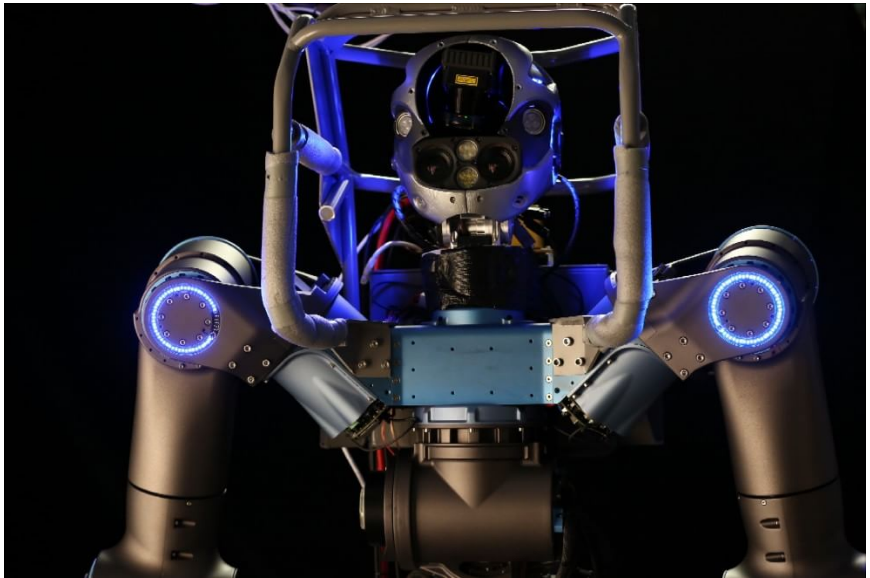
Hierarchical Planning for humanoid robots

Mirko Ferrati Lucia Pallottino



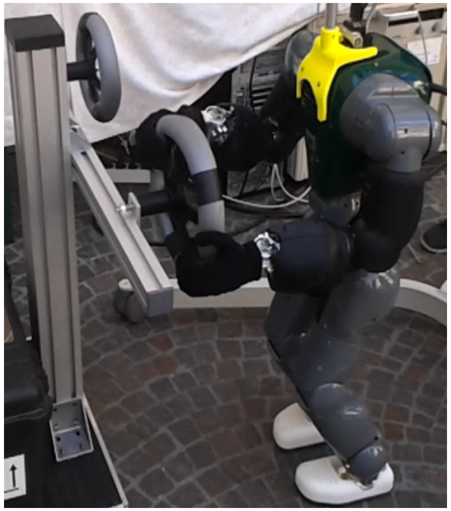
DARPA Robotics Challenge

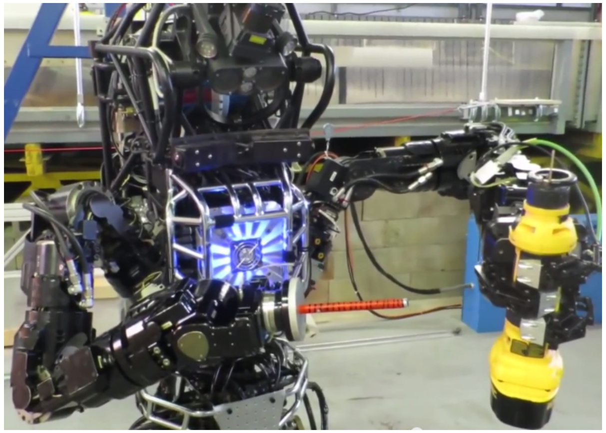
Recent disasters demonstrated that current robots barely assist humans

It is essential to work towards developing robots and control architectures with the capacity to be truly helpful

DARPA Robotics Challenge

- Climb stairs
- Traverse rough terrain
- Open doors
- Use human tools
- Rotate Valve
- …





