Autonomous Locomotion and Manipulation of Robotic Systems

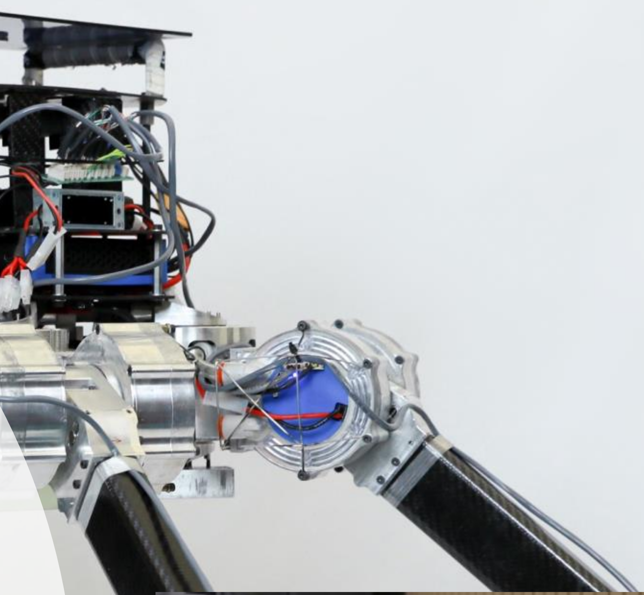
Alexander Schperberg

Advisor: Prof. Dennis Hong

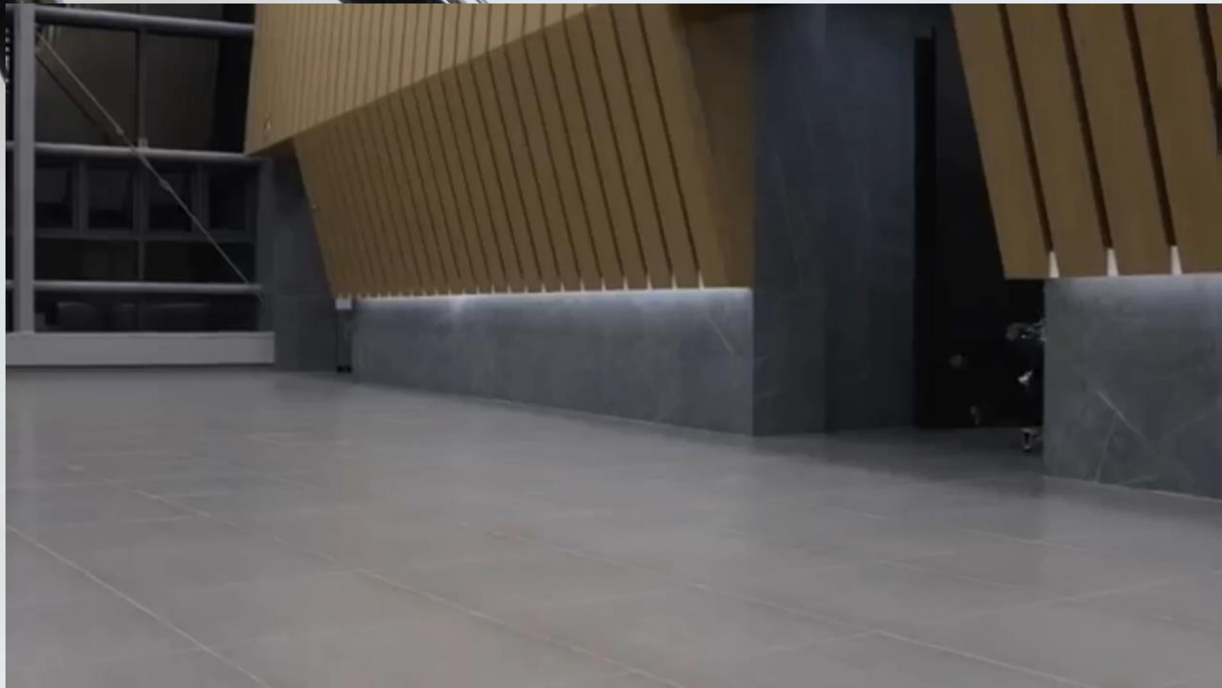
Robotics and Mechanisms Laboratory

Main Interest

Completely Autonomous Robots



- Robots are supposed to be autonomous, but to fulfill this requirement, they must constantly adapt and learn on their own
- How to use planning, vision, mapping, controls, and estimation simultaneously within an end-to-end framework for autonomous behavior



Risk-Averse MPC via Visual-Inertial Input and Recurrent Networks for Online Collision Avoidance

Alexander Schperberg, Kenny Chen, Stephanie Tsuei, Michael Jewett, Josh Hooks
Stefano Soatto, Ankur Mehta, and Dennis Hong

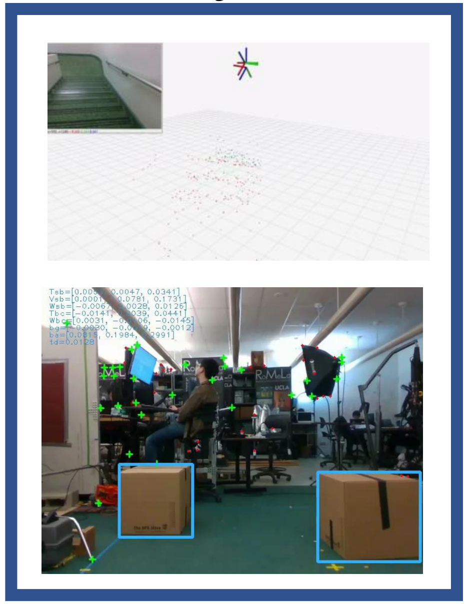
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, October 25-29, 2020

Project 1

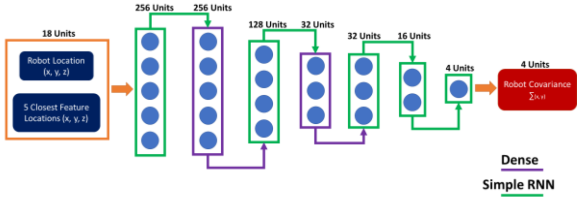
- Use **MPC** for high-level **online** path planning, informed by an **object detection system** and an **RNN** (trained on **SLAM** algorithms and estimations)

<u>Autonomous Planning</u>

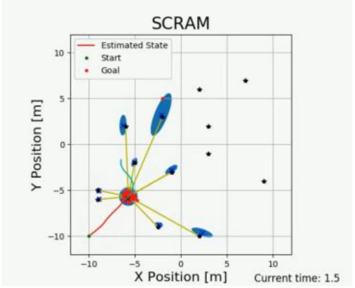
SLAM + Object Detector

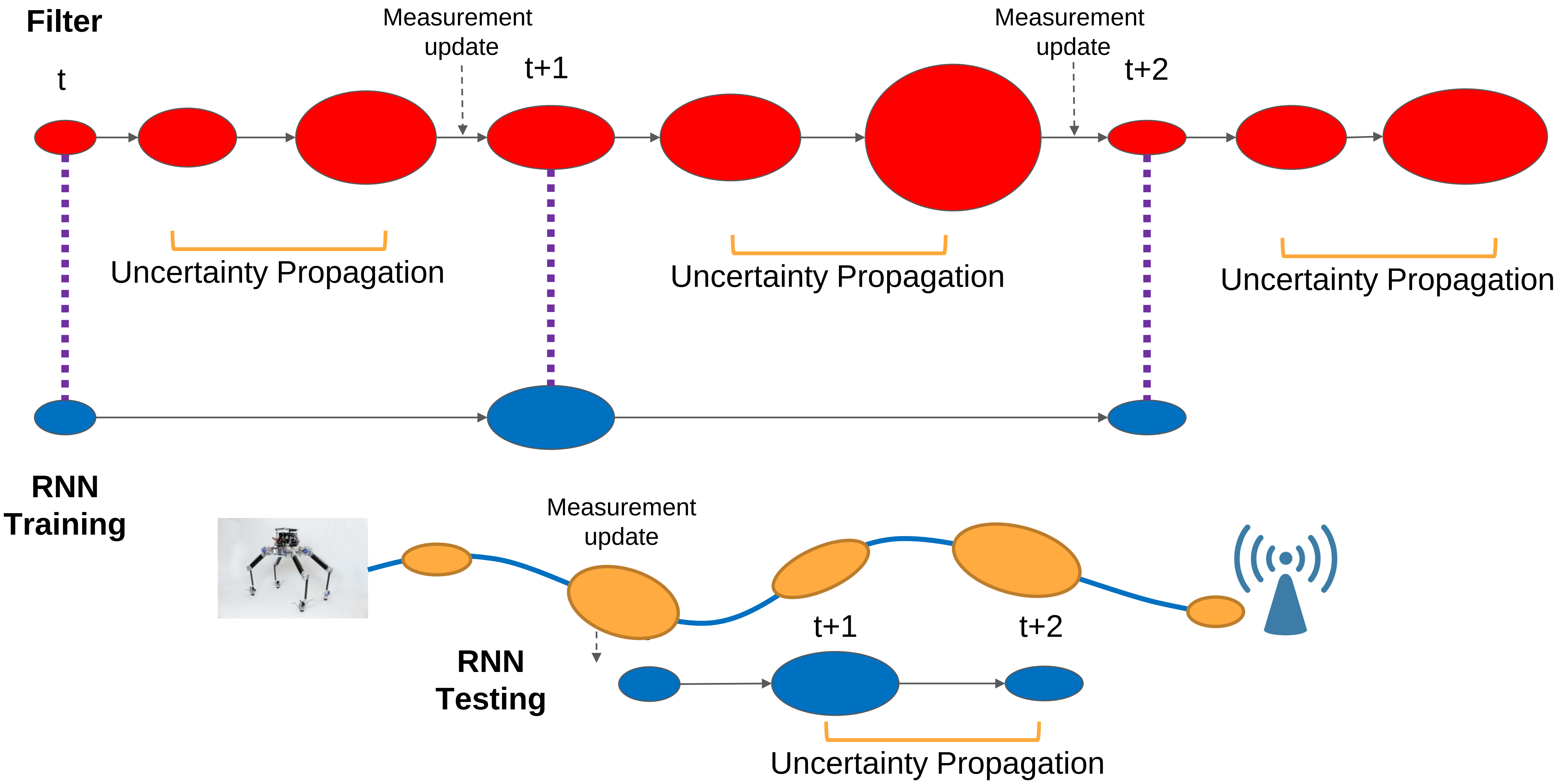


Machine learning



Path Planning

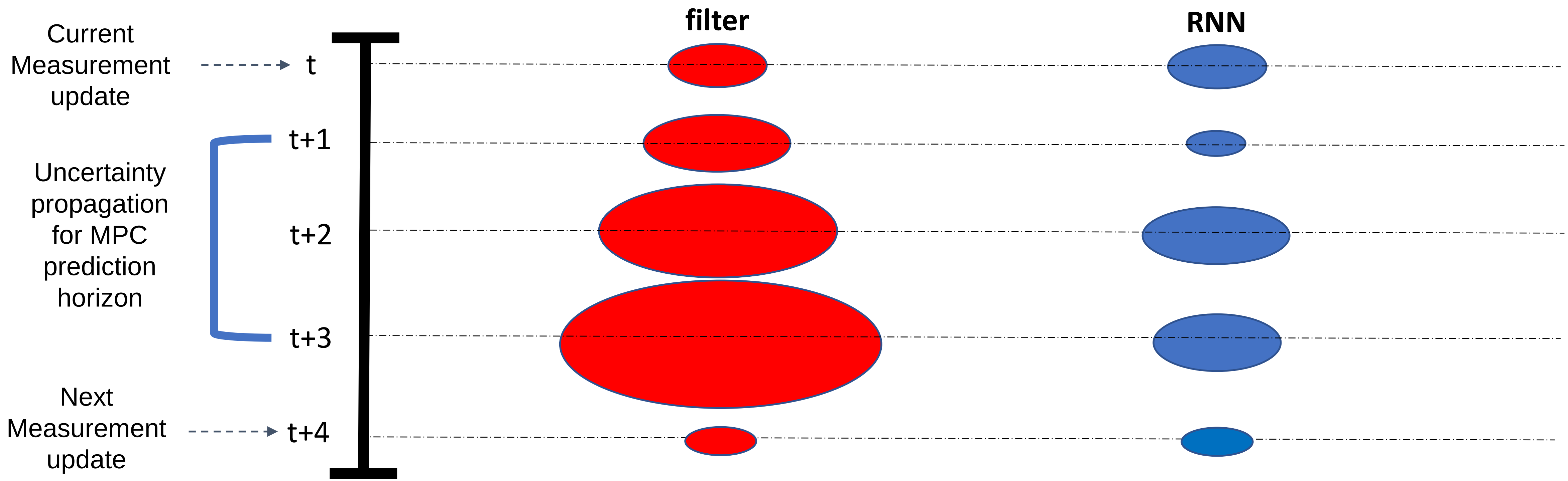


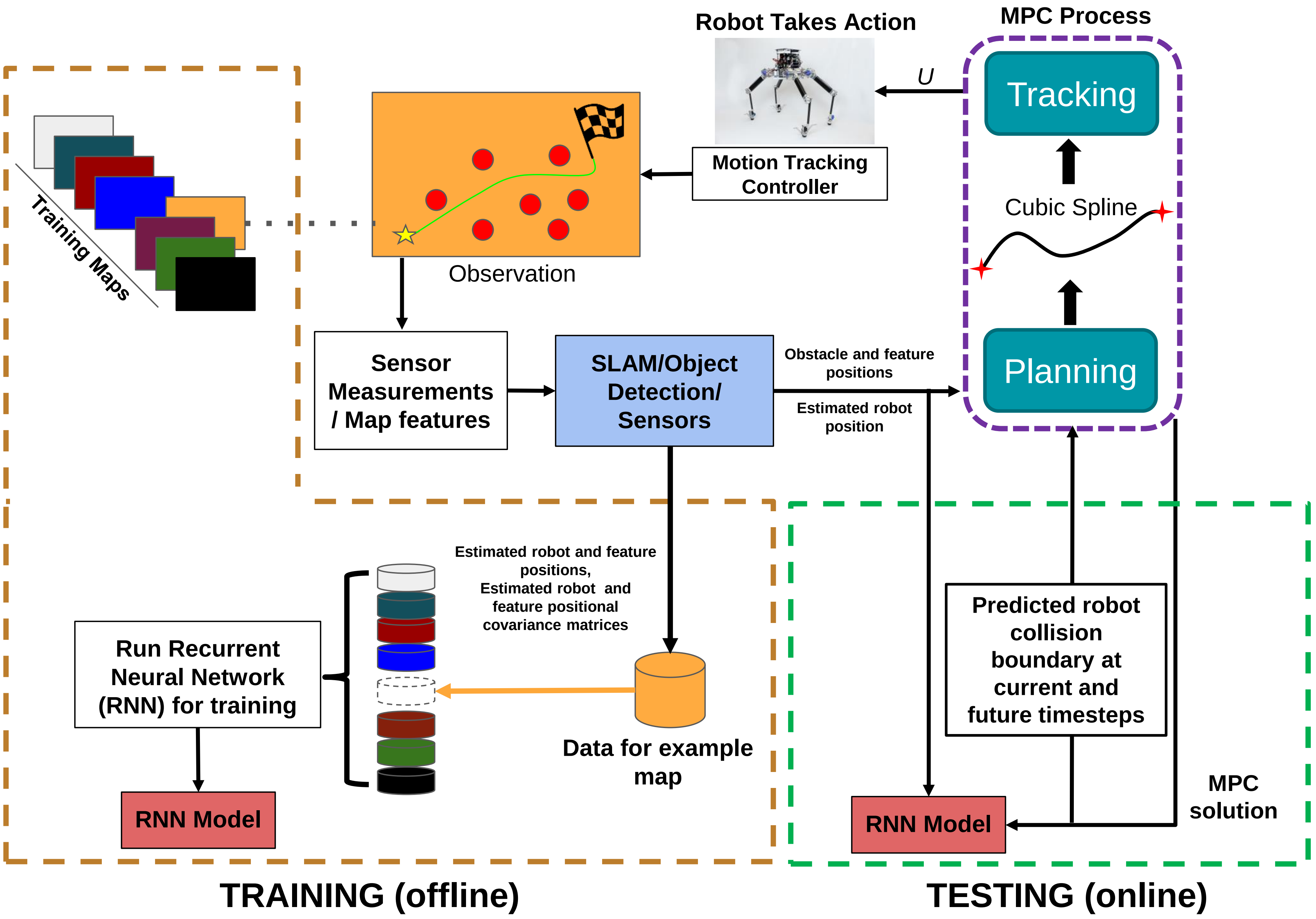

Filter
t
Measurement update
t+1
Measurement update
t+2
Uncertainty Propagation
Uncertainty Propagation
Uncertainty Propagation
RNN Training
Measurement update
RNN Testing
t+1
t+2
Uncertainty Propagation

