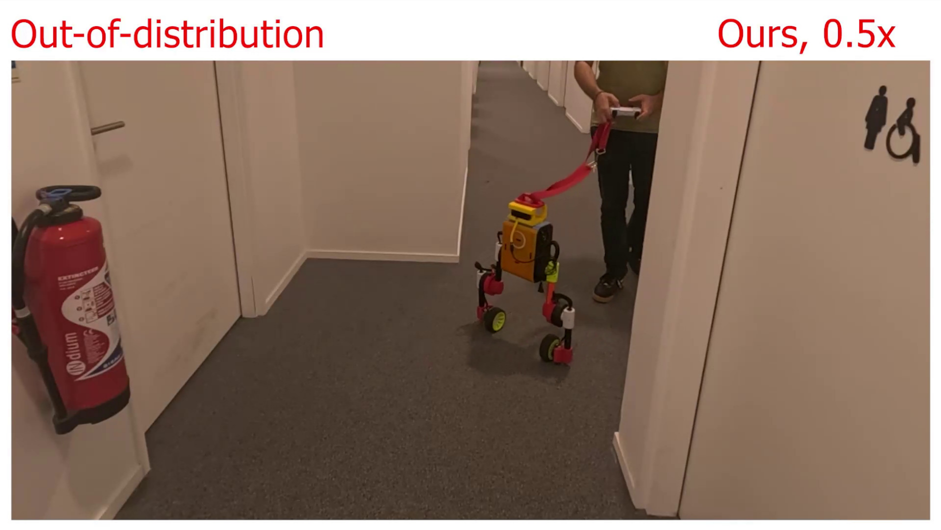


Figure 16: Comparison of corrections applied over the experimental setups.

REAL-ROBOT EXPERIMENTS

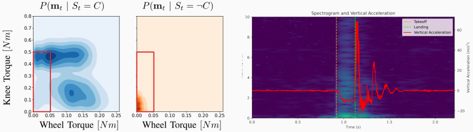


Conclusion

CONCLUSION

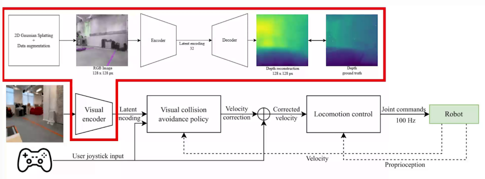
Contact detection:

- *Algorithm:* Kernel density estimation
- *Data:* Real-robot
- *Inductive bias:* Bayes-like filter



Collision avoidance:

- *Algorithm:* Policy gradient
- *Data:* Novel view synthesis
- *Inductive bias:* Depth estimation



THANK YOU FOR YOUR ATTENTION

Bibliography

[Bam+25] Antoine Bambade et al. **“ProxQP: an Efficient and Versatile Quadratic Programming Solver for Real-Time Robotics Applications and Beyond”.** In: *IEEE Transactions on Robotics* (2025).

[Car+25] Stéphane Caron et al. ***Upkie open source wheeled biped robot.*** Version 9.0.1. 2025. URL: `https://github.com/upkie/upkie`.

[FBH18] Peter Fankhauser, Michael Bloesch, and Marco Hutter. **“Probabilistic Terrain Mapping for Mobile Robots With Uncertain Localization”.** In: *IEEE Robotics and Automation Letters* 3.4 (Oct. 2018).

[GDC25] Ü. Bora Gökbakan, Frederike Dümbgen, and Stéphane Caron. **“A Data-Driven Contact Estimation Method for Wheeled-Biped Robots”.** In: *IEEE International Conference on Robotics and Automation*. 2025.

[Gri+20] Felix Grimminger et al. **“An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research”.** In: *IEEE Robotics and Automation Letters* 5.2 (2020).

[HSB22] Lukas Huber, Jean-Jacques Slotine, and Aude Billard. **“Fast Obstacle Avoidance Based on Real-Time Sensing”.** In: *arXiv preprint arXiv:2205.04928* (2022).

[Hua+24] Binbin Huang et al. **“2D Gaussian Splatting for Geometrically Accurate Radiance Fields”.** In: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers ’24*. SIGGRAPH ’24. ACM, July 2024.

[Hwa+16] Jemin Hwangbo et al. **“Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics”.** In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2016.

[Hwa+19] Jemin Hwangbo et al. **“Learning agile and dynamic motor skills for legged robots”.** In: *Science Robotics* 4.26 (2019).

[Jen+19] Fabian Jenelten et al. **“Dynamic Locomotion on Slippery Ground”.** In: *IEEE Robotics and Automation Letters* 4.4 (Oct. 2019). ISSN: 2377-3766, 2377-3774. (Visited on 06/13/2023).

[LHH19] Joonho Lee, Jemin Hwangbo, and Marco Hutter. **“Robust recovery controller for a quadrupedal robot using deep reinforcement learning”.** In: *arXiv preprint arXiv:1901.07517* (2019).

[Mik+22] Takahiro Miki et al. **“Learning robust perceptive locomotion for quadrupedal robots in the wild”.** In: *Science Robotics* 7.62 (Jan. 2022).

[RSR18] Nicholas Rotella, Stefan Schaal, and Ludovic Righetti. **“Unsupervised Contact Learning for Humanoid Estimation and Control”.** In: *IEEE International Conference on Robotics and Automation*. 2018.

[SF16] Johannes Lutz Schönberger and Jan-Michael Frahm. **“Structure-from-Motion Revisited”.** In: *Conference on Computer Vision and Pattern Recognition*. 2016.

[Wei+22] Xinyue Wei et al. **“Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search”.** In: *ACM Transactions on Graphics (TOG)* 41.4 (2022).

[You+24] Donghoon Youm et al. **“Legged Robot State Estimation With Invariant Extended Kalman Filter Using Neural Measurement Network”.** arXiv preprint arXiv:2402.00366. 2024.