Planning and Executing Humanoid Gaits in a World of Stairs

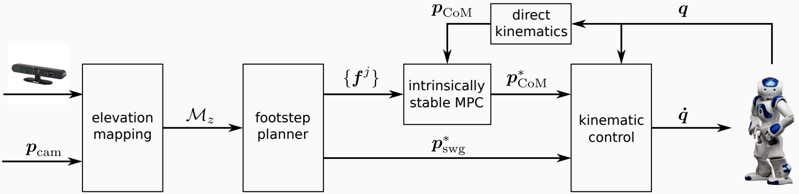
Candidate: Michele Cipriano
Thesis Advisor: Prof. Giuseppe Oriolo
January 21, 2020

Master Degree in Artificial Intelligence and Robotics
Department of Computer, Control and Management Engineering
Sapienza University of Rome

Introduction

- **goal:** humanoid robot locomotion in a *World of Stairs*
- **assumption:** humanoid is equipped with a **depth** sensor
- **assumption:** humanoid knows its location

Proposed Approach



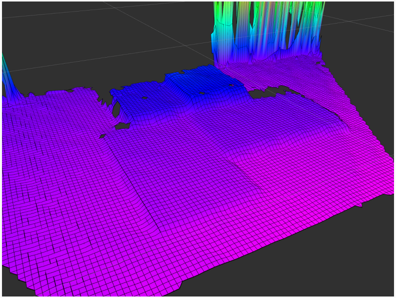
- **elevation map building**: autonomously build a map $\mathcal{M}_z$
- **footstep planner**: generates a footstep sequence $\{\boldsymbol{f}^j\}$ on $\mathcal{M}_z$ together with swing foot trajectories $\{\boldsymbol{p}^*_{\mathrm{swg}}\}$
- **variable-height IS-MPC**: computes a stable gait along the planned footsteps $\{\boldsymbol{f}^j\}$

Elevation Map Building: Framework

`elevation_mapping` [Fankhauser et al., 2018]

- robot-centric grid-based map: $\mathcal{M}_z$
- height estimate $\mathcal{N}(\hat{h}_i, \sigma^2_{h_i})$ for each cell i
- **Kalman filter** given new height and motion measurements
- **map fusion**: $(\hat{h}_i, h_{i,\mathrm{min}}, h_{i,\mathrm{max}})$ such that $h_i \in [h_{i,\mathrm{min}}, h_{i,\mathrm{max}}]$ with 95% confidence
- **dynamic environments** using visibility check based on ray tracing

- ASUS Xtion Pro (**depth** sensor)
- working range: 0.5–3.5 m



Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\max}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free
- **RRT-based** planner iteration
  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow Nearest(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow Rand(U)$
  4. **if** $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ **if** $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_\mathrm{f}^j - z_\mathrm{f}^{j-1}| \leq \Delta z_\mathrm{max}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_\mathrm{swg}^j$ is collision free
- RRT-based planner iteration
  1. $\boldsymbol{p}_\mathrm{rand} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_\mathrm{near} \leftarrow Nearest(\boldsymbol{p}_\mathrm{rand}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_\mathrm{cand} \leftarrow Rand(U)$
  4. **if** $\boldsymbol{f}_\mathrm{cand}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_\mathrm{swg}^\mathrm{cand} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_\mathrm{new}, v_\mathrm{near})$ **if** $\boldsymbol{p}_\mathrm{swg}^\mathrm{cand}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\mathsf{max}}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free
- RRT-based planner iteration
  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow Nearest(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow Rand(U)$
  4. **if** $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad\mathcal{T}.\mathrm{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ if $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\mathsf{max}}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free
- RRT-based planner iteration
  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow Nearest(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow Rand(U)$
  4. **if** $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ **if** $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_\mathrm{f}^j - z_\mathrm{f}^{j-1}| \leq \Delta z_{\max}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_\mathrm{swg}^j$ is collision free
- RRT-based planner iteration
  1. $\boldsymbol{p}_\mathrm{rand} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_\mathrm{near} \leftarrow Nearest(\boldsymbol{p}_\mathrm{rand}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_\mathrm{cand} \leftarrow Rand(U)$
  4. if $\boldsymbol{f}_\mathrm{cand}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_\mathrm{swg}^\mathrm{cand} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_\mathrm{new}, v_\mathrm{near})$ if $\boldsymbol{p}_\mathrm{swg}^\mathrm{cand}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\mathsf{max}}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free
- RRT-based planner iteration
  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow \mathit{Rand}(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow \mathit{Nearest}(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow \mathit{Rand}(U)$
  4. **if** $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ **if** $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements
  - R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\max}$
  - R2 footstep is fully in contact with the ground
  - R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free
- **RRT-based** planner iteration
  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow Nearest(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow Rand(U)$
  4. if $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ if $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3
- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements

  R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\mathsf{max}}$

  R2 footstep is fully in contact with the ground

  R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free

- RRT-based planner iteration

  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow Nearest(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow Rand(U)$
  4. **if** $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathsf{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathsf{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ if $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3

- U is the set of footstep primitives

Footstep Planner: Algorithm

- requirements

  R1 maximum footsteps height variation $|z_{\mathrm{f}}^{j} - z_{\mathrm{f}}^{j-1}| \leq \Delta z_{\mathrm{max}}$

  R2 footstep is fully in contact with the ground

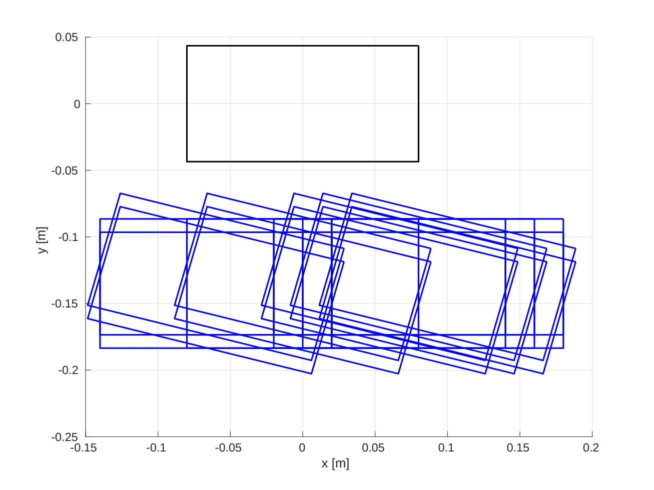
  R3 swing foot trajectory $\boldsymbol{p}_{\mathrm{swg}}^{j}$ is collision free

- **RRT-based** planner iteration

  1. $\boldsymbol{p}_{\mathrm{rand}} \leftarrow Rand(\mathcal{M}_z)$
  2. $v_{\mathrm{near}} \leftarrow Nearest(\boldsymbol{p}_{\mathrm{rand}}, \gamma, \mathcal{T})$
  3. $\boldsymbol{f}_{\mathrm{cand}} \leftarrow Rand(U)$
  4. **if** $\boldsymbol{f}_{\mathrm{cand}}$ feasible wrt R1-R2 **then**
  5. $\quad \boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}} \leftarrow \mathrm{BuildTrajectory}(\cdot)$
  6. $\quad \mathcal{T}.\mathrm{add}(v_{\mathrm{new}}, v_{\mathrm{near}})$ if $\boldsymbol{p}_{\mathrm{swg}}^{\mathrm{cand}}$ satisfies R3

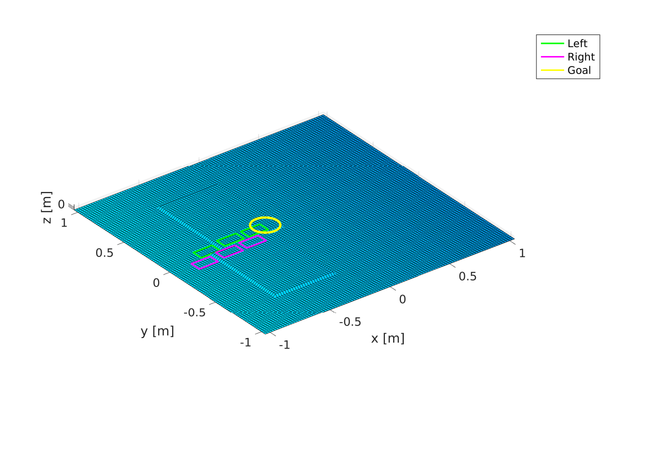
- U is the set of footstep primitives

Footstep Planner: Primitives Catalogue U

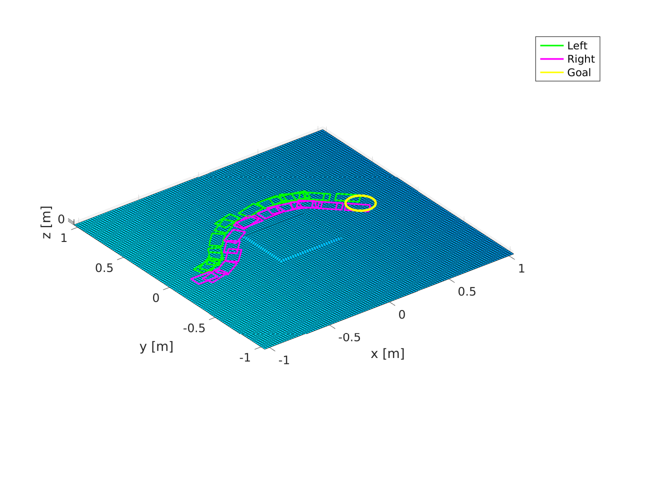


Footstep Planner: Results (stair climbing)