‘filter’ propagation vs RNN propagation for state uncertainties

- Filter (e.g., particle, KF, EKF): assumes constantly increasing uncertainty propagation between the current and next measurement update
- RNN: dynamically changing uncertainty propagation between the current and next measurement update

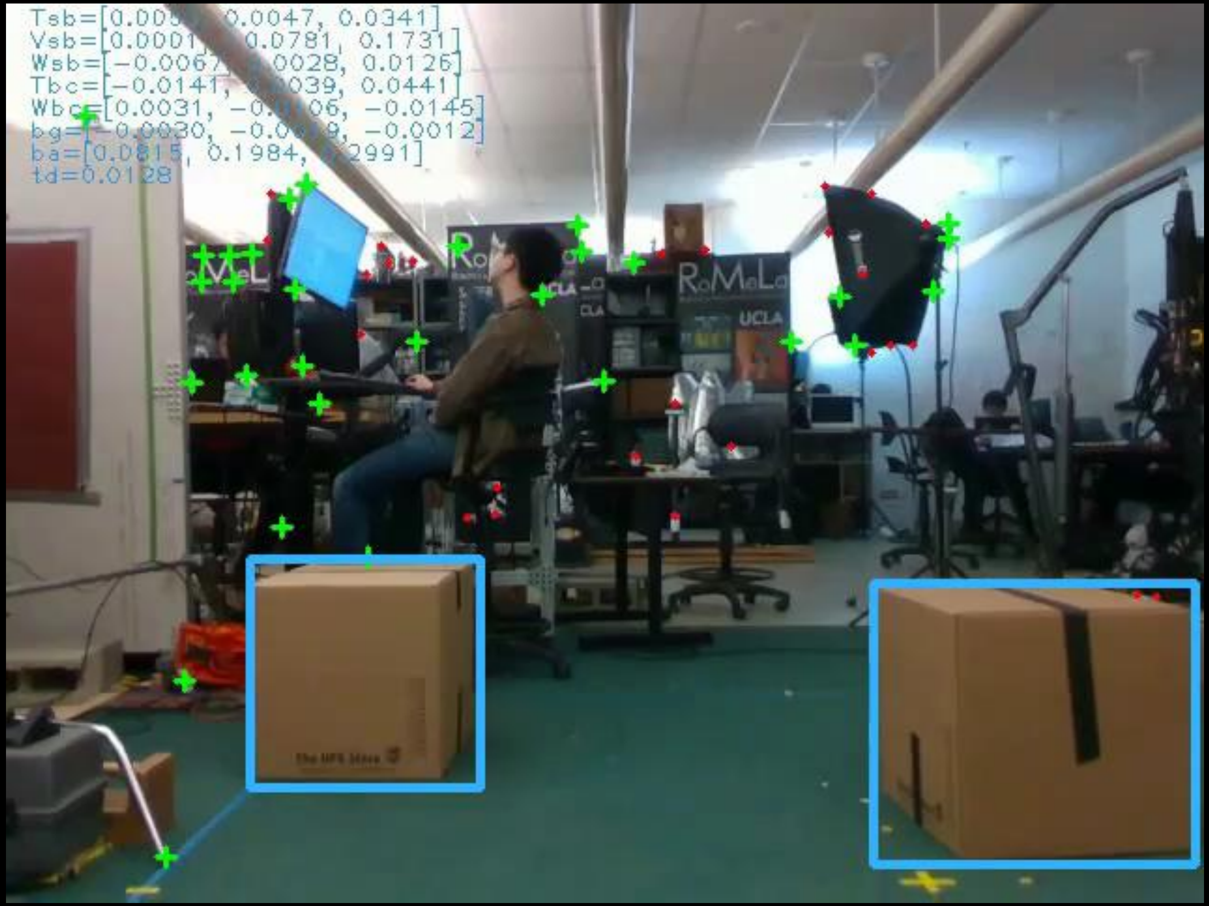
Example of uncertainty propagation between current and next measurement update



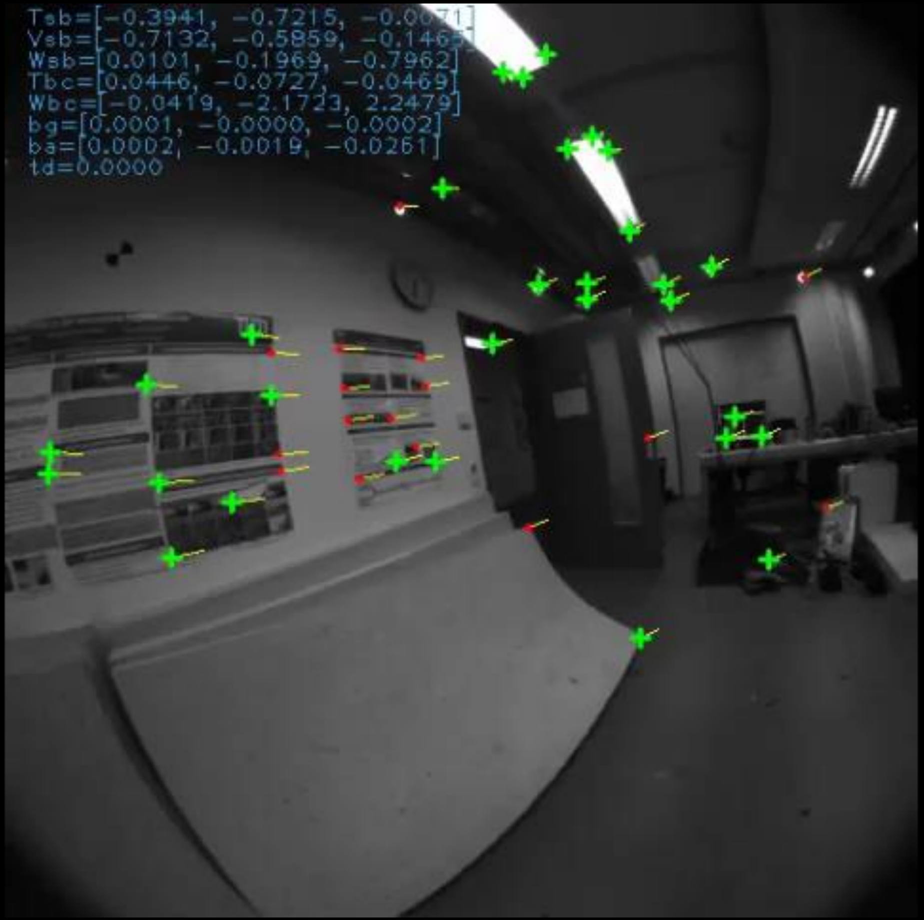

Robot Takes Action
MPC Process
U
Tracking
Cubic Spline
Planning
Motion Tracking Controller
Observation
Training Maps
Sensor Measurements / Map features
SLAM/Object Detection/ Sensors
Obstacle and feature positions
Estimated robot position
Estimated robot and feature positions, Estimated robot and feature positional covariance matrices
Run Recurrent Neural Network (RNN) for training
RNN Model
Data for example map
Predicted robot collision boundary at current and future timesteps
MPC solution
RNN Model
TRAINING (offline)
TESTING (online)

Building RNN model to predict future uncertainty with SLAM algorithm

Running XIVO[1] (SLAM) on our Quadruped



Running XIVO[1] (SLAM) on the TUM[2] Dataset

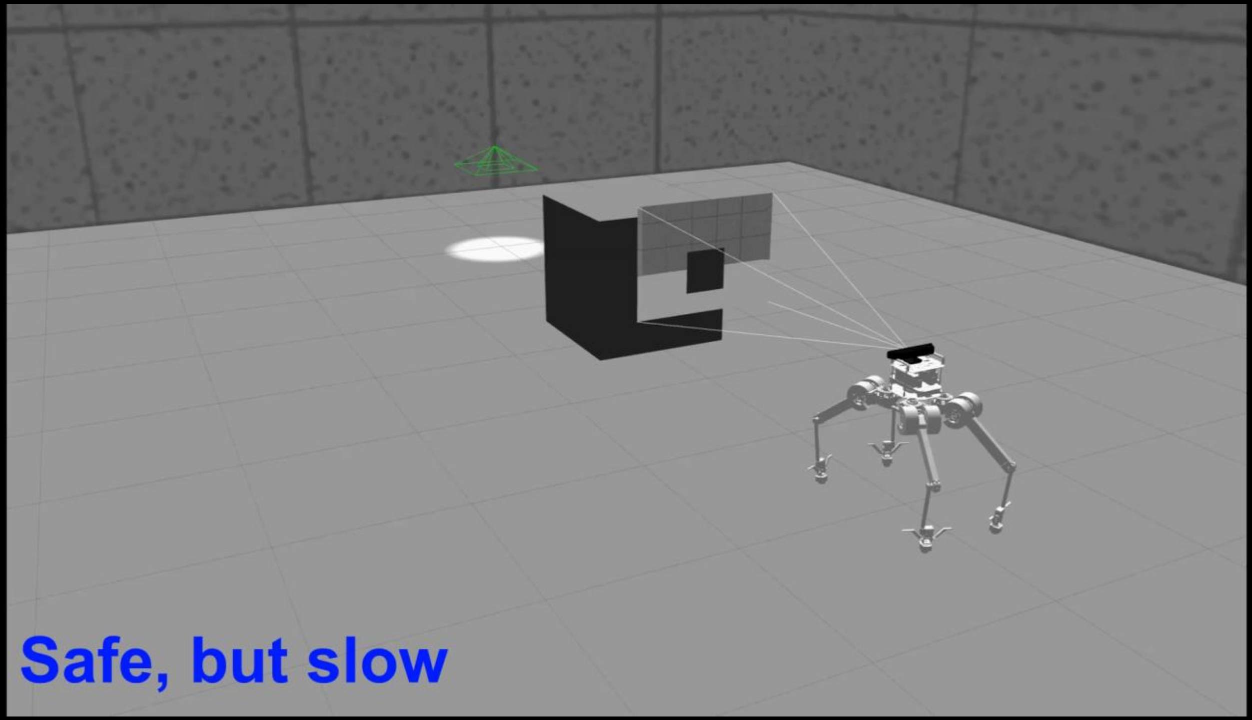


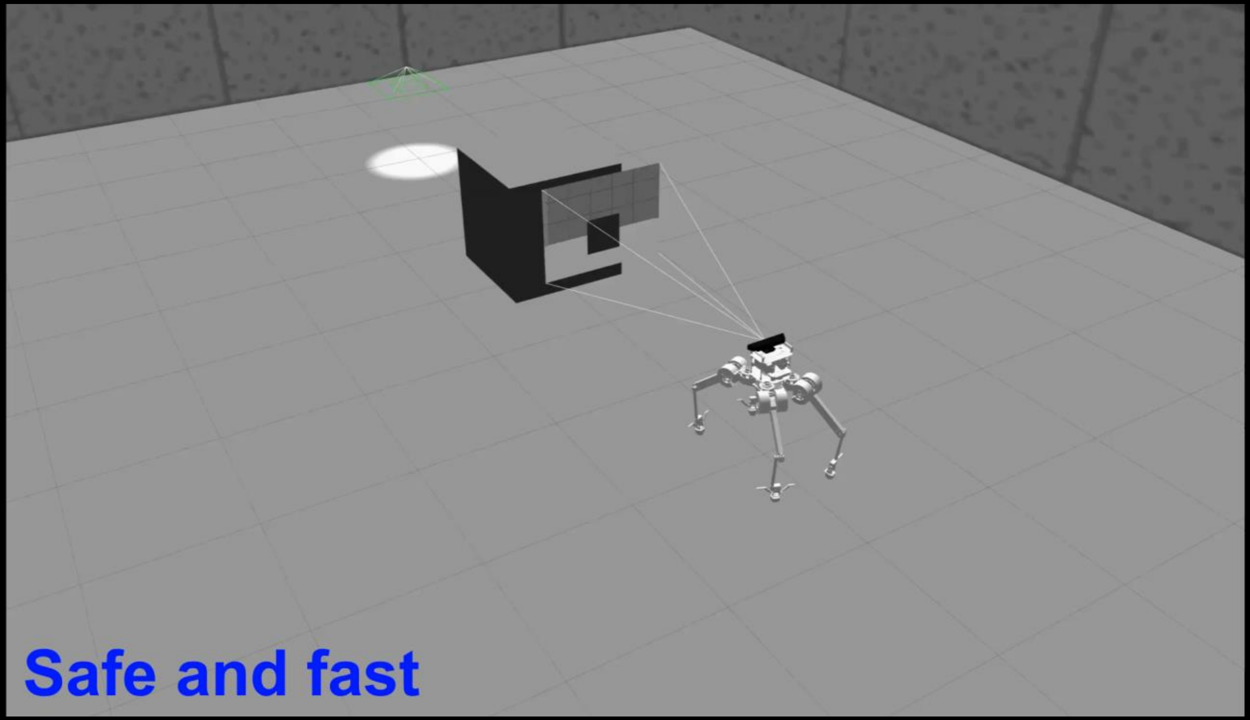
Our RNN model is built with **real** IMU + Camera data for Simulation

[1]https://github.com/ucla-vision
[2]https://vision.in.tum.de/data/datasets/rgbd-dataset

B) <u>Naïve MPC</u>
(uncertainty is static and inflated)

C) <u>MPC with RNN</u>
(uncertainty is dynamically propagated)





