Versatile and Robust Walking in Uneven Terrain

Prof. Dr. Ir. Daniel Rixen

Daniel Wahrmann

Arne-Christoph Hildebrandt

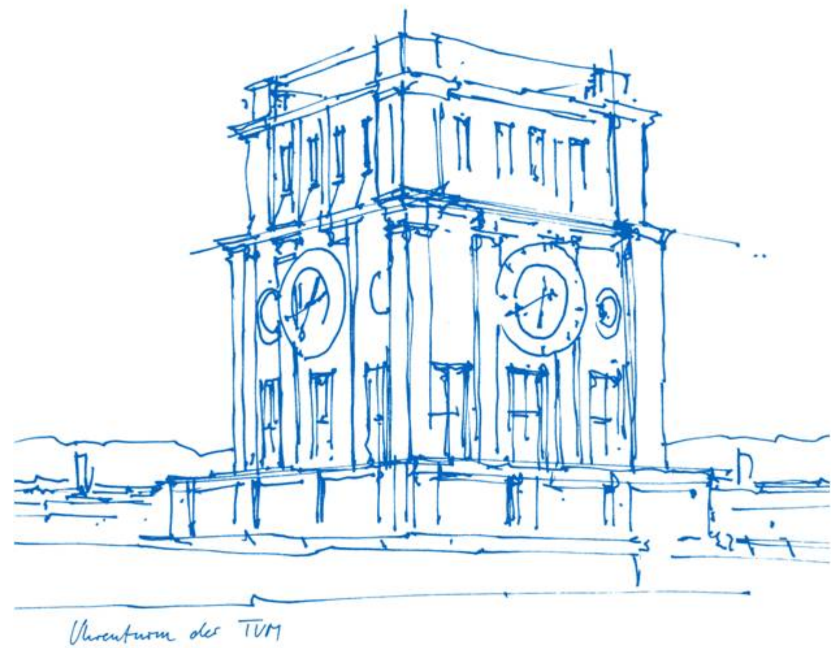
Robert Wittmann

Dr. habil. Ing. Thomas Buschmann

Technical University of Munich

Chair of Applied Mechanics

Garching, 13. January 2016



14:15 – 14:30 **The chair of Applied Mechanics and Lola**
Daniel Rixen

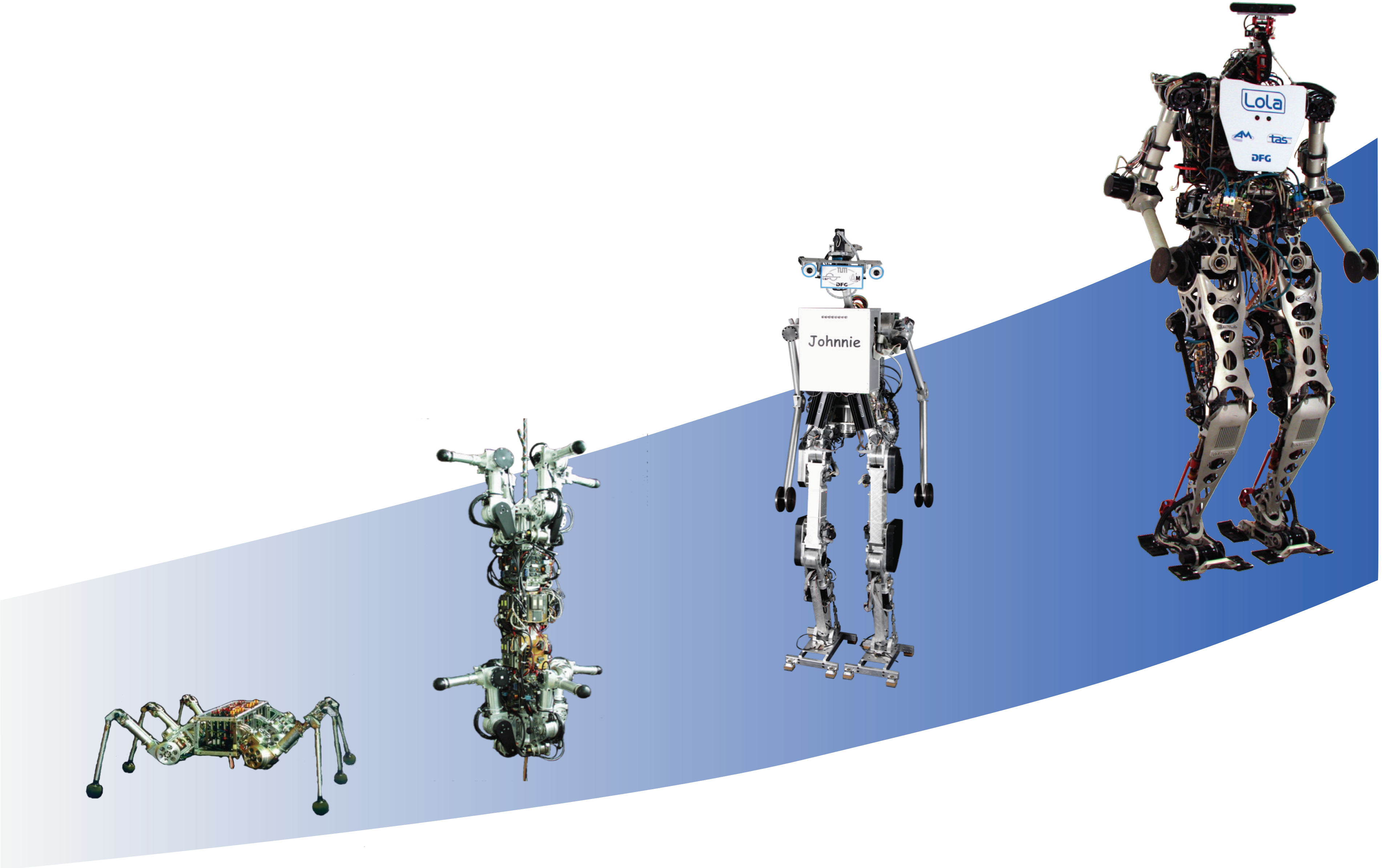
14:30 – 16:00 **Versatile and Robust Walking in unknown Terrain**

- **Environment Modelling & Vision System**
  Daniel Wahrmann (Daad)
- **Navigation & Motion generation**
  Arne Hildebrandt (DFG)

Pause

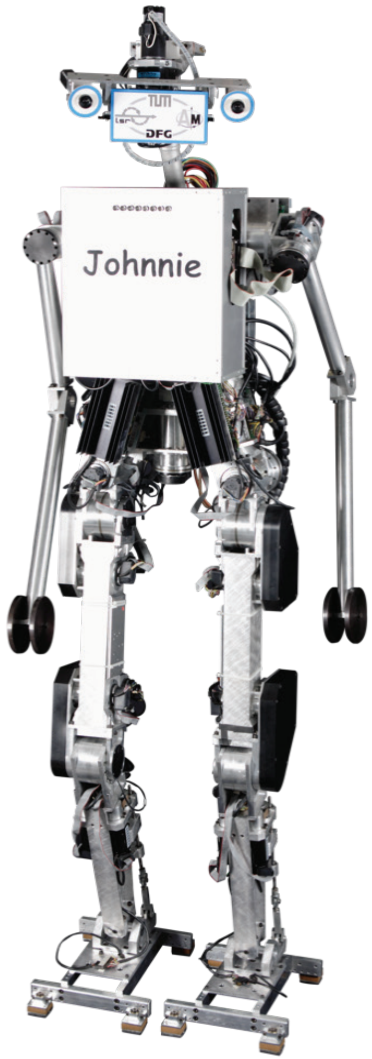
- **Environment Modelling & Vision System**
  Robert Wittmann (DFG)
- **Future Research & outlook**
  Daniel Rixen

16:00 – 17:00 **Lab demonstrations**


Johnnie
Lola
DFG

Humanoider Roboter: „Johnnie“

Prof. F. Pfeiffer



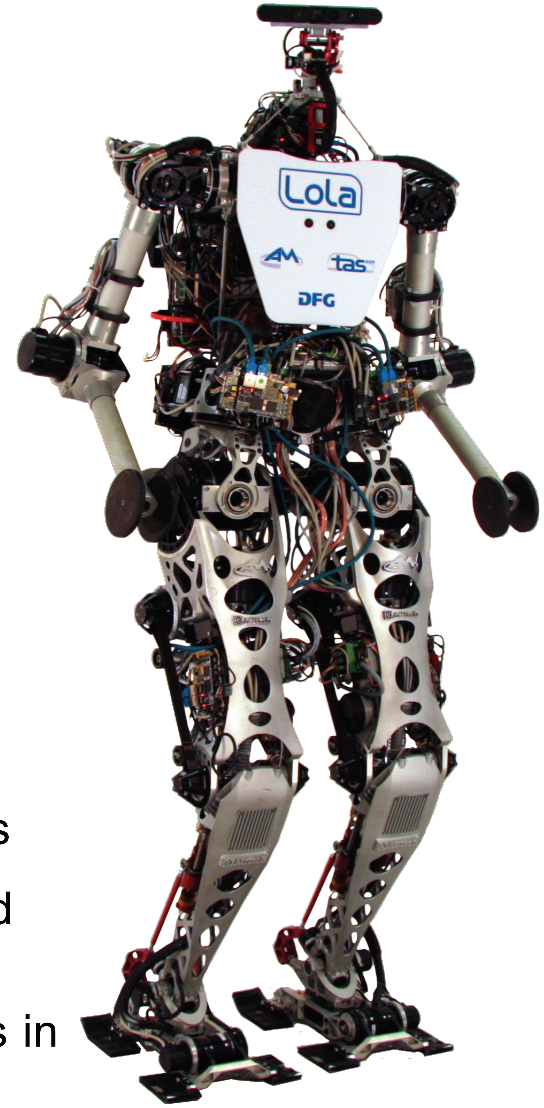
DFG-Schwerpunktprogramms
SPP 1039 - Autonomes Laufen wurde (1997 – 2002)

Dr. Gienger und Dr. Löffler

Humanoider Roboter: „Lola“

Prof. H. Ulbrich

2001 - 2012



DFG-Paketantrag: Natur und Technik intelligenten Laufens
2004-2010 (Sprecher Ansgar Büschges)

- ✓ **Dr. Sebastian Lohmeier:** Design and Realization of a Humanoid Robot for Fast and Autonomous Bipedal Walking
- ✓ **Dr. Thomas Buschmann:** Simulating and Controlling Biped Robots
- ✓ **Dr. Markus Schwienbacher:** Redundancy, Collision Avoidance and Dynamic Trajectory Planning for Humanoid Robots
- ✓ **Dr. Alexander Ewald**: Implementation of Neurobiological Principles in the Walking Control of Humanoid Robots
- ✓ **Dr. Valerio Favot – Dr. Bachmayer:** Electronics and Low-Level Control for Biped Robots

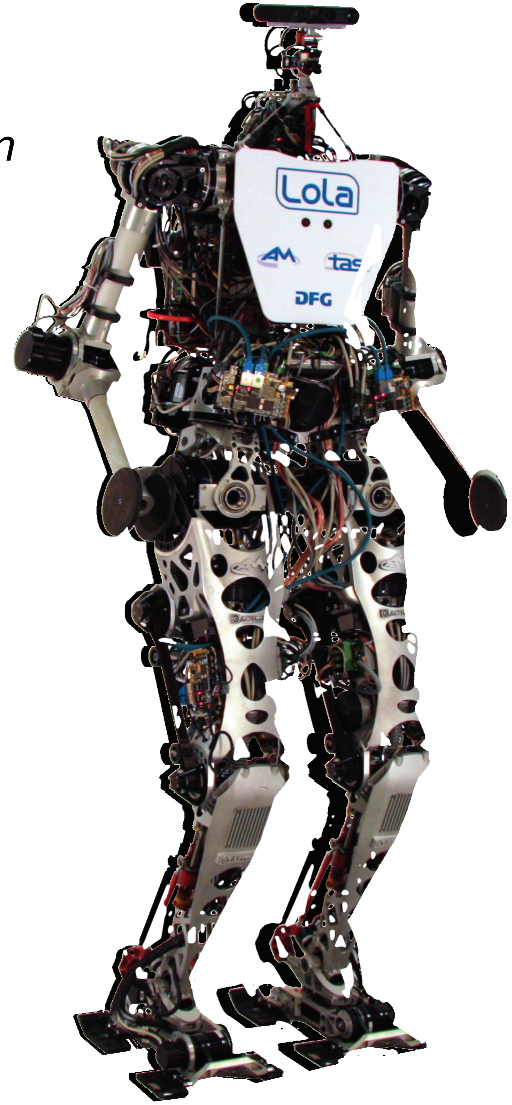
Humanoider Roboter: „Lola“

Dr. T. Buschmann

2012-2014

Prof. dr. D. Rixen

2012 -



DFG- **Flexibles und robustes Gehen in unebenem Gelände**
(2012-2016)

- ✓ **Robert Wittmann:**
  Robustes Gehen auf unebenem Terrain und unter großen Störungen
- ✓ **Arne Hildebrandt:**
  Autonome Navigation humanoider Roboter
- ✓ **Daniel Wahrmann:**
  Autonome Fortbewegung Umgebungsmodellierung
- ✓ **Felix Sygulla:**
  Robusteres Gehen auf unerkanntem unebenem Gelände mit Hilfe von taktilem Feedback
- ✓ **Philipp Seiwald**

Results are published in:

R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, D. Rixen and T. Buschmann, “Model-Based Predictive Bipedal Walking Stabilization”, in *IEEE-RAS International Conference on Humanoids Robots*, 2016

A.-C. Hildebrandt, M. Demmeler, R. Wittmann, D. Wahrmann, F. Sygulla, D. Rixen and T. Buschmann, “Real-Time Predictive Kinematic Evaluation and Optimization for Biped Robots”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015

R. Wittmann and D. Rixen, “A Prediction Model for State Observation and Model Predictive Control of Biped Robots”, in *Proc. Appl. Math. Mech., Wiley-Blackwell*, 2016

D. Wahrmann, A.-C. Hildebrandt, R. Wittmann, F. Sygulla, D. Rixen and T. Buschmann, “Fast Object Approximation for Real-Time 3D Obstacle Avoidance with Biped Robots”, in *IEEE International Conference on Advanced Intelligent Mechatronics*, 2016

A.-C. Hildebrandt, C. Schuetz, D. Wahrmann, R. Wittmann and D. Rixen, “A Flexible Robotic Framework for Autonomous Manufacturing Processes: Report from the European Robotics Challenge Stage 1”, in *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2016

A.-C. Hildebrandt, D. Wahrmann, R. Wittmann, D. Rixen and T. Buschmann, “Real-Time Pattern Generation Among Obstacles for Biped Robots”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015

R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, T. Buschmann and D. Rixen, “State Estimation for Biped Robots Using Multibody Dynamics”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015

R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, D. Rixen and T. Buschmann, “Real-Time Nonlinear Model Predictive Footstep Optimization for Biped Robots”, in *IEEE-RAS International Conference on Humanoids Robots*, 2015

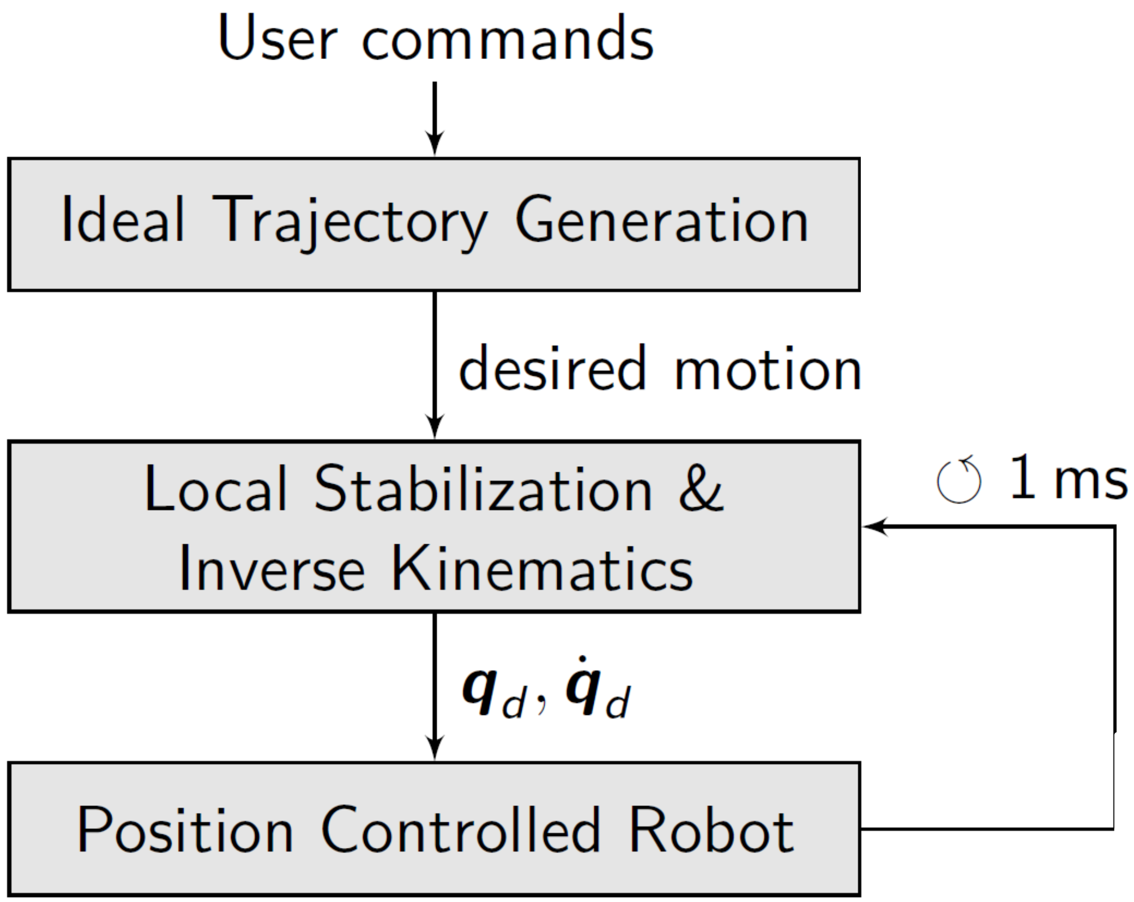
R. Wittmann, A.-C. Hildebrandt, A. Ewald and T. Buschmann, “An Estimation Model for Footstep Modifications of Biped Robots”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014

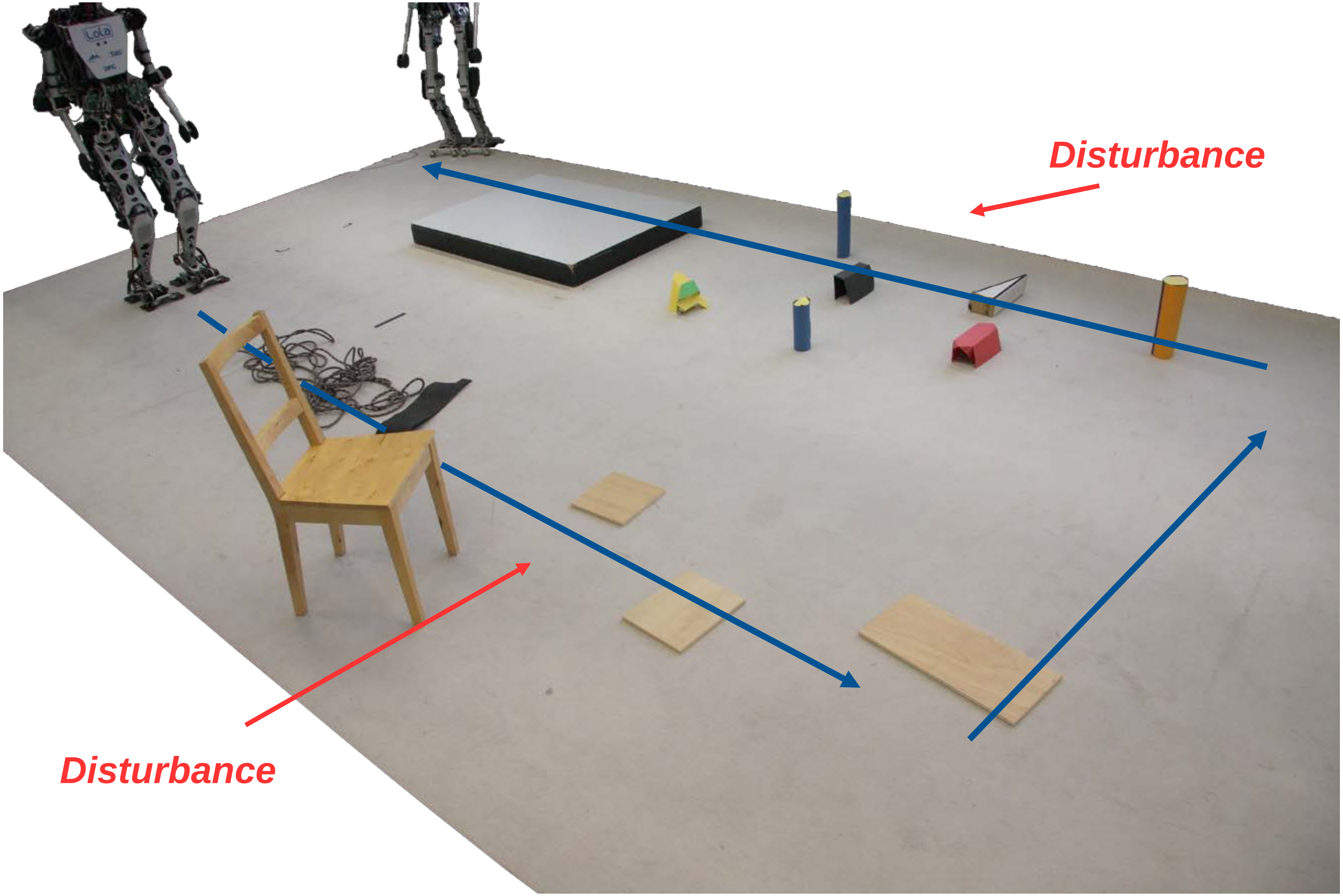
A.-C. Hildebrandt, R. Wittmann, D. Wahrmann, A. Ewald and T. Buschmann, “Real-Time 3D Collision Avoidance for Biped Robots”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014

http://www.lake-skadar.com/

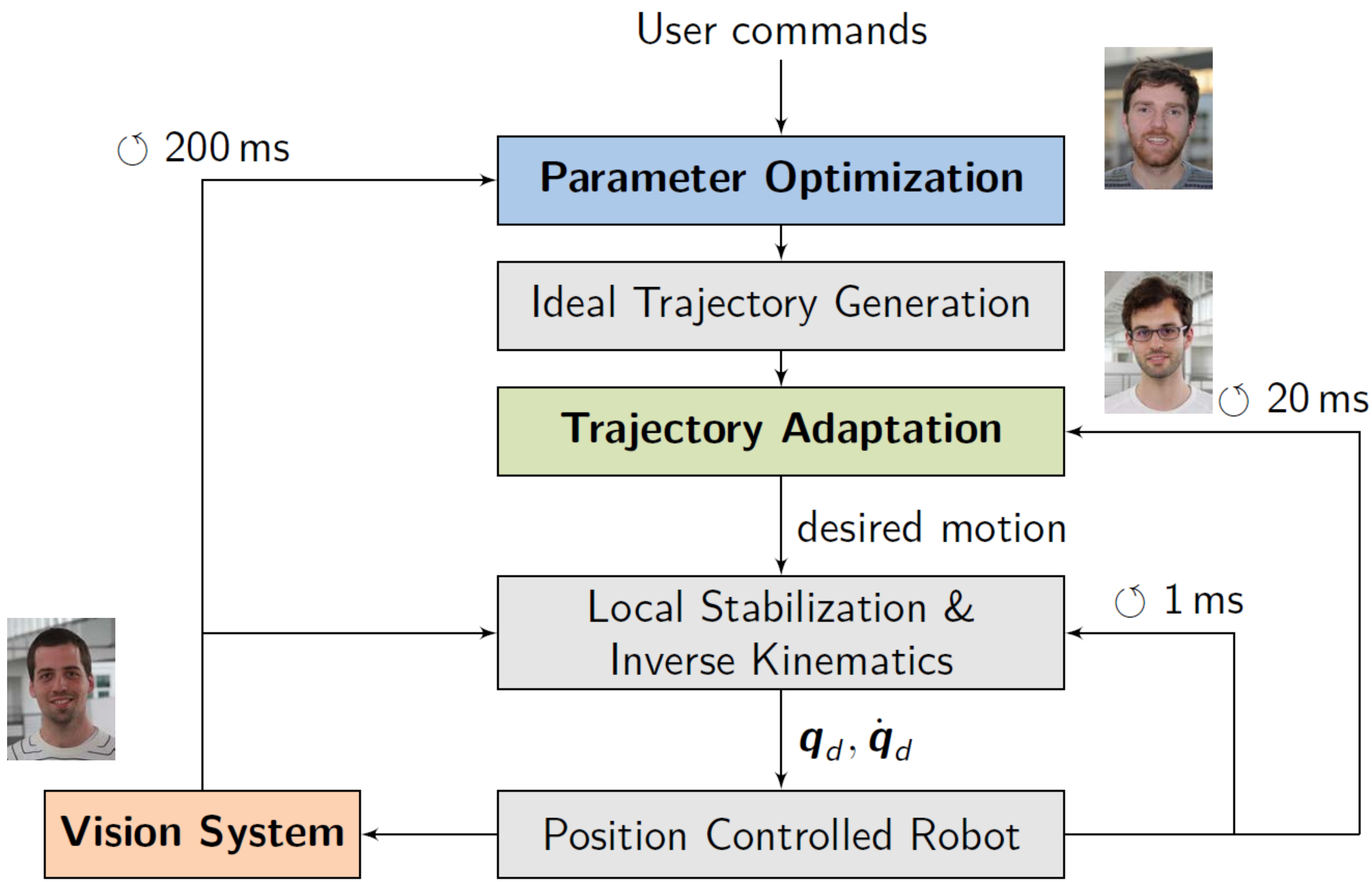
http://www.touristmaker.com/

Real-Time Control

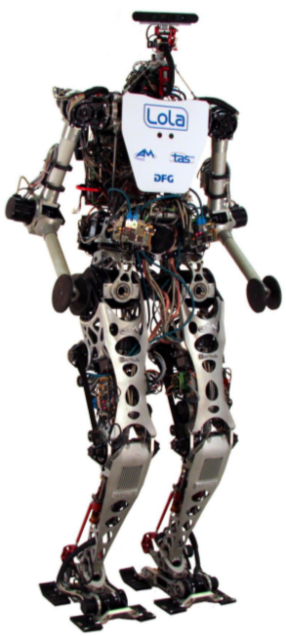


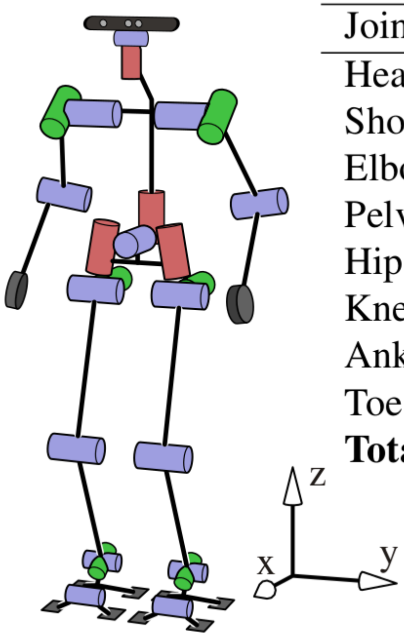

Disturbance
Disturbance

Real-Time Autonomous Navigation



The Humanoid Robot “Lola”