Hierarchical Planning Structure

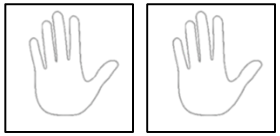
Task SPACE

↓

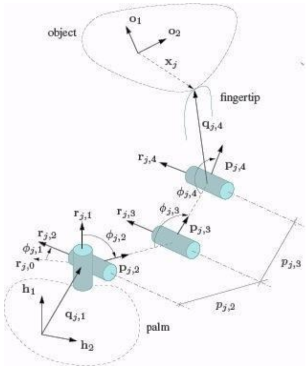
Cartesian SPACE

↓

Joint SPACE



x	*13.2*
y	*10.5*
z	*1.3*



Joint	*Degrees*
6	*5*

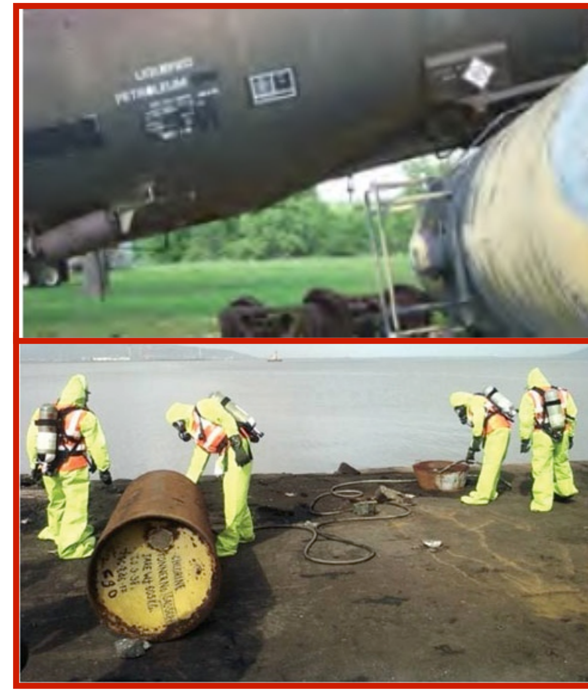
Planning and Control in disaster scenarios