Videos includes object detection and are sped up

RoMeLa
BOTICS & MECHANISMS LABORATORY

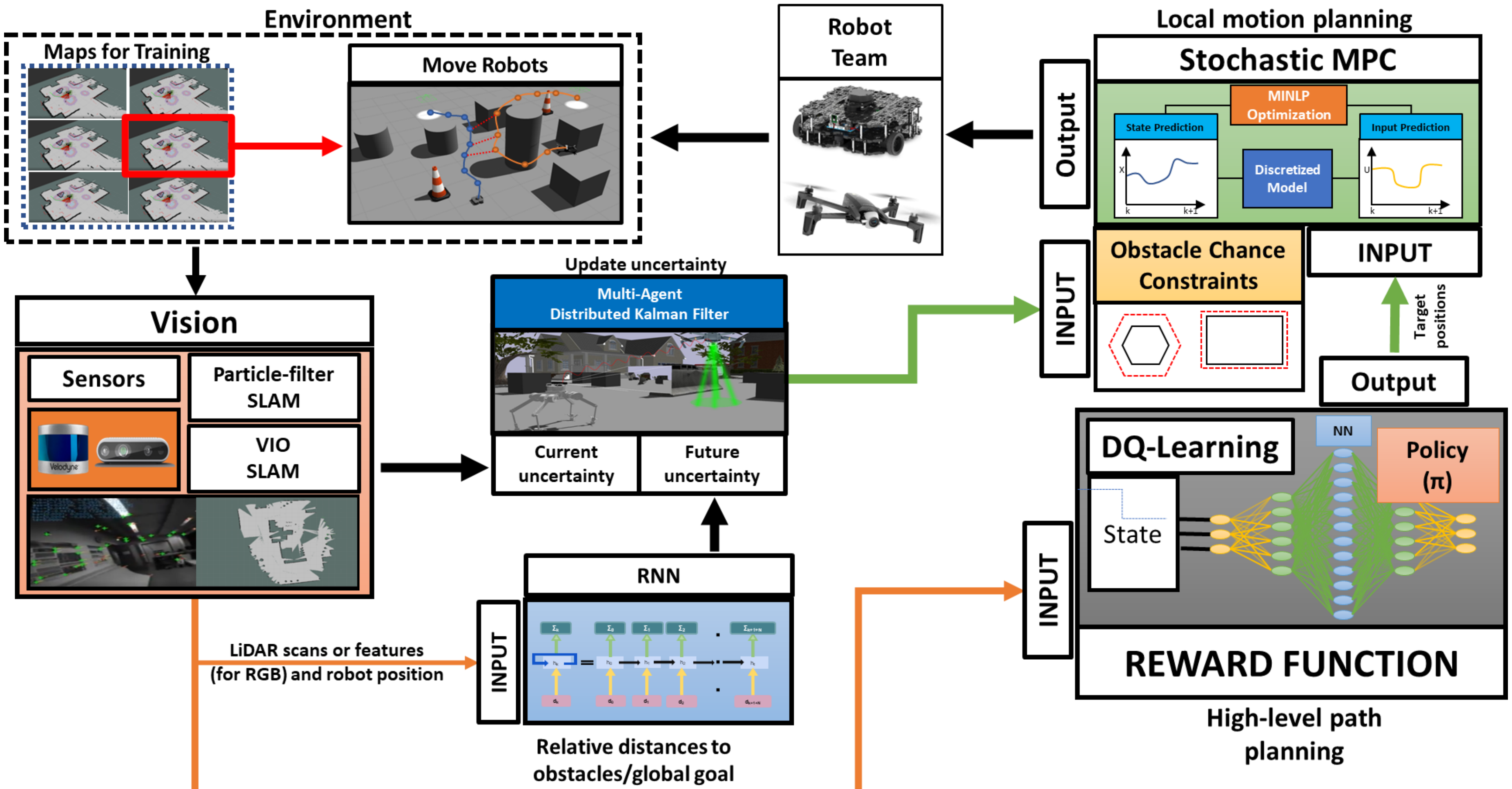
UCLAVISIONL

SABER: Data-Driven Motion Planner for Heterogeneous Robots

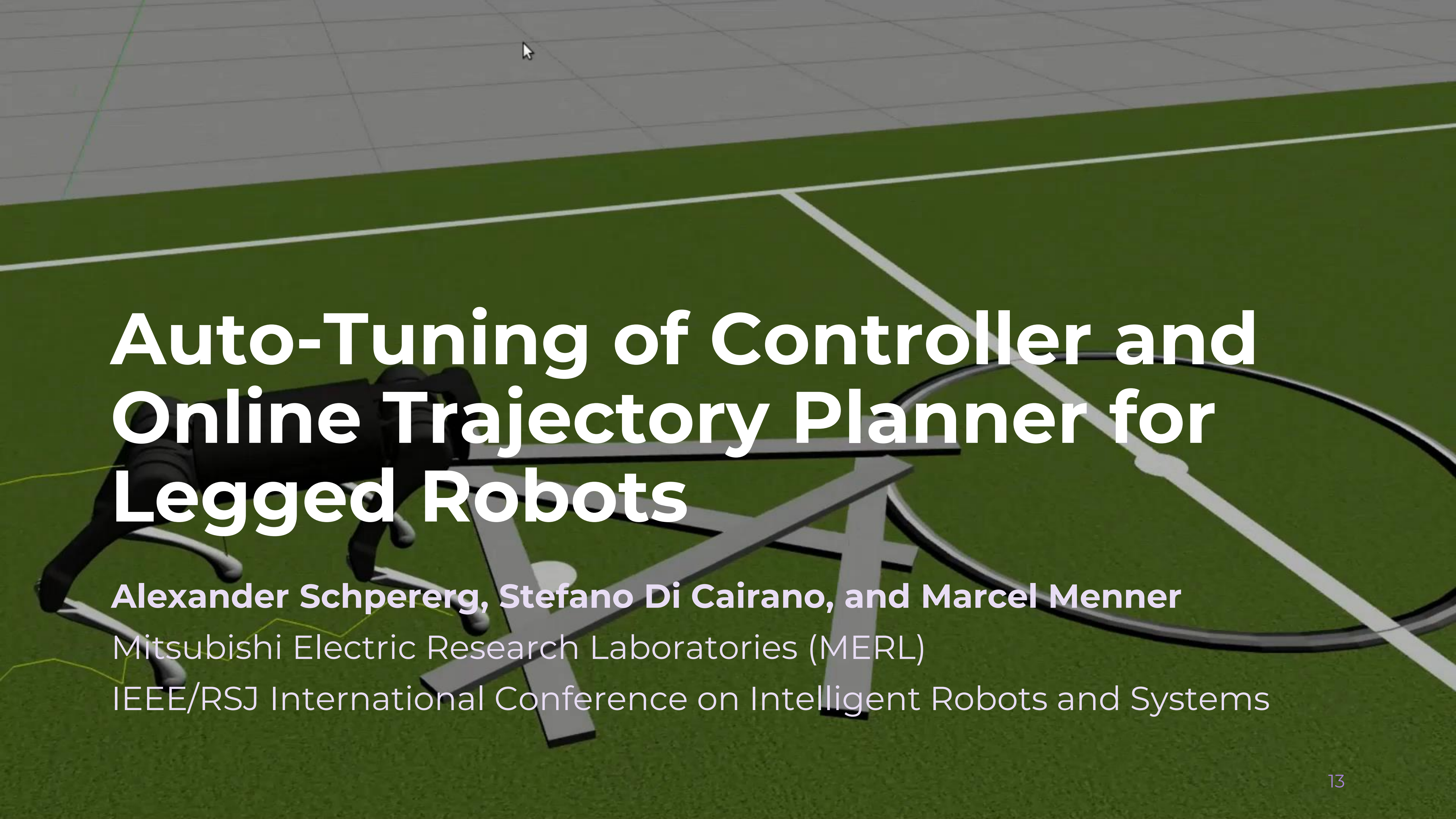
Alexander Schperberg, Stephanie Tsuei, Stefano Soatto, and Dennis Hong

This work was supported by a grant (N00014-15-1-2064) from the ONR.

All labs are affiliated with the University of California, Los Angeles, CA 90095, USA



Environment
Maps for Training
Move Robots
Robot Team
Local motion planning
Stochastic MPC
Output
MINLP Optimization
State Prediction
Input Prediction
Discretized Model
Obstacle Chance Constraints
INPUT
INPUT
Target positions
Output
Vision
Sensors
Particle-filter SLAM
VIO SLAM
Update uncertainty
Multi-Agent Distributed Kalman Filter
Current uncertainty
Future uncertainty
RNN
INPUT
LiDAR scans or features (for RGB) and robot position
Relative distances to obstacles/global goal
DQ-Learning
NN
Policy (π)
State
INPUT
REWARD FUNCTION
High-level path planning



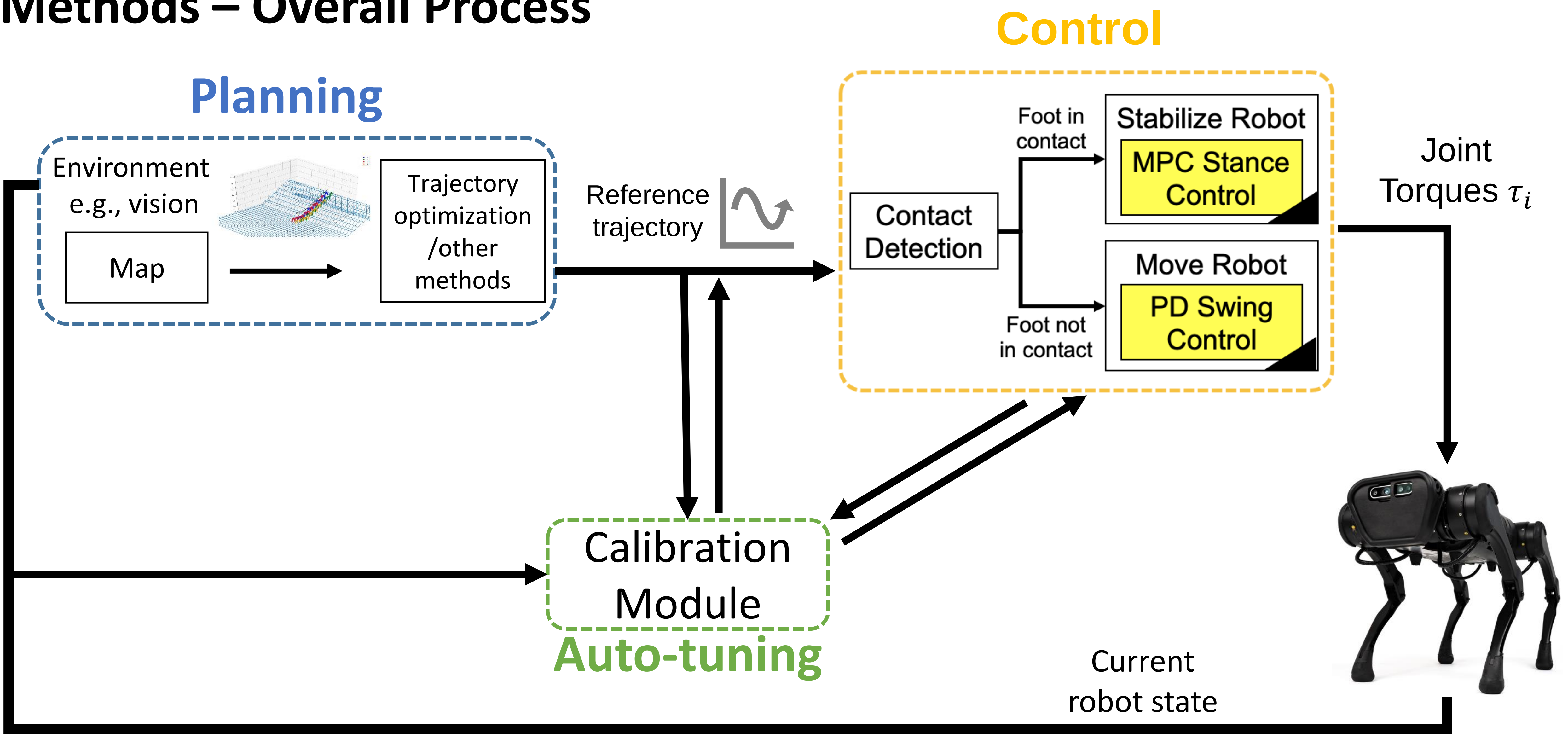
Auto-Tuning of Controller and Online Trajectory Planner for Legged Robots

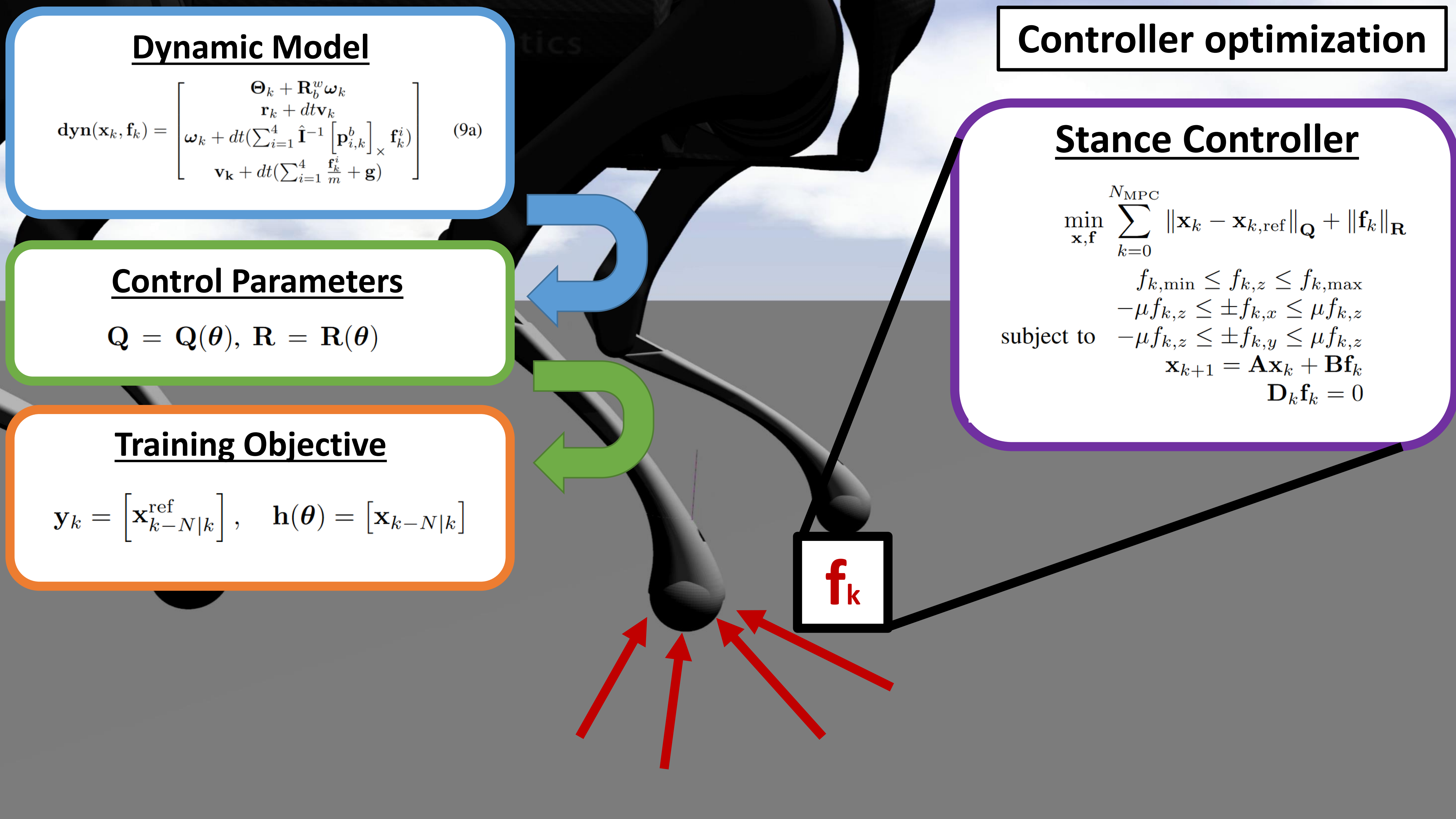
Alexander Schpererg, Stefano Di Cairano, and Marcel Menner

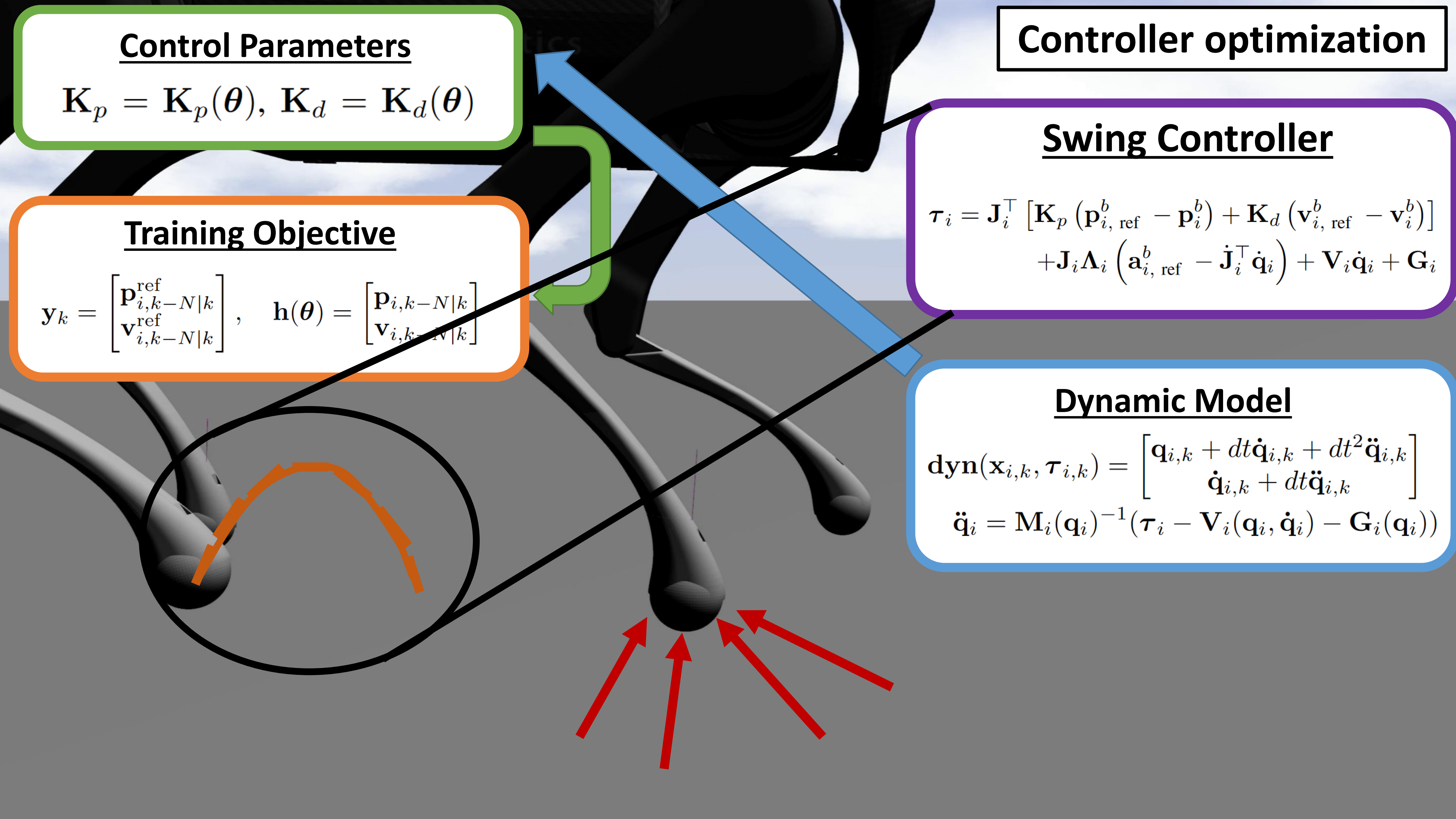
Mitsubishi Electric Research Laboratories (MERL)