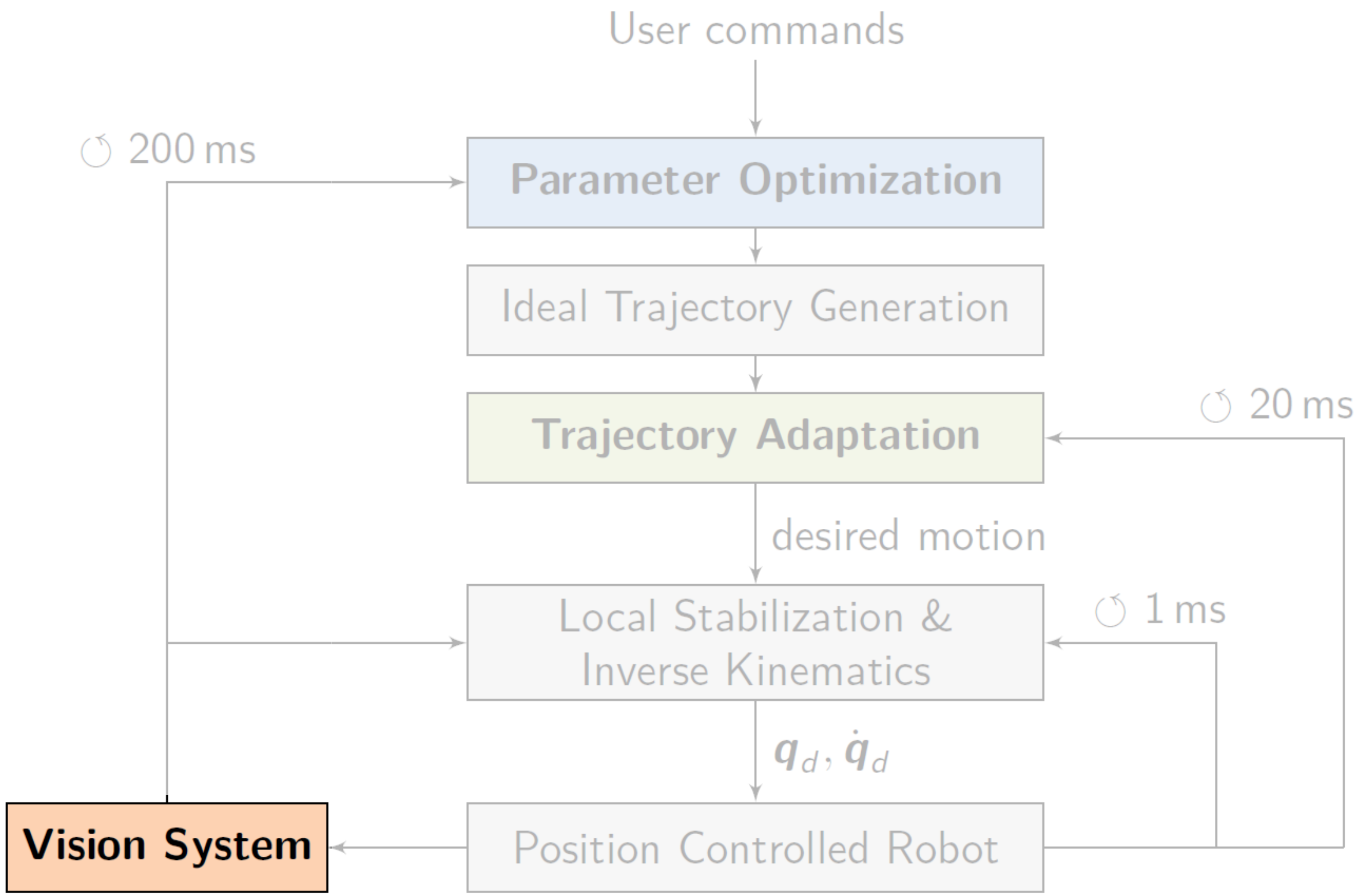




Joint	DoF
Head	2
Shoulder	2
Elbow	1
Pelvis	2
Hip	3
Knee	1
Ankle	2
Toe	1
Total	**24**

- 60 kg
- 1,8 m
- Asus Xtion PRO LIVE
- 2 Intel i7 Computers
  - Control (QNX)
  - Vision System (Linux)

Real-Time Autonomous Navigation



Contents

Introduction and Motivation

Environment Modeling

Vision System

Navigation and Motion Generation

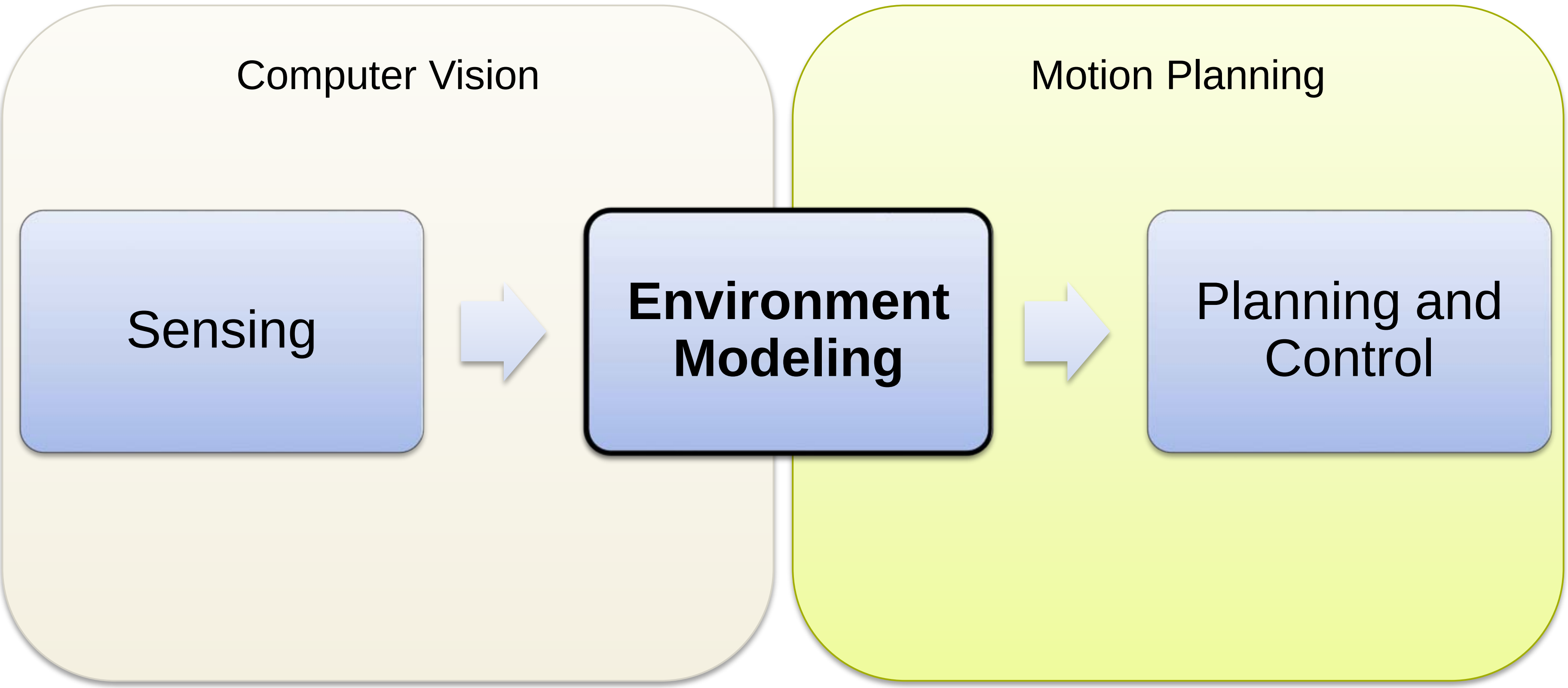
Stabilization

Results

Limitations

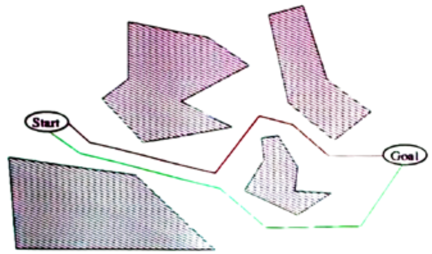
Perspectives

Autonomous Navigation in Unknown Environments



Autonomous Navigation: Examples

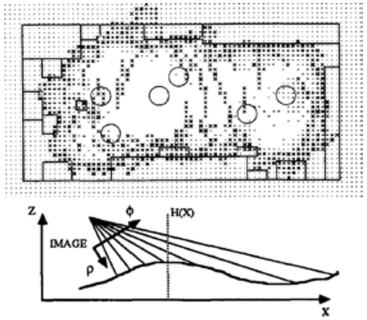
2D-Representation



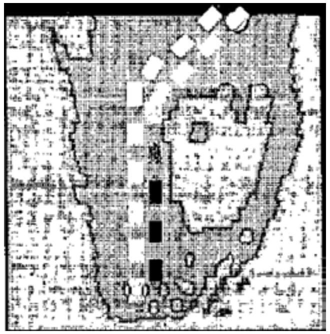
A* - Search Algorithm
(Hart et al. 1968)

Shakey the Robot
(Credit: SRI International)

2,5D-Representation

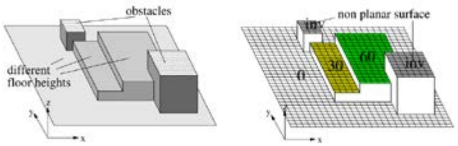


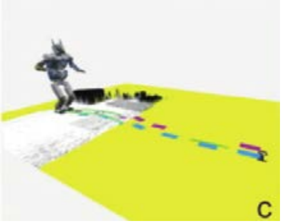
“Height Maps”
(Moravec et al. 1985,
Herbert et al. 1989)



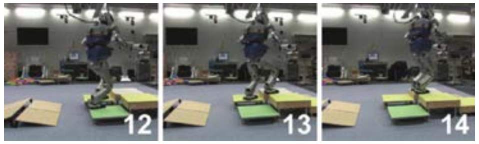
Humanoid Navigation
(Kagami et al. 2003)

Segmented 2,5D-Representation



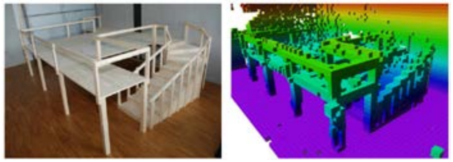


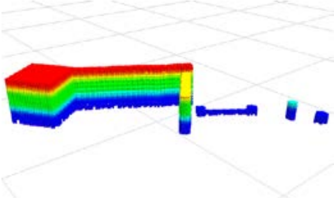
Sony Qrio
(Gutmann et al. 2005)

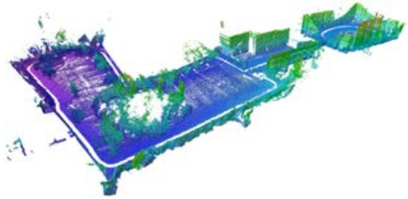


HRP 2
(Nishiwaki et al. 2012)

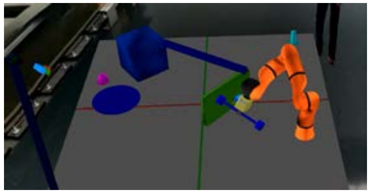
3D-Representation





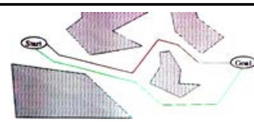
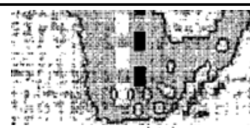
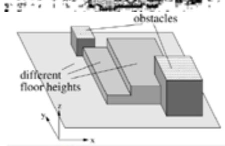
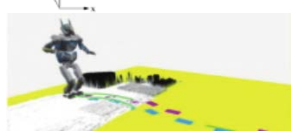
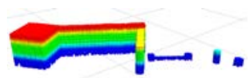


“Octomaps”
(Wurm et al. 2010)

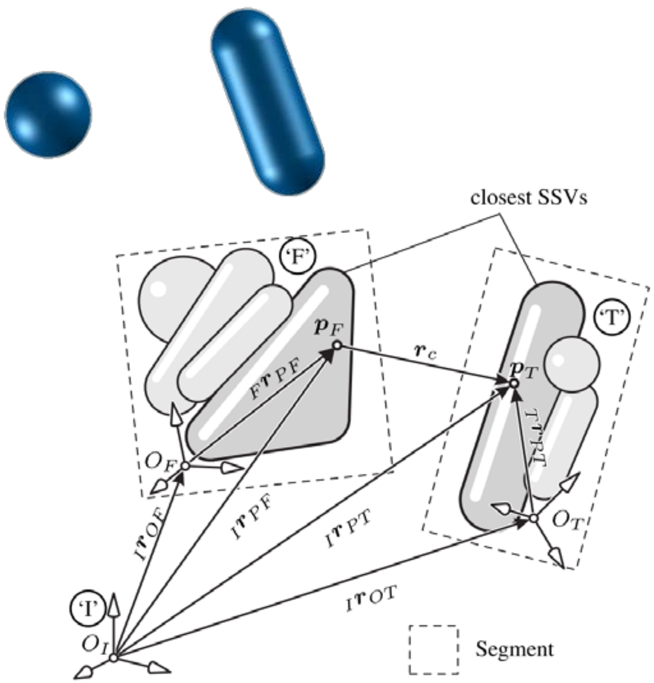


EuRoC
(ICARSC 2016)

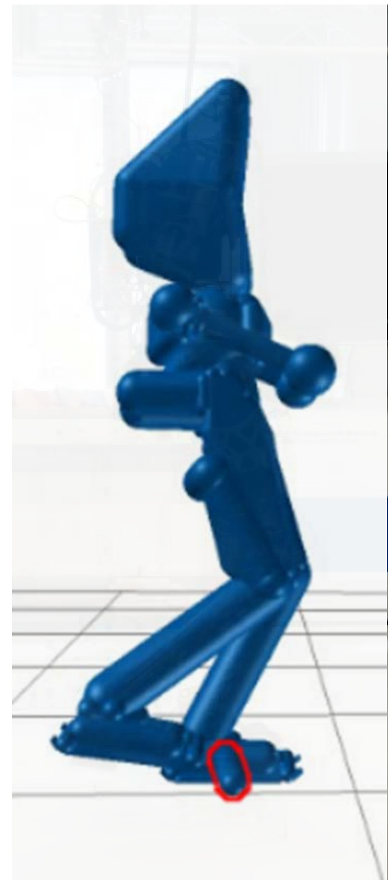
Environment Modeling

Art	Beispiel		Planung und Kollisionsvermeidung	Dynamische Umgebung?
2D	Shakey			~
2,5D	H7			~
	Qrio			~
	HRP 2			×
	…			
3D				✓
	Octomaps			×

Collision Model

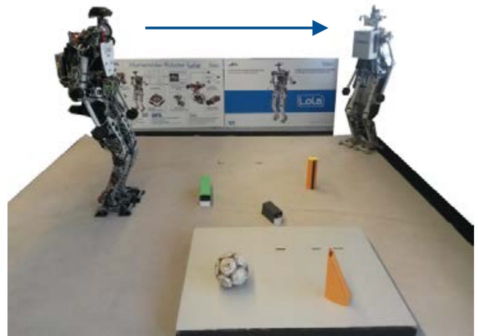


Swept-Sphere-Volumes (SSVs)
(ICRA 2011)

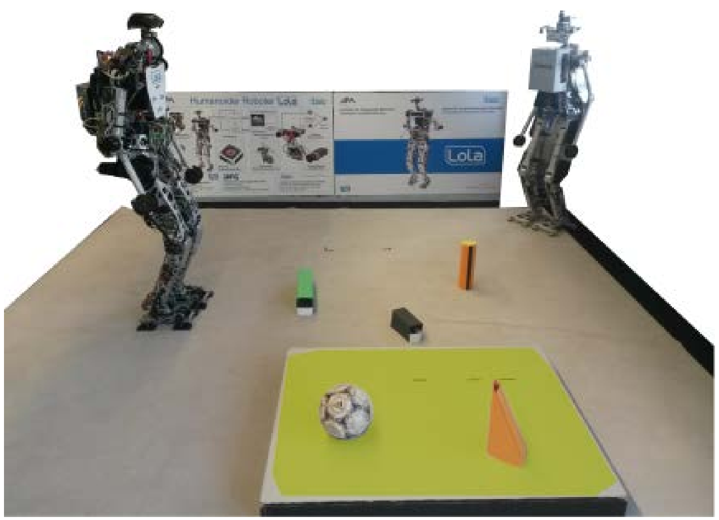


Obstacle Avoidance
(IROS 2015, AIM 2016)

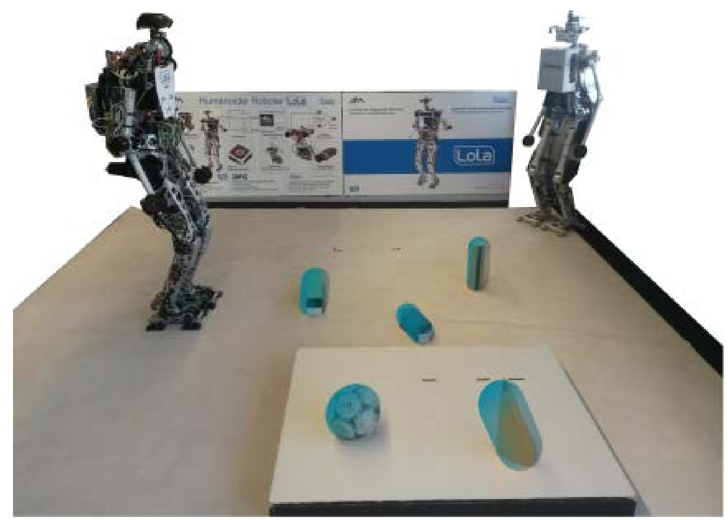
Environment Model



Walkable Surfaces



Collision Objects



Contents

Introduction and Motivation

Environment Modeling

Vision System

Navigation and Motion Generation

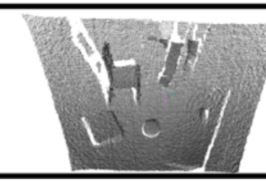
Stabilization

Results

Limitations

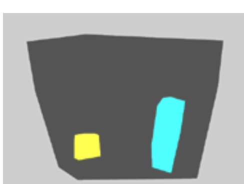
Perspectives

Vision System



Surface Approximation

- Segmentation
- Clustering
- Approximation
- Tracking

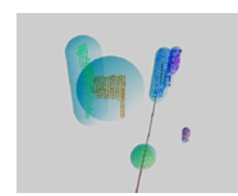


Plane Recognition

not-so-RANSAC

Obstacle Approximation

- Segmentation
- Clustering
- Approximation
- Tracking

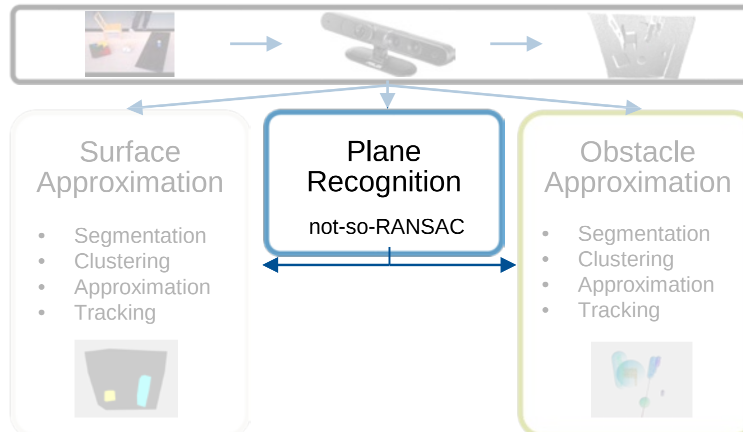


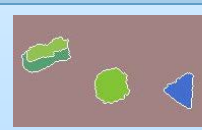
http://github.com/am-lola/lepp3

Vision System

Plane Recognition

Recognize walkable surfaces from the scenario.



Input	Method	Output	Time/Frame [ms]	Proprieties
	Depth-Map		90-110	easy, inexact, difficulty with multiple surfaces
	Normal-Based		800-1100	robust, slow, problem with far-away objects
	RANSAC		50-70	fast, multiple surfaces, parameter-sensible
	…			

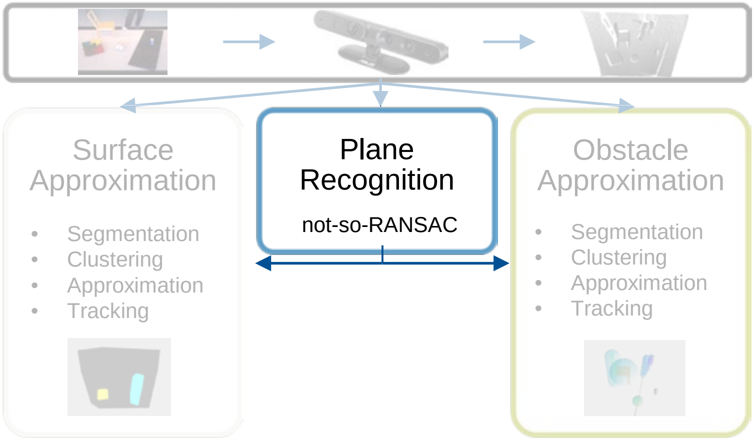
Vision System

RANdom SAmple Consensus (RANSAC)

Iterative test of plane parameters

$\boldsymbol{a}x + \boldsymbol{b}y + \boldsymbol{c}z + \boldsymbol{d} = 0$

until enough inliers are found



Generates a list of plane parameters

a_1, b_1, c_1, d_1

a_2, b_2, c_2, d_2

...

a_i, b_i, c_i, d_i

Modifications:

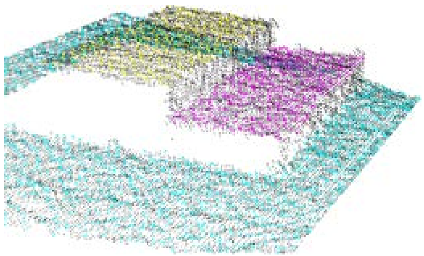
not-so-RANSAC

Initialized with known planes

Classification

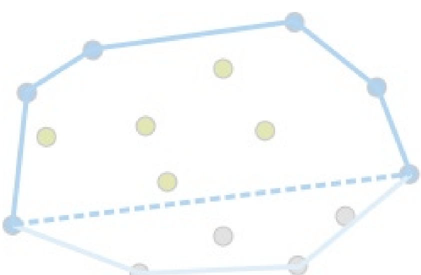
Separate clusters based on inclination

Vision System



Euclidean Clustering

- Classification based on plane inclination

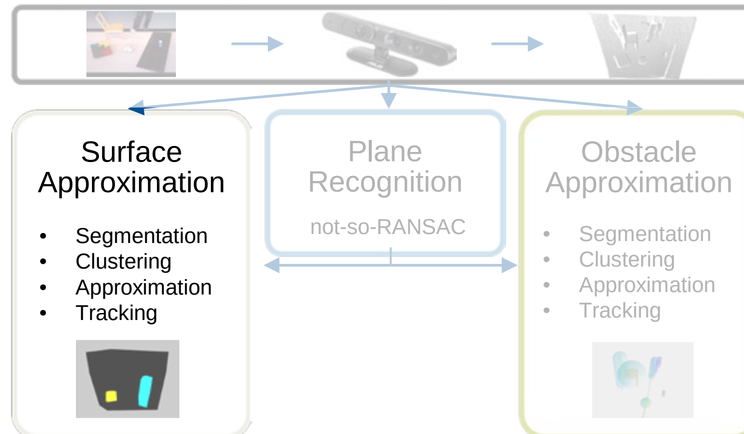


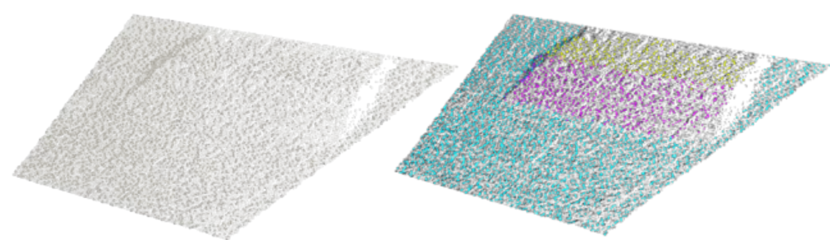
Polygon Approximation

- Quickhull Algorithm
- Iterative reduction to a n-sided polygon

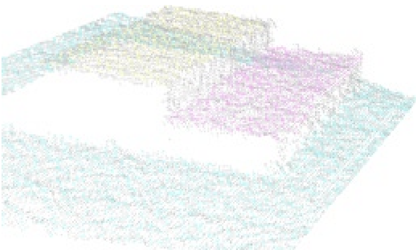
Polygon Tracking

- Geometric low-pass filter

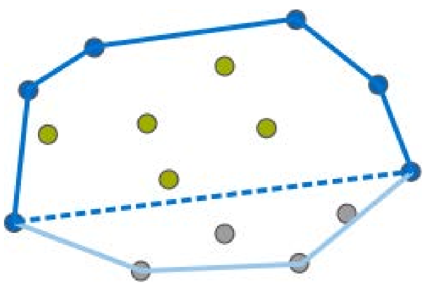




Vision System

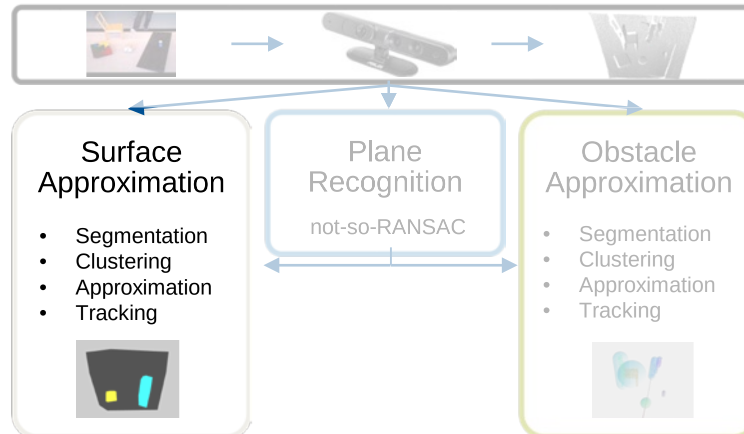


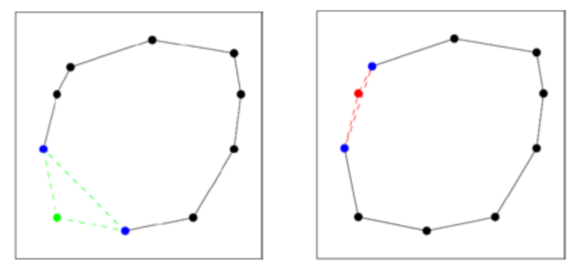
Euclidean Clustering
- Classification based on plane inclination

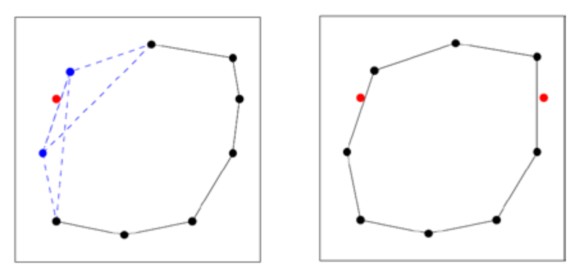


Polygon Approximation
- Quickhull Algorithm
- Iterative reduction to a n-sided polygon

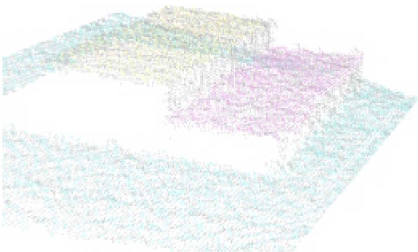
Polygon Tracking
- Geometric low-pass filter



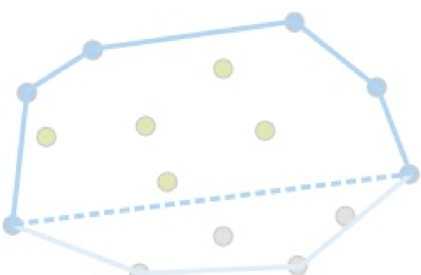




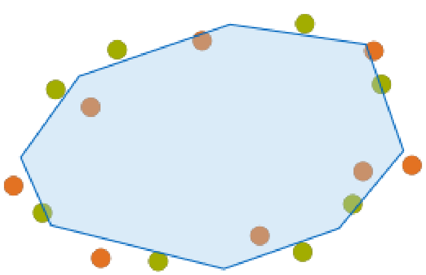
Vision System



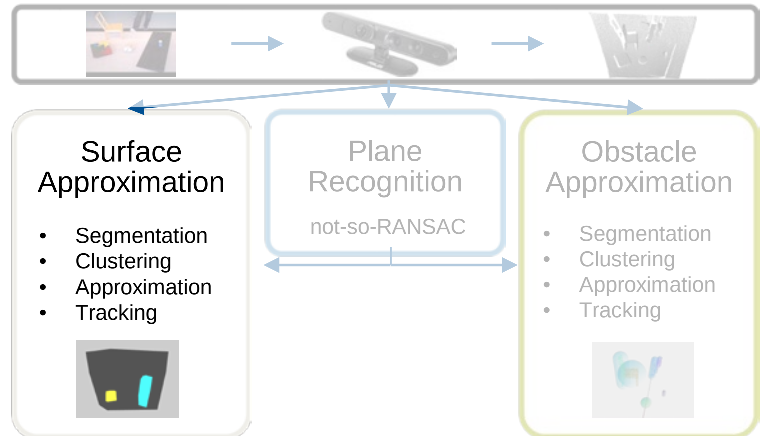
Euclidean Clustering
- Classification based on plane inclination