Entering and inspecting an unstable factory building after an earthquake event

Enter the building

Cross an uneven terrain

Reach and grasp tools



Hazardous material inspection and retrieval after a transportation accident

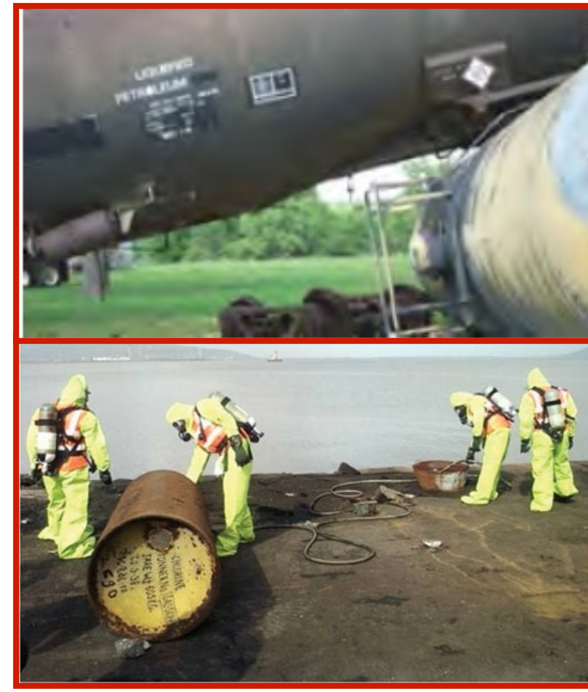
Task SPACE

Planning and Control in disaster scenarios

Entering and inspecting an unstable factory building after an earthquake event

Sequence of foot and hand positions for stairs climbing

Position of the hands on a pipe or tool



Hazardous material inspection and retrieval after a transportation accident

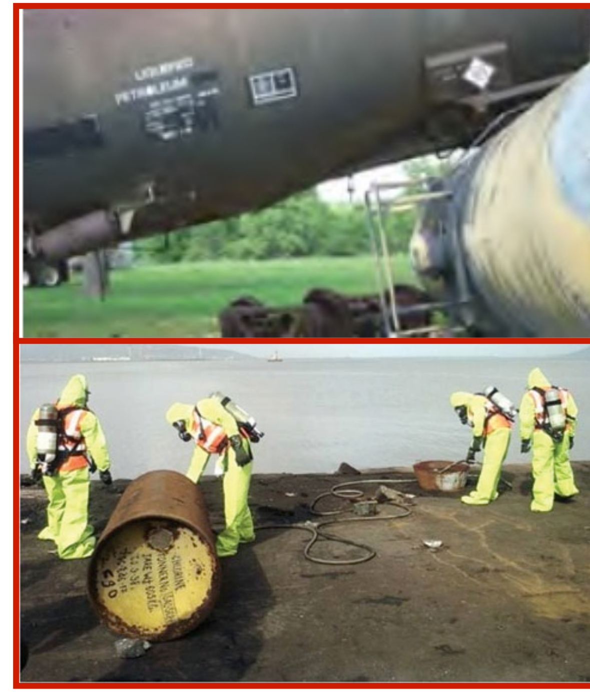
Cartesian SPACE

Planning and Control in disaster scenarios

Entering and inspecting an unstable factory building after an earthquake event

Perform step

Perform grasp



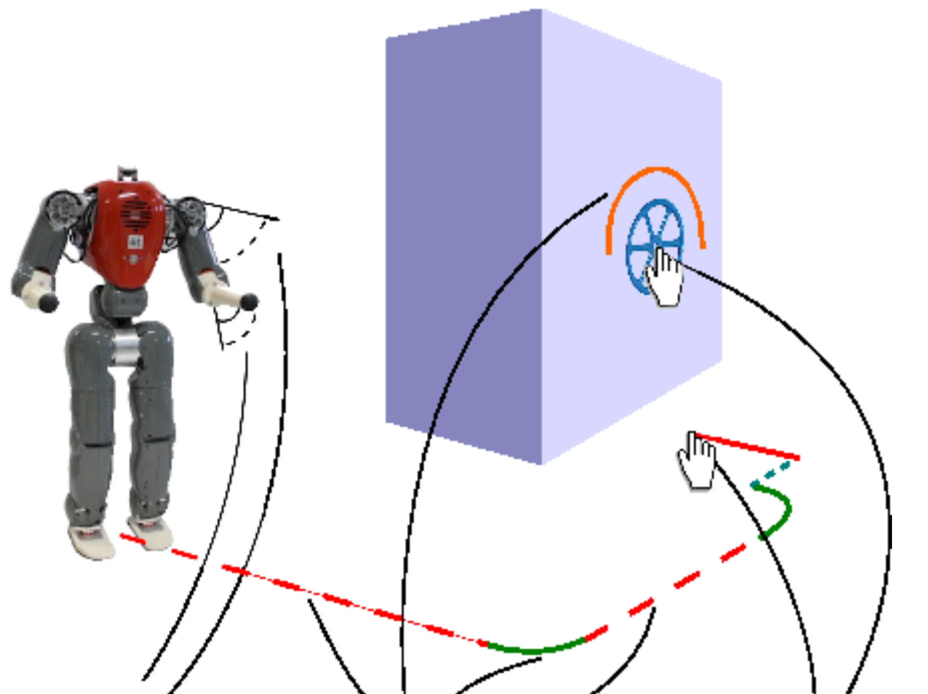
Hazardous material inspection and retrieval after a transportation accident