IEEE/RSJ International Conference on Intelligent Robots and Systems

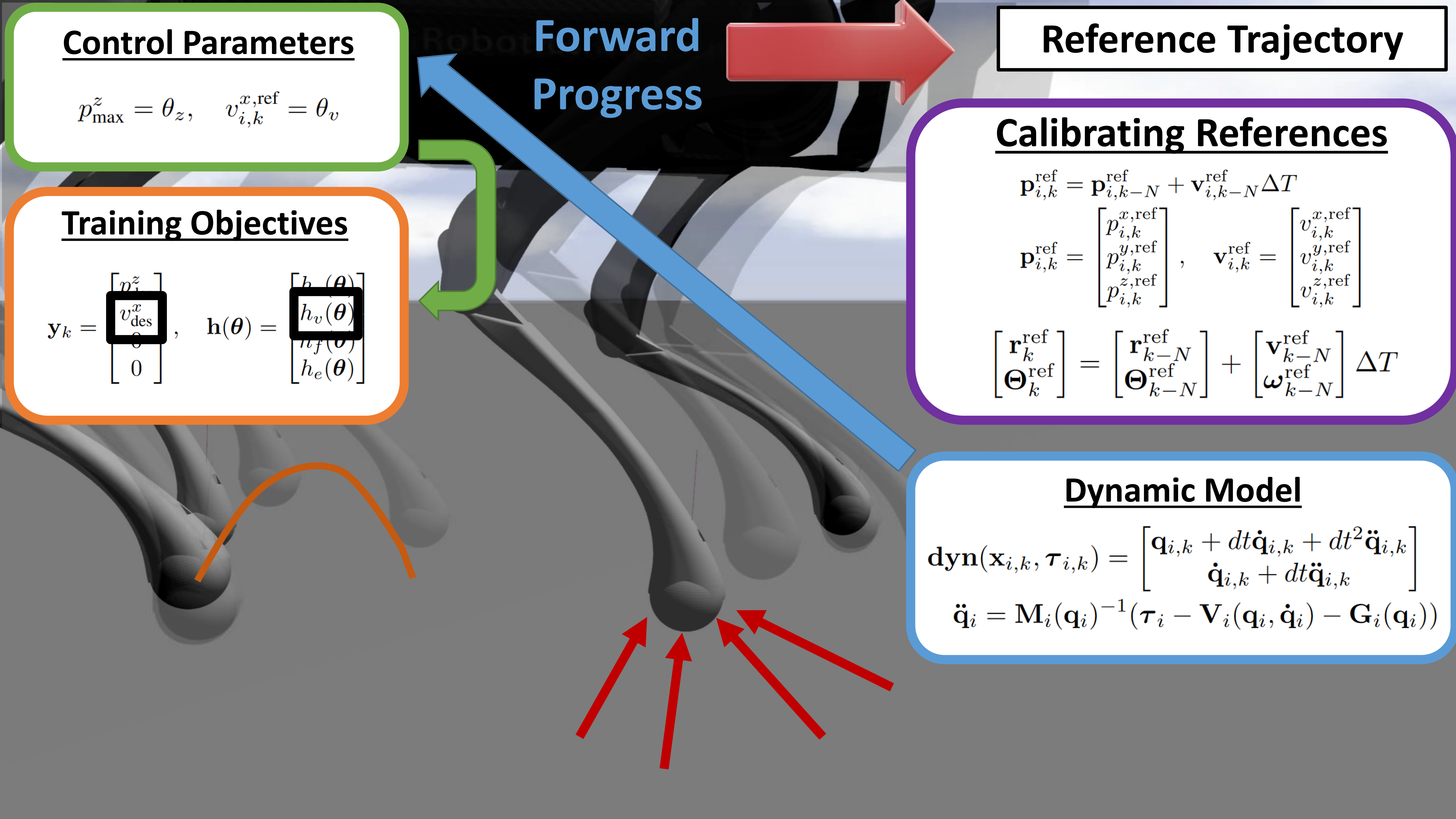
Methods – Overall Process

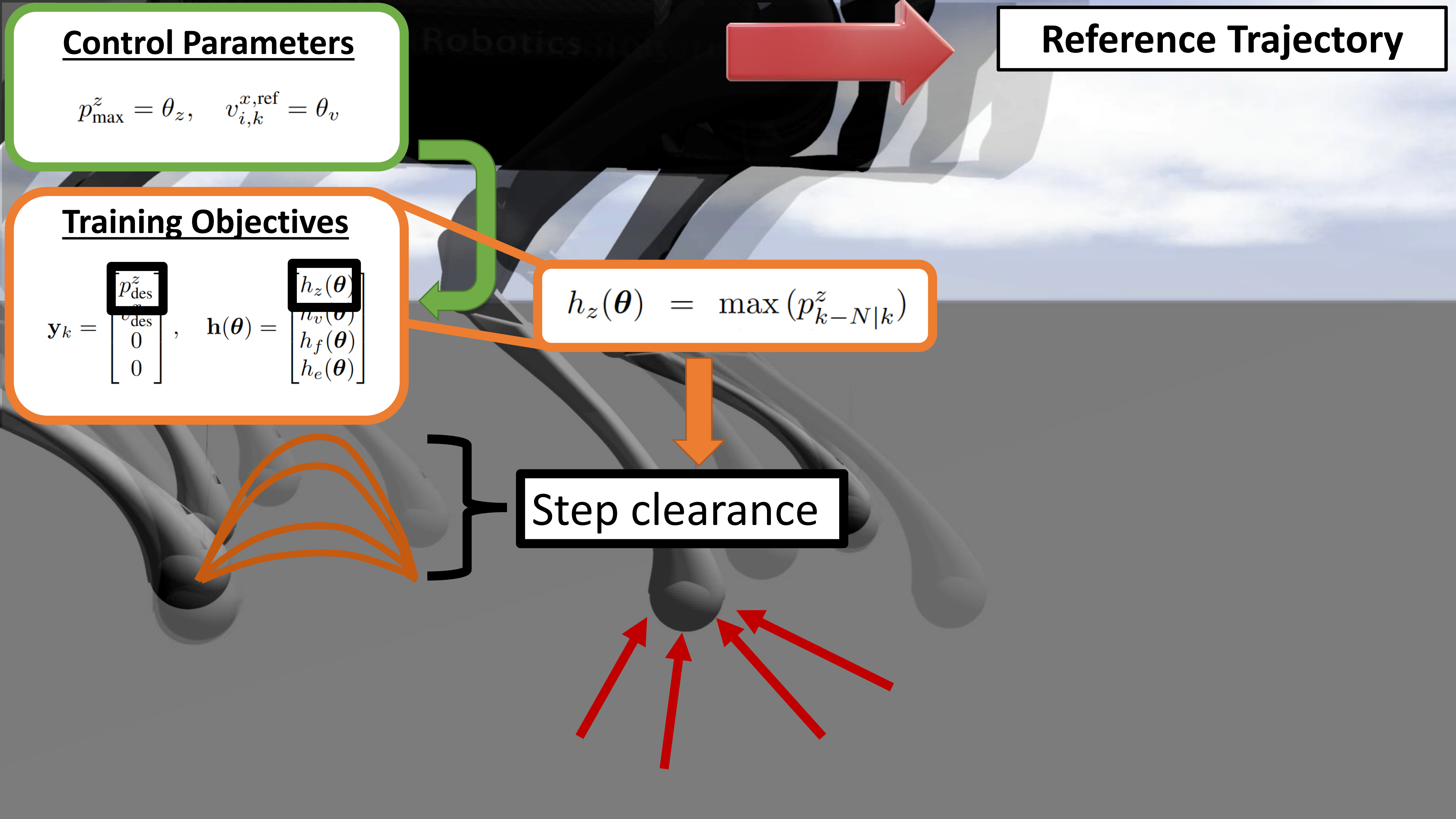



Dynamic Model
$\mathbf{dyn}(\mathbf{x}_k, \mathbf{f}_k) = \begin{bmatrix} \boldsymbol{\Theta}_k + \mathbf{R}_b^w \boldsymbol{\omega}_k \\ \mathbf{r}_k + dt\mathbf{v}_k \\ \boldsymbol{\omega}_k + dt(\sum_{i=1}^4 \hat{\mathbf{I}}^{-1} \left[\mathbf{p}_{i,k}^b\right]_\times \mathbf{f}_k^i) \\ \mathbf{v_k} + dt(\sum_{i=1}^4 \frac{\mathbf{f}_k^i}{m} + \mathbf{g}) \end{bmatrix}$ (9a)
Control Parameters
$\mathbf{Q} = \mathbf{Q}(\boldsymbol{\theta}), \ \mathbf{R} = \mathbf{R}(\boldsymbol{\theta})$
Training Objective
$\mathbf{y}_k = \left[\mathbf{x}_{k-N|k}^{\text{ref}}\right], \quad \mathbf{h}(\boldsymbol{\theta}) = \left[\mathbf{x}_{k-N|k}\right]$
Controller optimization
Stance Controller
$\min_{\mathbf{x},\mathbf{f}} \sum_{k=0}^{N_{\text{MPC}}} \|\mathbf{x}_k - \mathbf{x}_{k,\text{ref}}\|_{\mathbf{Q}} + \|\mathbf{f}_k\|_{\mathbf{R}}$
subject to $f_{k,\min} \leq f_{k,z} \leq f_{k,\max}$
$-\mu f_{k,z} \leq \pm f_{k,x} \leq \mu f_{k,z}$
$-\mu f_{k,z} \leq \pm f_{k,y} \leq \mu f_{k,z}$
$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{f}_k$
$\mathbf{D}_k \mathbf{f}_k = 0$
$\mathbf{f}_\mathbf{k}$

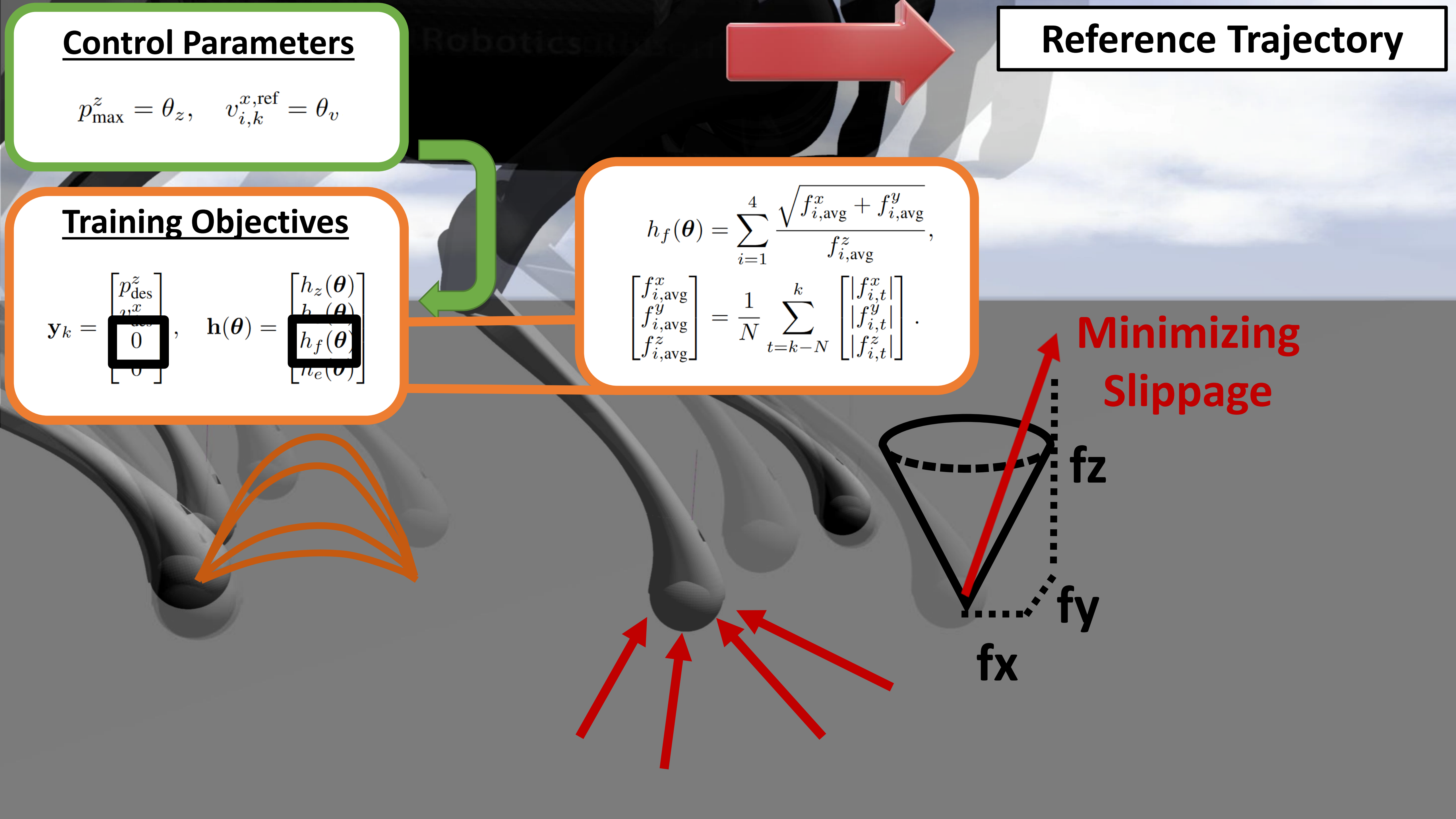


Controller optimization
Control Parameters
$\mathbf{K}_p = \mathbf{K}_p(\boldsymbol{\theta}),\ \mathbf{K}_d = \mathbf{K}_d(\boldsymbol{\theta})$
Training Objective
$\mathbf{y}_k = \begin{bmatrix} \mathbf{p}^{\text{ref}}_{i,k-N|k} \\ \mathbf{v}^{\text{ref}}_{i,k-N|k} \end{bmatrix}, \quad \mathbf{h}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{p}_{i,k-N|k} \\ \mathbf{v}_{i,k-N|k} \end{bmatrix}$
Swing Controller
$\boldsymbol{\tau}_i = \mathbf{J}_i^\top \left[\mathbf{K}_p \left(\mathbf{p}^b_{i,\text{ ref}} - \mathbf{p}^b_i\right) + \mathbf{K}_d \left(\mathbf{v}^b_{i,\text{ ref}} - \mathbf{v}^b_i\right)\right] + \mathbf{J}_i \boldsymbol{\Lambda}_i \left(\mathbf{a}^b_{i,\text{ ref}} - \dot{\mathbf{J}}_i^\top \dot{\mathbf{q}}_i\right) + \mathbf{V}_i \dot{\mathbf{q}}_i + \mathbf{G}_i$
Dynamic Model
$\mathbf{dyn}(\mathbf{x}_{i,k}, \boldsymbol{\tau}_{i,k}) = \begin{bmatrix} \mathbf{q}_{i,k} + dt\dot{\mathbf{q}}_{i,k} + dt^2\ddot{\mathbf{q}}_{i,k} \\ \dot{\mathbf{q}}_{i,k} + dt\ddot{\mathbf{q}}_{i,k} \end{bmatrix}$
$\ddot{\mathbf{q}}_i = \mathbf{M}_i(\mathbf{q}_i)^{-1}(\boldsymbol{\tau}_i - \mathbf{V}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) - \mathbf{G}_i(\mathbf{q}_i))$