tree size: 10 – solution size: 6 – runtime: 0.8 ms



tree size: 488 – solution size: 31 – runtime: 70 ms



Variable Height IS-MPC: 3D Motion Model

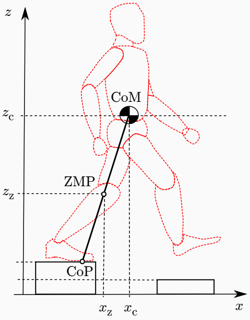
- **LIP** model not suitable for gait generation over **uneven terrain** due to constant height assumption
- linearity can be maintained by constraining vertical motion such that

$$\frac{\ddot{z}_c + g}{z_c - z_z} = \omega^2$$

- CoM dynamics become

$$\ddot{x}_c = \omega^2 (x_c - x_z)$$
$$\ddot{y}_c = \omega^2 (y_c - y_z)$$
$$\ddot{z}_c = \omega^2 (z_c - z_z) - g$$

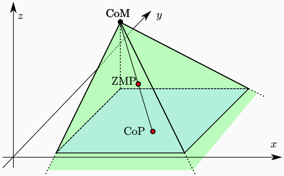
- x_c, y_c, z_c: CoM coordinates
- x_z, y_z, z_z: ZMP coordinates



Variable Height IS-MPC: MPC Formulation

- **constrain ZMP** into subregion of polyhedral cone (box)

$$R^T_{k+i} \begin{pmatrix} x_z^{k+i} - x_f^{k+i} \\ y_z^{k+i} - y_f^{k+i} \\ z_z^{k+i} - y_f^{k+i} \end{pmatrix} \leq \frac{1}{2} \begin{pmatrix} \tilde{d}_x^z \\ \tilde{d}_y^z \\ d_z^z \end{pmatrix}$$



- bound CoM wrt ZMP (**LIP stability**)

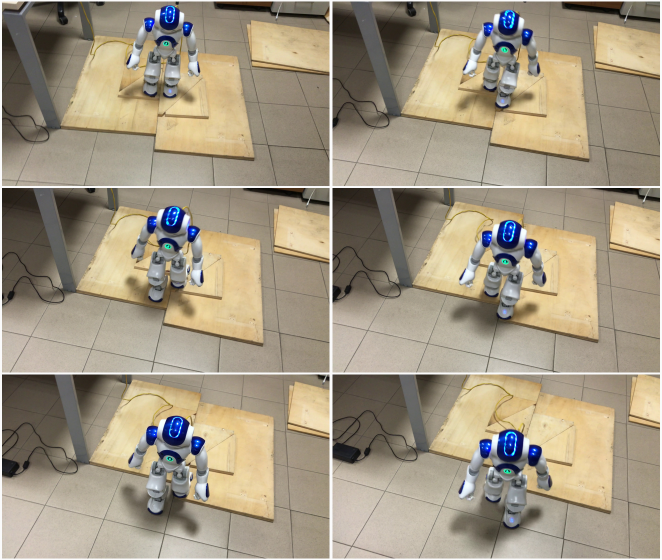
$$\frac{1}{\omega} \frac{1 - e^{-\delta\omega}}{1 - e^{-N\delta\omega}} \sum_{i=0}^{N-1} e^{-i\delta\omega} \dot{x}_z^{k+i} = x_c^k + \frac{\dot{x}_c^k}{\omega} - x_z^k$$

- solve **QP problem** using **MPC** scheme

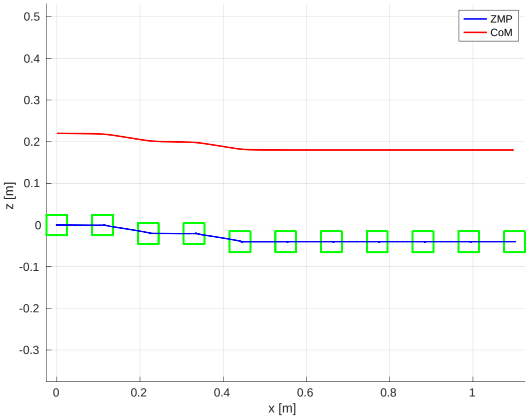
$$\min_{\dot{X}_z^k, \dot{Y}_z^k, \dot{Z}_z^k} \|\dot{X}_z^k\|_2^2 + \|\dot{Y}_z^k\|_2^2 + \|\dot{Z}_z^k\|_2^2 + \beta(\|\Delta X_f^{k+1}\|_2^2 + \|\Delta Y_f^{k+1}\|_2^2 + \|\Delta Z_f^{k+1}\|_2^2)$$