Joint SPACE

Hierarchical Planning

- Joints
- Footsteps
- Trajectory

Hierarchical Control Structure



Joint SPACE

Kinematic constraint planning

Cartesian SPACE

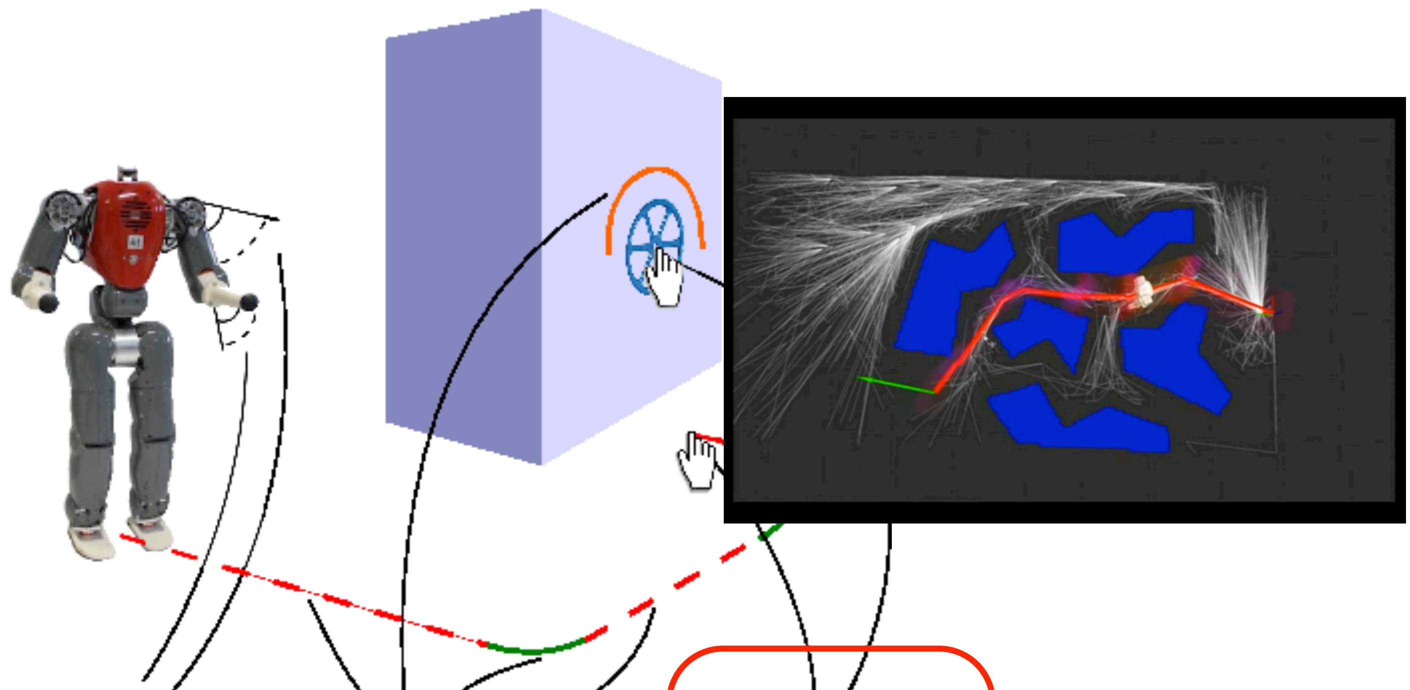
Vision based Footstep planner

Task SPACE

Gait based Planner

Increasing Autonomy

Hierarchical Control Structure



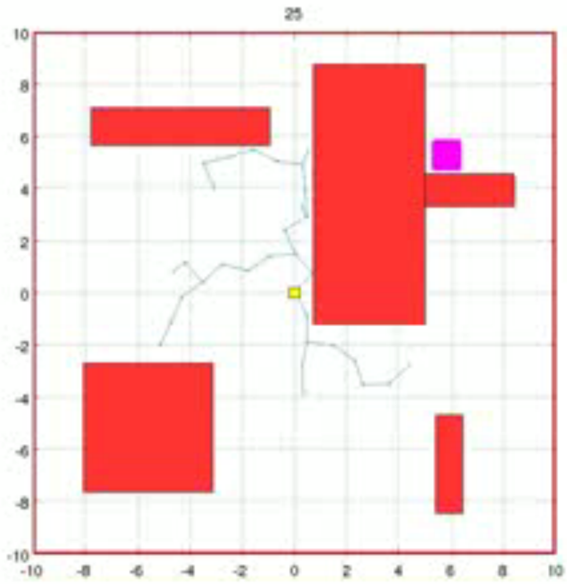
Joint SPACE	Cartesian SPACE	Task SPACE
Kinematic constraint planning	Vision based Footstep planner	Gait based Planner

T6.1

Increasing Autonomy

Idea: RRT*-like algorithms

RRT*- like planners: anytime optimal sample based motion planning (Frazzoli – Karaman)