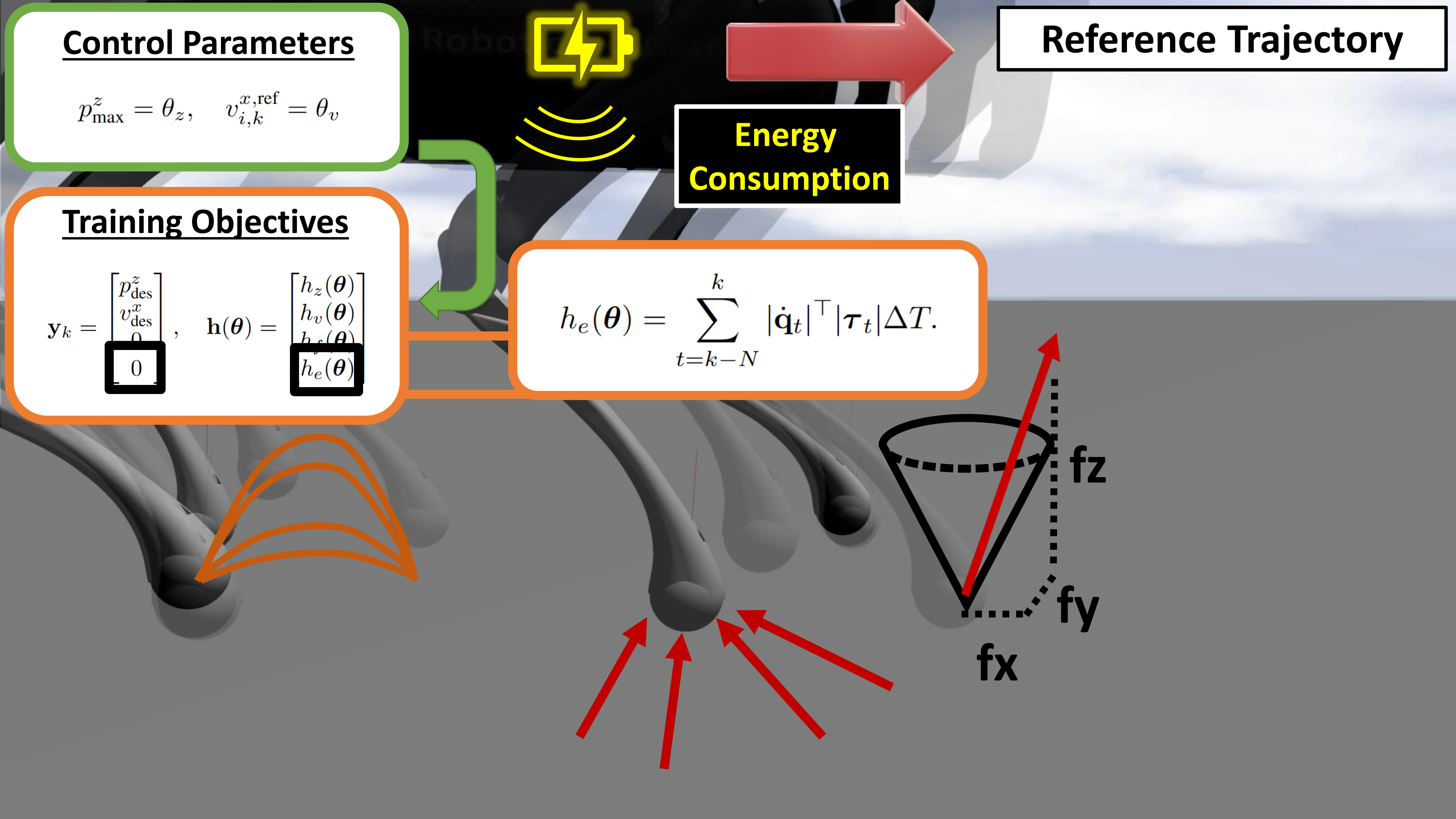

Control Parameters
$p^z_{\max} = \theta_z, \quad v^{x,\mathrm{ref}}_{i,k} = \theta_v$
Training Objectives
$\mathbf{y}_k = \begin{bmatrix} p^z_i \\ \boxed{v^x_{\mathrm{des}}} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{h}(\boldsymbol{\theta}) = \begin{bmatrix} h(\boldsymbol{\theta}) \\ \boxed{h_v(\boldsymbol{\theta})} \\ h_f(\boldsymbol{\theta}) \\ h_e(\boldsymbol{\theta}) \end{bmatrix}$
Forward Progress
Reference Trajectory
Calibrating References
$\mathbf{p}^{\mathrm{ref}}_{i,k} = \mathbf{p}^{\mathrm{ref}}_{i,k-N} + \mathbf{v}^{\mathrm{ref}}_{i,k-N}\Delta T$
$\mathbf{p}^{\mathrm{ref}}_{i,k} = \begin{bmatrix} p^{x,\mathrm{ref}}_{i,k} \\ p^{y,\mathrm{ref}}_{i,k} \\ p^{z,\mathrm{ref}}_{i,k} \end{bmatrix}, \quad \mathbf{v}^{\mathrm{ref}}_{i,k} = \begin{bmatrix} v^{x,\mathrm{ref}}_{i,k} \\ v^{y,\mathrm{ref}}_{i,k} \\ v^{z,\mathrm{ref}}_{i,k} \end{bmatrix}$
$\begin{bmatrix} \mathbf{r}^{\mathrm{ref}}_k \\ \boldsymbol{\Theta}^{\mathrm{ref}}_k \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{\mathrm{ref}}_{k-N} \\ \boldsymbol{\Theta}^{\mathrm{ref}}_{k-N} \end{bmatrix} + \begin{bmatrix} \mathbf{v}^{\mathrm{ref}}_{k-N} \\ \boldsymbol{\omega}^{\mathrm{ref}}_{k-N} \end{bmatrix} \Delta T$
Dynamic Model
$\mathbf{dyn}(\mathbf{x}_{i,k}, \boldsymbol{\tau}_{i,k}) = \begin{bmatrix} \mathbf{q}_{i,k} + dt\dot{\mathbf{q}}_{i,k} + dt^2\ddot{\mathbf{q}}_{i,k} \\ \dot{\mathbf{q}}_{i,k} + dt\ddot{\mathbf{q}}_{i,k} \end{bmatrix}$
$\ddot{\mathbf{q}}_i = \mathbf{M}_i(\mathbf{q}_i)^{-1}(\boldsymbol{\tau}_i - \mathbf{V}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) - \mathbf{G}_i(\mathbf{q}_i))$



Control Parameters
$p^z_{\max} = \theta_z, \quad v^{x,\mathrm{ref}}_{i,k} = \theta_v$
Training Objectives
$\mathbf{y}_k = \begin{bmatrix} \boxed{p^z_{\mathrm{des}}} \\ v^x_{\mathrm{des}} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{h}(\boldsymbol{\theta}) = \begin{bmatrix} \boxed{h_z(\boldsymbol{\theta})} \\ h_v(\boldsymbol{\theta}) \\ h_f(\boldsymbol{\theta}) \\ h_e(\boldsymbol{\theta}) \end{bmatrix}$
Reference Trajectory
$h_z(\boldsymbol{\theta}) = \max\left(p^z_{k-N|k}\right)$
Step clearance



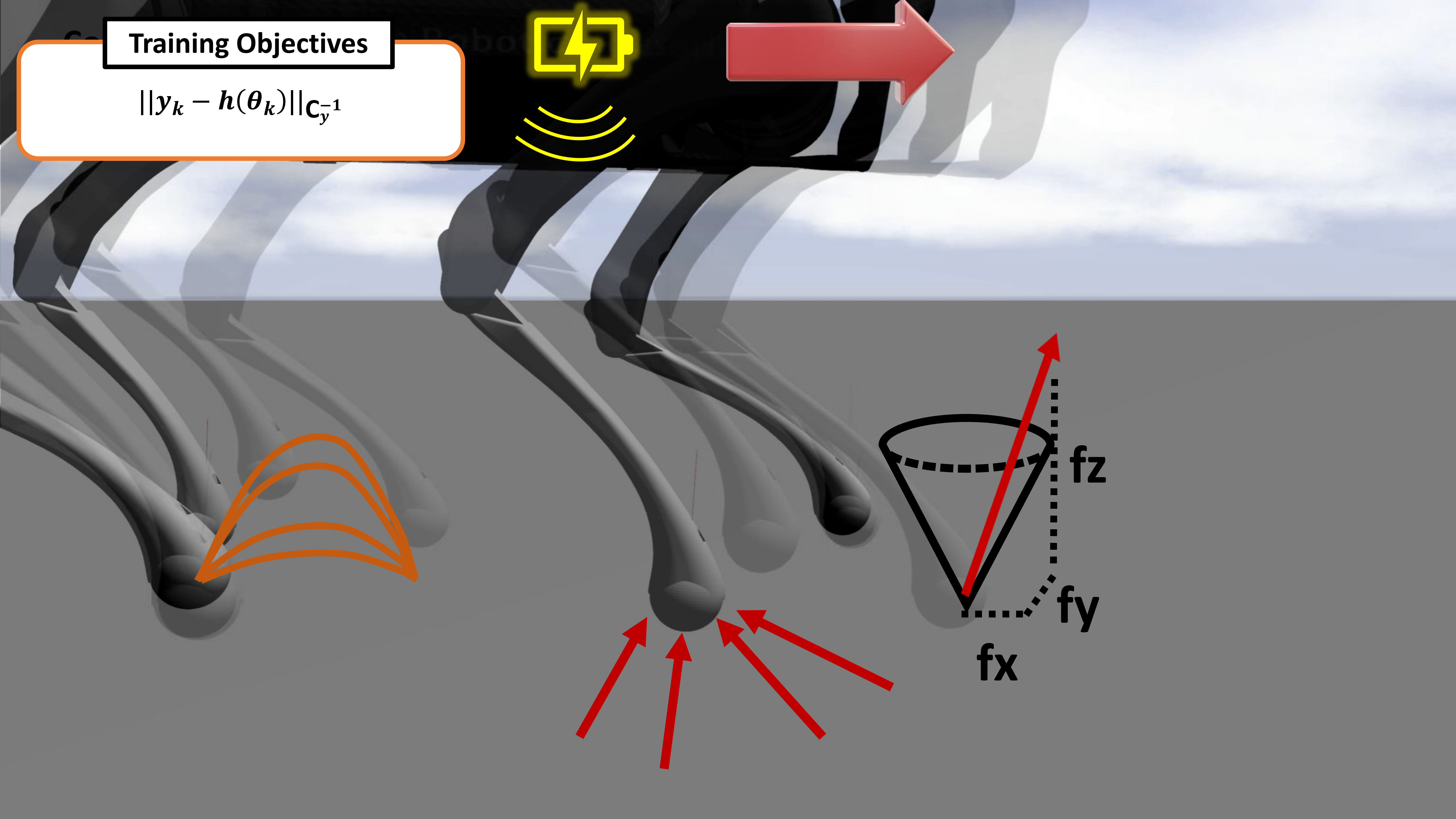
Control Parameters
$p^z_{\max} = \theta_z, \quad v^{x,\mathrm{ref}}_{i,k} = \theta_v$
Training Objectives
$\mathbf{y}_k = \begin{bmatrix} p^z_{\mathrm{des}} \\ v^x_{\mathrm{des}} \\ \boxed{0} \\ 0 \end{bmatrix}, \quad \mathbf{h}(\boldsymbol{\theta}) = \begin{bmatrix} h_z(\boldsymbol{\theta}) \\ h_v(\boldsymbol{\theta}) \\ \boxed{h_f(\boldsymbol{\theta})} \\ h_e(\boldsymbol{\theta}) \end{bmatrix}$
$h_f(\boldsymbol{\theta}) = \sum_{i=1}^{4} \frac{\sqrt{f^x_{i,\mathrm{avg}} + f^y_{i,\mathrm{avg}}}}{f^z_{i,\mathrm{avg}}},$
$\begin{bmatrix} f^x_{i,\mathrm{avg}} \\ f^y_{i,\mathrm{avg}} \\ f^z_{i,\mathrm{avg}} \end{bmatrix} = \frac{1}{N} \sum_{t=k-N}^{k} \begin{bmatrix} |f^x_{i,t}| \\ |f^y_{i,t}| \\ |f^z_{i,t}| \end{bmatrix}.$
Reference Trajectory
Minimizing Slippage
fz
fy
fx



Control Parameters
$p^z_{\max} = \theta_z, \quad v^{x,\mathrm{ref}}_{i,k} = \theta_v$
Training Objectives
$\mathbf{y}_k = \begin{bmatrix} p^z_{\mathrm{des}} \\ v^x_{\mathrm{des}} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{h}(\boldsymbol{\theta}) = \begin{bmatrix} h_z(\boldsymbol{\theta}) \\ h_v(\boldsymbol{\theta}) \\ h_f(\boldsymbol{\theta}) \\ h_e(\boldsymbol{\theta}) \end{bmatrix}$
Energy Consumption
$h_e(\boldsymbol{\theta}) = \sum_{t=k-N}^{k} |\dot{\mathbf{q}}_t|^\top |\boldsymbol{\tau}_t| \Delta T.$
Reference Trajectory
fz
fy
fx

Training Objectives

$$||y_k - h(\theta_k)||_{C_y^{-1}}$$



RoMeLa
ROBOTICS & MECHANISMS LABORATORY

Real-to-Sim: Deep Learning with Auto-Tuning to Predict Residual Errors using Sparse Data

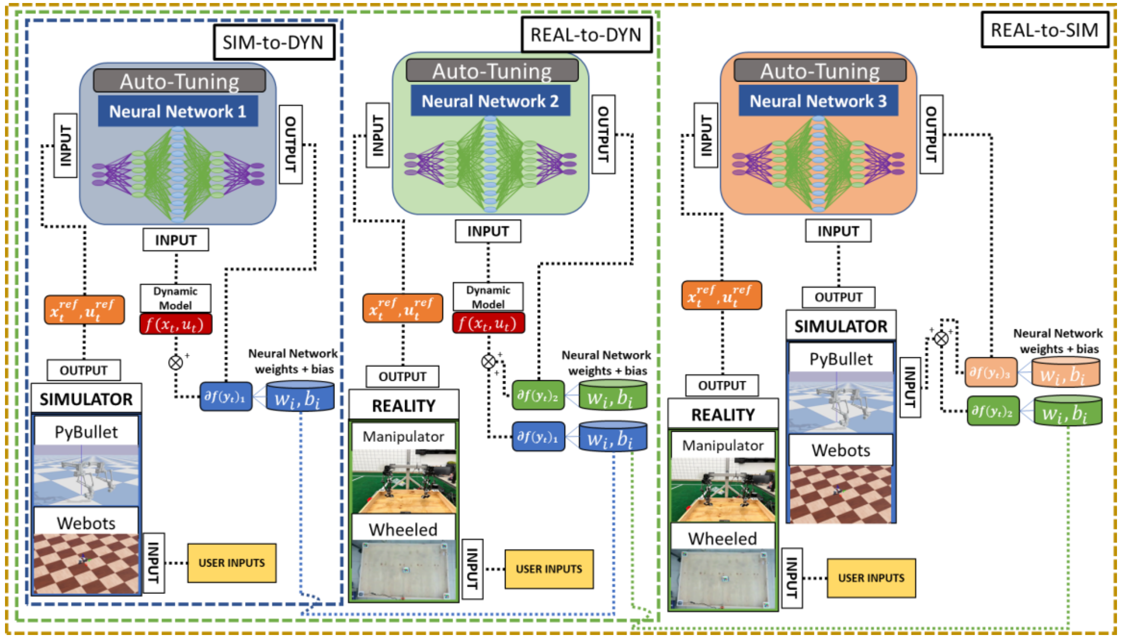
Alexander Schperberg*, Yusuke Tanaka, Feng Xu

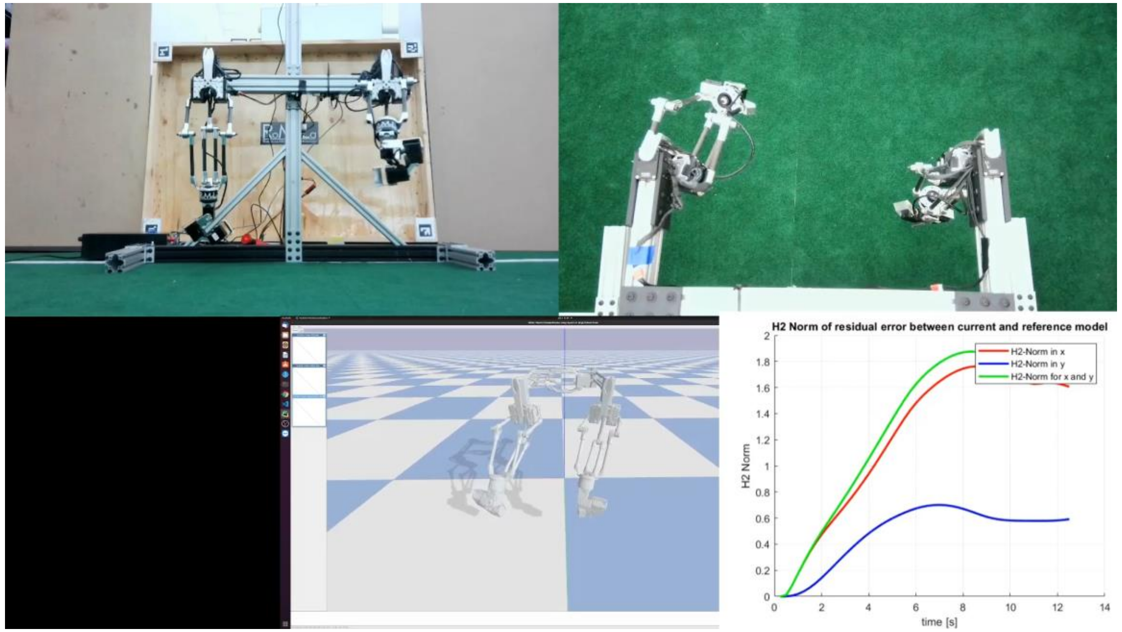
Marcel Menner, and Dennis Hong

*Corresponding author

email: aschperb@gmail.com

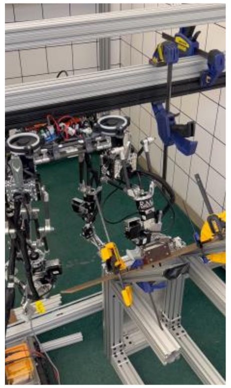
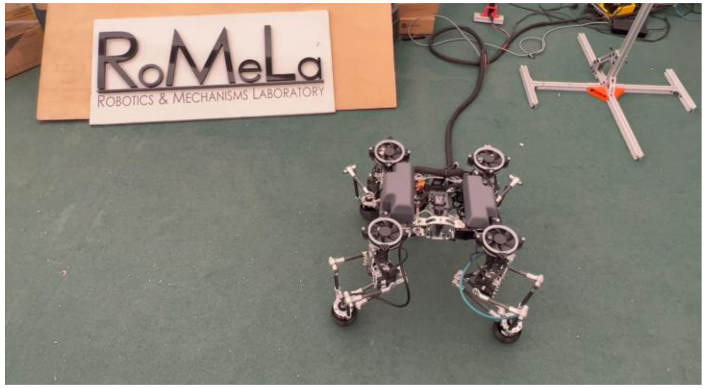
UCLA