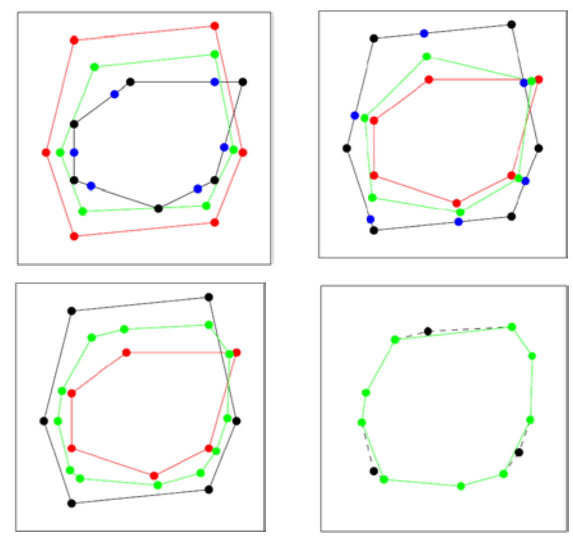


Polygon Approximation
- Quickhull Algorithm
- Iterative reduction to a n-sided polygon

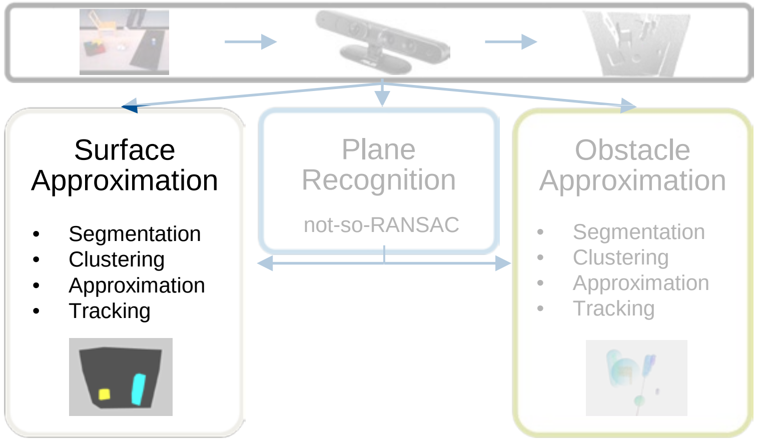


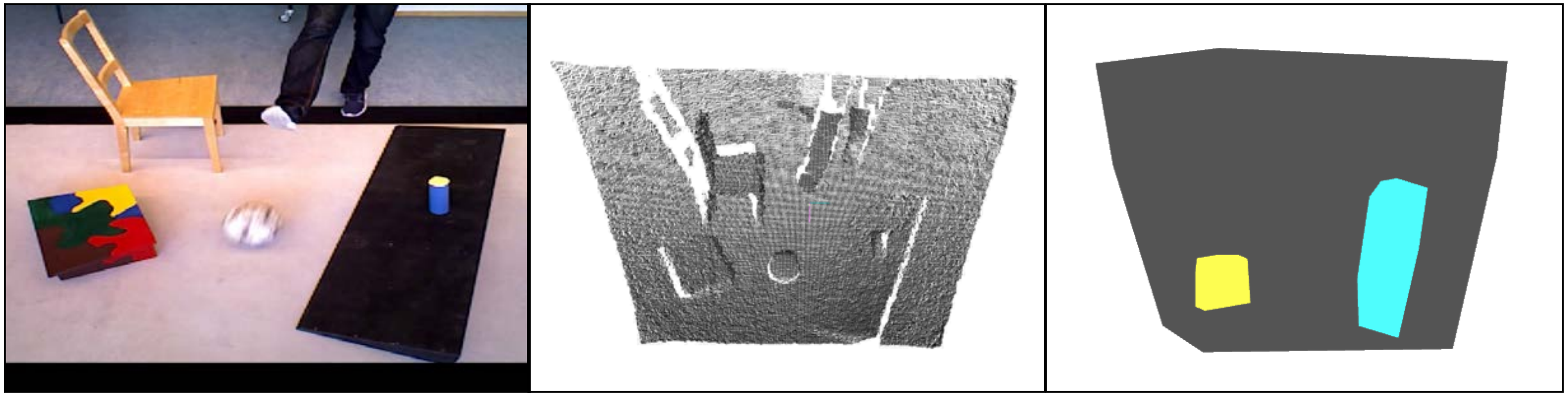
Polygon Tracking
- Geometric low-pass filter





Vision System



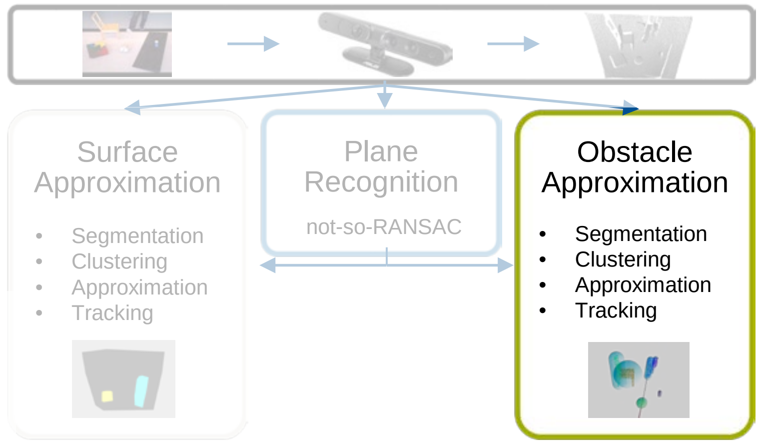


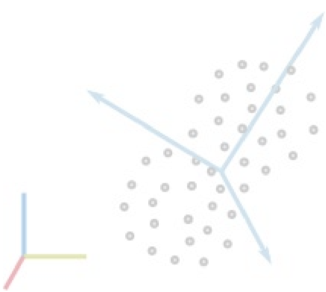
Vision System

Gaussian Clustering

„Gaussian Mixture Models“

Probabilistic fit of a 3D normal distribution





„Swept-Sphere-Volumes“ (SSVs) Approximation

Inertia tensor analysis (PAD)

Main moments of inertia

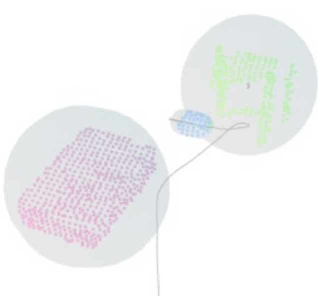
GMM: *K* gaussians with the following form:

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

Expectation Step (iteration through each point x_i and Gaussian k): “responsibility”

Maximization Step (iteration through each Gaussian k): new mean μ_k^{new}, covariance Σ_k^{mle} and mixing coefficient π_k^{new}

Regularization: add a low-pass filter to the covariance: $\Sigma_k^{new} = \lambda\Sigma_k + (1-\lambda)\Sigma_k^{mle}$



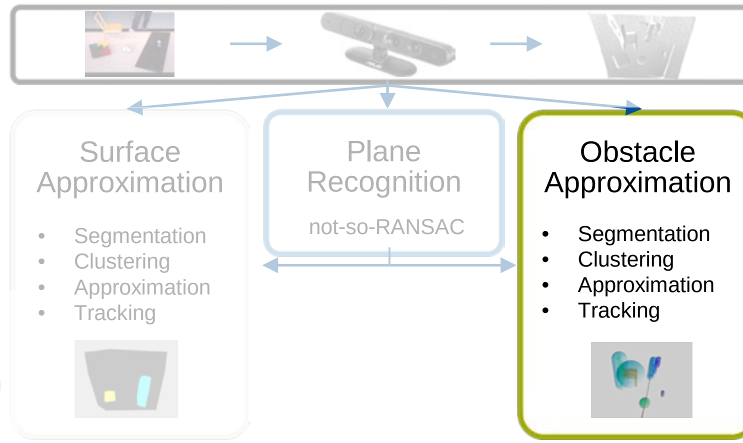
Obstacle-Tracking

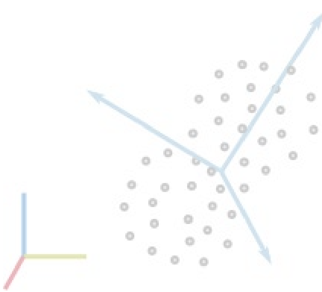
Kalman-Filter on the obstacle’s centroid with constant $\dot{x}$ and Gaussian noise on x and $\dot{x}$

Vision System

Gaussian Clustering

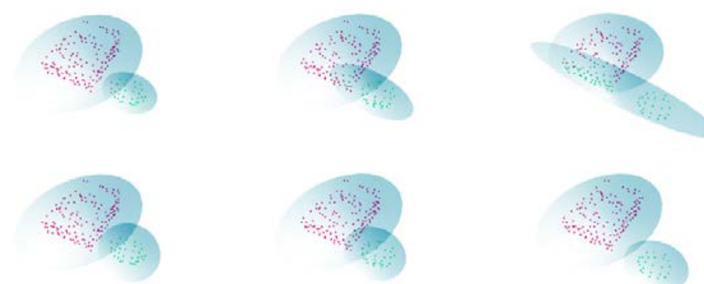
- „Gaussian Mixture Models“
- Probabilistic fit of a 3D normal distribution





„Swept-Sphere-Volumes“ (SSVs) Approximation

- Inertia tensor analysis (PAD)
- Main moments of inertia

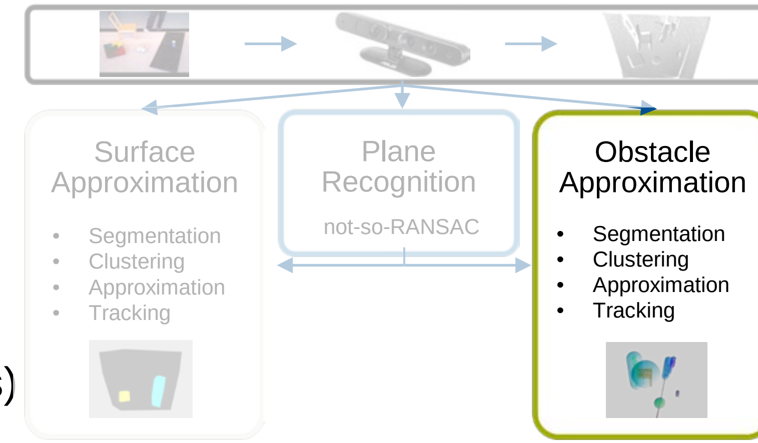


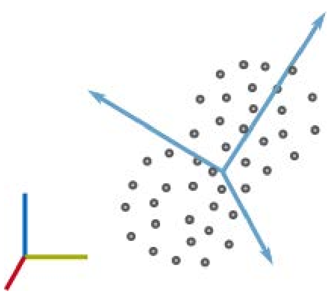
Obstacle-Tracking

- Kalman-Filter on the obstacle’s centroid with constant $\dot{x}$ and Gaussian noise on x and $\dot{x}$

Vision System

Gaussian Clustering
- „Gaussian Mixture Models“
- Probabilistic fit of a 3D normal distribution





„Swept-Sphere-Volumes“ (SSVs) Approximation
- Inertia tensor analysis (PAD)
- Main moments of inertia

Inertia Matrix I

eigenvalues $I_{max} \geq I_{mid} \geq I_{min}$

Geometric invariants:

$$\xi_1 = \frac{I_{min}}{I_{max}} \qquad \xi_2 = \frac{I_{mid}}{I_{max}}$$

Point-SSV: $I_{max} \cong I_{mid} \cong I_{min}$

$\rightarrow \boldsymbol{\xi_1 \cong \xi_2 \cong 1}$

Line-SSV: $I_{max} \gg I_{mid} \cong I_{min}$

$\rightarrow \boldsymbol{\xi_1 < \xi_2 \cong 1}$

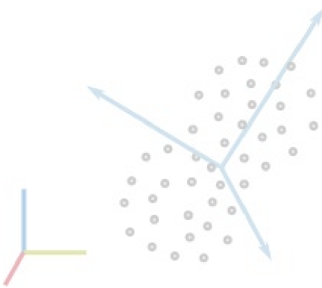
Obstacle-Tracking
- Kalman-Filter on the obstacle’s centroid with constant $\dot{x}$ and Gaussian noise on x and $\dot{x}$

Vision System

Gaussian Clustering

„Gaussian Mixture Models“

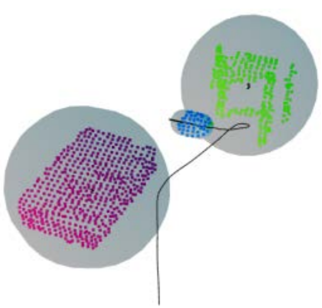
Probabilistic fit of a 3D normal distribution



„Swept-Sphere-Volumes“ (SSVs) Approximation

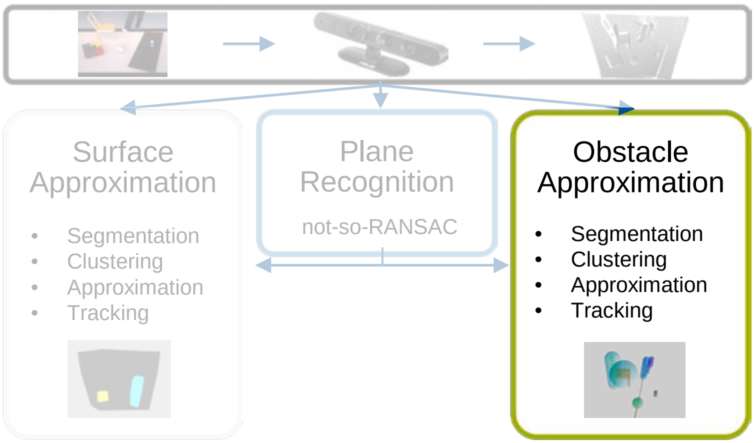
Inertia tensor analysis (PAD)

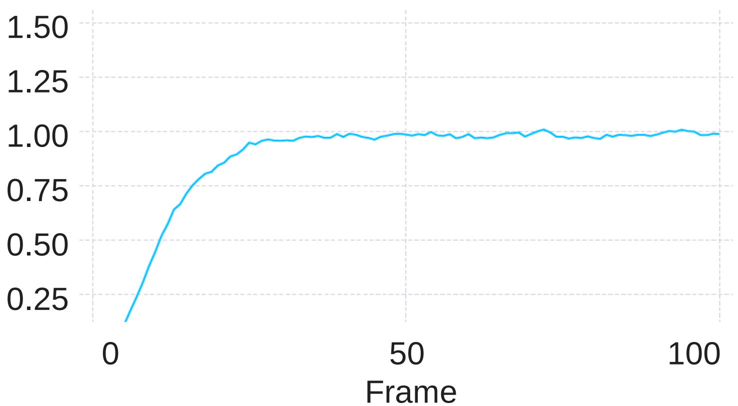
Main moments of inertia



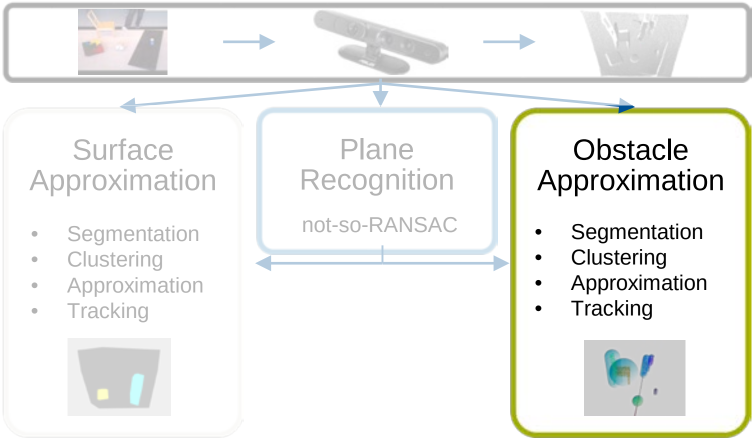
Obstacle-Tracking

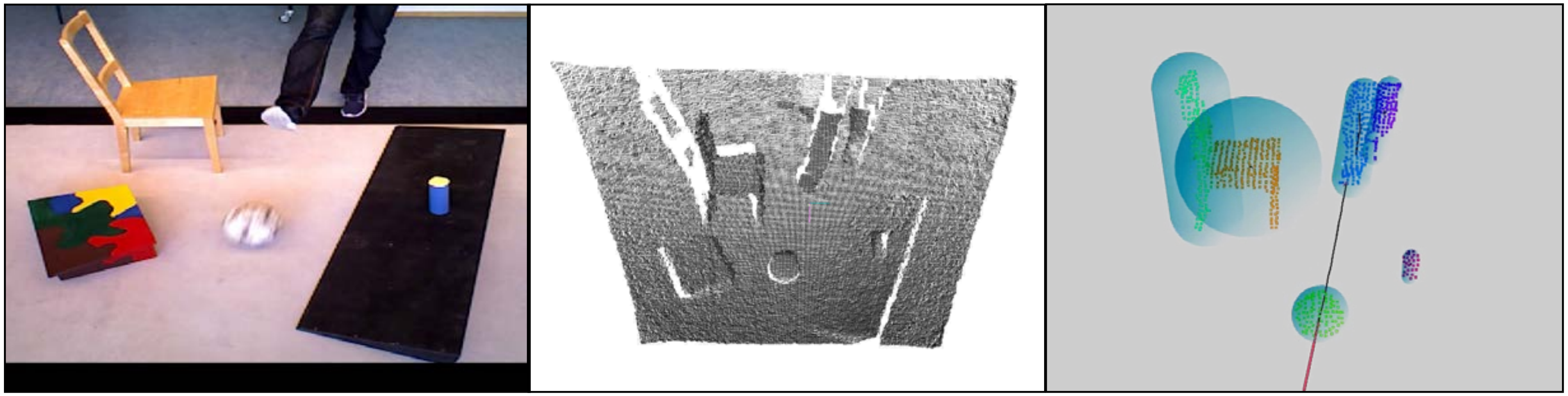
Kalman-Filter on the obstacle's centroid with constant $\dot{x}$ and Gaussian noise on x and $\dot{x}$