s.t. ZMP and stability constraints

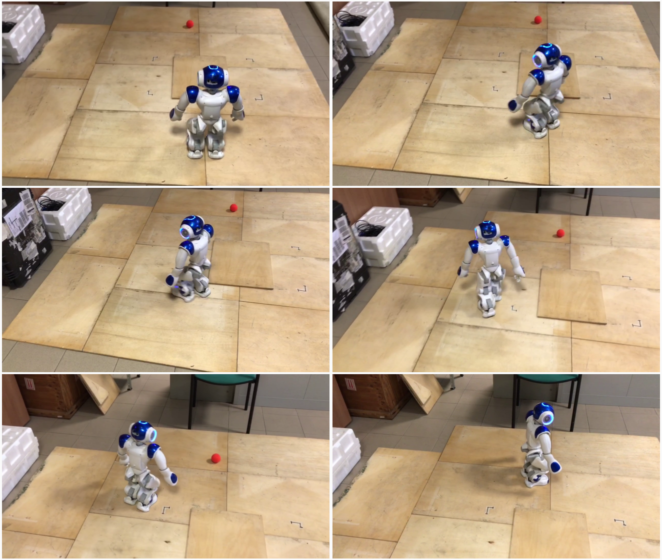
Experiment: Downstairs



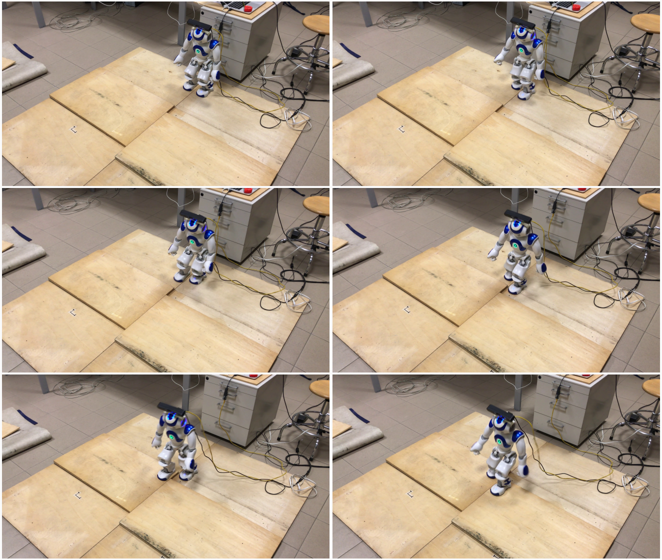
Experiment: Downstairs



Experiment: Obstacle Avoidance

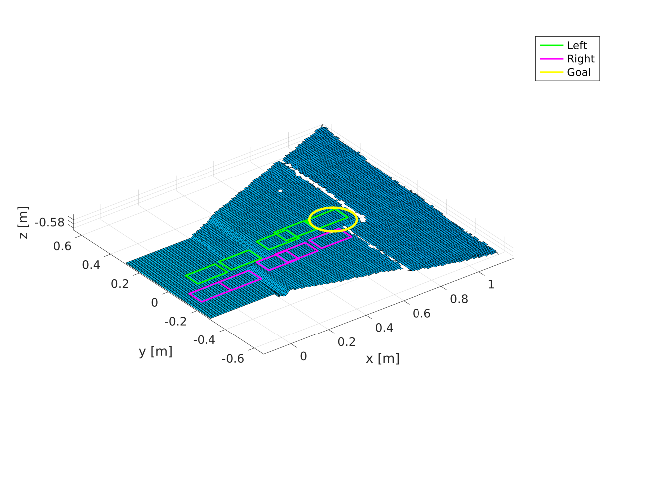


Experiment: Unknown Environment



Experiment: Unknown Environment

tree size: 454 – solution size: 10 – runtime: 331 ms



Video

Conclusions

- NAO is able to autonomously climb the stairs in an unknown *World of Stairs* environment
- future works
    - **localization** module and **continuous mapping**
    - **replanning** phase
    - dynamic and rough environments

Q&A

References

Fankhauser, P., Bloesch, M., and Hutter, M. (2018).
Probabilistic terrain mapping for mobile robots with uncertain localization.
IEEE Robotics and Automation Letters (RA-L), 3(4):3019–3026.

Ferrari, P., Scianca, N., Lanari, L., and Oriolo, G. (2019).
An integrated motion planner/controller for humanoid robots on uneven ground.
In *18th European Control Conference, ECC 2019, Naples, Italy, June 25-28, 2019*, pages 1598–1603.

Zamparelli, A., Scianca, N., Lanari, L., and Oriolo, G. (2018).
Humanoid Gait Generation on Uneven Ground using Intrinsically Stable MPC.
IFAC-PapersOnLine, 51:393–398.