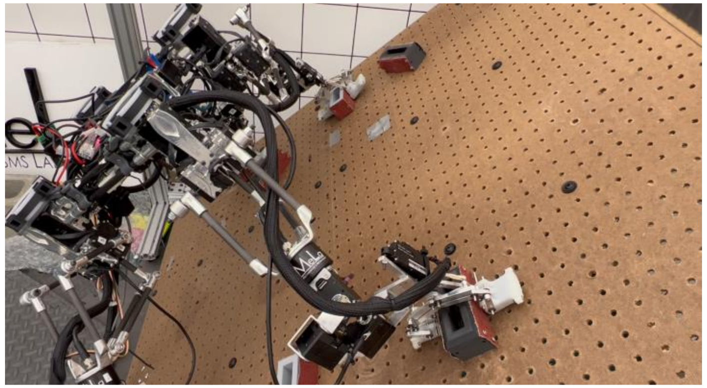

SIM-to-DYN
REAL-to-DYN
REAL-to-SIM
Auto-Tuning
Neural Network 1
Neural Network 2
Neural Network 3
INPUT
OUTPUT
Dynamic Model
Neural Network weights + bias
SIMULATOR
PyBullet
Webots
REALITY
Manipulator
Wheeled
USER INPUTS


H2 Norm of residual error between current and reference model
H2-Norm in x
H2-Norm in y
H2-Norm for x and y
H2 Norm
time [s]

Auto-Tuning Force Controller for Contact-Rich Robotic Systems using an Unscented Kalman Filter

Alexander Schperberg*, Yuki Shirai, Xuan Lin, Yusuke Tanaka, and Dennis Hong

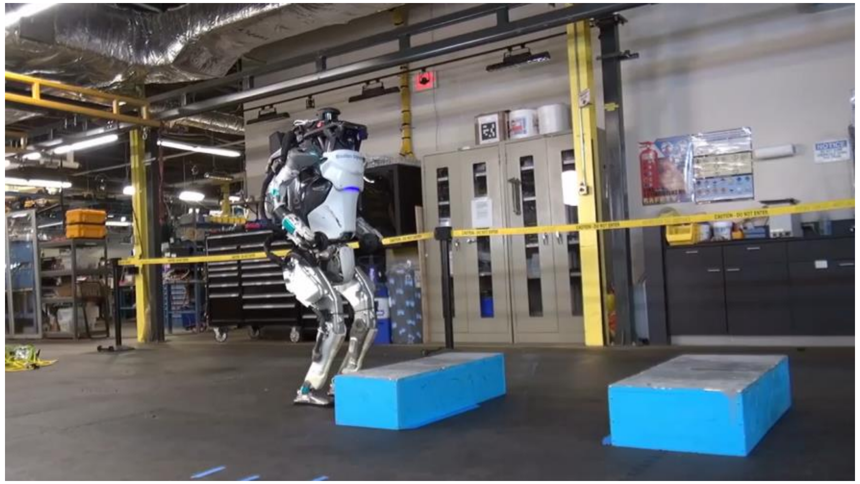


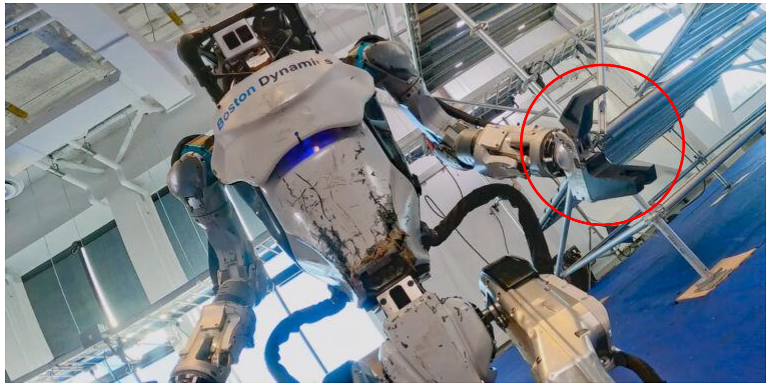
***Corresponding author**

email: aschperb@gmail.com

Combining the experience of past works towards autonomous locomotion and manipulation

- For robots to be not just autonomous but useful for diverse sets of tasks we need to do loco-manipulation

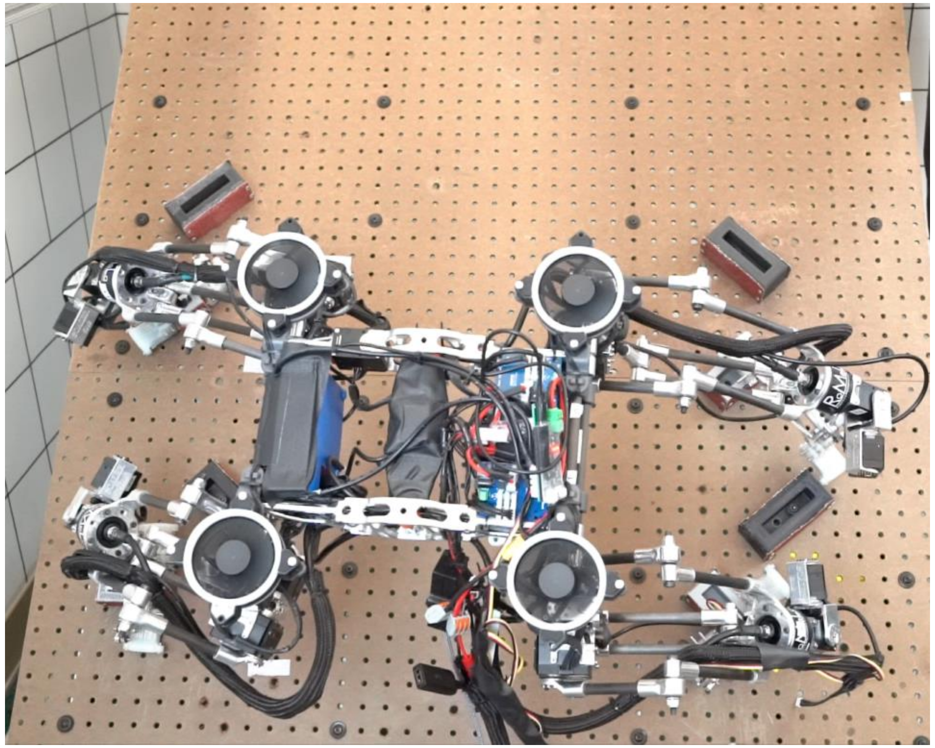




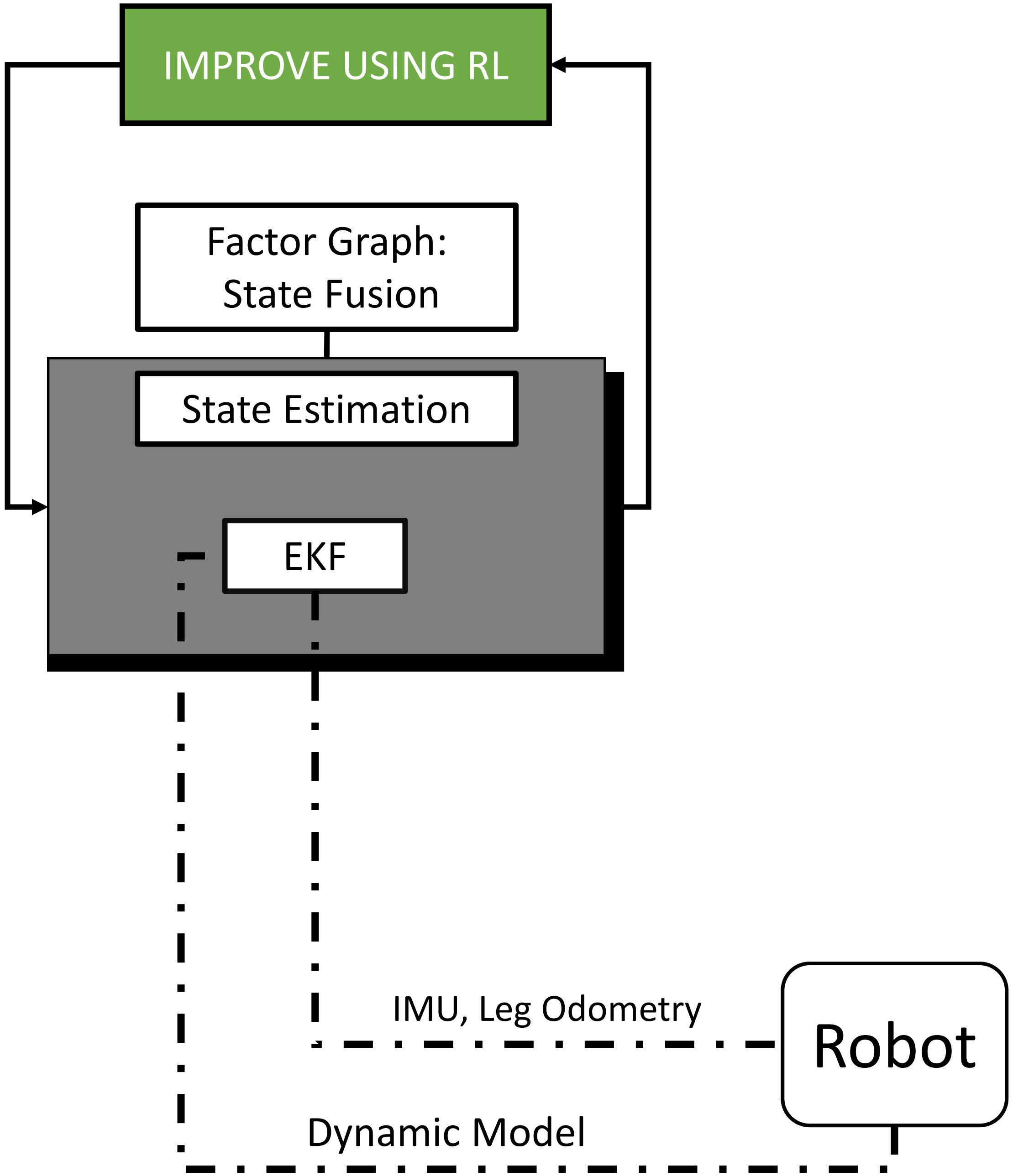
Project Objective

- A climbing robot is the perfect to test loco-manipulation problems
- Must autonomously climb over a **discrete** surface (locomotion and manipulation through grasping and object)
- Need to address **estimation**, **vision** (i.e., mapping the environment), **planning** (finding a graspable hold), and **control** (moving a limb to the bouldering hold in the correct configuration and smoothly) simultaneously