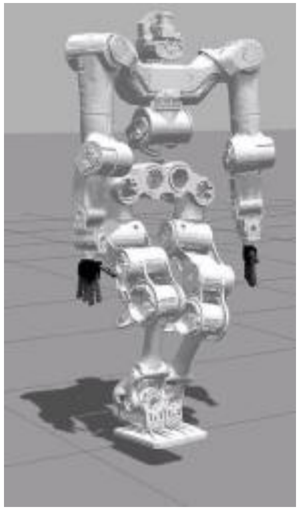


A preliminary solution provided very fast

Solution improved with time

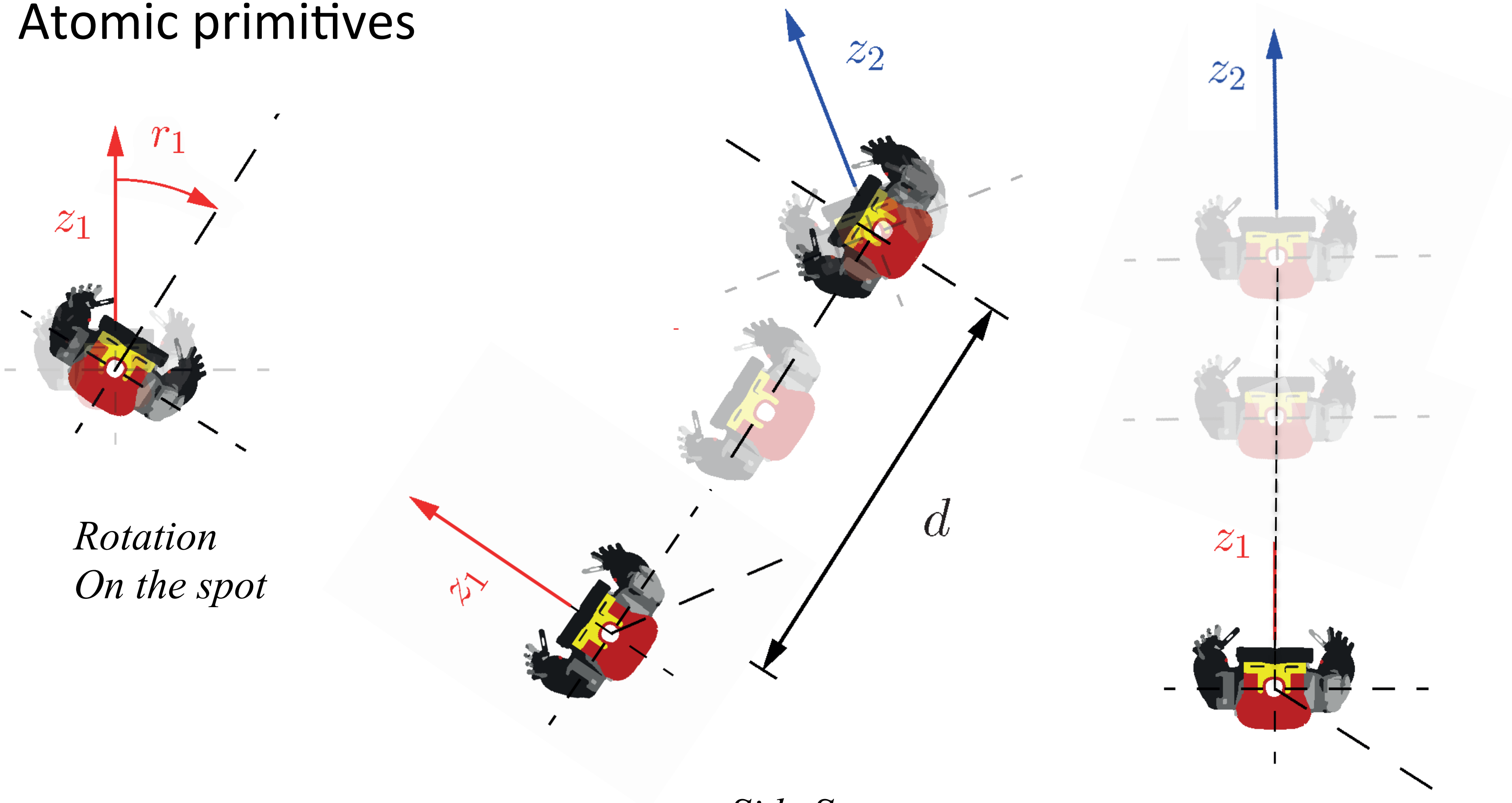
Optimal in probability

Sample: configuration
Arc: path