Vision System

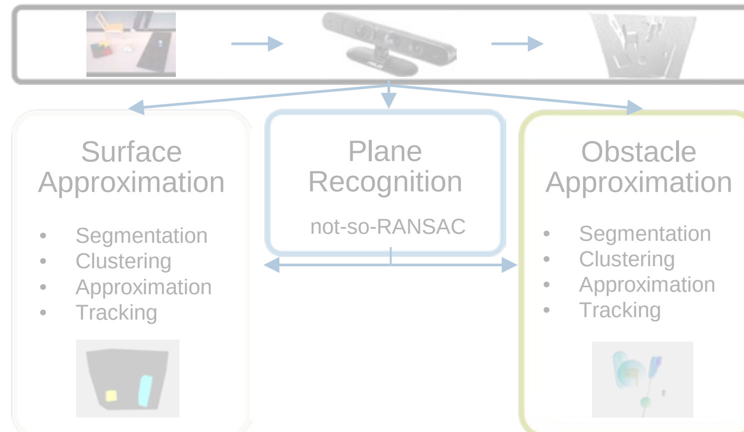


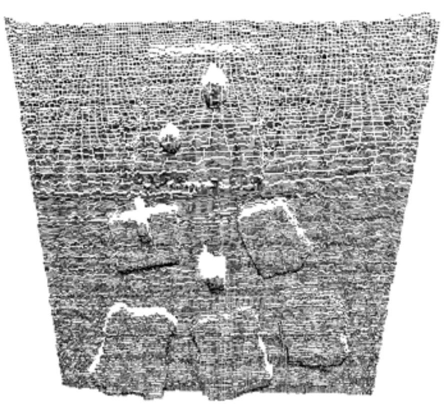


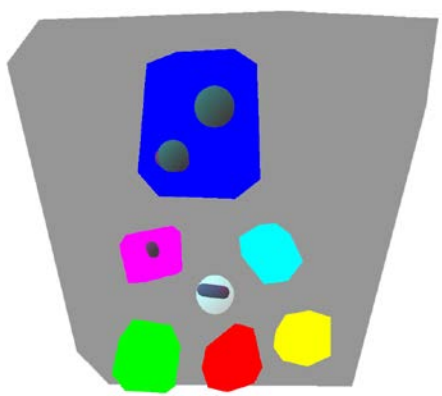
Vision System

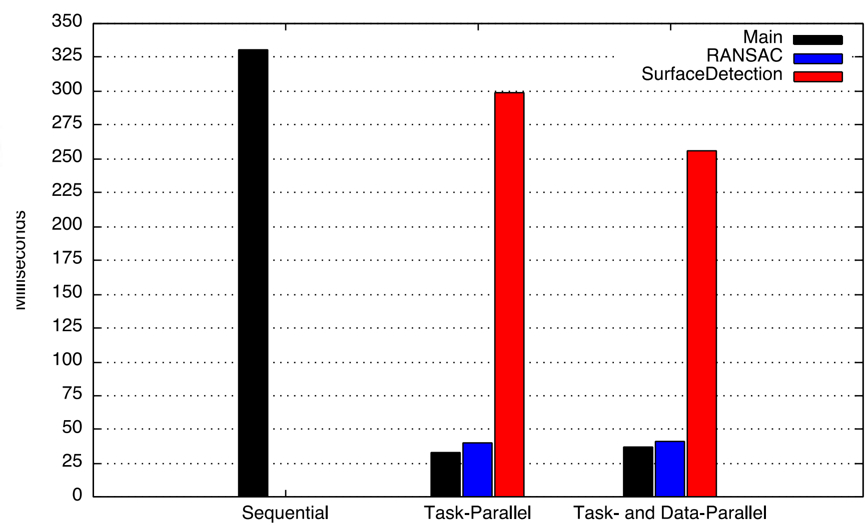
Vision System

Parallel Processes









Contents

Introduction and Motivation

Environment Modeling

Vision System

Navigation and Motion Generation

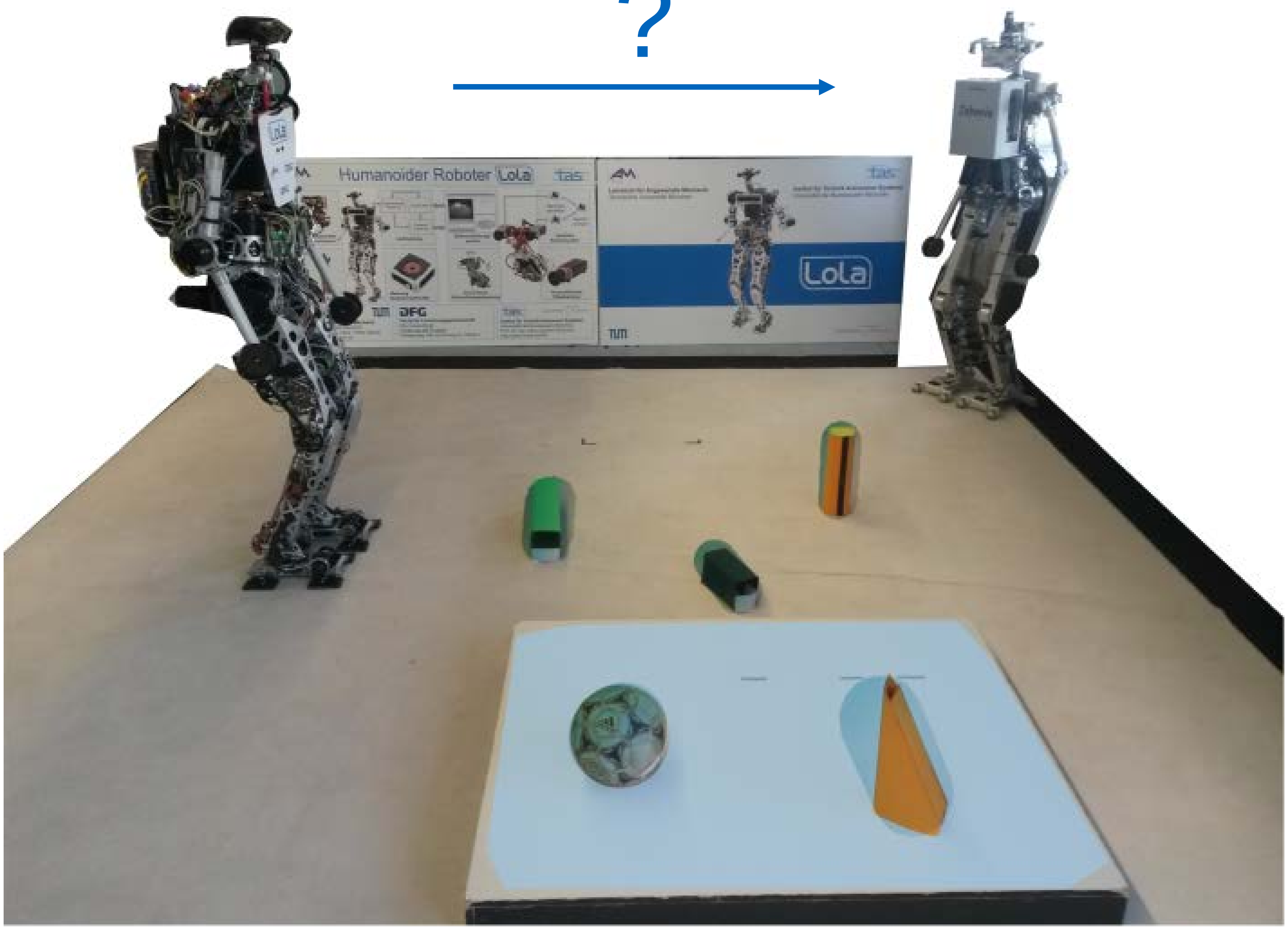
Stabilization

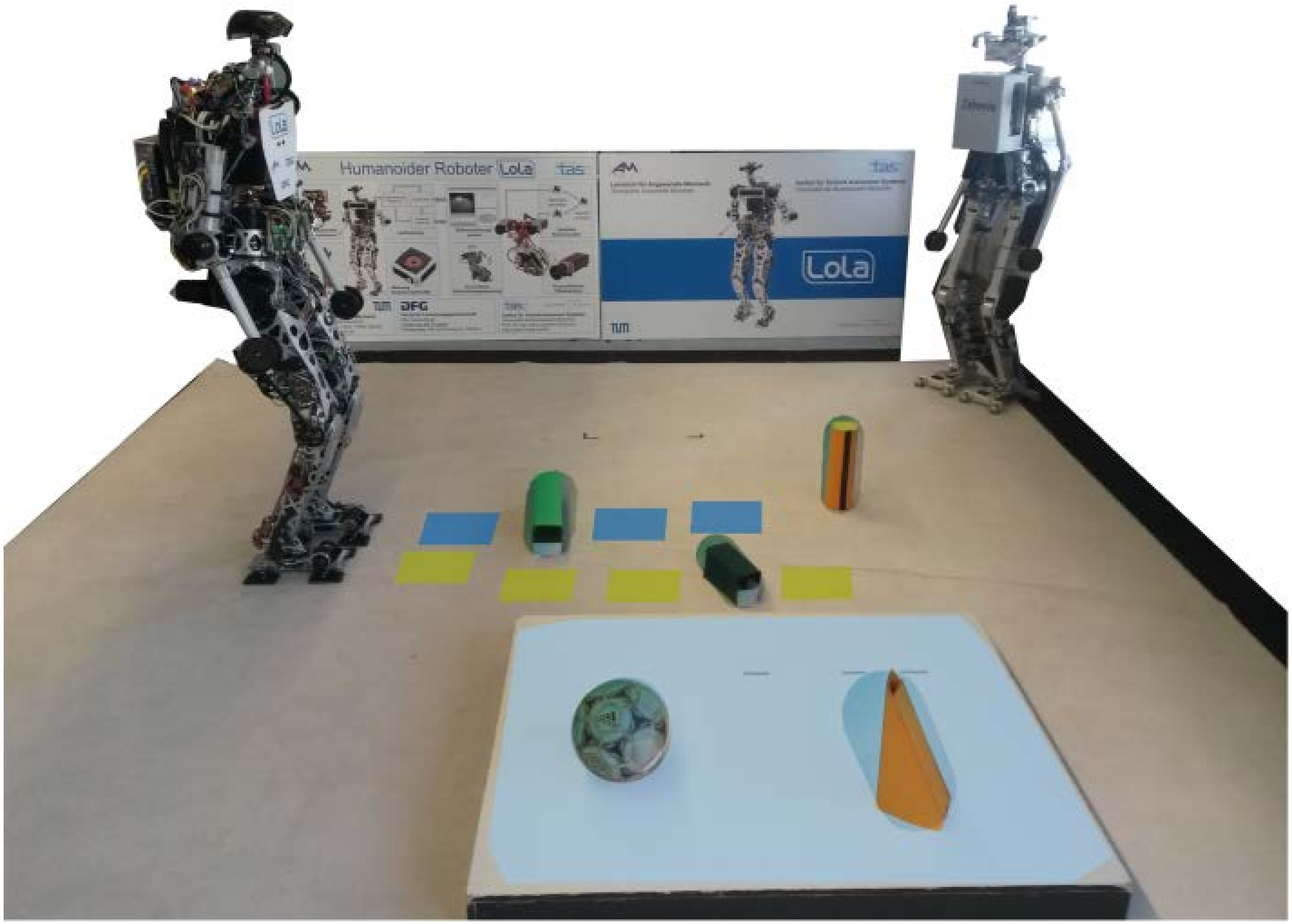
Results

Limitations

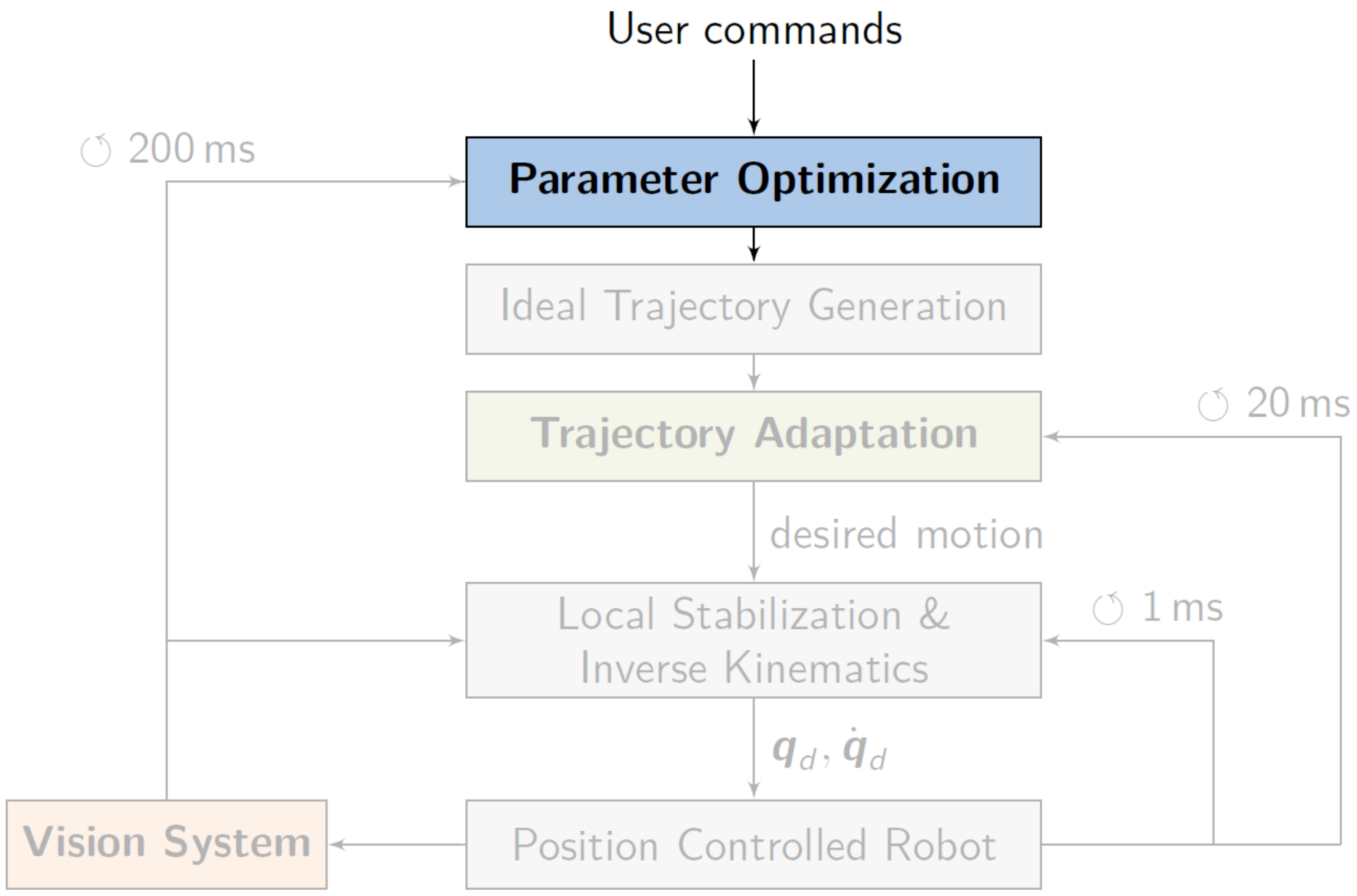
Perspectives

?


Humanoider Roboter Lola
Lola


Humanoider Roboter Lola
Lola

Real-Time Autonomous Navigation



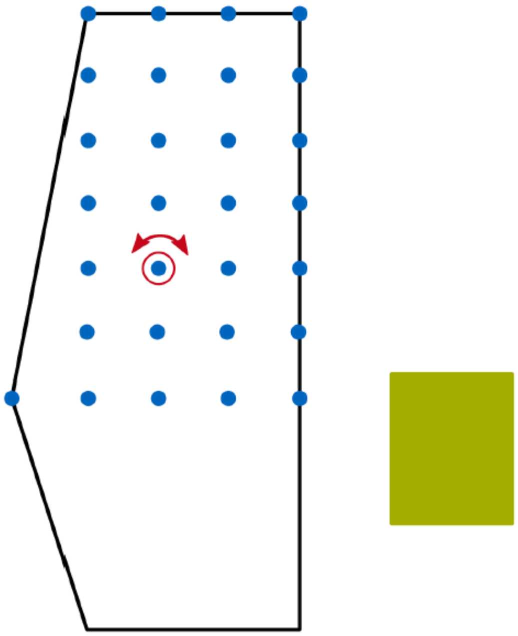
Navigation

Discrete Footholds

Navigation

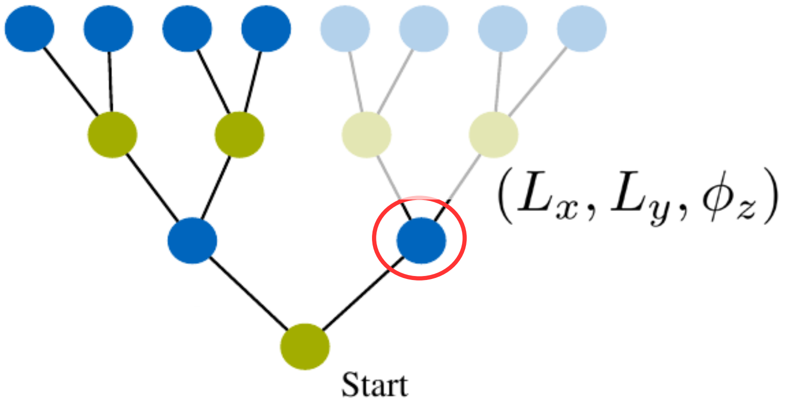
Discrete Footholds

Full-Discretization

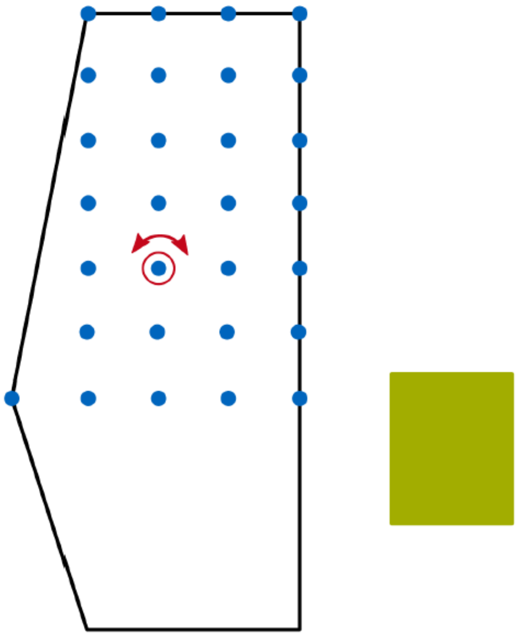


Navigation

Search Tree – A*-Search

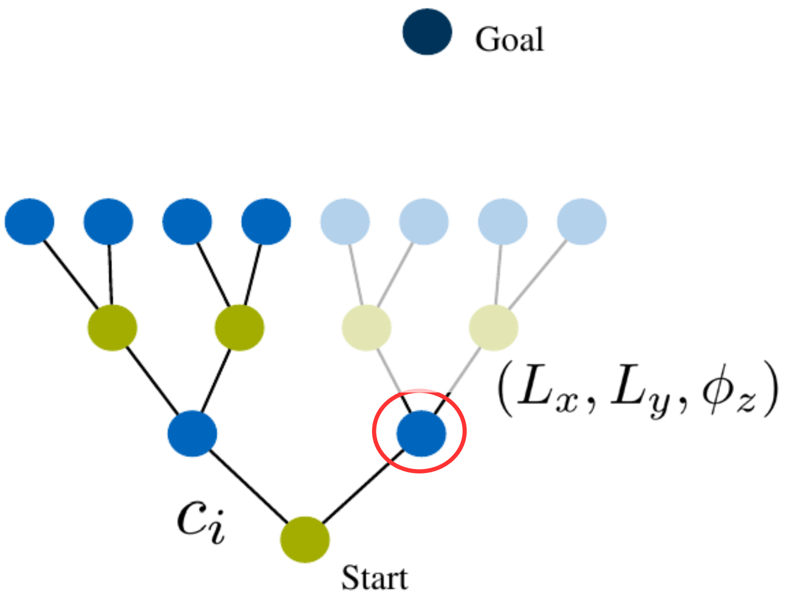


Full-Discretization

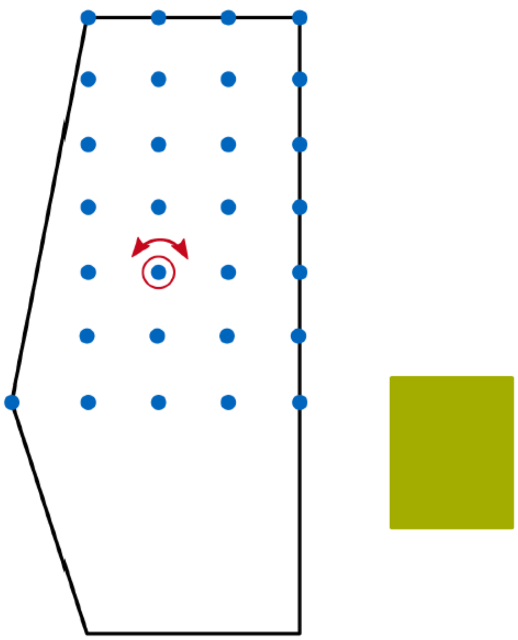


Navigation

Search Tree – A*-Search

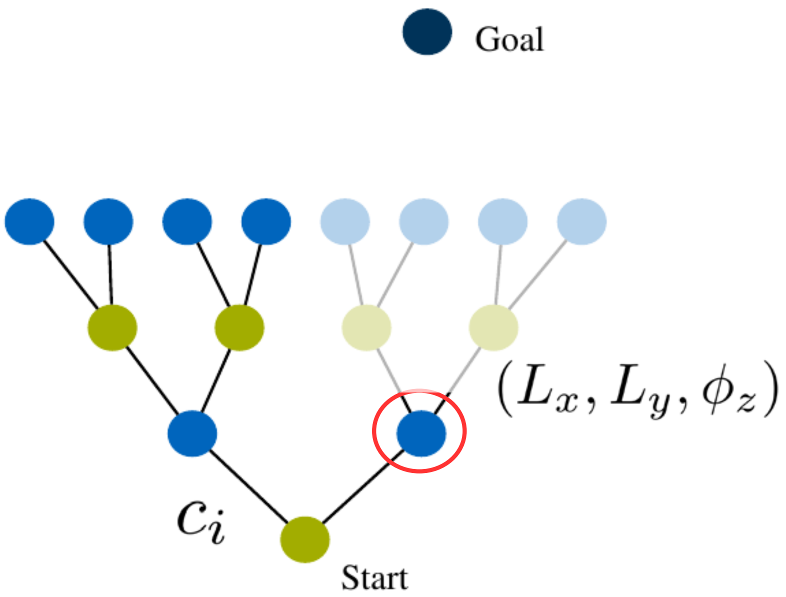


Full-Discretization



Node Analyses

Search Tree – A*-Search

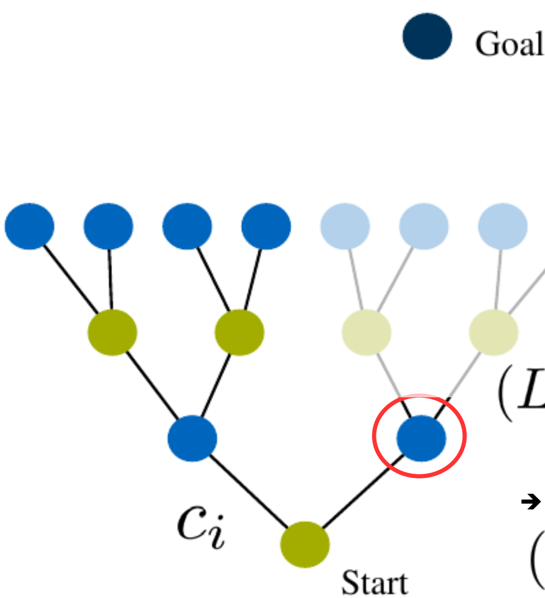


➔ 6D Pose? (L_z, ϕ_y, ϕ_z)

➔ Geometrically Valid?

Node Analyses

Search Tree – A*-Search

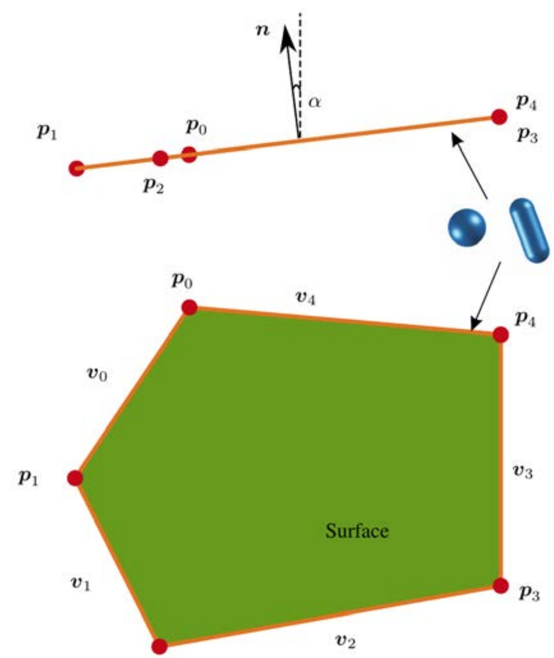


(L_x, L_y, ϕ_z)

➔ 6D Pose?

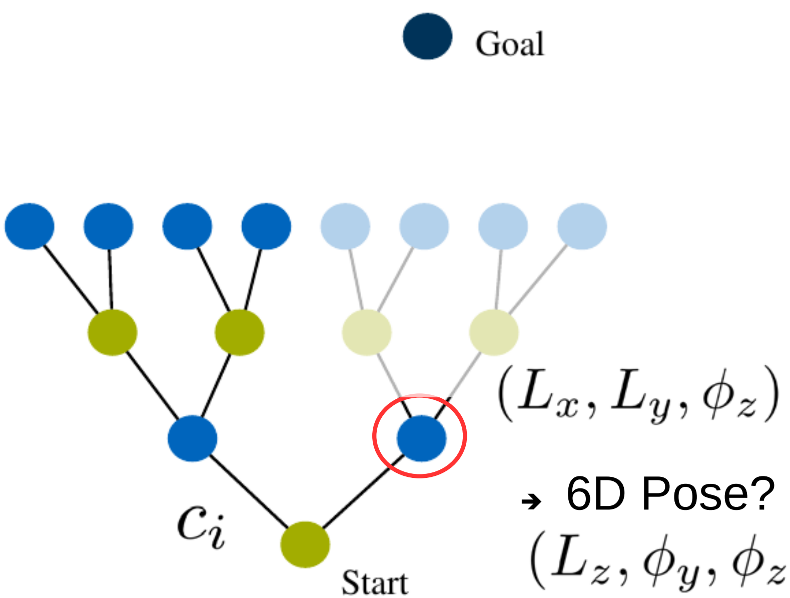
(L_z, ϕ_y, ϕ_z)

6D – Pose : Surface Modelling



Node Analyses

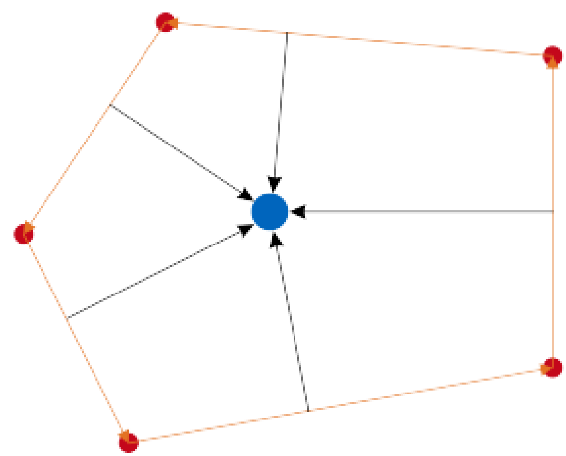
Search Tree – A*-Search



→ 6D Pose?

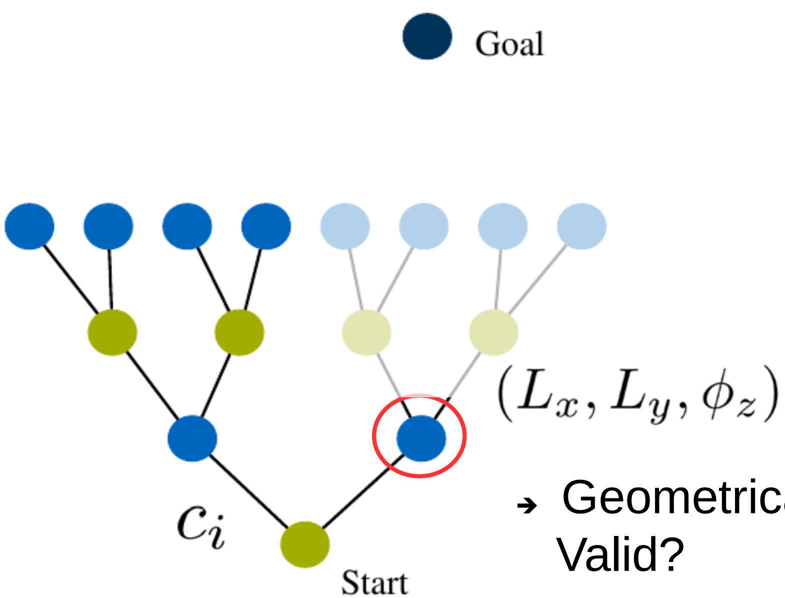
(L_z, ϕ_y, ϕ_z)

6D – Pose : Geometric Position



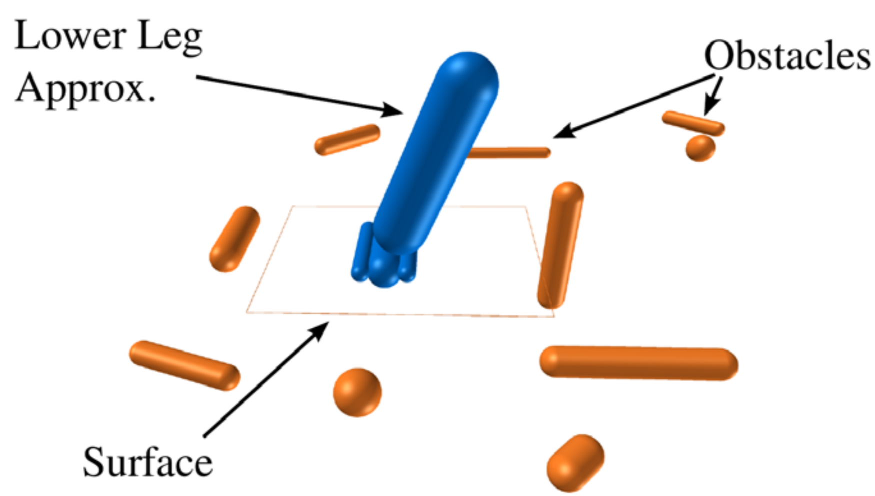
Node Analyses

Search Tree – A*-Search



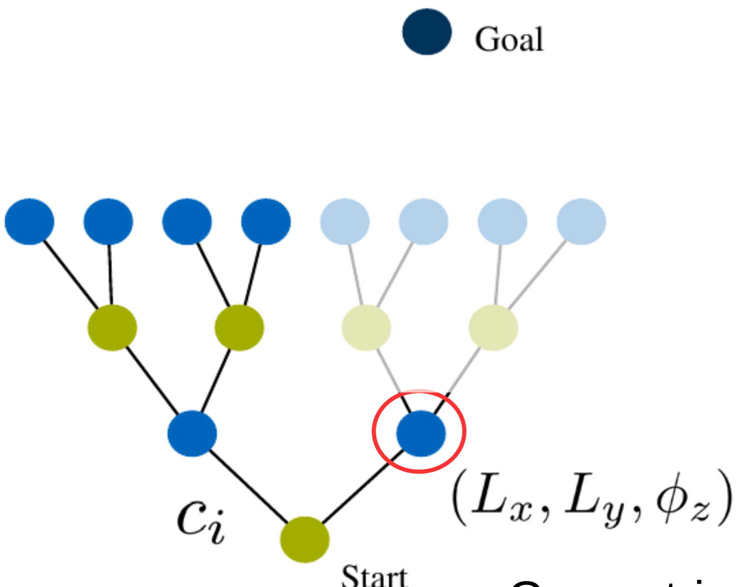
→ Geometrically Valid?

3D - Collision Checking



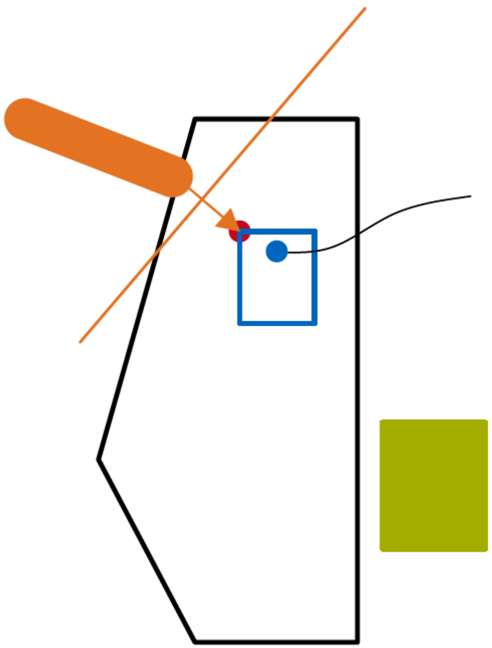
Real Time – Node Optimization

Search Tree – A*-Search

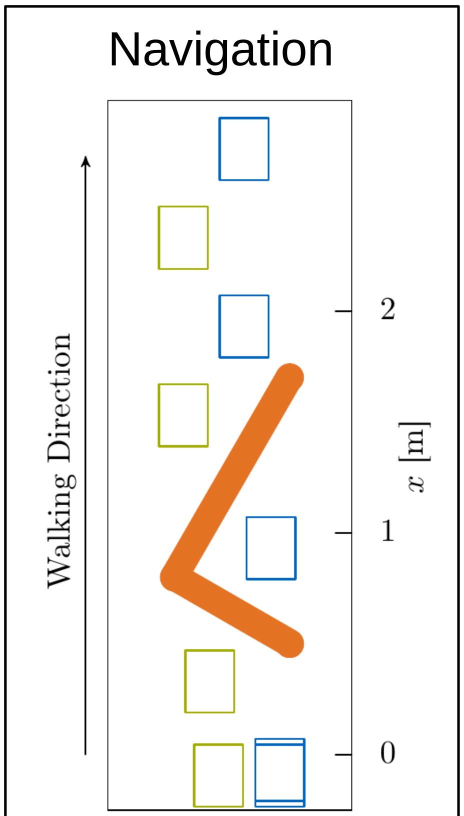


→ Geometrically Valid?

Local Adaption



Motion Generation

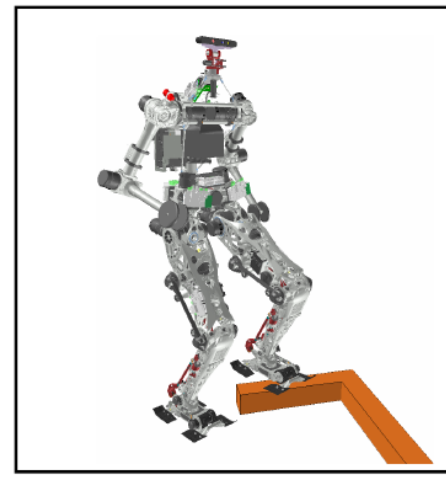


Full-Kinematic Motion?