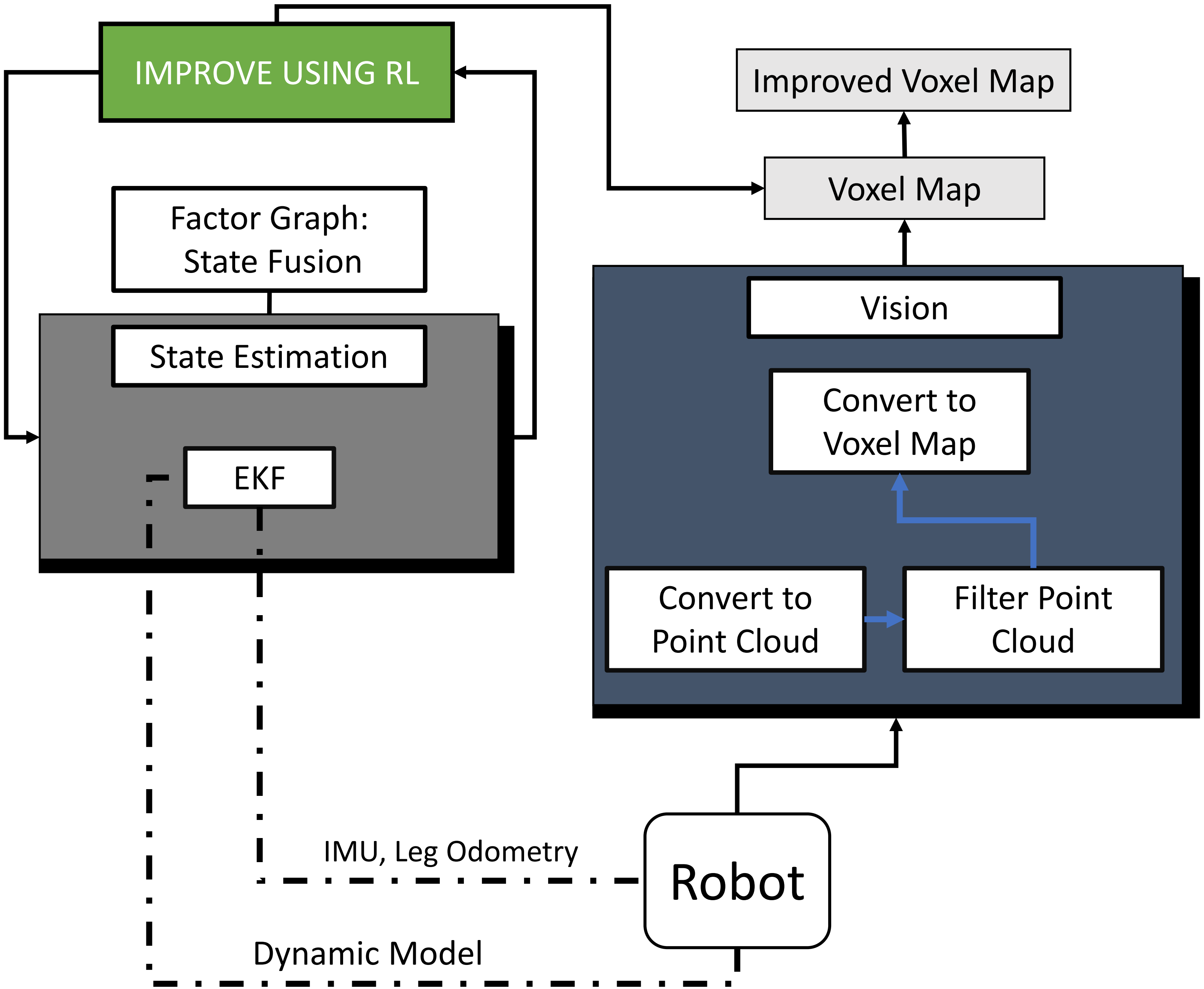
Robot to be used - SCALER



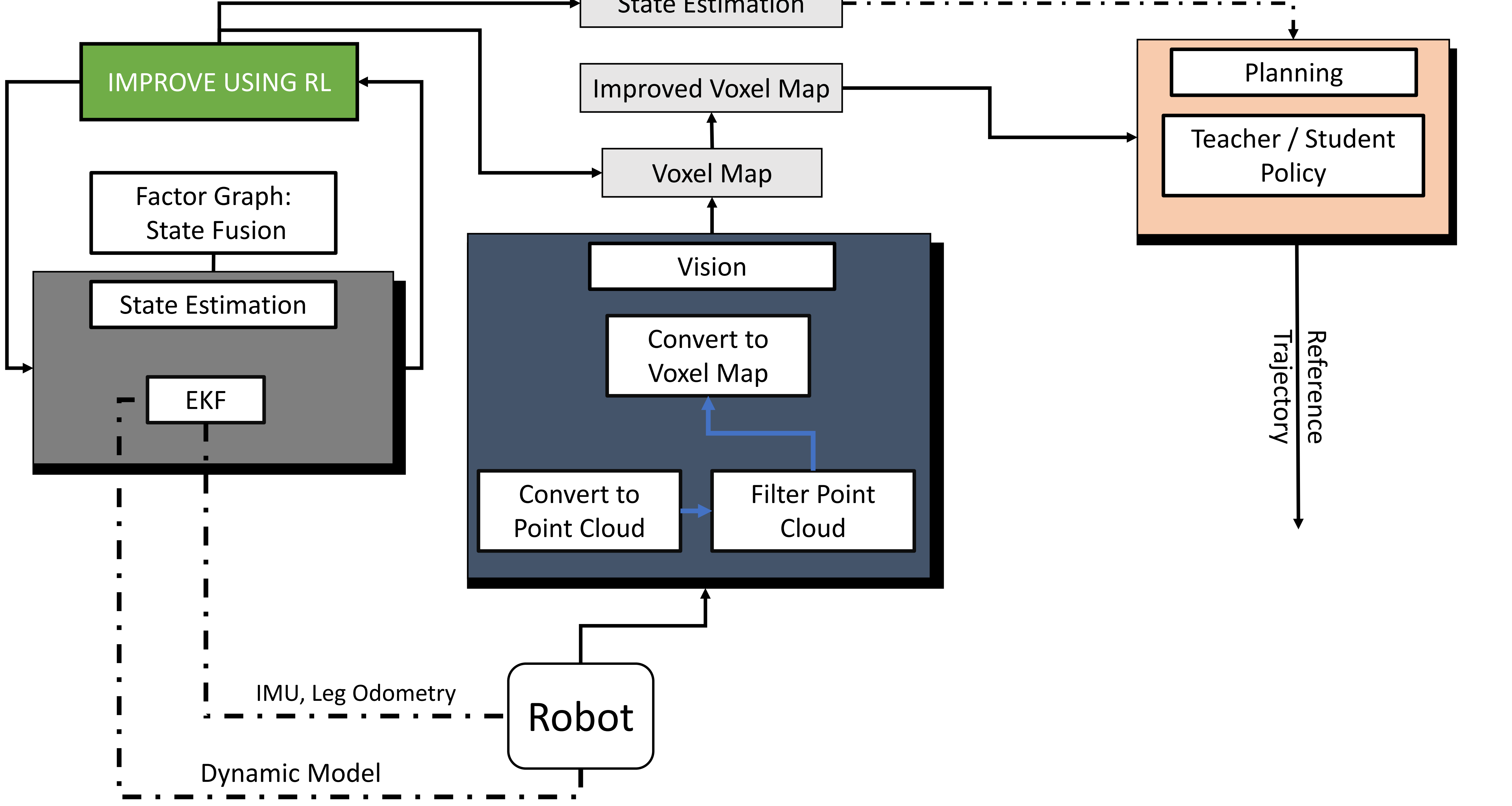
Method flow chart



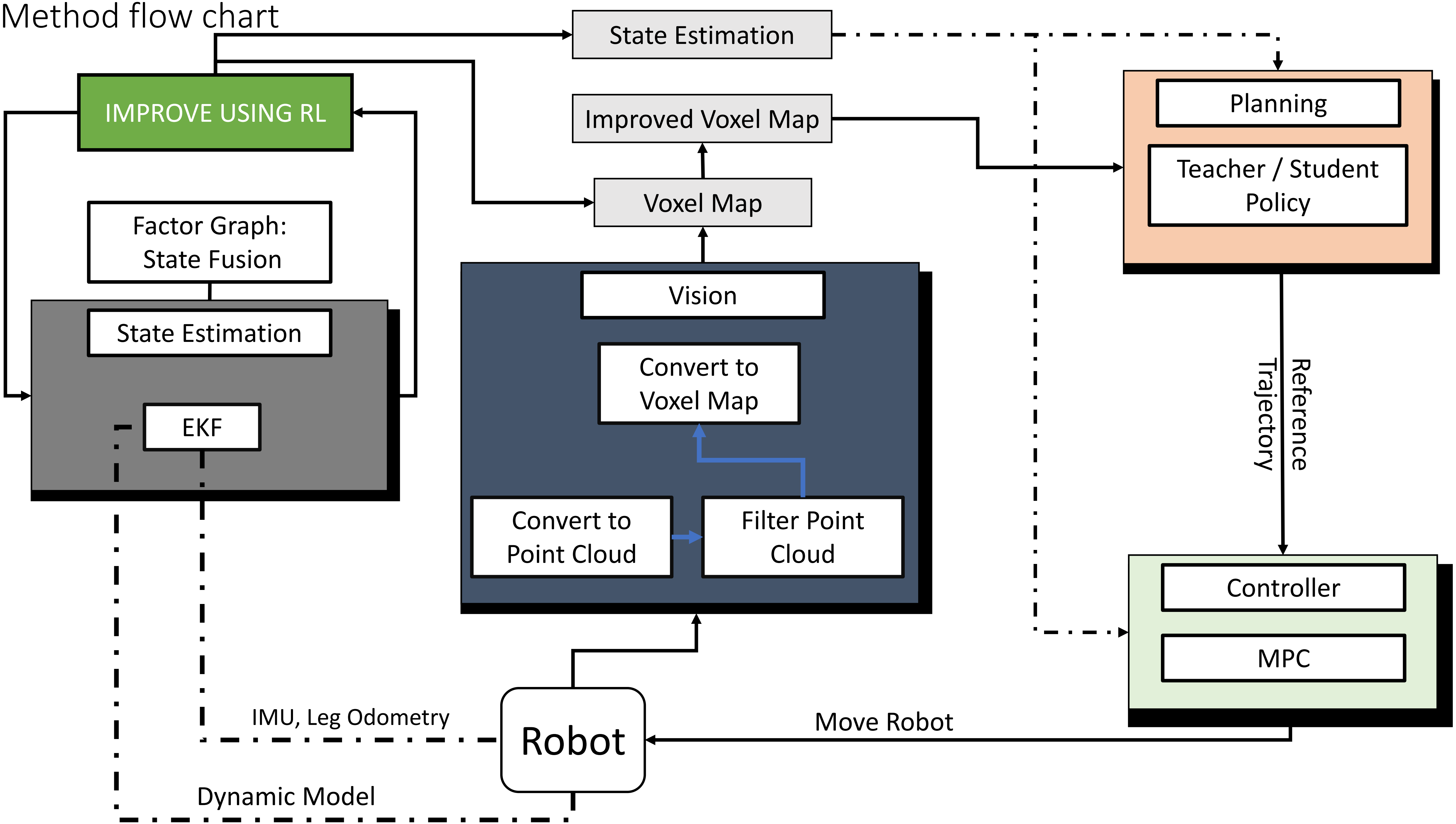
Method flow chart



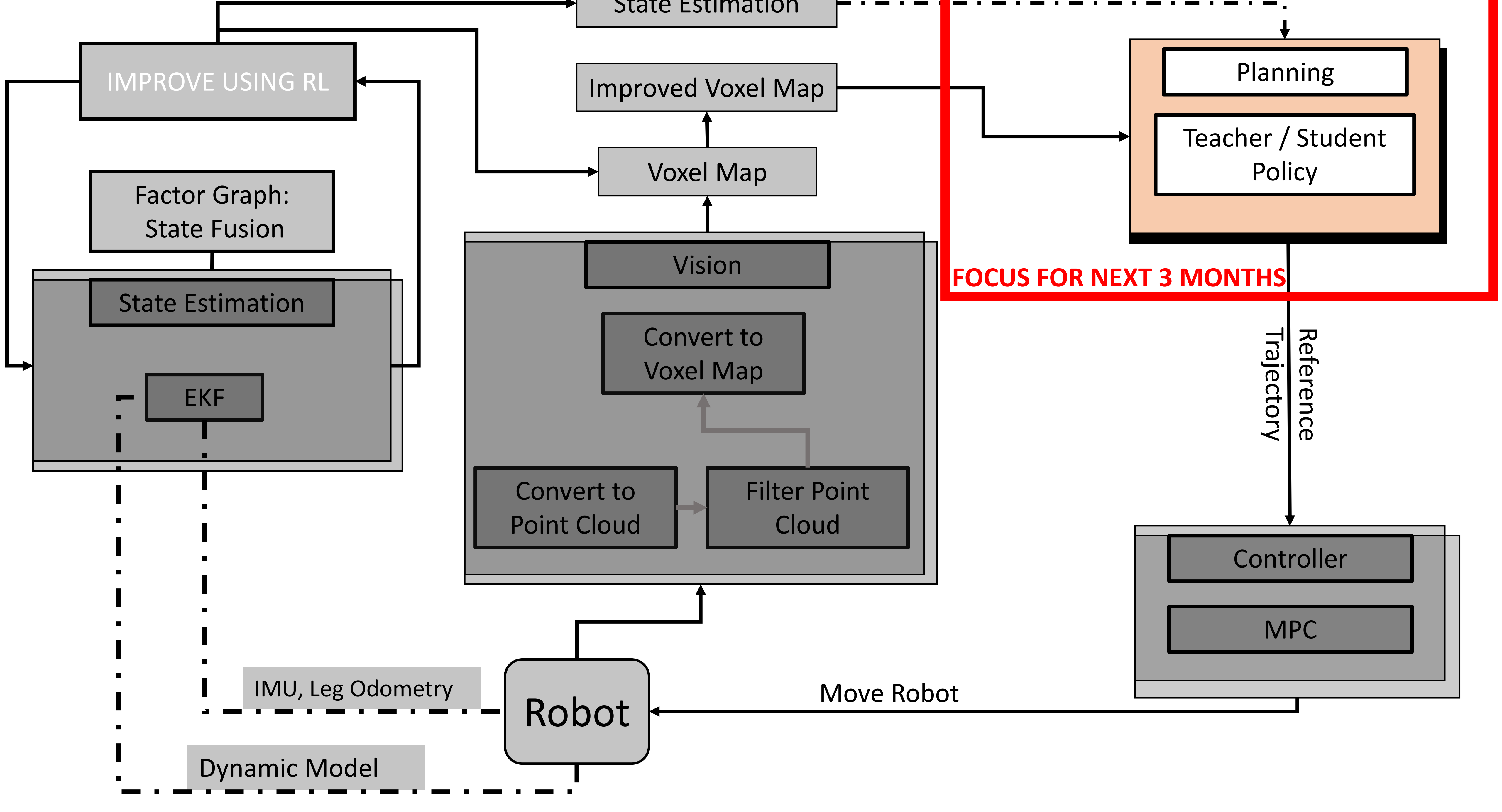
Method flow chart



Method flow chart

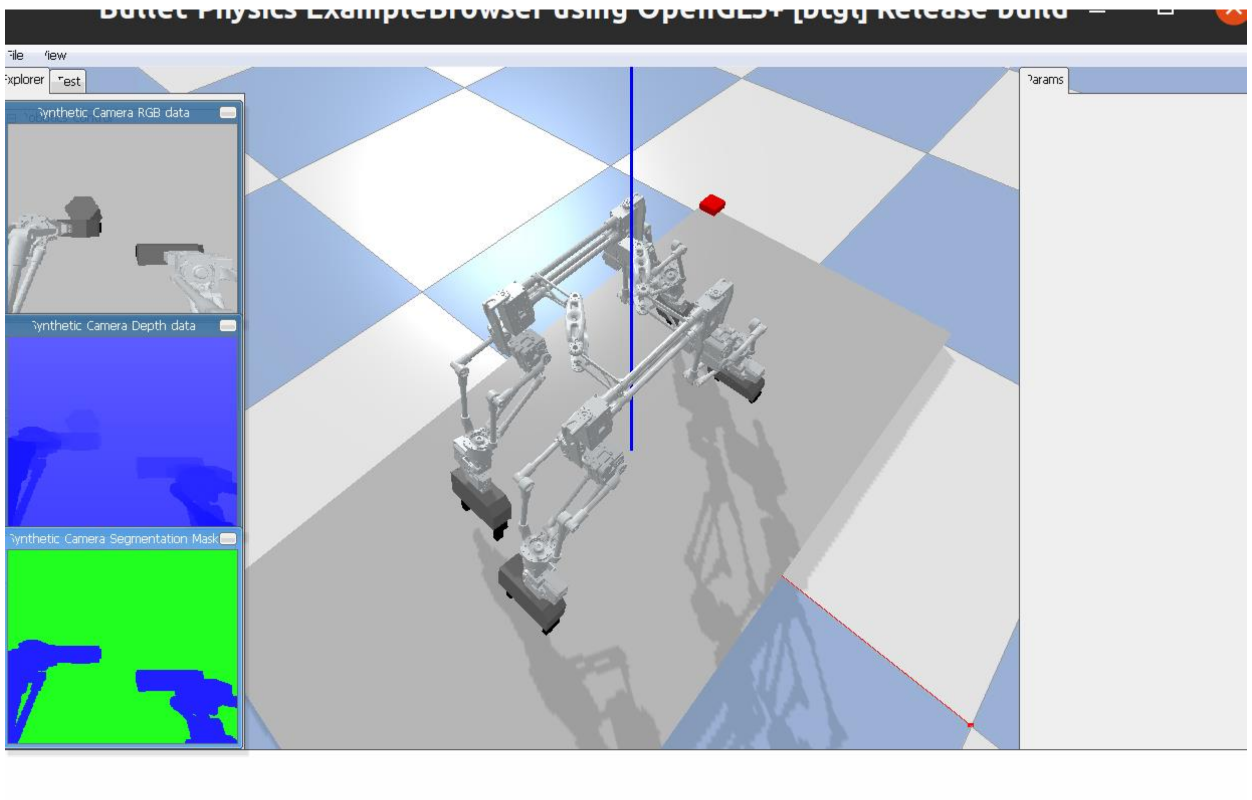


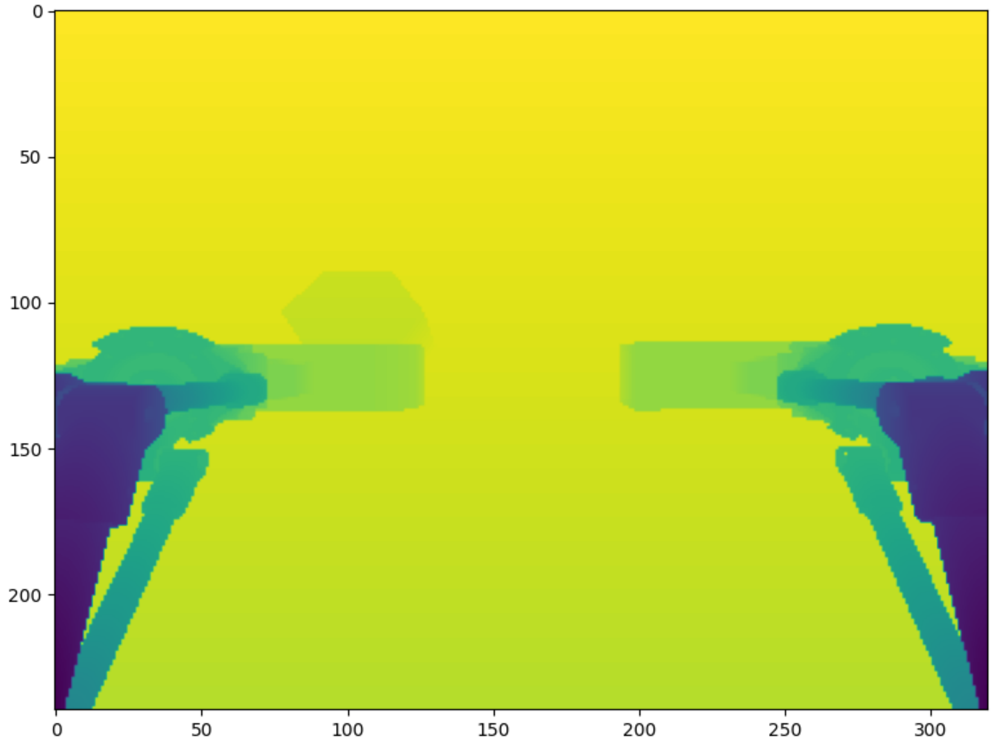
Method flow chart



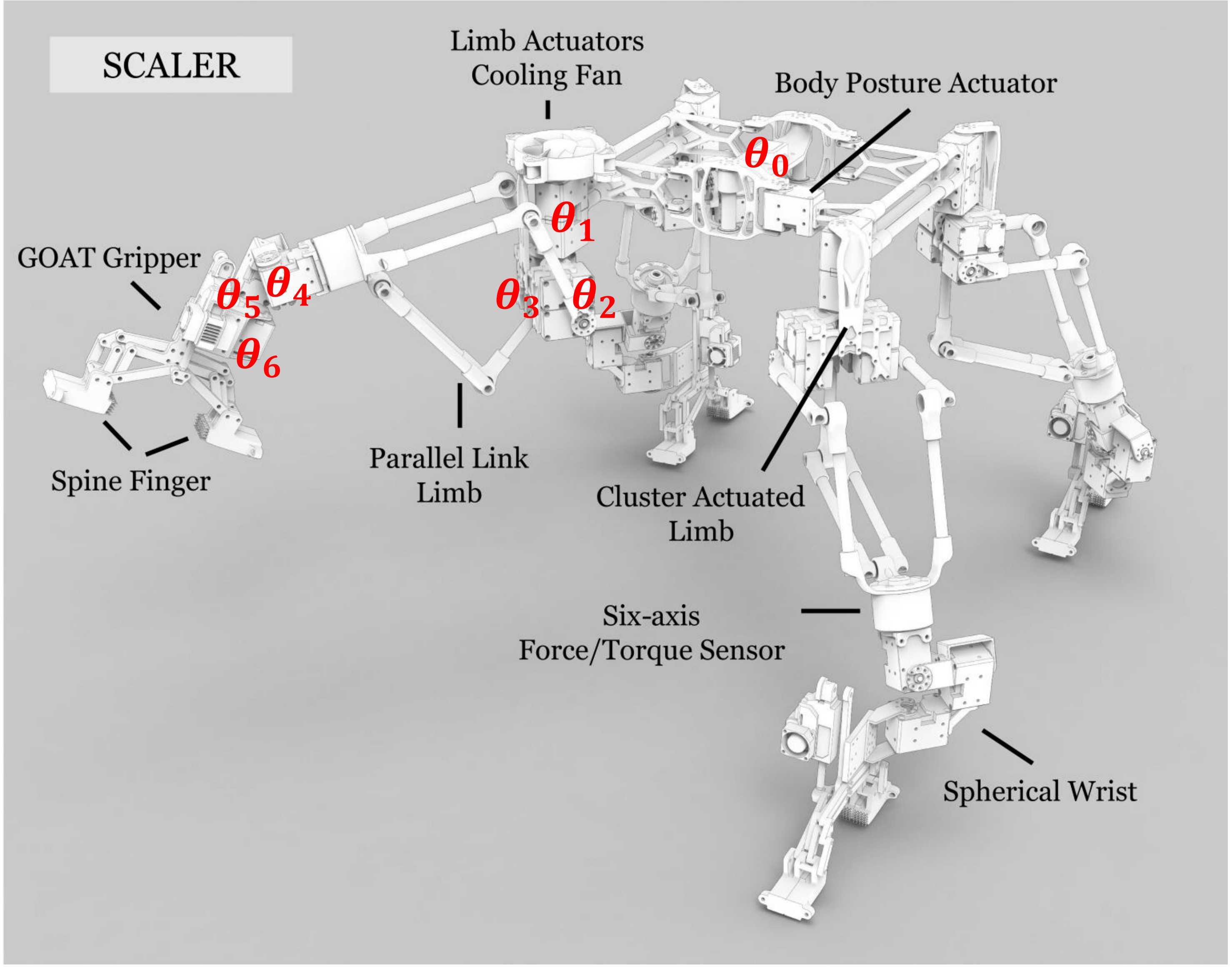
Building the Simulator

Dealing with Depth Images

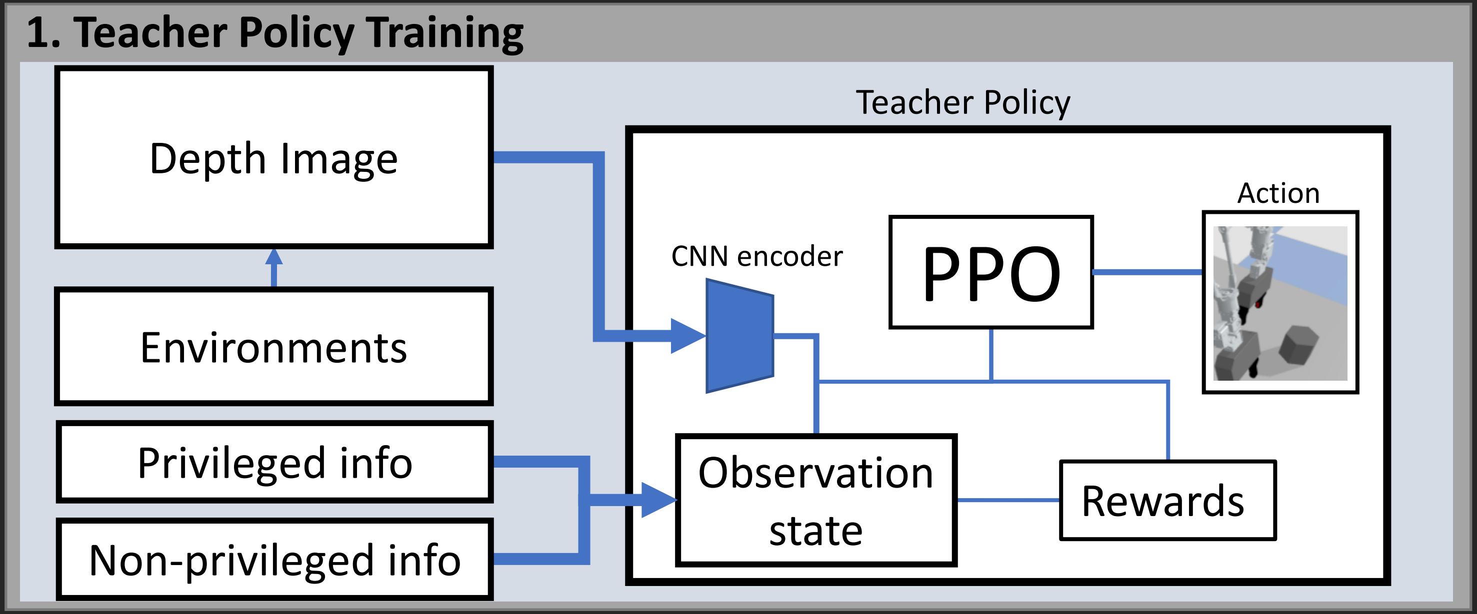




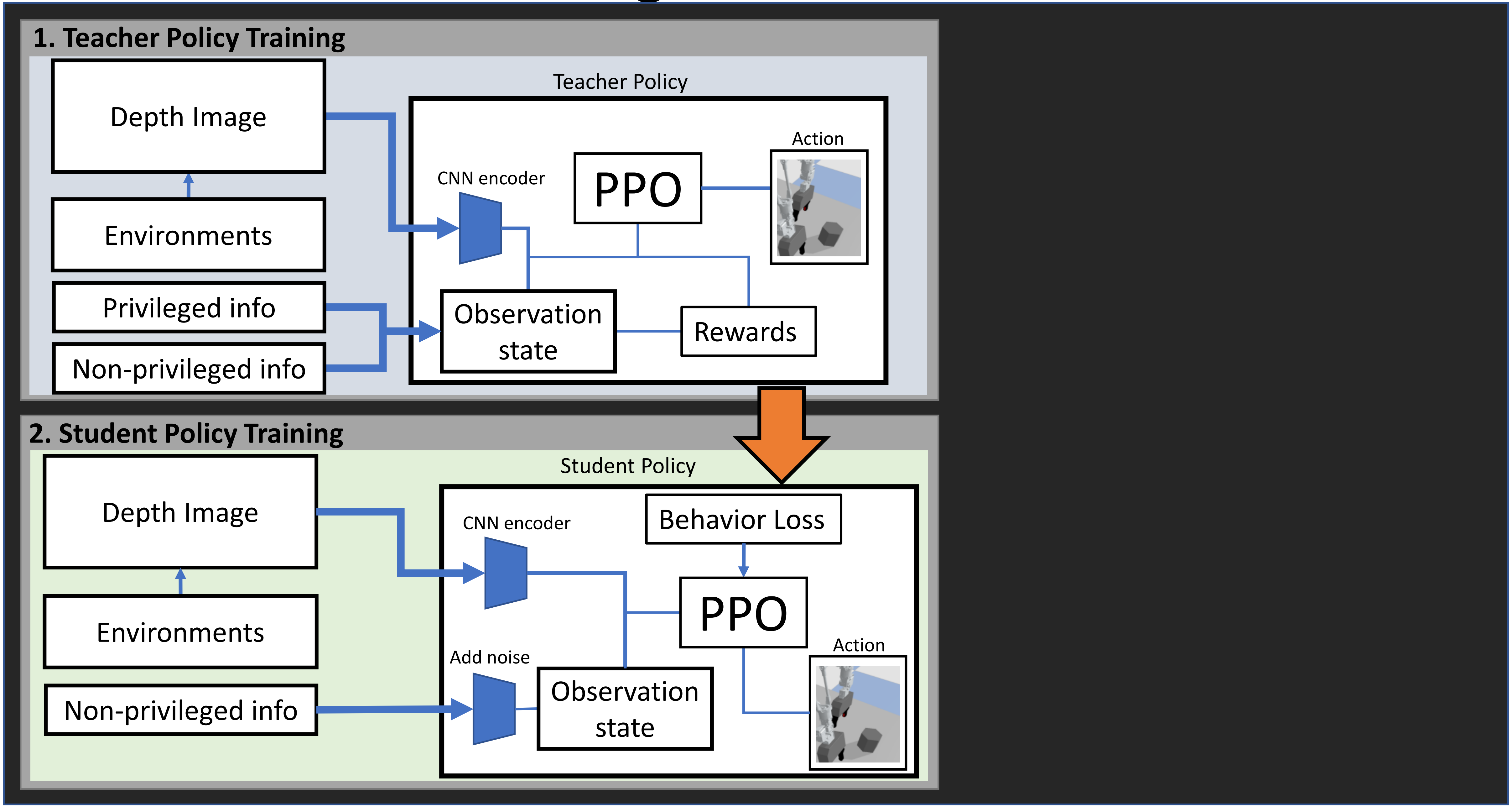
Reinforcement Learning



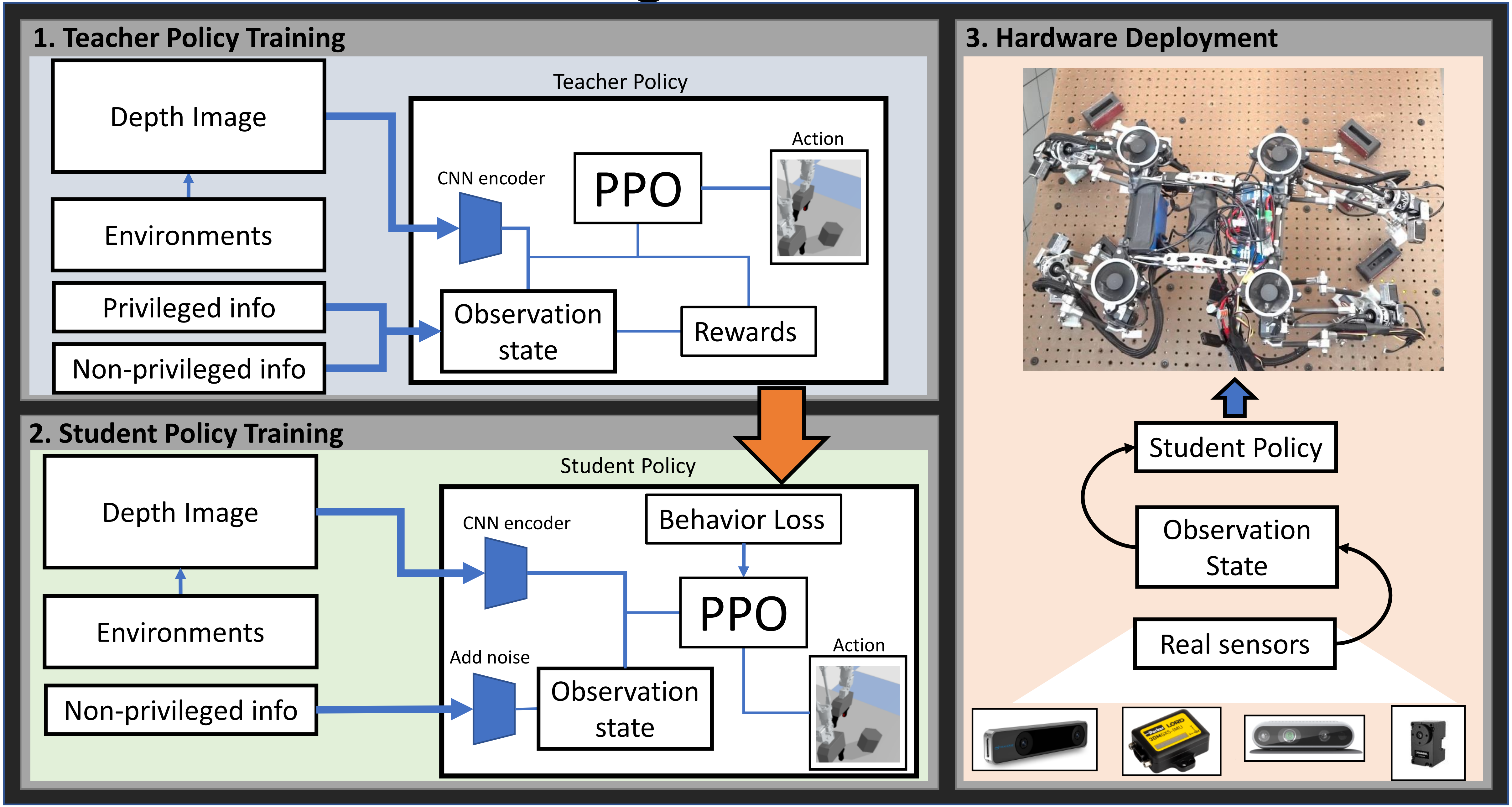
Reinforcement Learning – Flow chart



Reinforcement Learning – Flow chart



Reinforcement Learning – Flow chart



Reinforcement Learning – Reward functions

Cost function rewards

Cost function

$$J = \sum_{t=0}^{t=N} \left(||x_t - x_g||\right)^2 Q + (||u_t||)^2 R + p_t^z \beta$$

Dynamic Model

$$M\ddot{r}_t - \sum_{i=1}^{n_l} \lambda_t^i + Mg = 0$$

$$I\dot{\omega}_t + \omega_t \times I\omega_t - \sum_{i=1}^{n_l} (r_t - p_t^i) \times \lambda_t^i + \tau_t^i = 0$$

States/Actions

$$x_t = [r_t, \theta_t, p_t\, q_t, d_t]^T$$

$$u_t = [\lambda_t, \tau_t]^T$$

Joint Motion Rewards

1. Limit sudden velocity and acceleration changes

$$r_s = -\sum_{i}^{7} (0.01 q_i^2 + \ddot{q}_i^2)$$

2. Avoid being outside workspace

$$r_{jt,i} = \begin{cases} -(q_i - q_{i+1})^2 & \text{If } q_{min} < q_i < q_{max} \\ 0.0 & \text{otherwise} \end{cases}$$

3. Energy Efficiency

$$r_{\tau,i} = -\sum_{i}^{7} \tau_i^2$$

Variable definitions

r_t	-- Center of Mass Pos	λ_t	-- ground reaction force
θ_t	-- Center of Mass Angle	τ_t	-- ground reaction torque
p_t	-- Wrist position	x_g	-- goal state