Cost



Atomic Controlled Motion Primitives

Atomic primitives



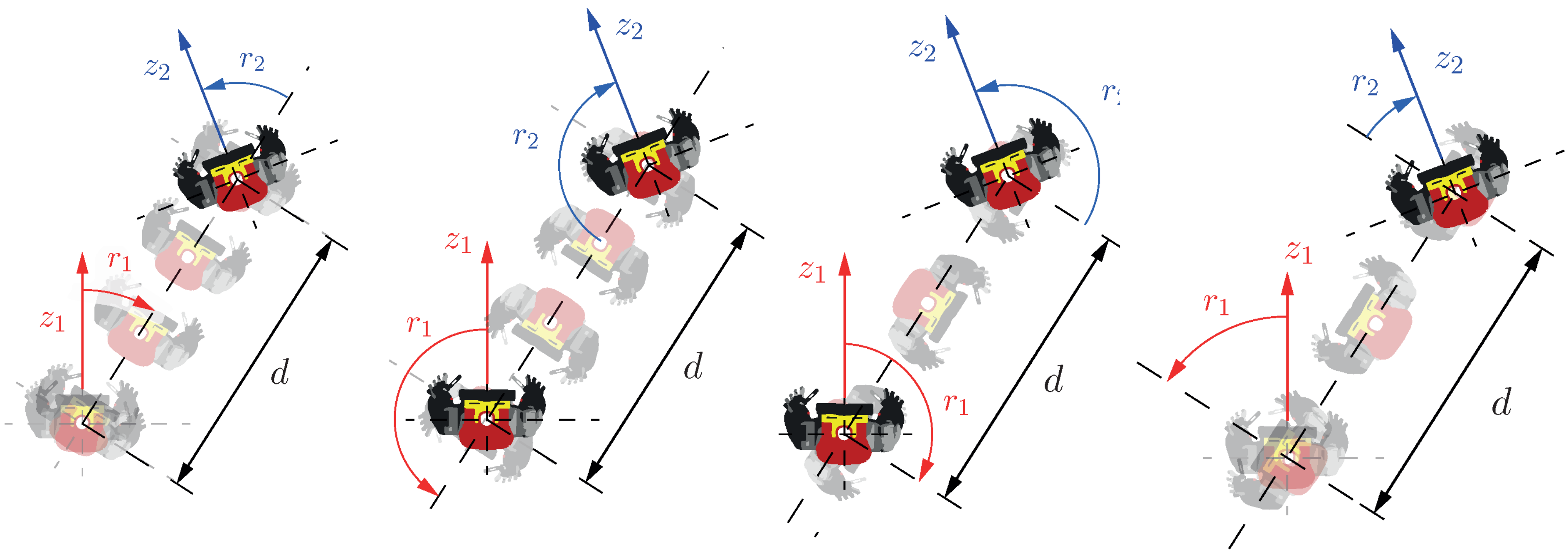
Rotation On the spot

Side Steps

Straight walking

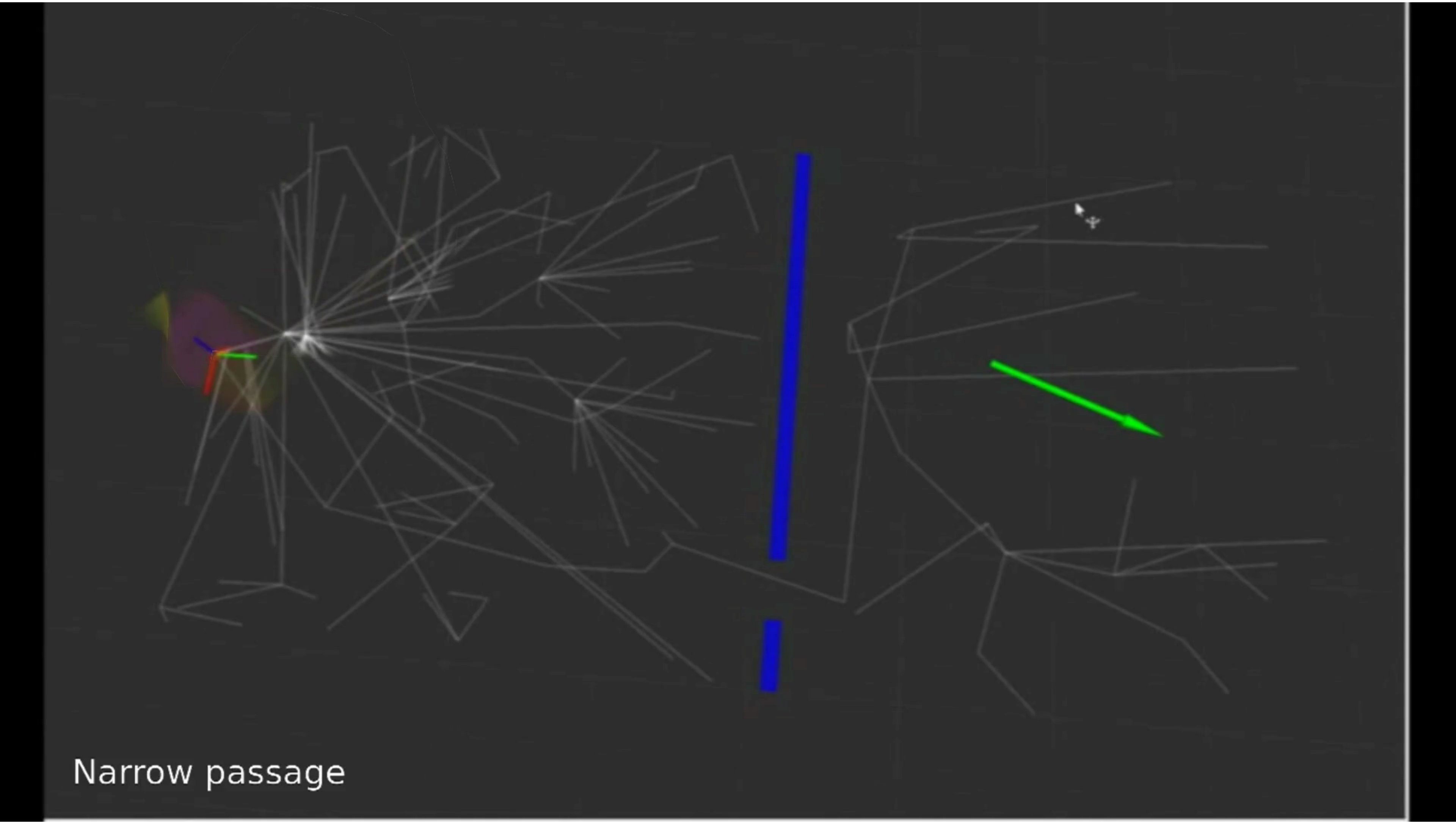