Stable Walking?

$$T_{Calc} < T_{Step}$$

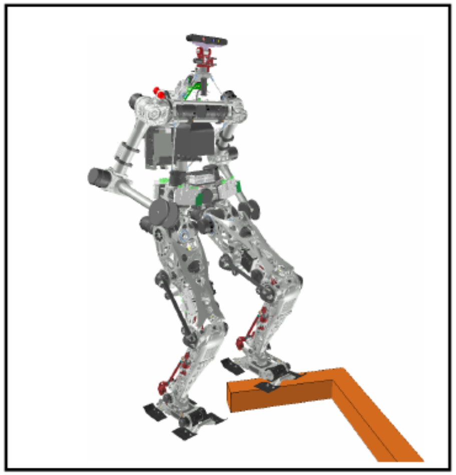
Local Stabilization & Inverse Kinematics



Motion Generation

$$\dot{q} = J^{\#}\dot{w}(t) + Nu(t)$$

Local Stabilization &
Inverse Kinematics

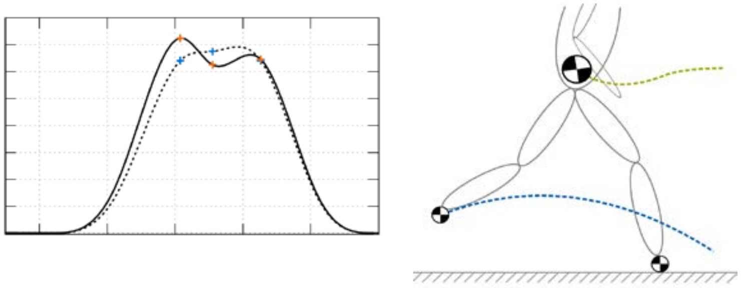


Motion Generation

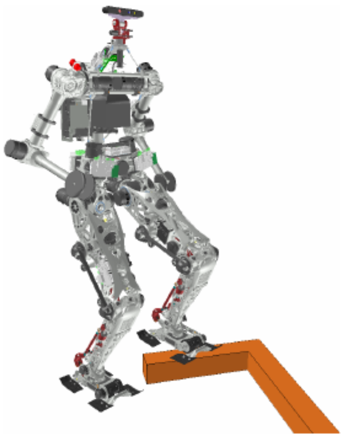
$$\dot{q} = J^{\#}\dot{w}(t) + Nu(t)$$

$$w(t) = w(t, p)$$

Dynamic Constraints



Local Stabilization & Inverse Kinematics

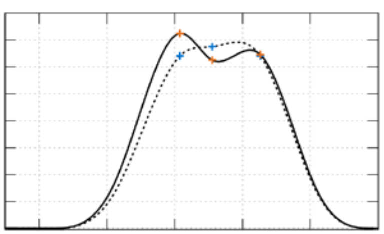


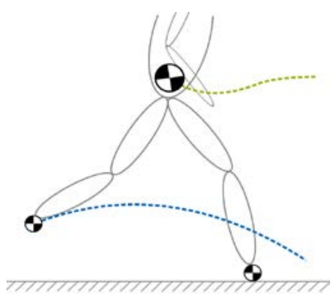
Motion Generation

$$\dot{q} = J^{\#}\dot{w}(t) + Nu(t)$$

$$w(t) = w(t,p)$$

Dynamic Constraints

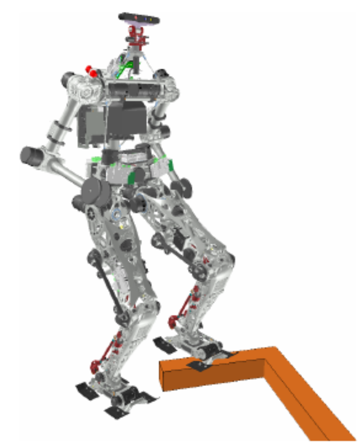




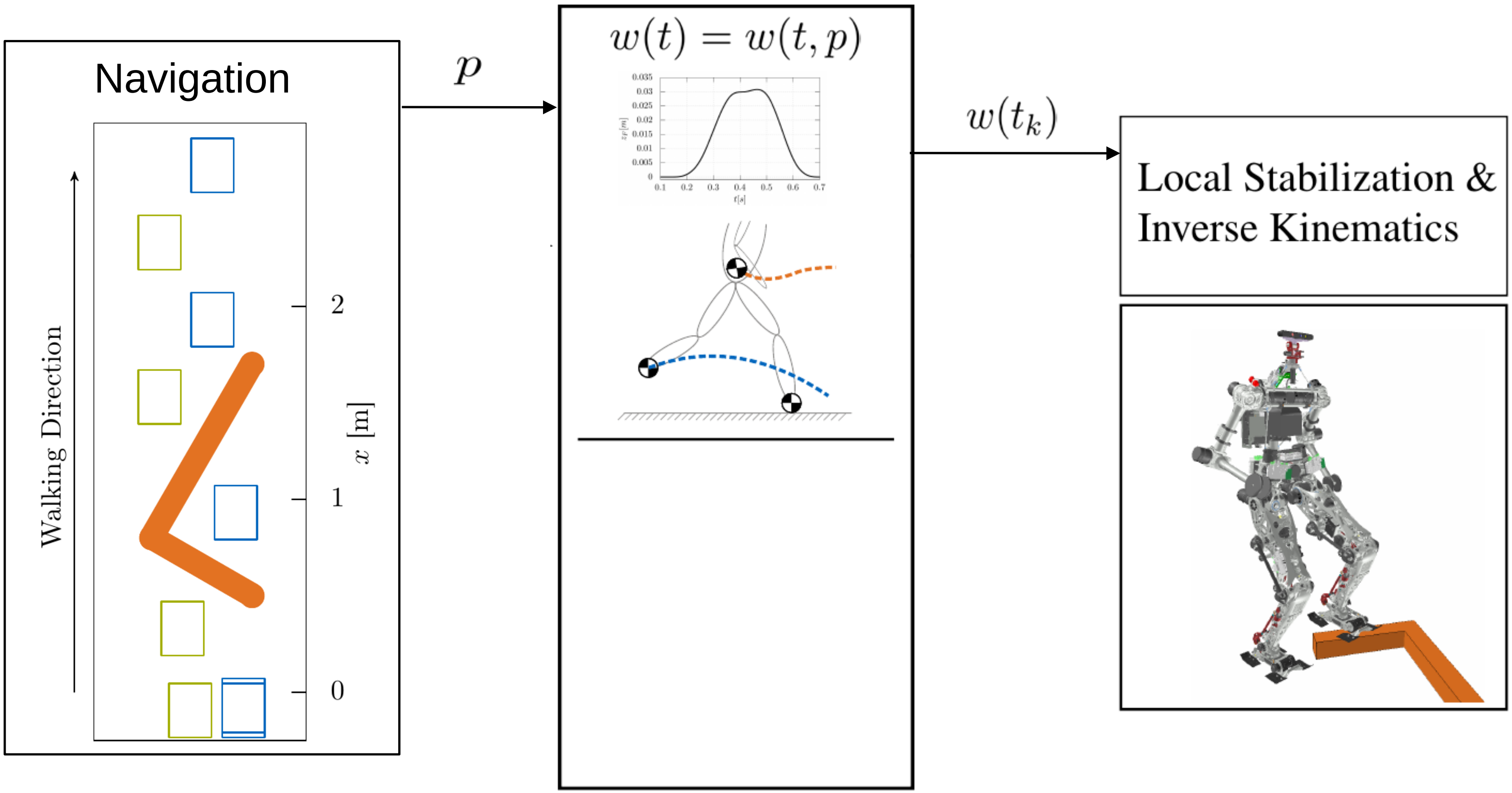
$$u = \alpha \frac{\partial L(q)}{\partial q}$$

Locally

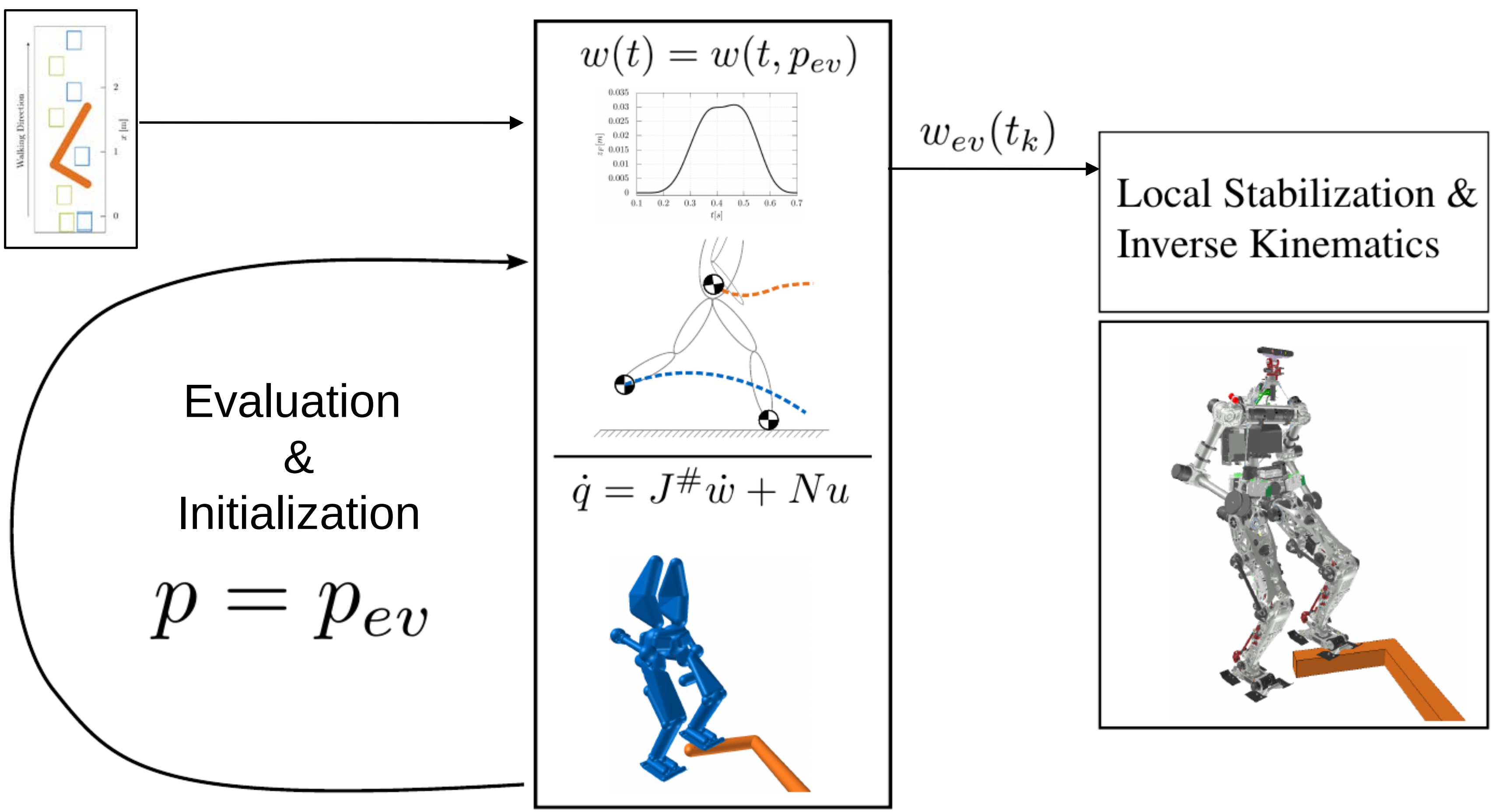
Local Stabilization & Inverse Kinematics