q_t	-- Wrist orientation	p_t^G	-- Fingertip position
d_t	-- distance between fingertips	q_t^G	-- Fingertip orientation

Reinforcement Learning – Parameter description

- Privileged info

$$prev = \{\mu, x_g, \theta_t^{f1}, \theta_t^{f2}, w_t^{i,j}, \forall i,j\}$$

- Non-privileged info

$$non\ prev = \{\theta_t^{0:6}, x_t, u_t, o_t^{img}\}$$

Teacher Policy

- Observation State

$$obs = \{\theta_t^{0:6}, x_t, u_t, x_g, \mu, \theta_t^{f1}, \theta_t^{f2}, w_t^{i,j}, o_t^{img}, \forall i, j\}$$

Student Policy

- Observation State

$$obs = \{\theta_t^{0:6}, x_t, u_t, o_t^{img}\}$$

- Actions

$$x_t = [r_t, \theta_t, p_t\ q_t, d_t]^T$$

$$u_t = [\lambda_t, \tau_t]^T$$

- New Variable definitions

$\theta_t^{0:6}$ -- Joint angles

μ -- coefficient of friction

o_t^{img} -- Depth/ image

w -- wrench

θ_t^{f1} -- angle normal to bouldering hold, finger 1

θ_t^{f2} -- angle normal to bouldering hold, finger 2

PRELIMINARY RESULTS

- Privileged info

$$prev = \{\mu, \boldsymbol{x_g}, \theta_t^{f1}, \theta_t^{f2}, w_t^{i,j}, \forall i,j\}$$

- Non-privileged info

$$non\ prev = \{\theta_t^{0:6}, x_t, u_t, o_t^{img}\}$$

- Actions

$$x_t = [r_t, \theta_t, \boldsymbol{p_t}, q_t, d_t]^T$$

$$u_t = [\lambda_t, \tau_t]^T$$

Teacher Policy

- Observation State

$$obs = \{\theta_t^{0:6}, \boldsymbol{x_t}, u_t, \boldsymbol{x_g}, \mu, \theta_t^{f1}, \theta_t^{f2}, w_t^{i,j}, o_t^{img}, \forall i,j\}$$

Student Policy

- Observation State

$$obs = \{\theta_t^{0:6}, x_t, u_t, o_t^{img}\}$$

- New Variable definitions

$\theta_t^{0:6}$	-- Joint angles	μ	-- coefficient of friction
o_t^{img}	-- Depth/ image	w	-- wrench
θ_t^{f1}	-- angle normal to bouldering hold, finger 1		
θ_t^{f2}	-- angle normal to bouldering hold, finger 2		