Complex Motion Primitives

Steering primitives as sequences of atomic ones

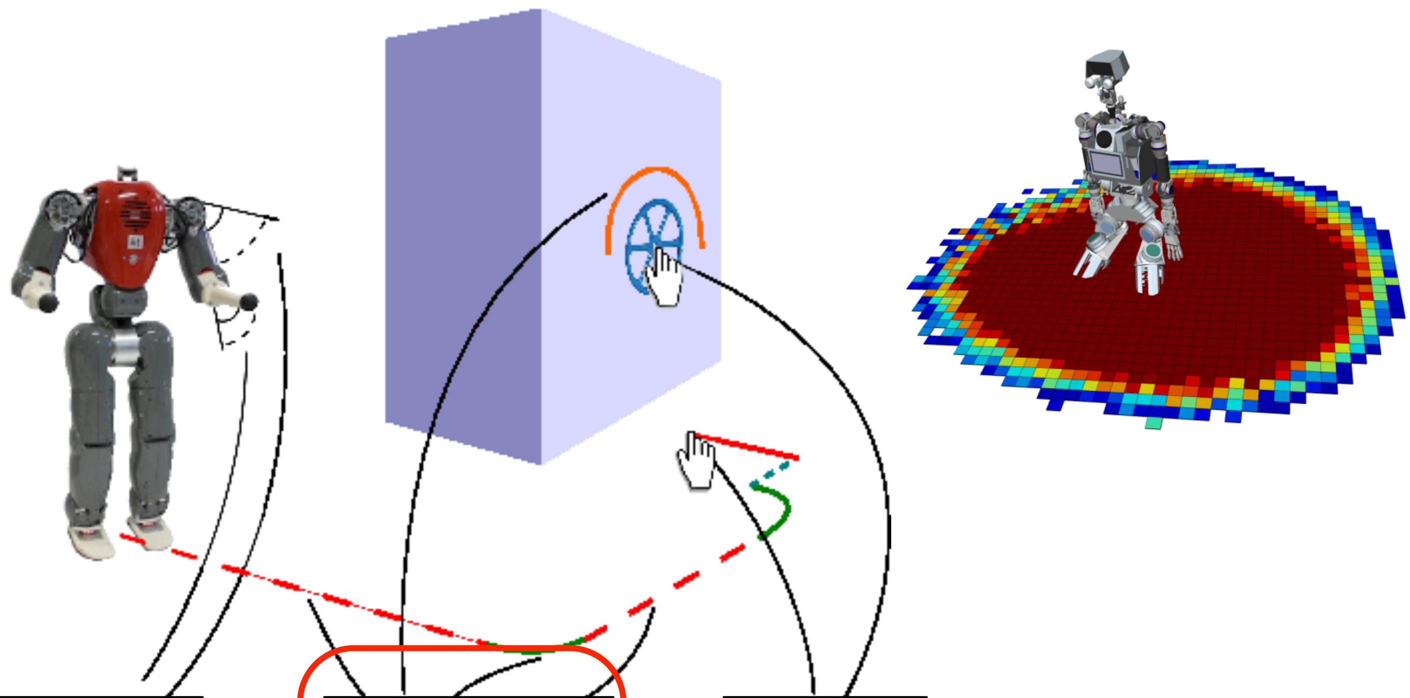


Steer₁	*Steer₂*	*Steer₃*	*Steer₄*
Turn, Fwd, Turn	Turn, Bwd, Turn	Turn, Left, Turn	Turn, Right, Turn