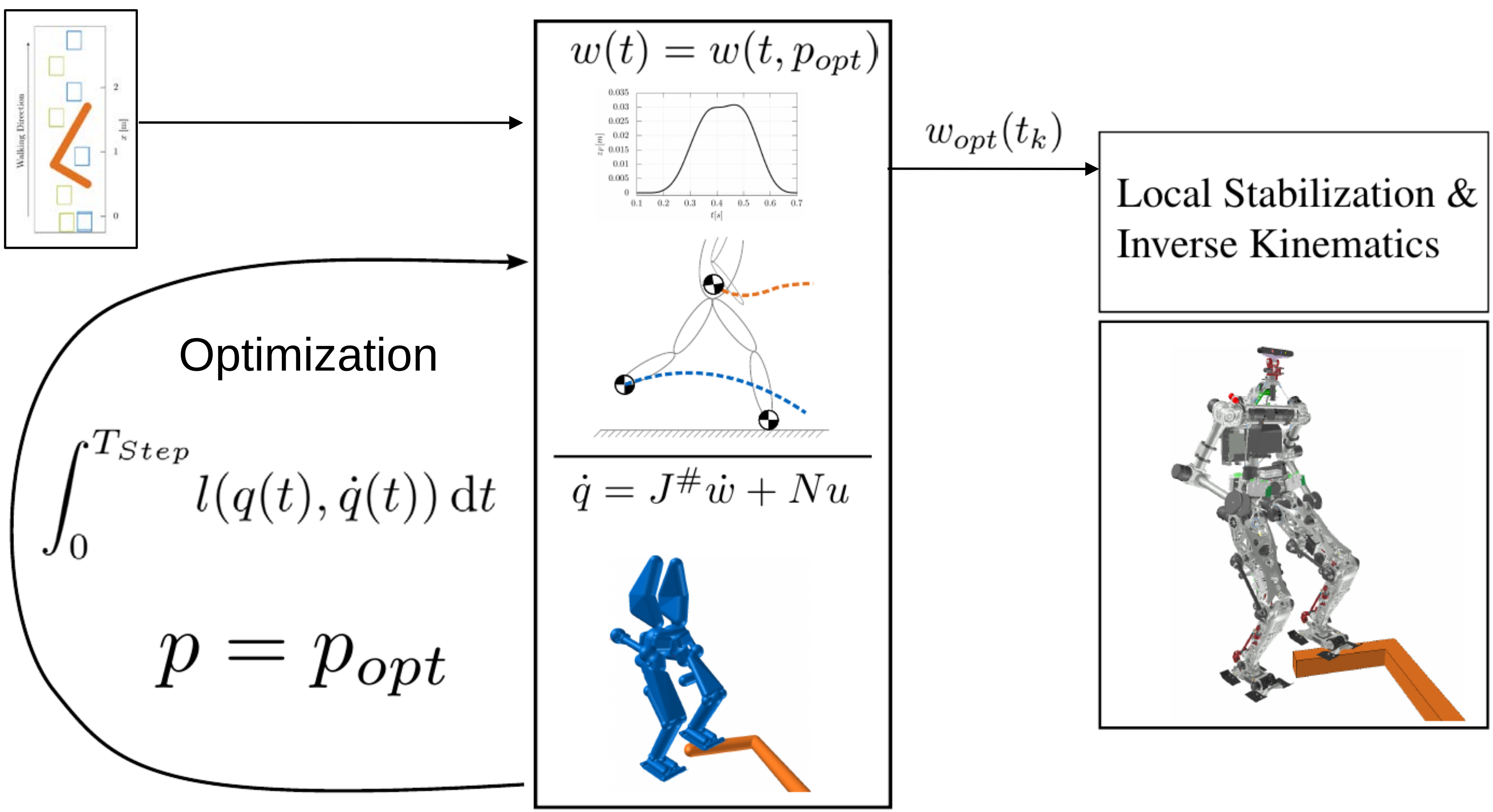
Predictive Kinematic Planning



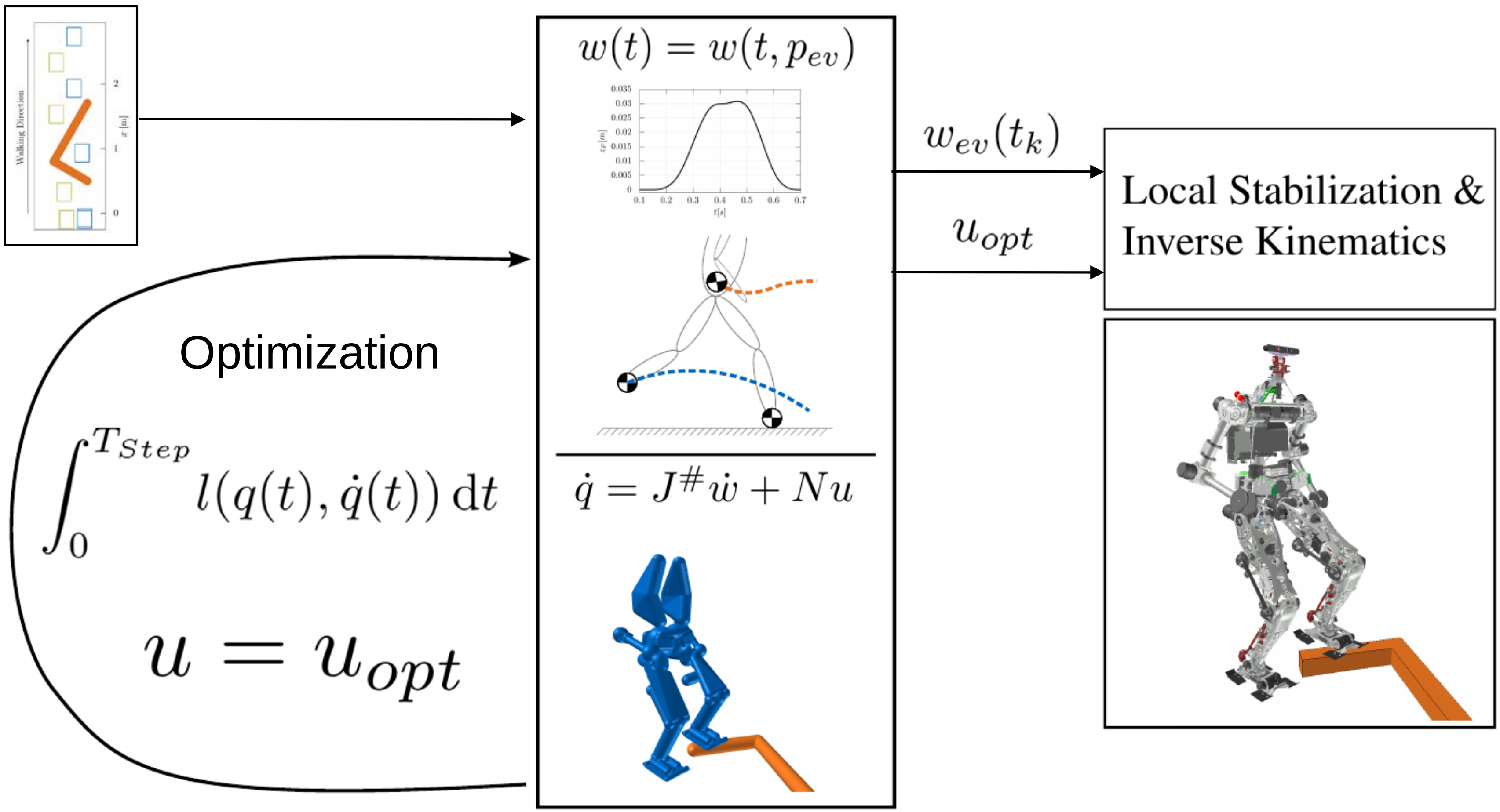
Predictive Kinematic Planning



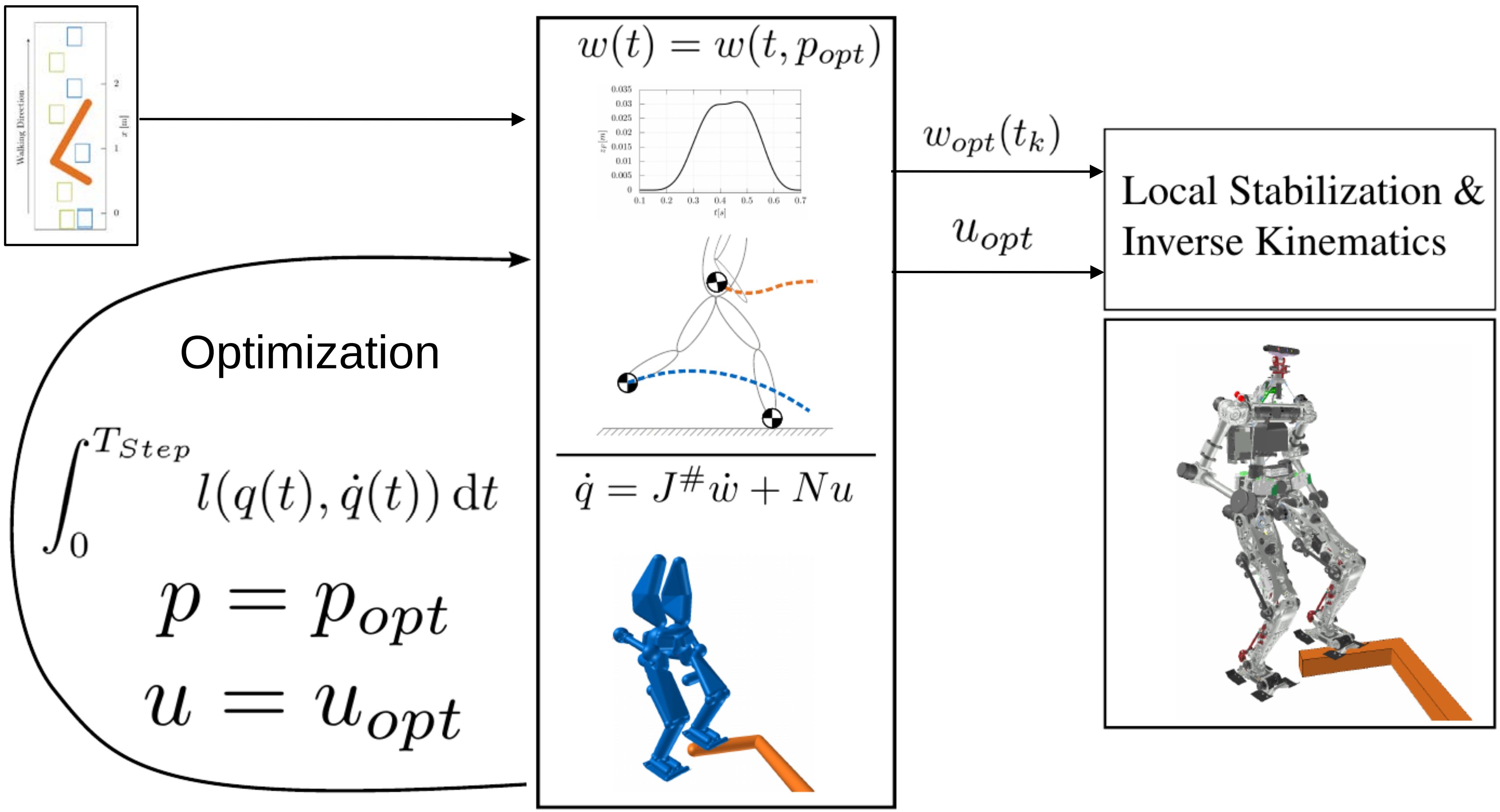
Predictive Kinematic Planning



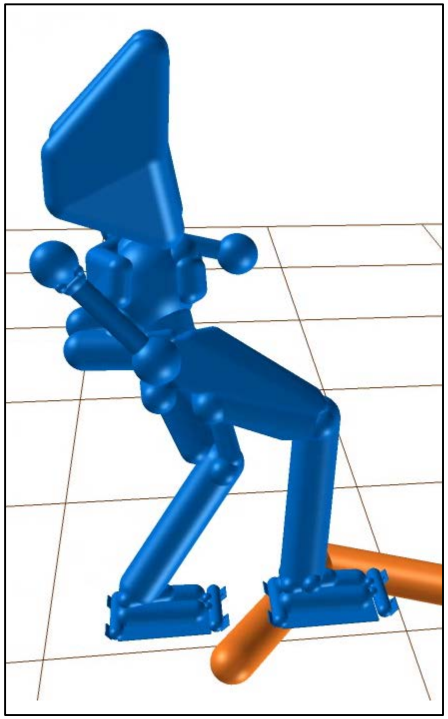
Predictive Kinematic Planning

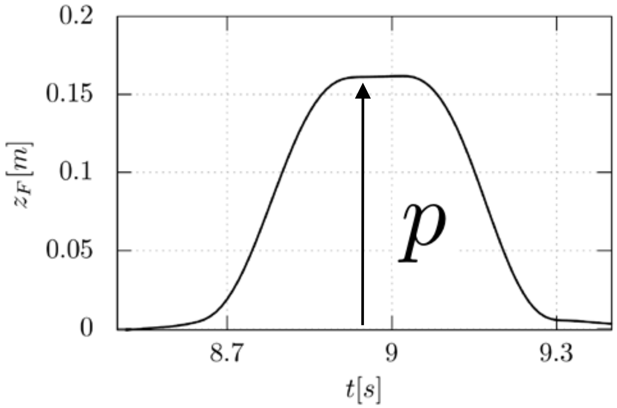


Predictive Kinematic Planning

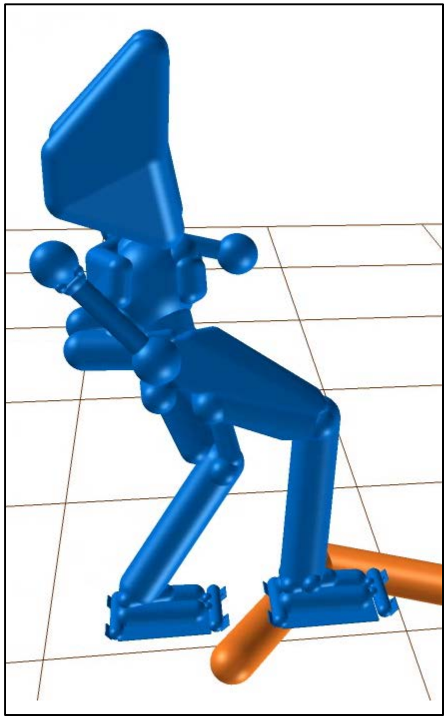


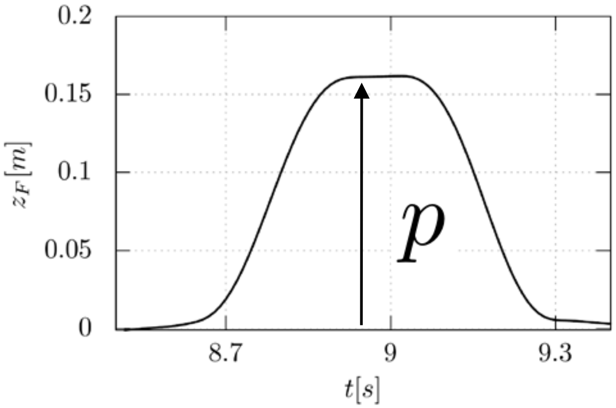
Reactive Adaptation

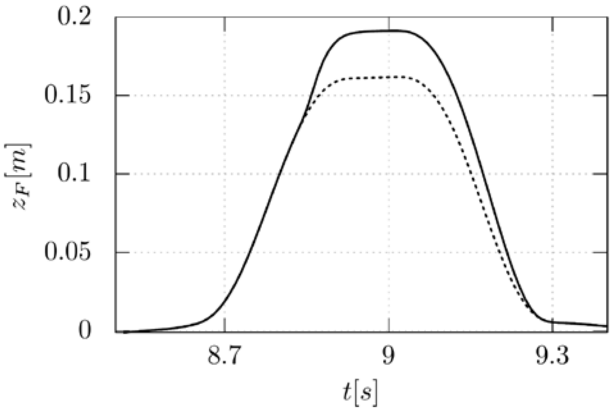




Predictive Kinematic Planning

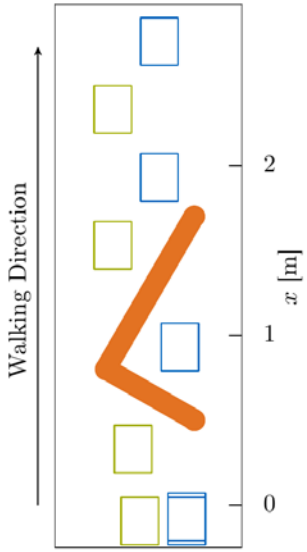




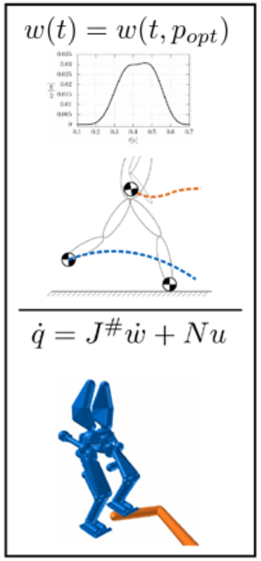


Navigation and Motion Generation

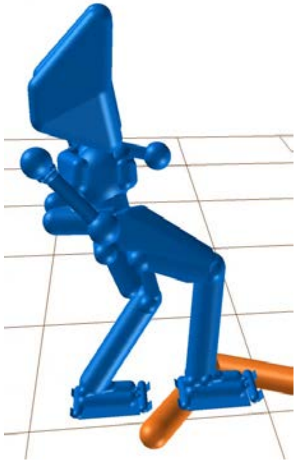
Navigation



Predictive Kinematics



Local Adaptation



Contents

Introduction and Motivation

Environment Modeling

Vision System

Navigation and Motion Generation

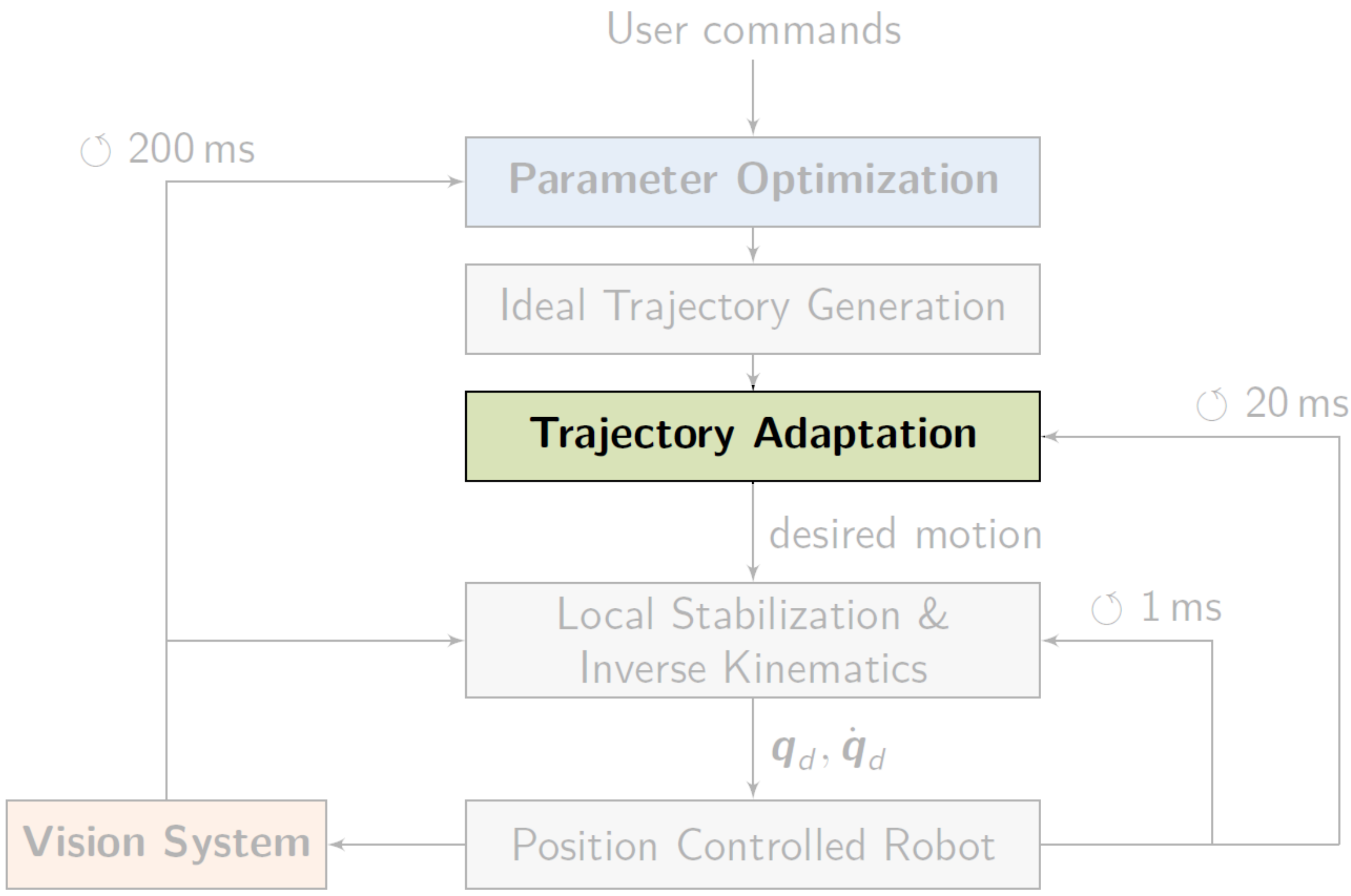
Stabilization

Results

Limitations

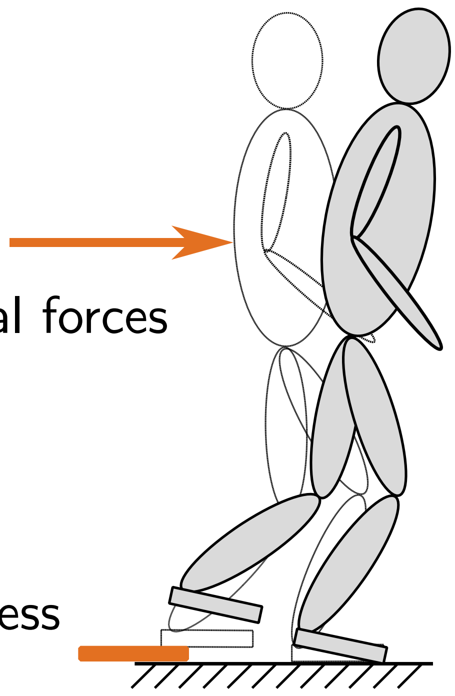
Perspectives

Real-Time Autonomous Navigation



Motivation

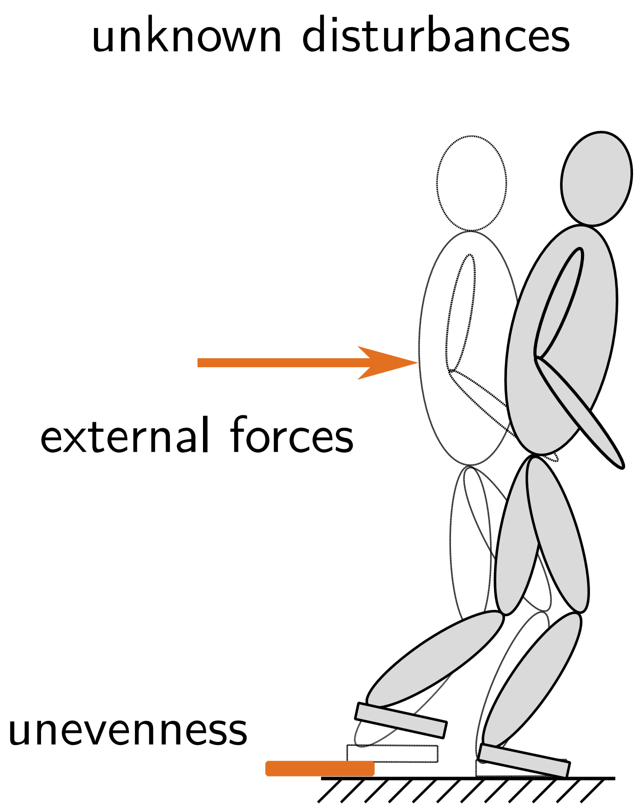
unknown disturbances



external forces

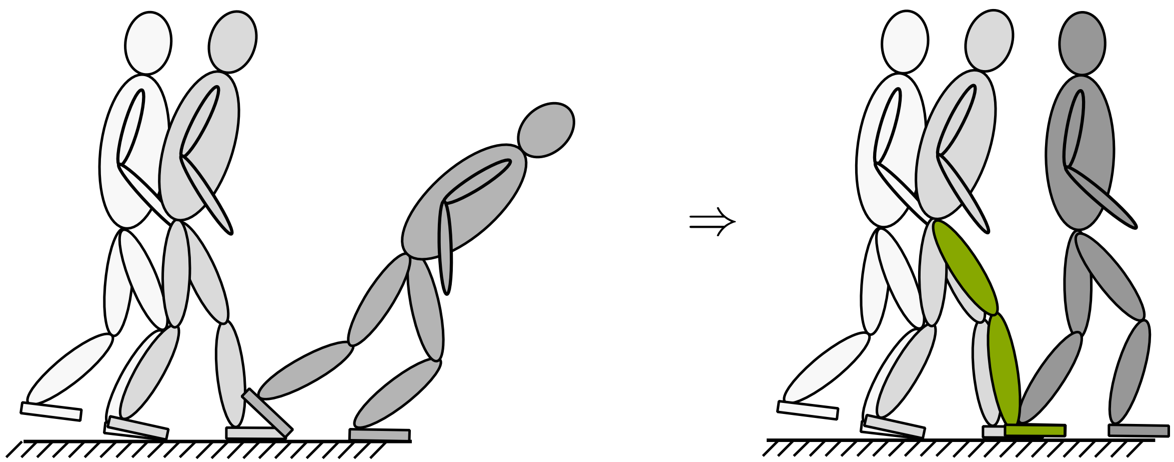
unevenness

Motivation



- current state?
- divergence?
- adapt **future** motion?

Solution?



Solution

- current state? $\longrightarrow$ state estimator
- divergence? $\longrightarrow$ prediction model
- adapt **future** motion? $\longrightarrow$ trajectory optimization

Solution

- current state? $\rightarrow$ state estimator
- divergence? $\rightarrow$ prediction model
- adapt **future** motion? $\rightarrow$ trajectory optimization

---

- input?
  - inertial measurements (IMU)
  - ideal planned motion

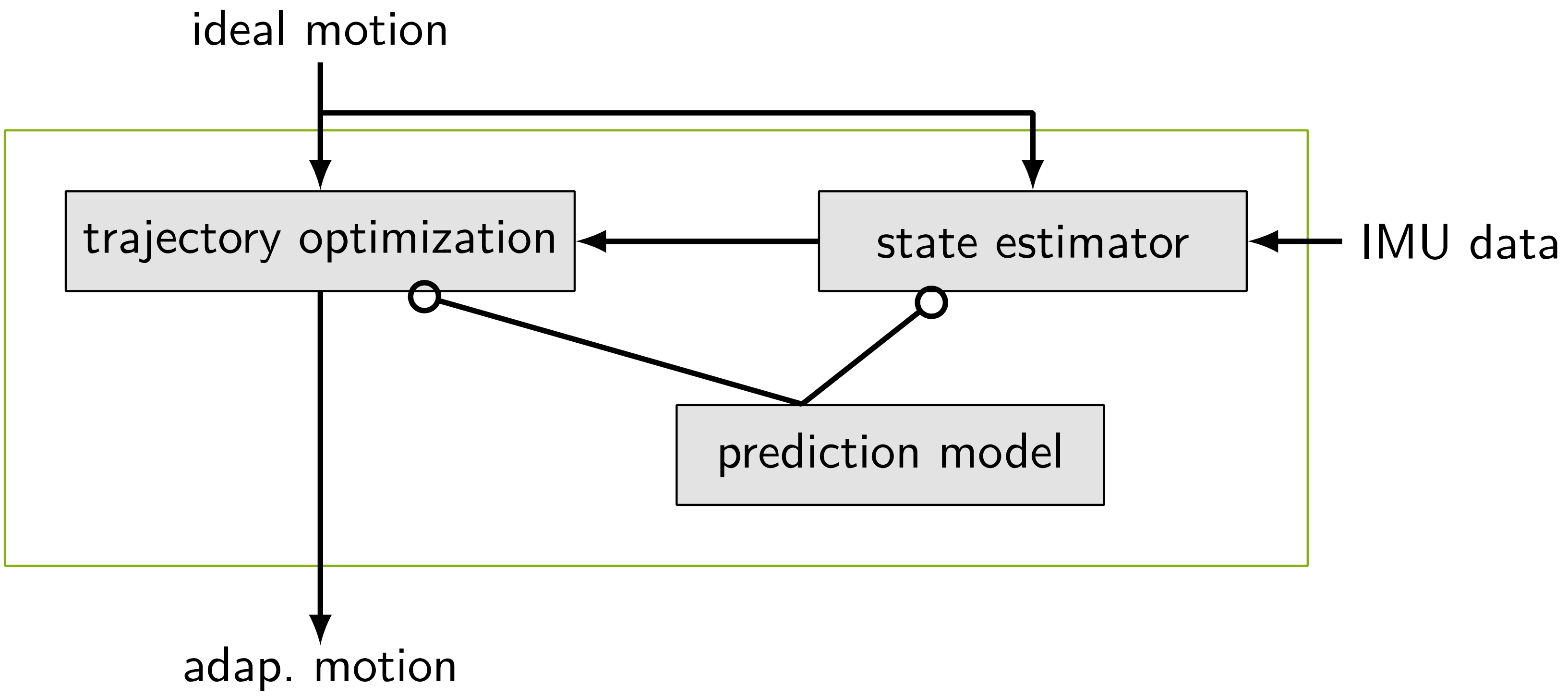
Solution

- current state? $\rightarrow$ state estimator
- divergence? $\rightarrow$ prediction model
- adapt **future** motion? $\rightarrow$ trajectory optimization

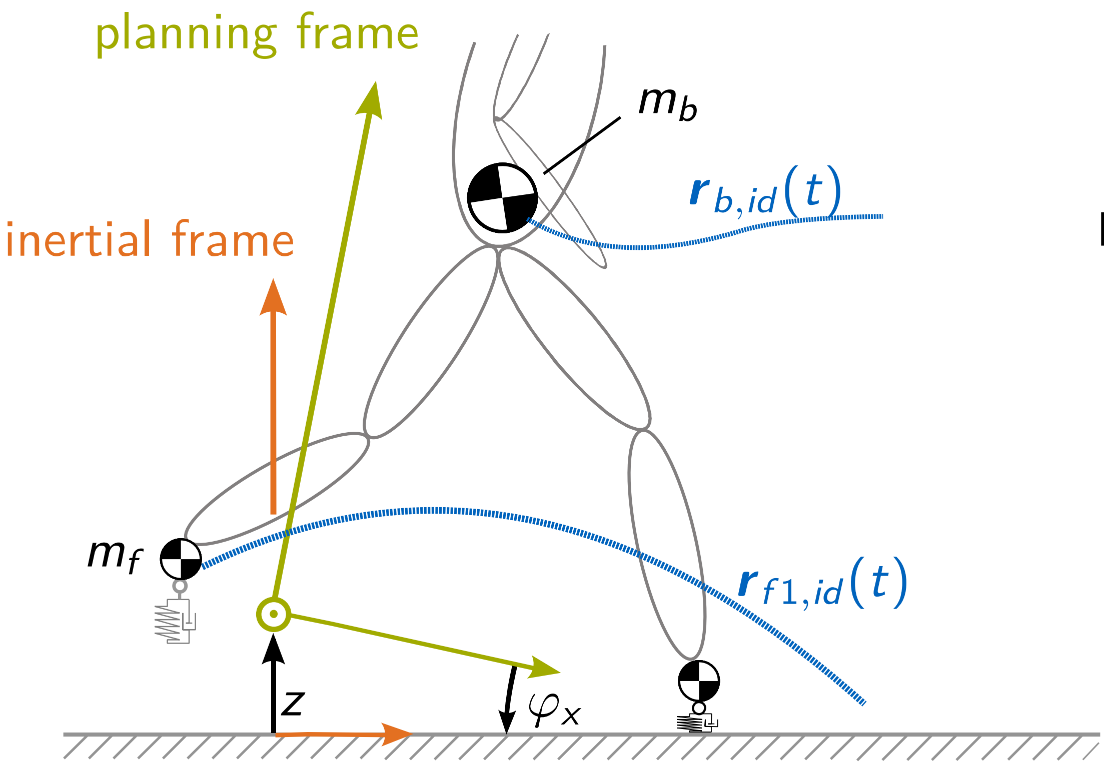
---

- input?
  - inertial measurements (IMU)
  - ideal planned motion
- output?
  - modified foot trajectories
  - modified center of gravity trajectories

Trajectory Adaptation



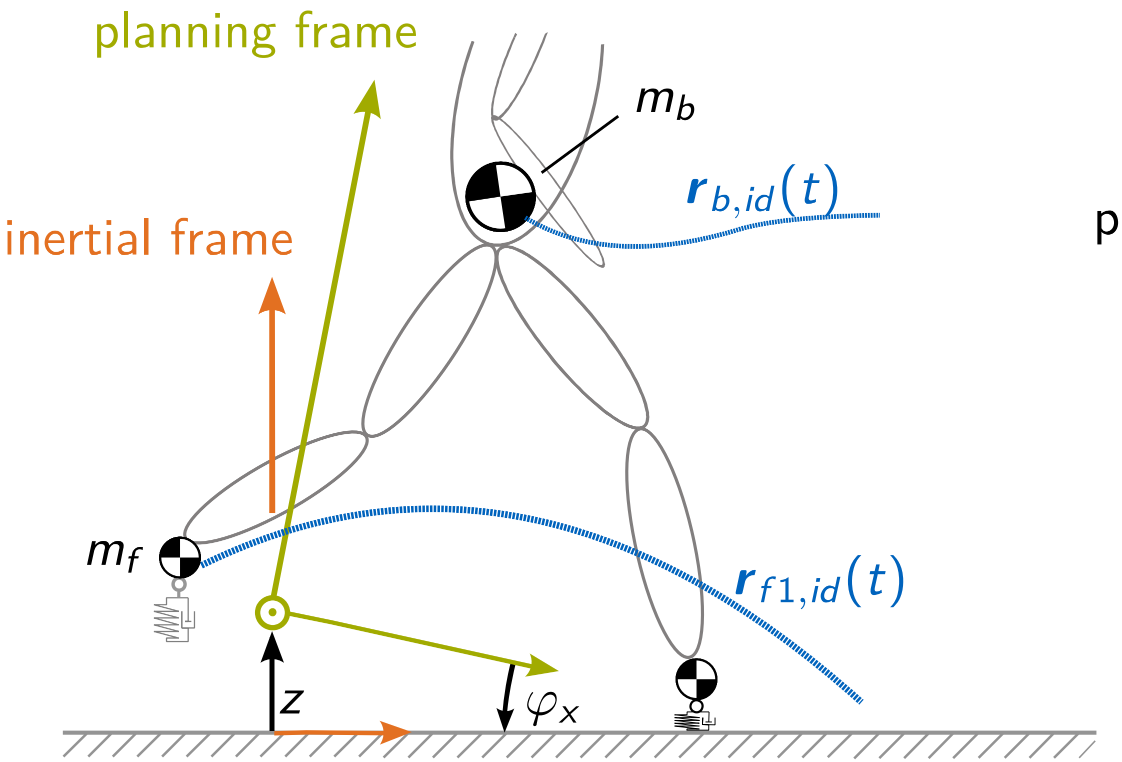
Dynamic Prediction Model



properties:

- planar
- 3 point masses

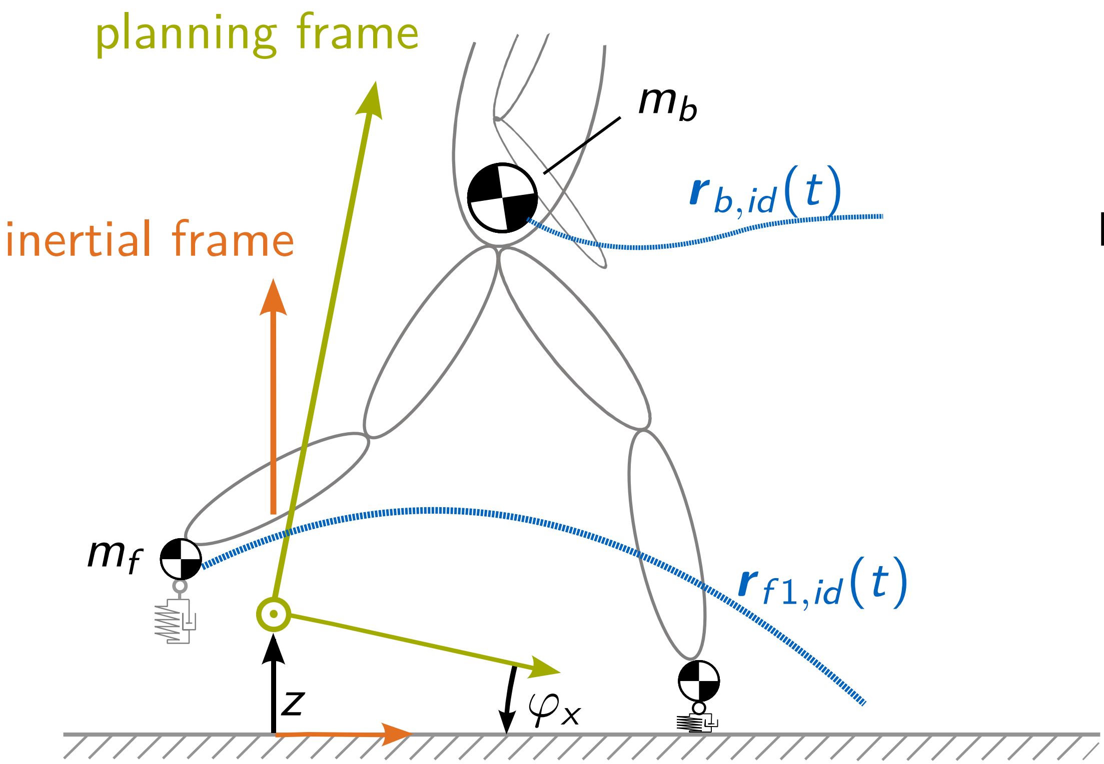
Dynamic Prediction Model



properties:

- planar
- 3 point masses
- 2 unilateral contacts
- 2 passive dofs

Dynamic Prediction Model



properties:

- planar
- 3 point masses
- 2 unilateral contacts
- 2 passive dofs
- local contact force control

Dynamic Prediction Model

- nonlinear dynamics

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \boldsymbol{\lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) + \boldsymbol{T}_s(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

Dynamic Prediction Model

- nonlinear dynamics

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \boldsymbol{\lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) + \boldsymbol{T}_s(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

- switching contact states

$$\boldsymbol{\lambda} = \sum_{i \in N_C} \left(\frac{\partial \boldsymbol{r}_{Fi}}{\partial \boldsymbol{q}} \right)^T \boldsymbol{F}_i$$

Dynamic Prediction Model

- nonlinear dynamics

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \boldsymbol{\lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) + \boldsymbol{T}_s(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

- switching contact states

$$\boldsymbol{\lambda} = \sum_{i \in N_C} \left(\frac{\partial \boldsymbol{r}_{Fi}}{\partial \boldsymbol{q}} \right)^T \boldsymbol{F}_i$$

- local force control

$$\boldsymbol{T}_s = [0, \, \mathrm{sat}(-K_p \Delta\varphi_x - K_d \Delta\dot{\varphi}_x)]^T$$

Dynamic Prediction Model

- nonlinear dynamics

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \boldsymbol{\lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) + \boldsymbol{T}_s(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

- switching contact states

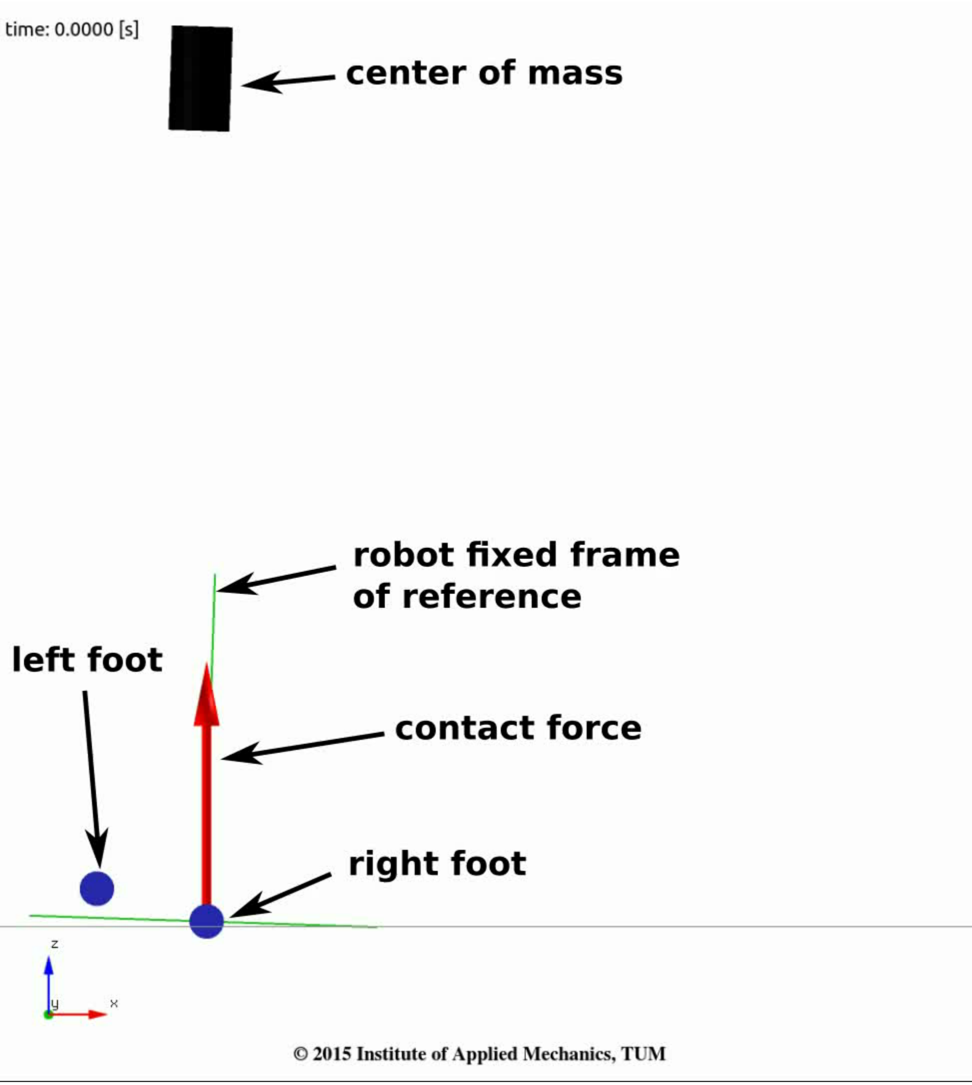
$$\boldsymbol{\lambda} = \sum_{i \in N_C} \left(\frac{\partial \boldsymbol{r}_{Fi}}{\partial \boldsymbol{q}} \right)^T \boldsymbol{F}_i$$

- local force control

$$\boldsymbol{T}_s = [0, \mathrm{sat}(-K_p \Delta\varphi_x - K_d \Delta\dot{\varphi}_x)]^T$$

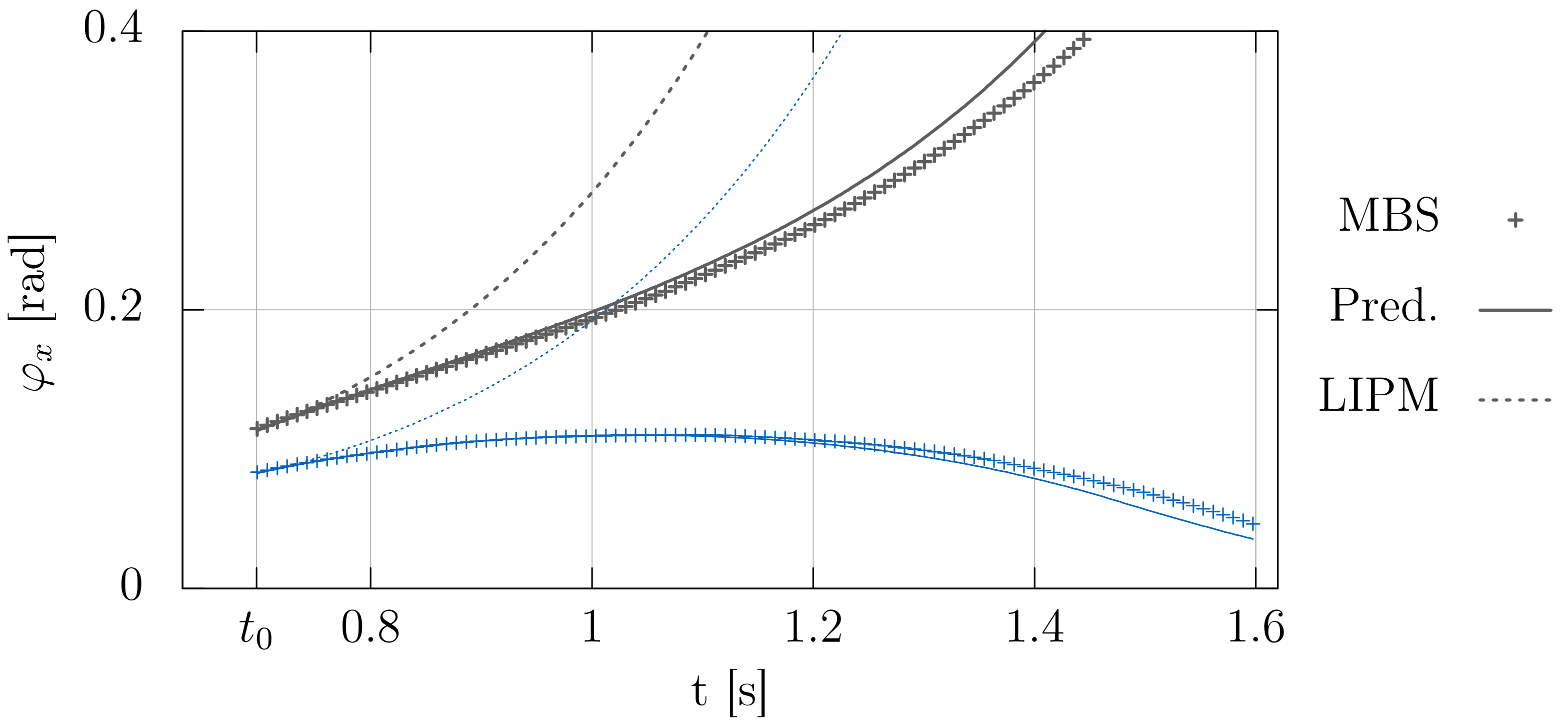
- computation time (Horizon $T = 1\,\mathrm{s}$, forward Euler $\triangle t = 3\,\mathrm{ms}$)
  $\rightarrow \quad \approx 50\,\mu\mathrm{s}$

Prediction Result



Verification – Simulation

proposed model vs. pendulum:

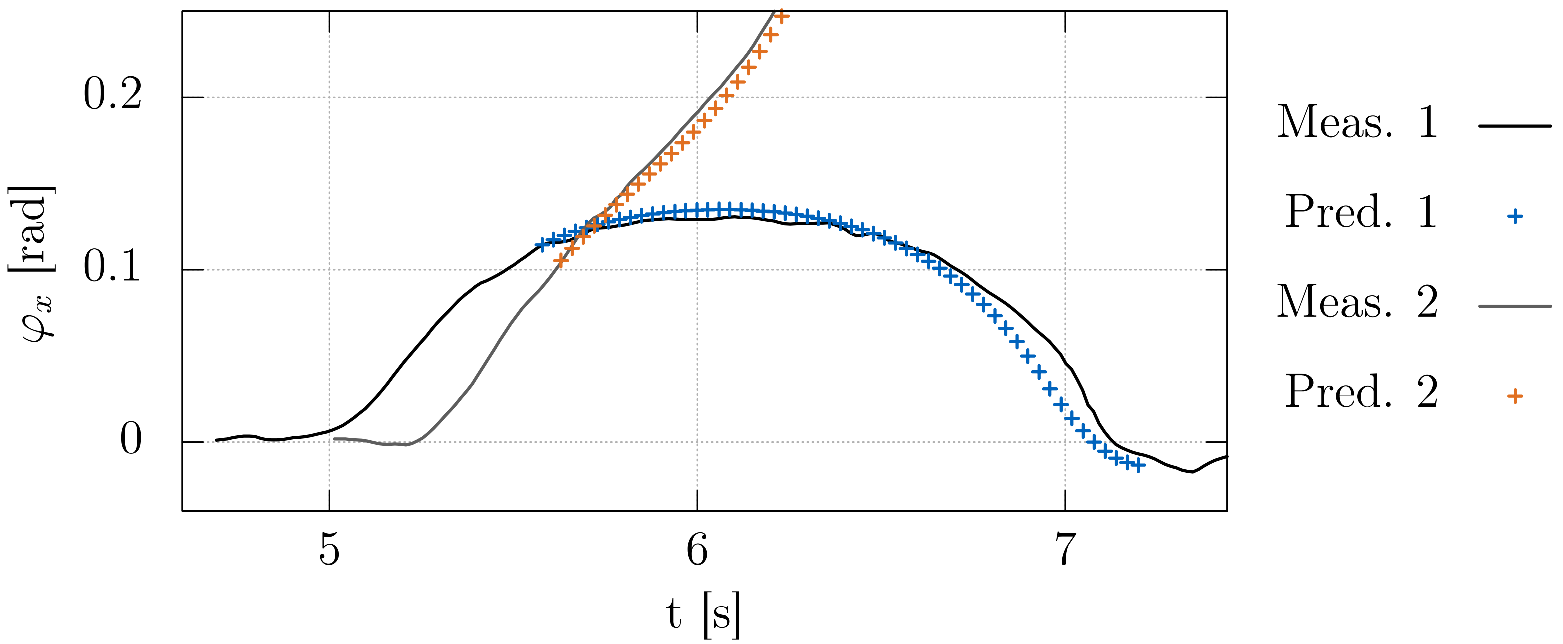


Verification – Measurements

- absolute inclination φ
- measurement vs. prediction model

Verification – Measurements

- absolute inclination φ
- measurement vs. prediction model

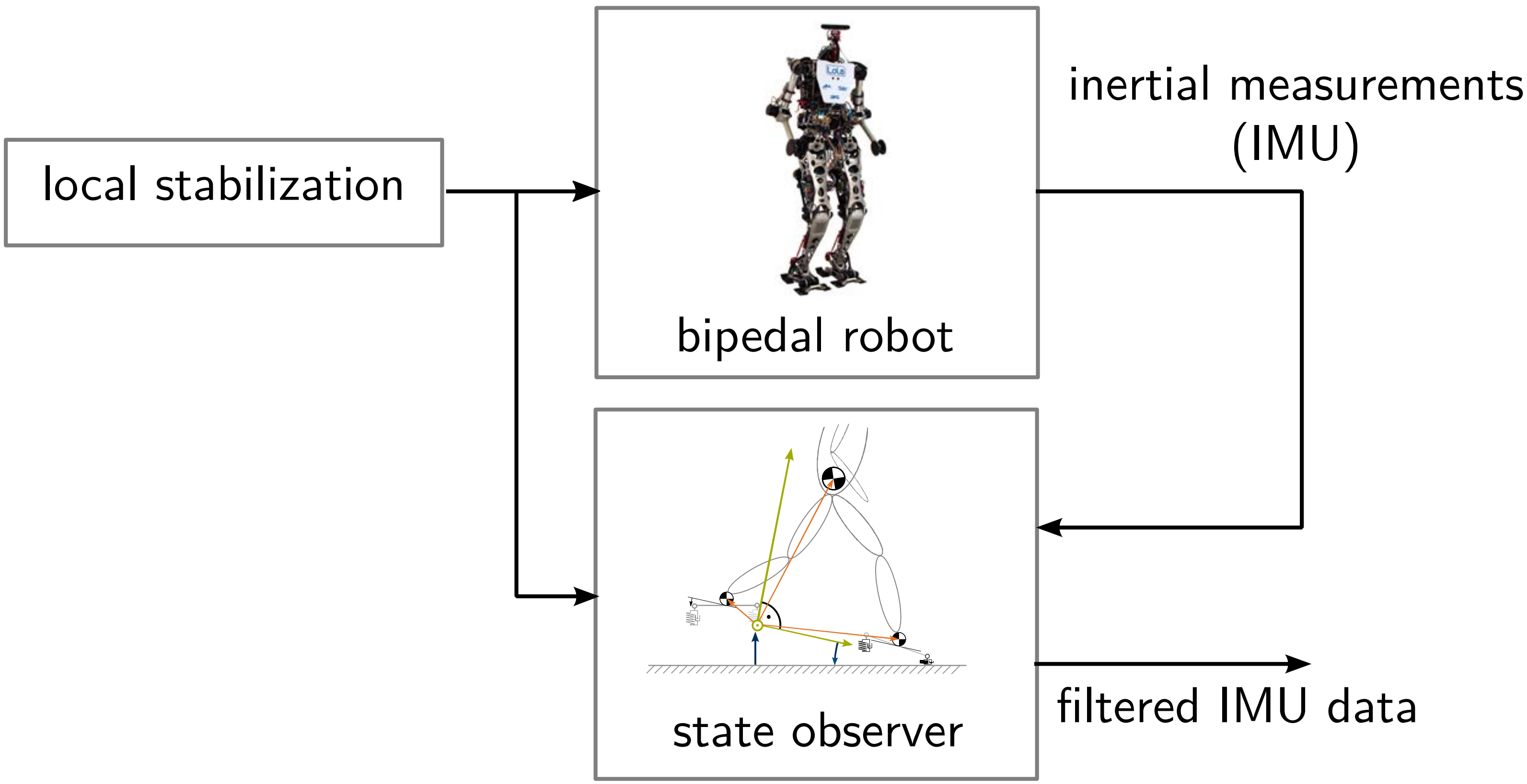


State Estimation - Extended Kalman Filter

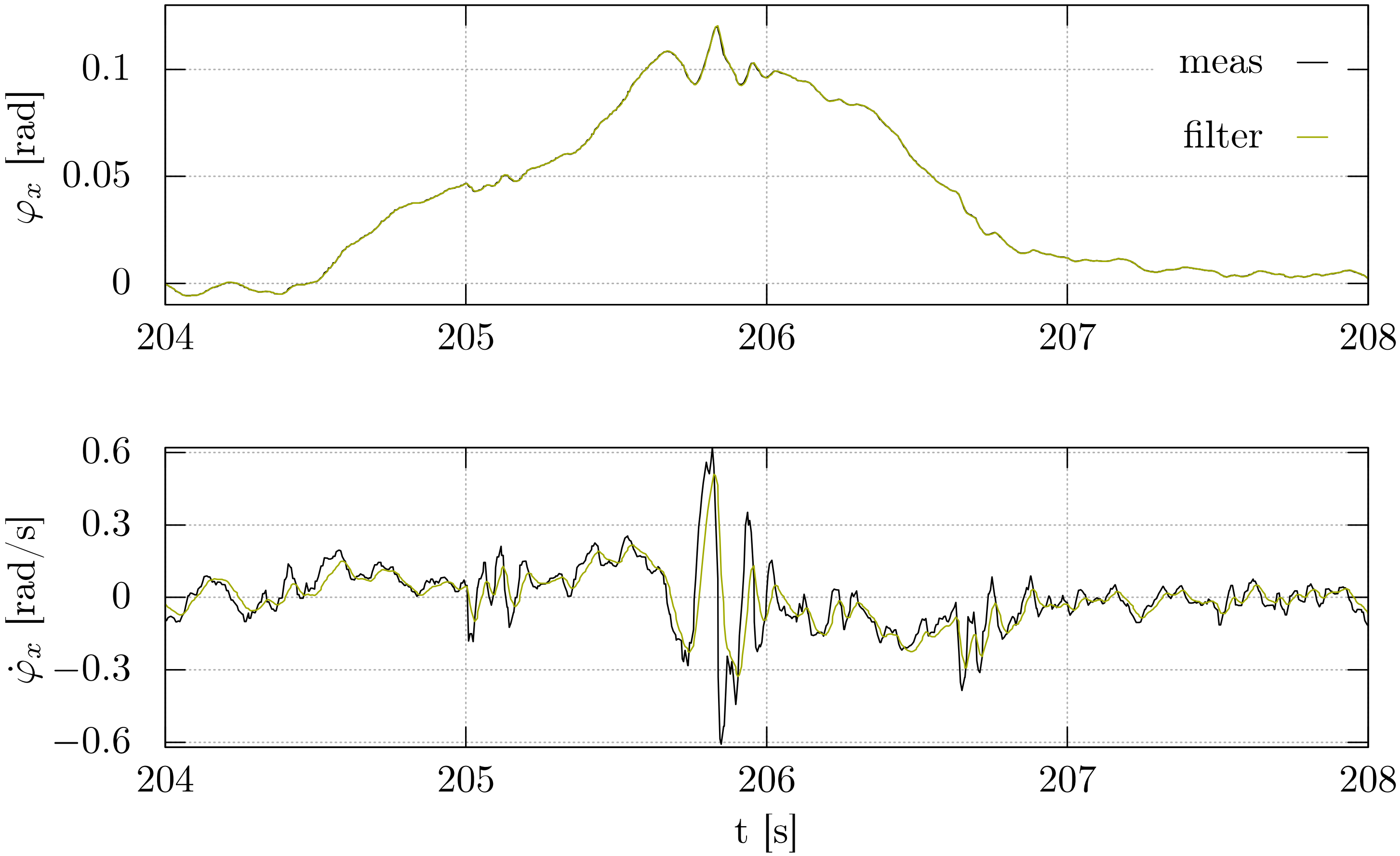
estimate for inertial/absolute pose

State Estimation - Extended Kalman Filter

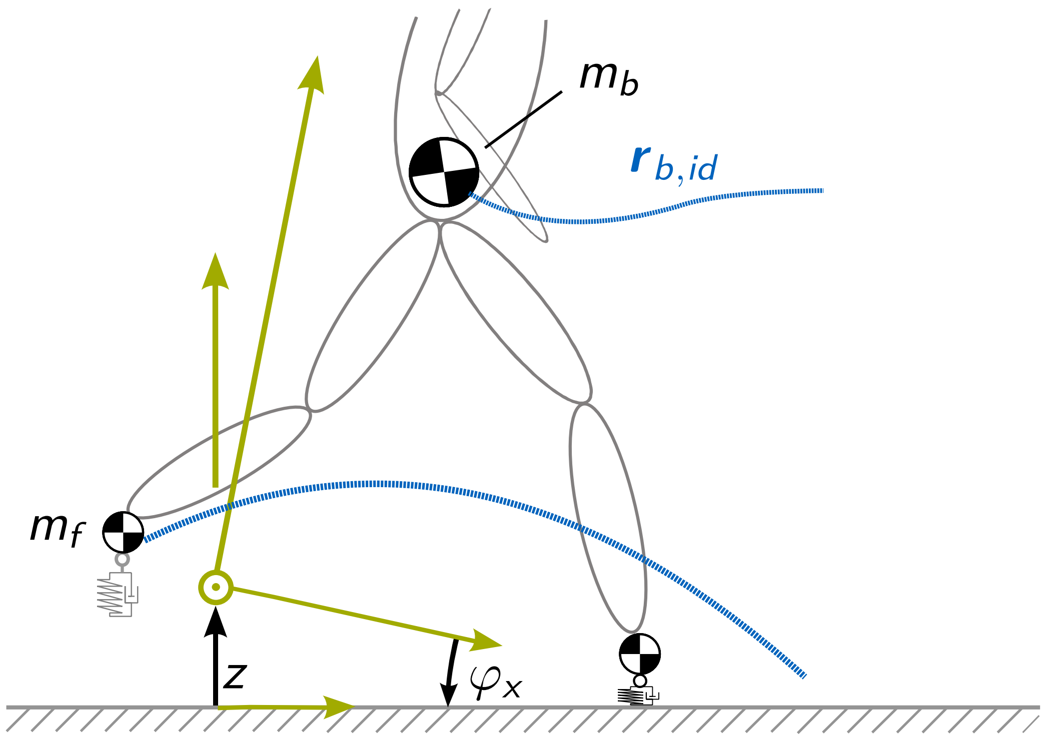
estimate for inertial/absolute pose



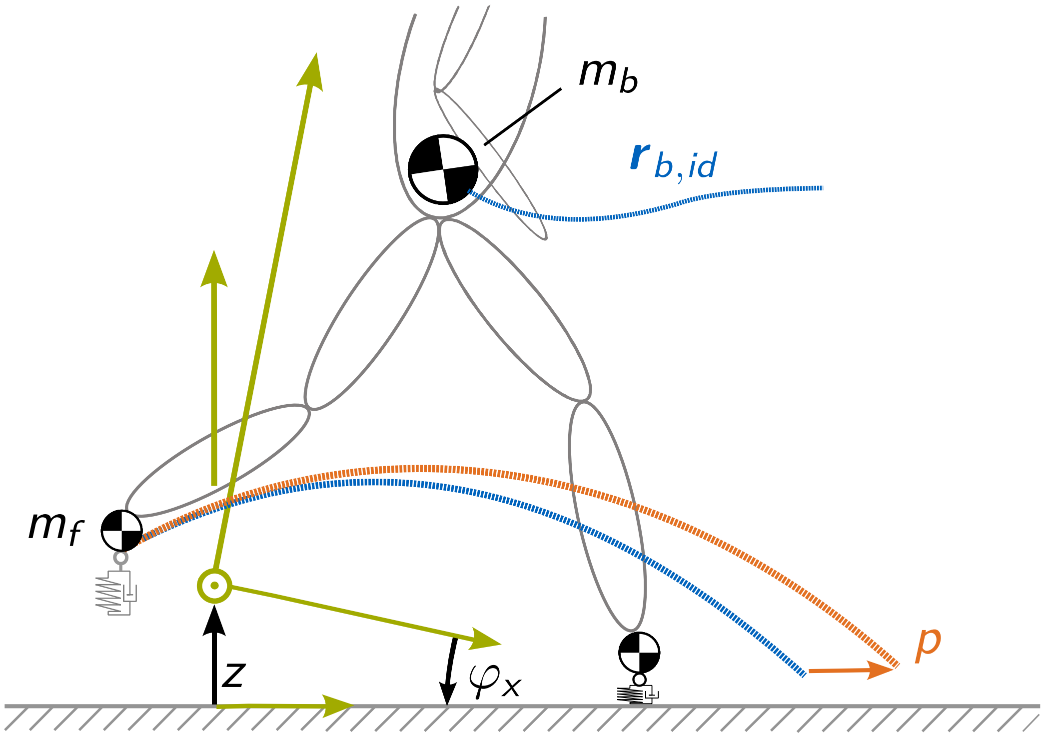
State Estimator - Kalman Filter



Model with Modifications



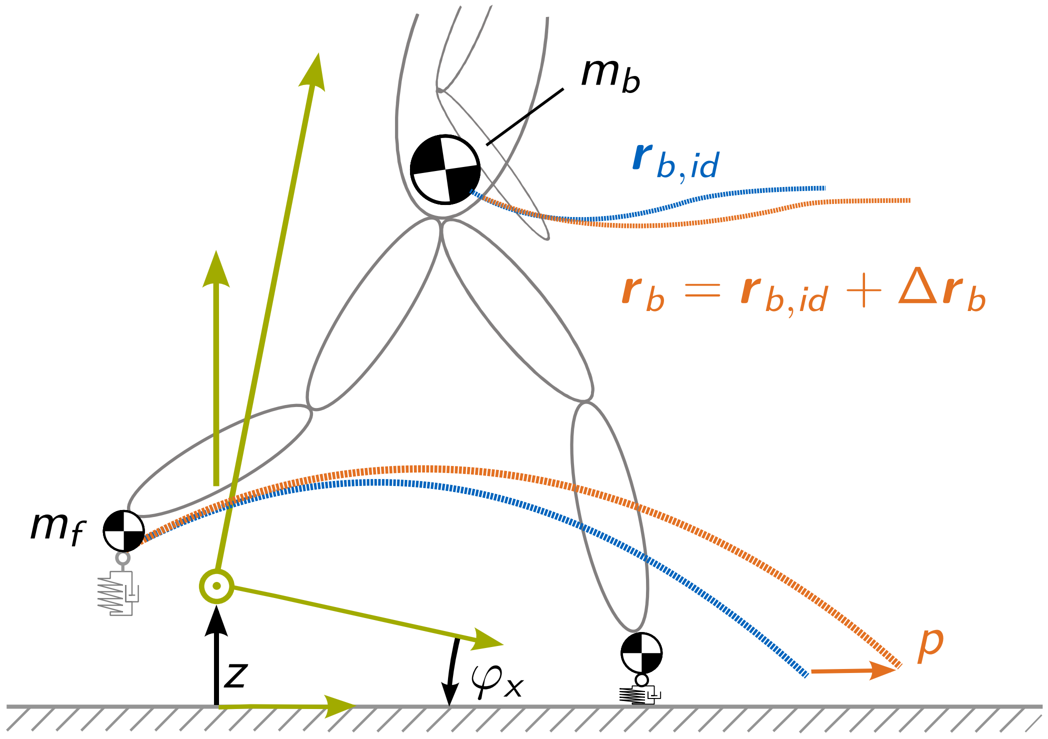
Model with Modifications



modifications:

- final swing foot position p

Model with Modifications



modifications:

- final swing foot position p
- CoG trajectory $\Delta \boldsymbol{r}_b(t)$

Model with Modifications

- swing foot modification (5th order polynomial):

$$\boldsymbol{r}_{f1} = \boldsymbol{r}_{f1,id} + \Delta\boldsymbol{r}_{f1}(p)$$

Model with Modifications

- swing foot modification (5th order polynomial):

$$\boldsymbol{r}_{f1} = \boldsymbol{r}_{f1,id} + \Delta \boldsymbol{r}_{f1}(p)$$

- CoG Modification:

$$\boldsymbol{r}_{b} = \boldsymbol{r}_{b,id} + \Delta \boldsymbol{r}_{b} = \begin{bmatrix} x_{b,id} \\ 0 \\ z_{b,id} \end{bmatrix} + \begin{bmatrix} \Delta x_{b} \\ 0 \\ 0 \end{bmatrix}, \qquad u = \Delta \ddot{x}_{b}$$

Model with Modifications

- swing foot modification (5th order polynomial):

$$\boldsymbol{r}_{f1} = \boldsymbol{r}_{f1,id} + \Delta\boldsymbol{r}_{f1}(p)$$

- CoG Modification:

$$\boldsymbol{r}_b = \boldsymbol{r}_{b,id} + \Delta\boldsymbol{r}_b = \begin{bmatrix} x_{b,id} \\ 0 \\ z_{b,id} \end{bmatrix} + \begin{bmatrix} \Delta x_b \\ 0 \\ 0 \end{bmatrix}, \qquad u = \Delta\ddot{x}_b$$

- EoM with modifications

$$\dot{\boldsymbol{x}} = \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ \Delta x_b \\ \Delta\dot{x}_b \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}[\lambda(p) + \boldsymbol{T}_s - \boldsymbol{h}] \\ \Delta\dot{x}_b \\ u \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}, t, p, u)\,.$$

Optimization Problem

$$
\begin{aligned}
\min_{p, u(t)} J = \min_{p, u(t)} & \left[s(\boldsymbol{x}(t_f), p) + \int\limits_{t_0}^{t_f} h(\boldsymbol{x}, p, u, t) \mathrm{d}t \right], \\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, p, u(t), t), \quad \text{(prediction model)} \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad \text{(estimated state)} \\
& u(t) \in \mathcal{U}, \\
& p \in \mathcal{P}.
\end{aligned}
$$

Optimization Problem

$$\min_{p,u(t)} J = \min_{p,u(t)} \left[s(\boldsymbol{x}(t_f), p) + \int_{t_0}^{t_f} h(\boldsymbol{x}, p, u, t) \mathrm{d}t \right],$$

$$\begin{aligned} \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, p, u(t), t), \quad \text{(prediction model)} \\ & \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad \text{(estimated state)} \\ & u(t) \in \mathcal{U}, \\ & p \in \mathcal{P}. \end{aligned}$$

- real-time solution with conjugate gradient method

Optimization Problem

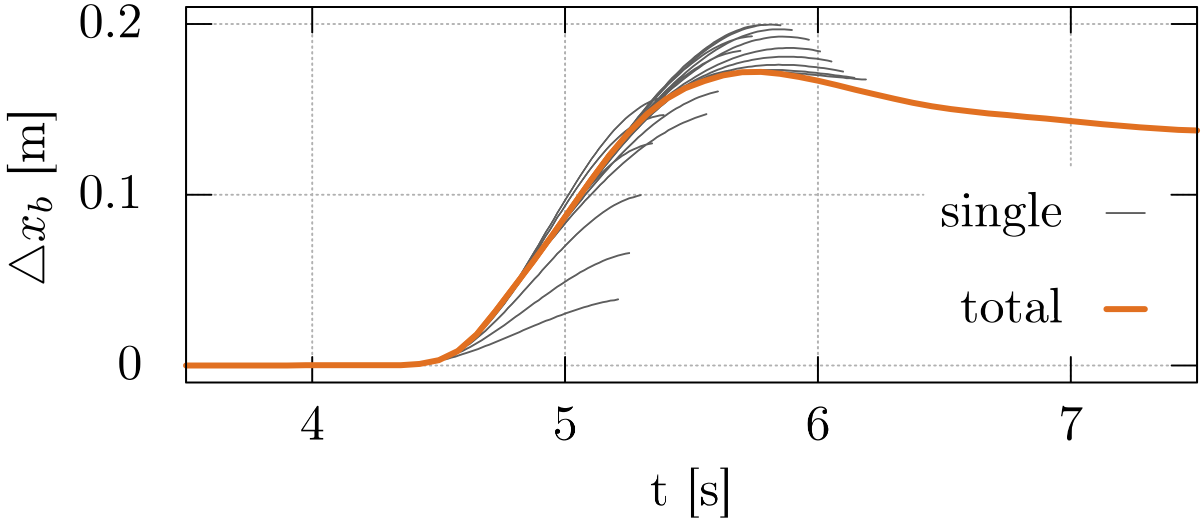
$$\min_{p,u(t)} J = \min_{p,u(t)} \left[s(\boldsymbol{x}(t_f), p) + \int_{t_0}^{t_f} h(\boldsymbol{x}, p, u, t)\mathrm{d}t \right],$$

$$\begin{aligned} \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, p, u(t), t), \quad \text{(prediction model)} \\ & \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad \text{(estimated state)} \\ & u(t) \in \mathcal{U}, \\ & p \in \mathcal{P}. \end{aligned}$$

- real-time solution with conjugate gradient method
- receeding horizon with 50 Hz
- 3 iterations take max. 3.5 ms

Results

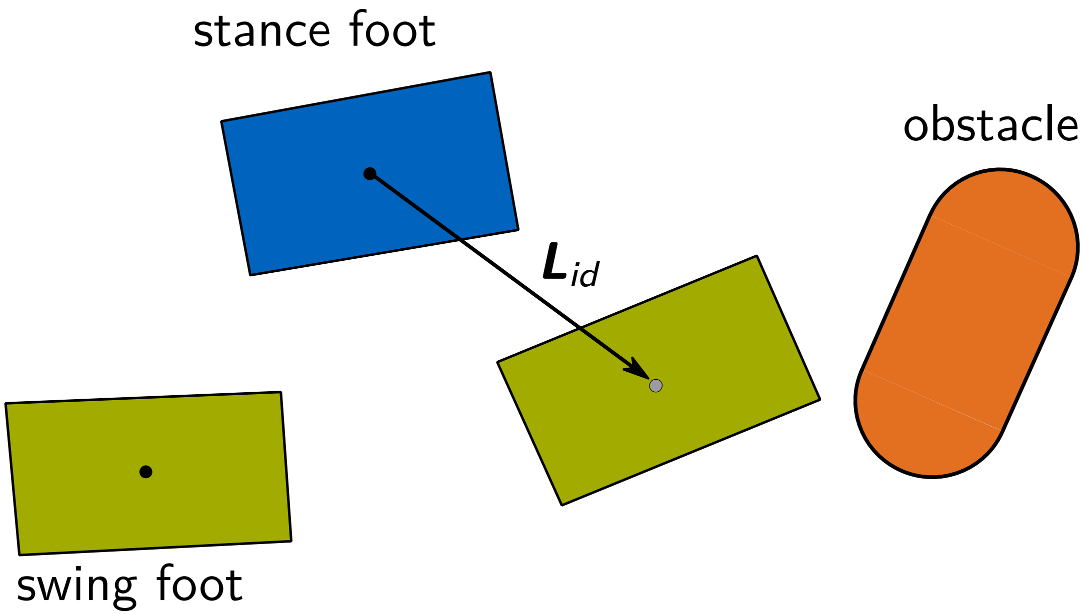
receeding horizon for CoG modification:



Combination with collision avoidance

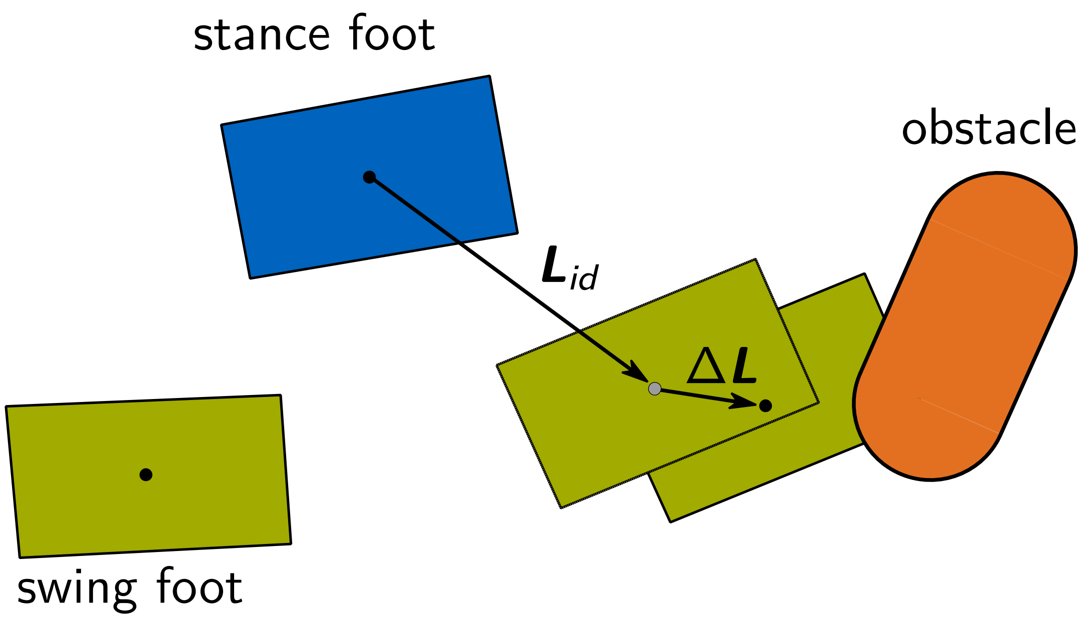
Vision + planning + stabilization?

Problem Description



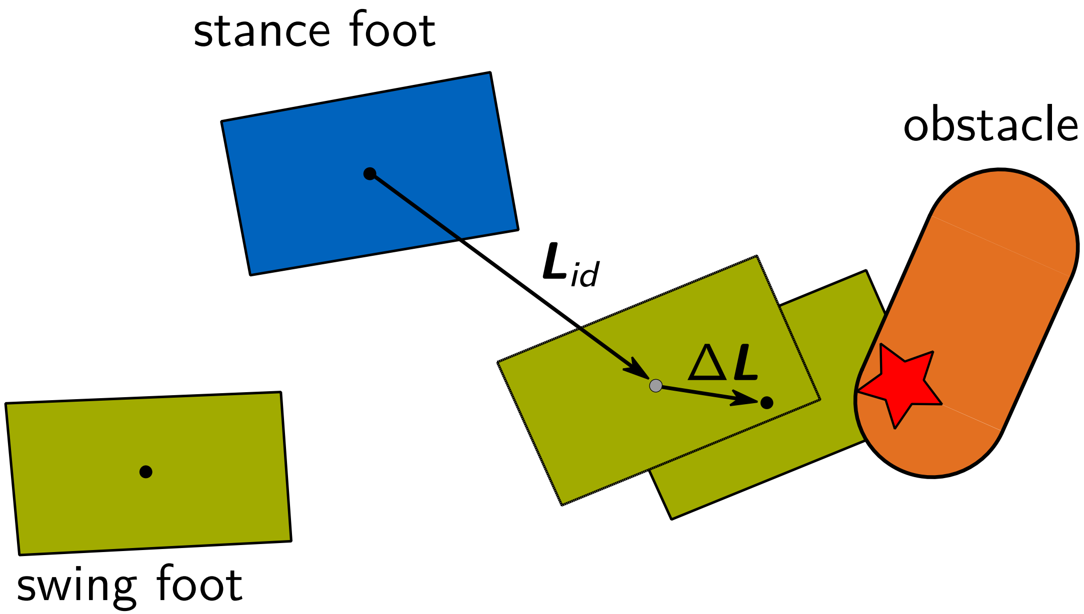
- $\boldsymbol{L}_{id}$: ideal step

Problem Description



- $\boldsymbol{L}_{id}$: ideal step
- $\Delta \boldsymbol{L}$: adaptation (disturbance)

Problem Description



- $\boldsymbol{L}_{id}$: ideal step
- $\Delta \boldsymbol{L}$: adaptation (disturbance)
- $\boldsymbol{L}_{id} + \Delta \boldsymbol{L}$: collision!

Strategy

- preserve hierarchy
- collision avoidance $>$ stabilization

Strategy

- preserve hierarchy
- collision avoidance $>$ stabilization
- foot position in optimization:

$$p \in \mathcal{P}$$

Optimization (again)

$$\begin{aligned} \min_{u,p} \, & J(u,p) \\ \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, u, p) \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \\ u &\in \mathcal{U} \\ & \boxed{p \in \mathcal{P}} \end{aligned}$$

Strategy

- preserve hierarchy
- collision avoidance $>$ stabilization
- foot position in optimization:

$$p \in \mathcal{P}$$

- question:

$$\mathcal{P} = ??$$

Optimization (again)

$$\begin{aligned} \min_{u,p} \; & J(u,p) \\ \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, u, p) \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \\ u &\in \mathcal{U} \\ & \boxed{p \in \mathcal{P}} \end{aligned}$$

Strategy

- preserve hierarchy
- collision avoidance $>$ stabilization
- foot position in optimization:

$$\boldsymbol{p} \in \mathcal{P}$$

- question:

$$\mathcal{P} = ??$$

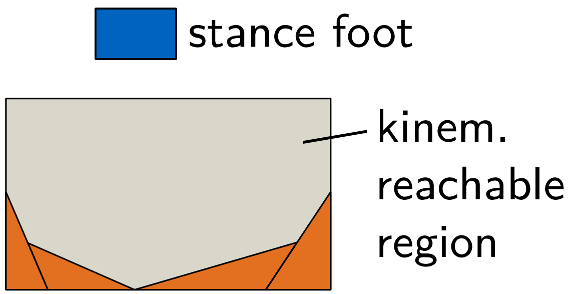
Optimization (again)

$$\begin{aligned} \min_{u,p} \; & J(u,p) \\ \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, u, p) \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \\ u &\in \mathcal{U} \\ & \boxed{p \in \mathcal{P}} \end{aligned}$$

$\Rightarrow$ geometric constraints $\rightarrow$ inequality constraints

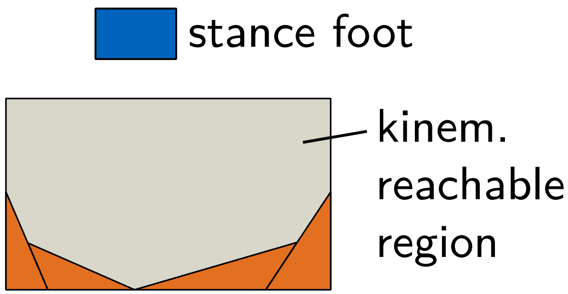
Geometric Constraints

kinematic limits

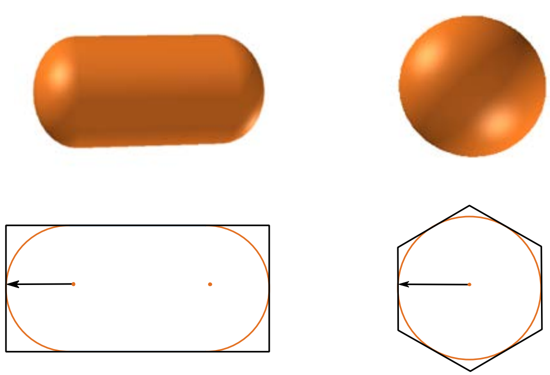


Geometric Constraints

kinematic limits

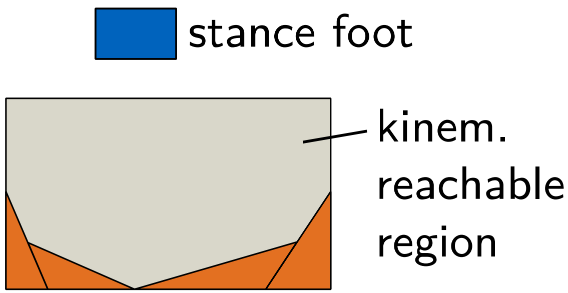


obstacles

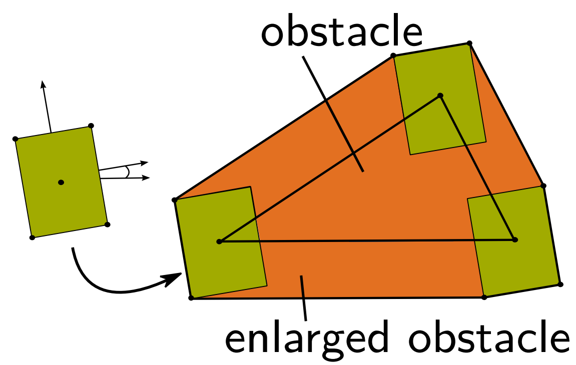


Geometric Constraints

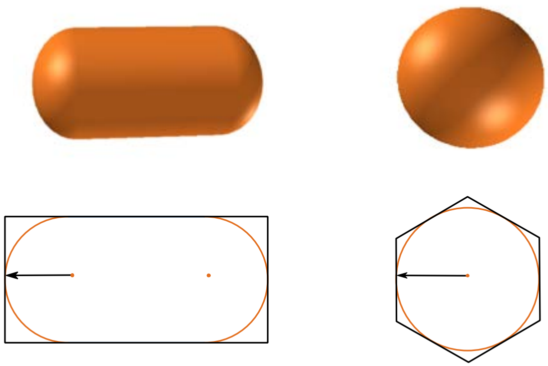
kinematic limits



foot geometry

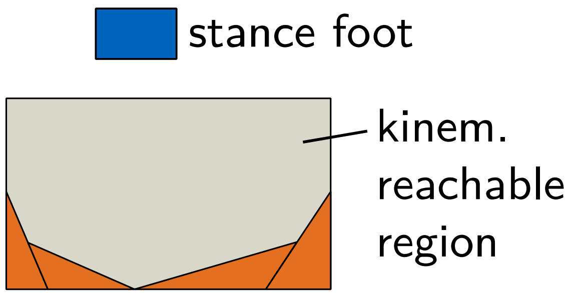


obstacles

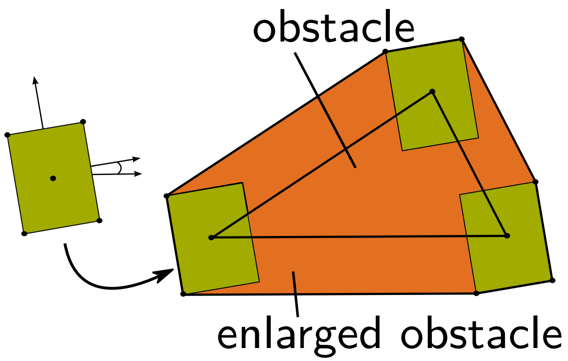


Geometric Constraints

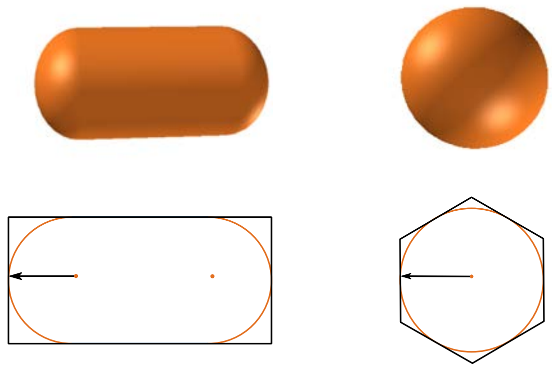
kinematic limits



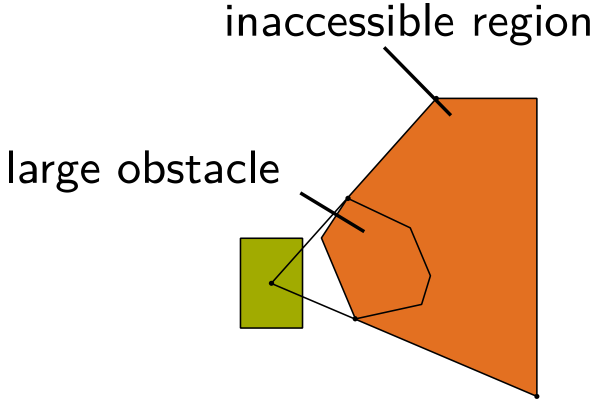
foot geometry



obstacles



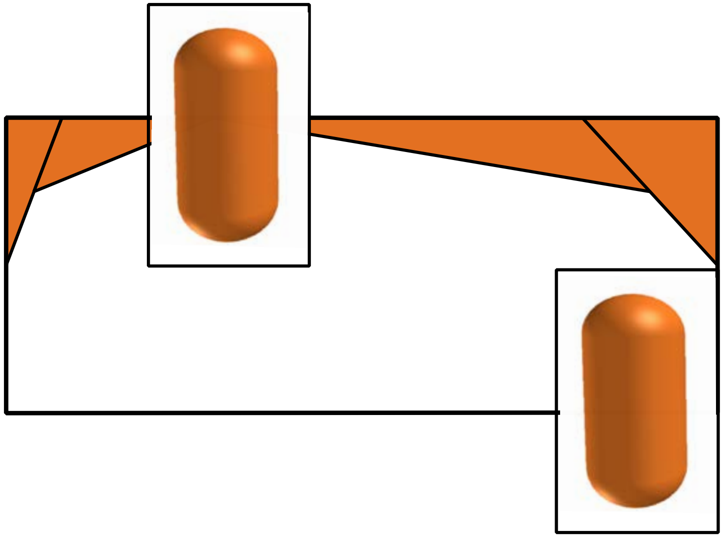
large obstacles



Finding Safe Footholds

- set of all inequality constraints
- invalid points $\boldsymbol{x}$:

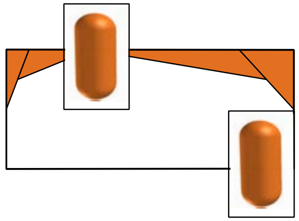
$$\mathcal{C}_{I,k} = \boldsymbol{A}_k \boldsymbol{x} > \boldsymbol{b}_k$$



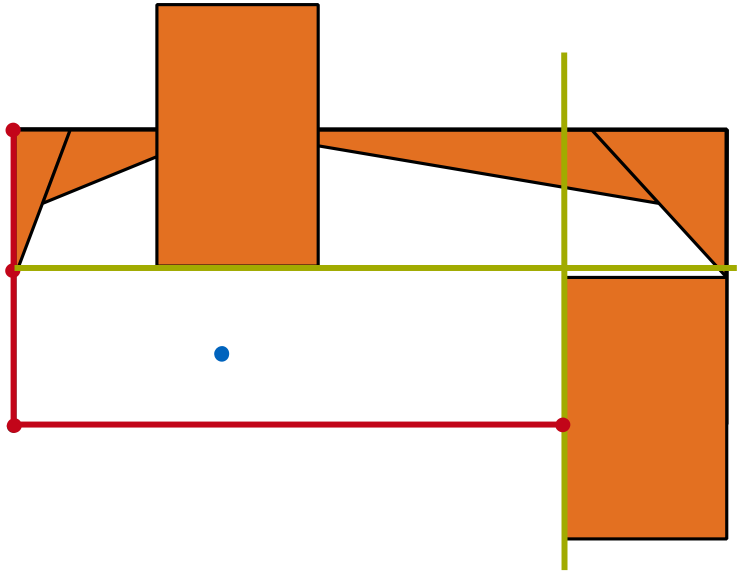
Finding Safe Footholds

- set of all inequality constraints
- invalid points $\boldsymbol{x}$:

$$\mathcal{C}_{I,k} = \boldsymbol{A}_k \boldsymbol{x} > \boldsymbol{b}_k$$



- search safe regions with different seed points



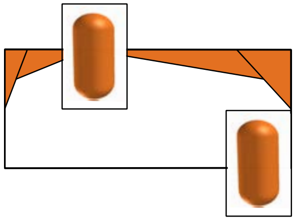
Finding Safe Footholds

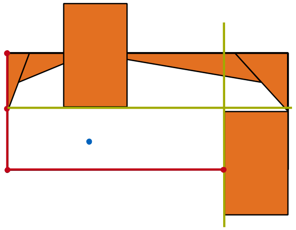
- set of all inequality constraints
- invalid points $\boldsymbol{x}$:

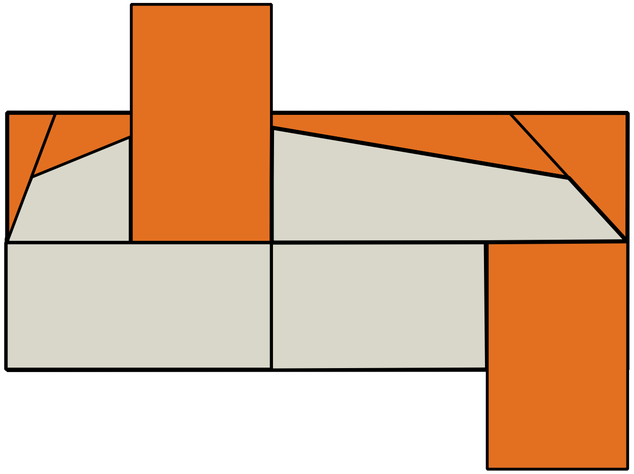
$$\mathcal{C}_{I,k} = \boldsymbol{A}_k \boldsymbol{x} > \boldsymbol{b}_k$$

- search safe regions with different seed points
- set of convex safe regions

$$\mathcal{P} = \{\mathcal{C}_{V,1}, \ldots \mathcal{C}_{V,n}\}$$







Finding Safe Footholds

- set of all inequality constraints
- invalid points $\boldsymbol{x}$:

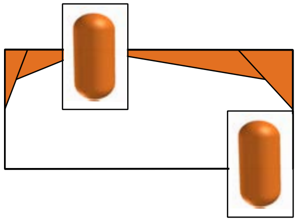
$$\mathcal{C}_{I,k} = \boldsymbol{A}_k \boldsymbol{x} > \boldsymbol{b}_k$$

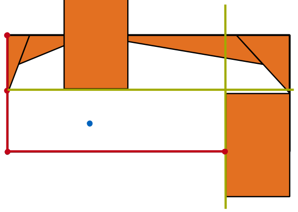
- search safe regions with different seed points
- set of convex safe regions

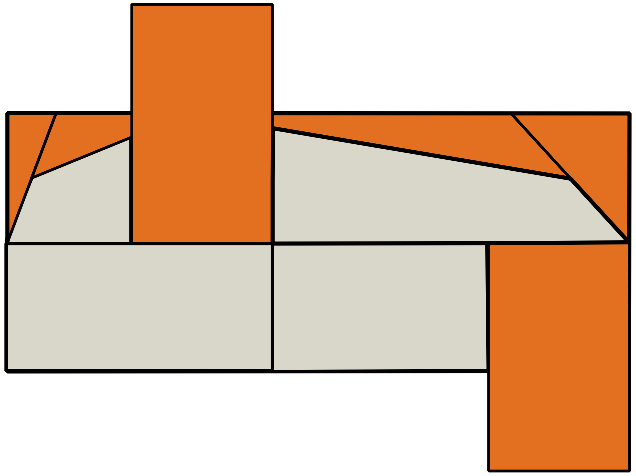
$$\mathcal{P} = \{\mathcal{C}_{V,1}, \ldots \mathcal{C}_{V,n}\}$$

- compute closest point

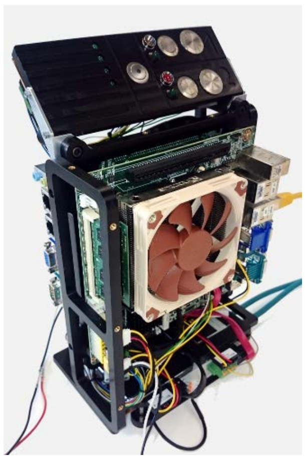
$$p \in \mathcal{P}$$





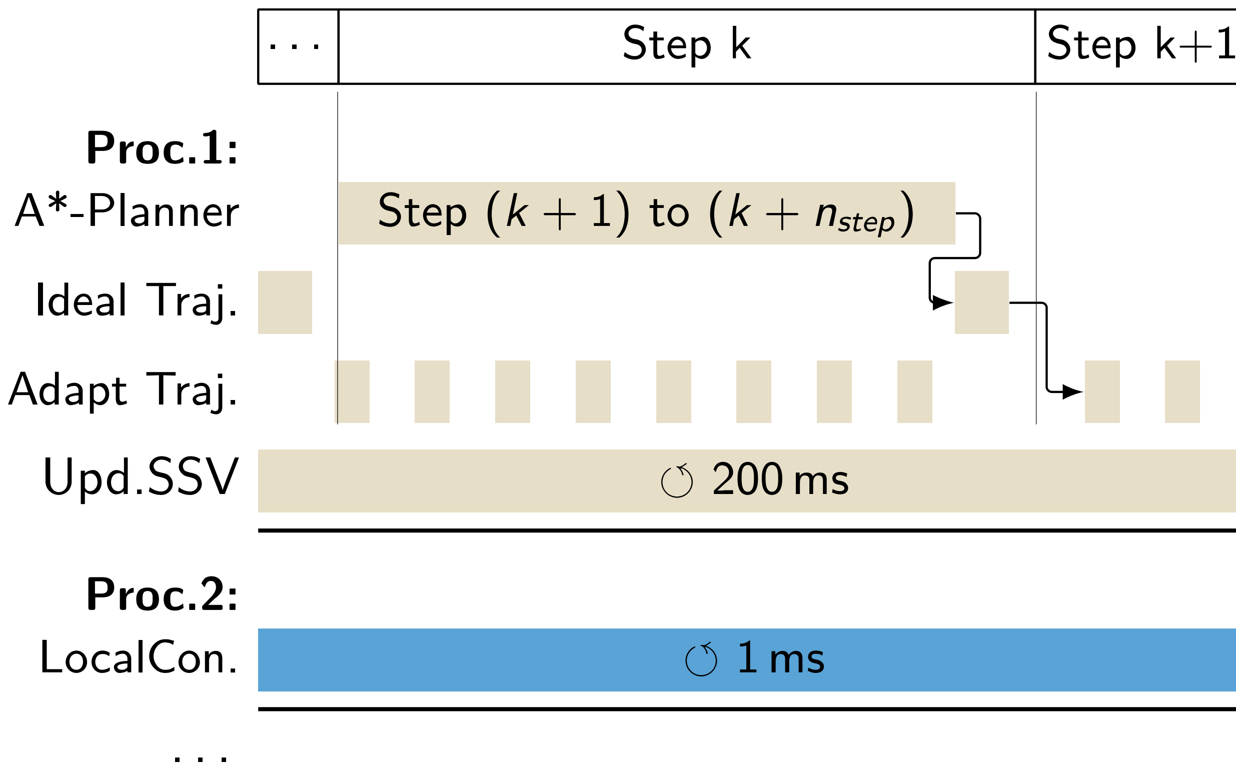


Real-time System

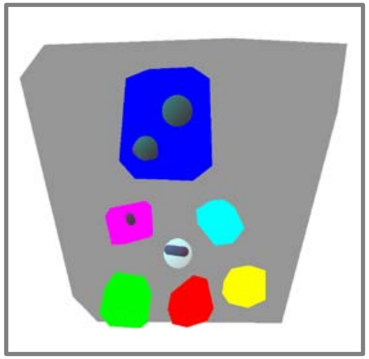


Onboard computer (2x Intel i7)

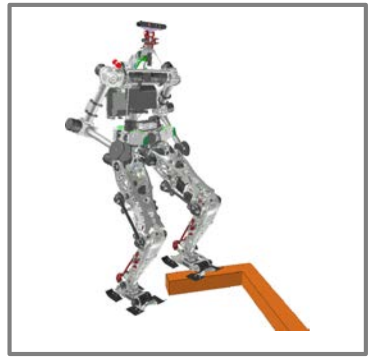
- 1x Linux (vision)
- 1x QNX (control)



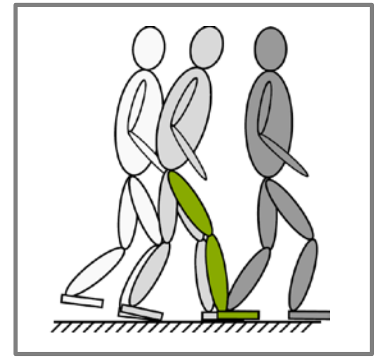
Summary



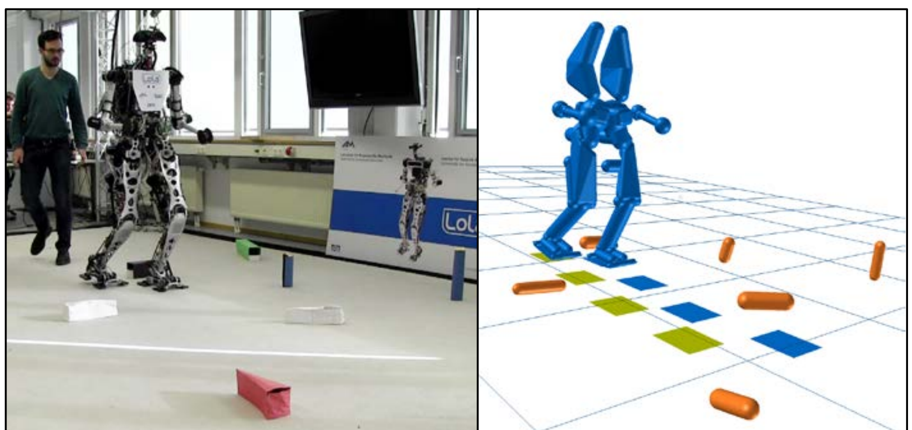
Environment Modeling



Motion Generation



Stabilization



Contents

Introduction and Motivation

Environment Modeling

Vision System

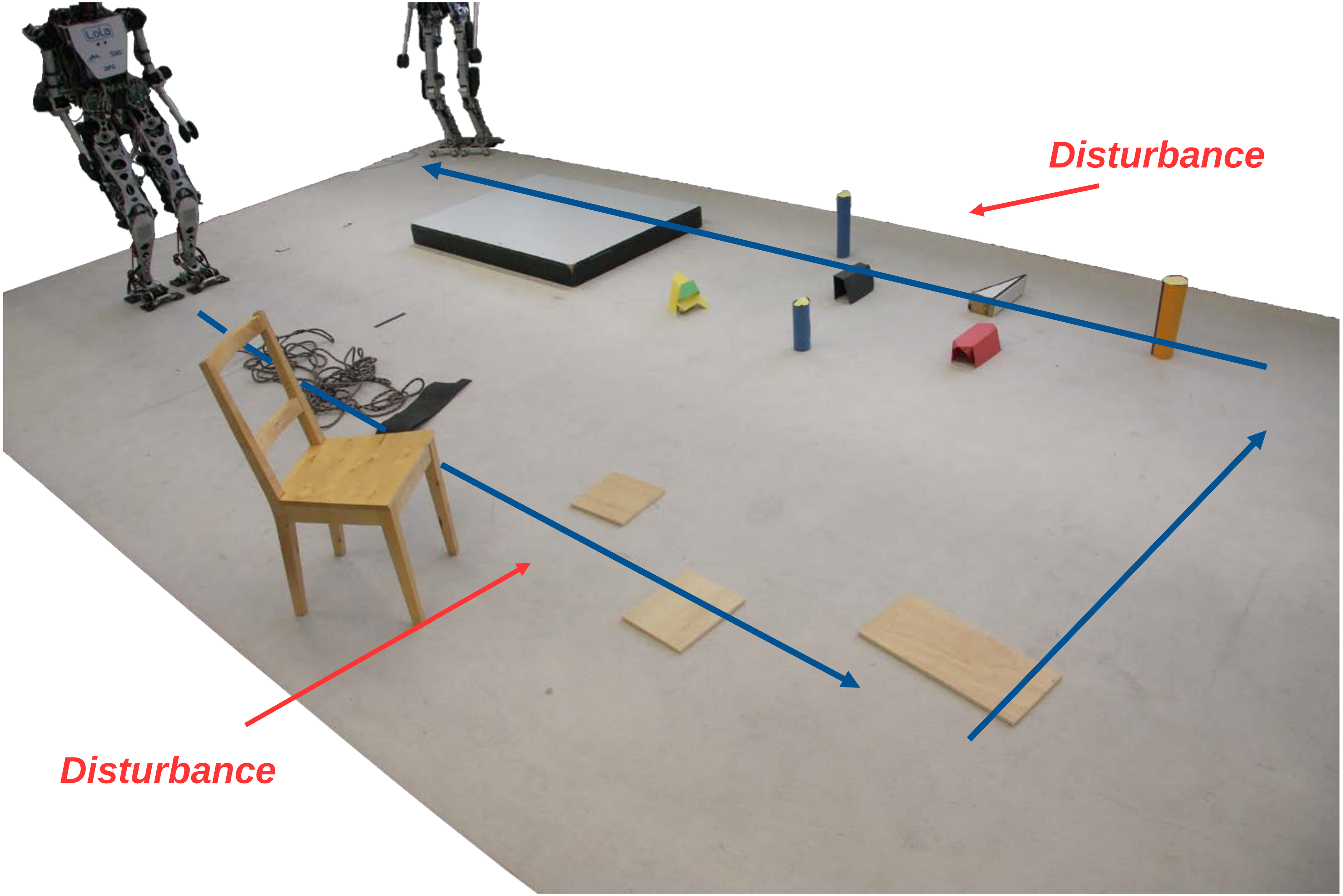
Navigation and Motion Generation

Stabilization

Results

Limitations

Perspectives


Disturbance
Disturbance

Results

Videos can be found at:

www.youtube.com/appliedmechanicstum

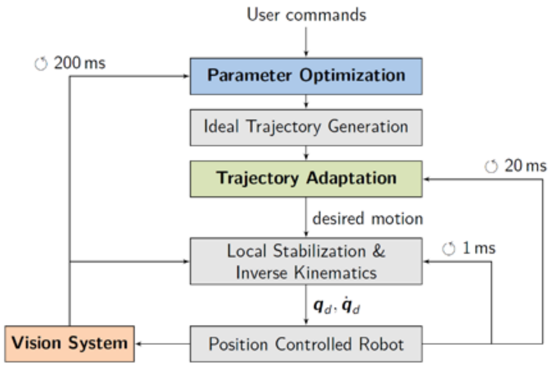
Lola – Playlist

https://www.youtube.com/playlist?list=PLVAvoOVYkpkMFNgz8cercTVvVryz1eh0a

Limitations

Hardware:

- Prototype state
- Cabling issues
- Camera accuracy
- Rare communication errors



Software:

- More software testing
- Advanced error handling
    - Methods fail
    - Other fails
- Robustness
- Tuning

Sources

P. E. Hart, N. J. Nilsson and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” in *IEEE Transactions on Systems Science and Cybernetics*, 1968

N. J. Nilsson, “Shakey the Robot,” in *SRI International: Technical Note 323*, 1984

H. P. Moravec and A. Elfes, “High Resolution Maps from Wide Angle Sonar,” in *IEEE International Conference on Robotics and Automation*, 1985

M. Herbert, C. Caillas, E. Krotkov, I. S. Kweon and T. Kanade, “Terrain Mapping for a Roving Planetary Explorer,” in *IEEE International Conference on Robotics and Automation*, 1989

S. Kagami, K. Nishiwaki, J. J. Kuffner, K. Okada, M. Inaba and H. Inoue, “Vision-based 2.5D terrain modeling for humanoid locomotion,” in *IEEE International Conference on Robotics and Automation*, 2003

J.-S. Gutmann, M. Fukuchi and M. Fujita, “A Floor and Obstacle Height Map for 3D Navigation of a Humanoid Robot,” in *IEEE International Conference on Robotics and Automation*, 2005

K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss and W. Burgard, “OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems,” in *IEEE ICRA workshop on best practice in 3D perception and modeling for mobile manipulation*, 2010

M. Schwienbacher, T. Buschmann, S. Lohmeier, V. Favot and H. Ulbrich, “Self-Collision Avoidance and Angular Momentum Compensation for a Biped Humanoid Robot,” in *IEEE International Conference on Robotics and Automation*, 2011

K. Nishiwaki, J. Chestnutt and S. Kagami, “Autonomous Navigation of a Humanoid Robot over Unknown Rough Terrain using a Laser Range Sensor,” in *The International Journal of Robotics Research*, 2012