RRT* for Walk-Man Navigation



Hierarchical Control Structure

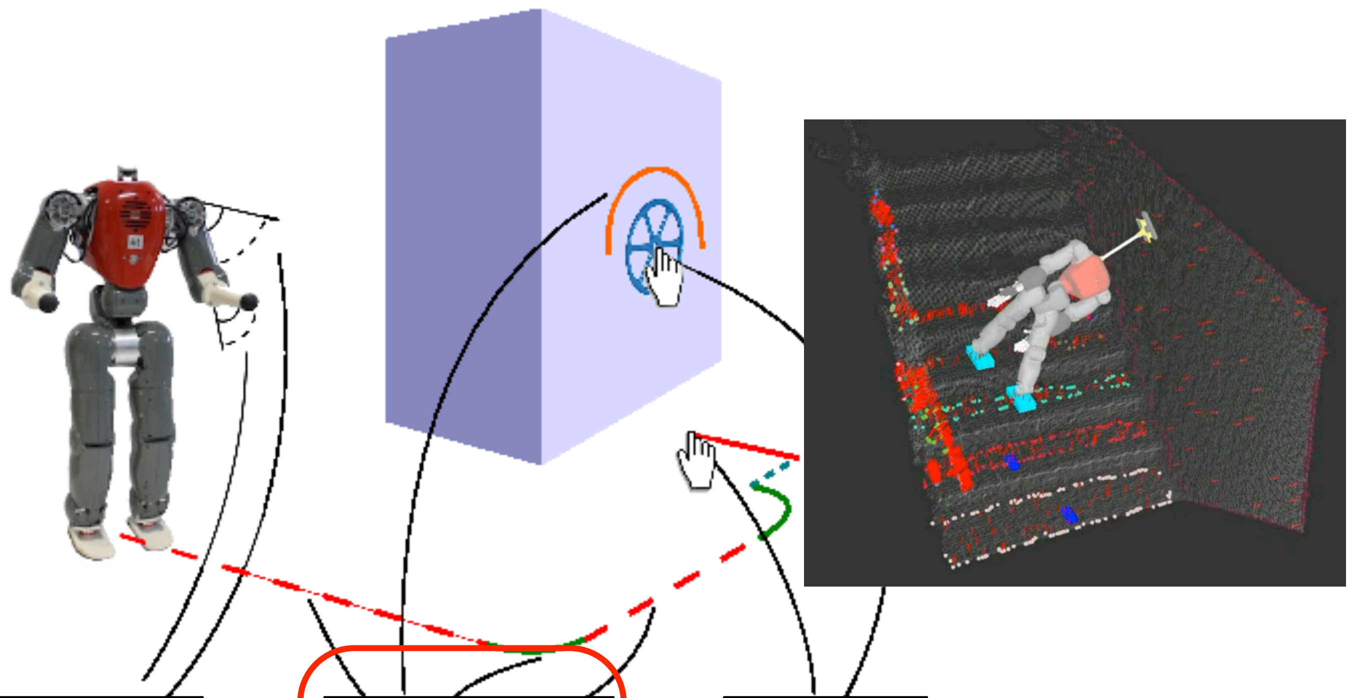


Joint SPACE	Cartesian SPACE	Task SPACE
Kinematic constraint planning	IK Solver	Gait based Planner

T6.1

Increasing Autonomy

Hierarchical Control Structure



Joint SPACE	Cartesian SPACE	Task SPACE
Kinematic constraint planning	Vision based Footstep planner	Gait based Planner

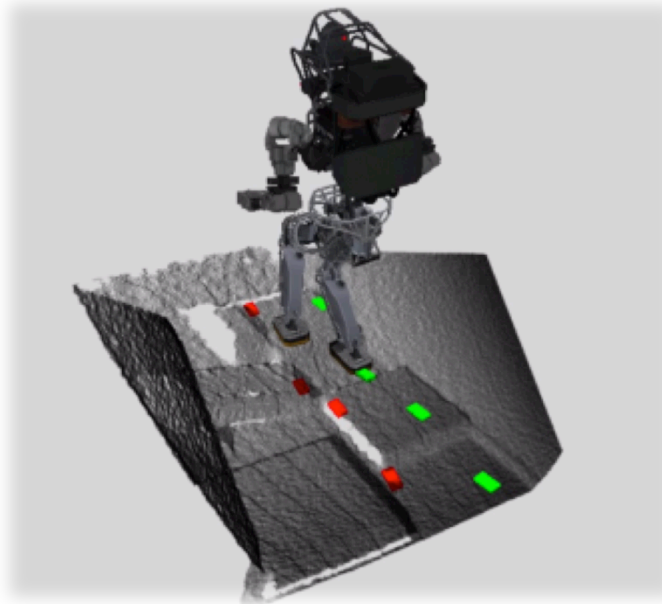
T6.1

Increasing Autonomy

Foot Planning for Humanoid Robots

- **Challenge:**
- Many DOFs, dynamical constraints, intrinsic instability.

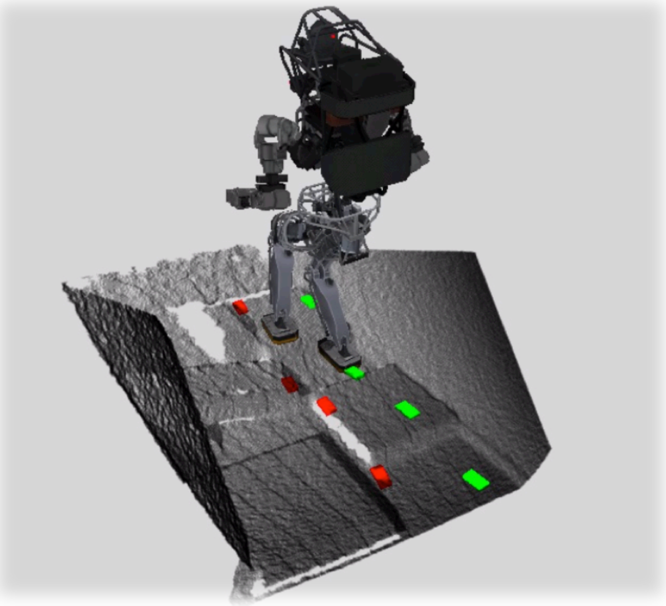
- **Approaches:**
- Division between planning and walking control.
  - Discrete search:
    - Fixed valid set of footsteps computed offline.
    - Search on a tree.
  - Limitations: Lack of generality.
  - Continuous search:
    - Infinite set of footsteps.
    - Sample based method
  - Limitations: Large computational time.
  - Other approaches:
    - QP, MPC, Inverse Dynamic.
    - Computation of steps and walking control.



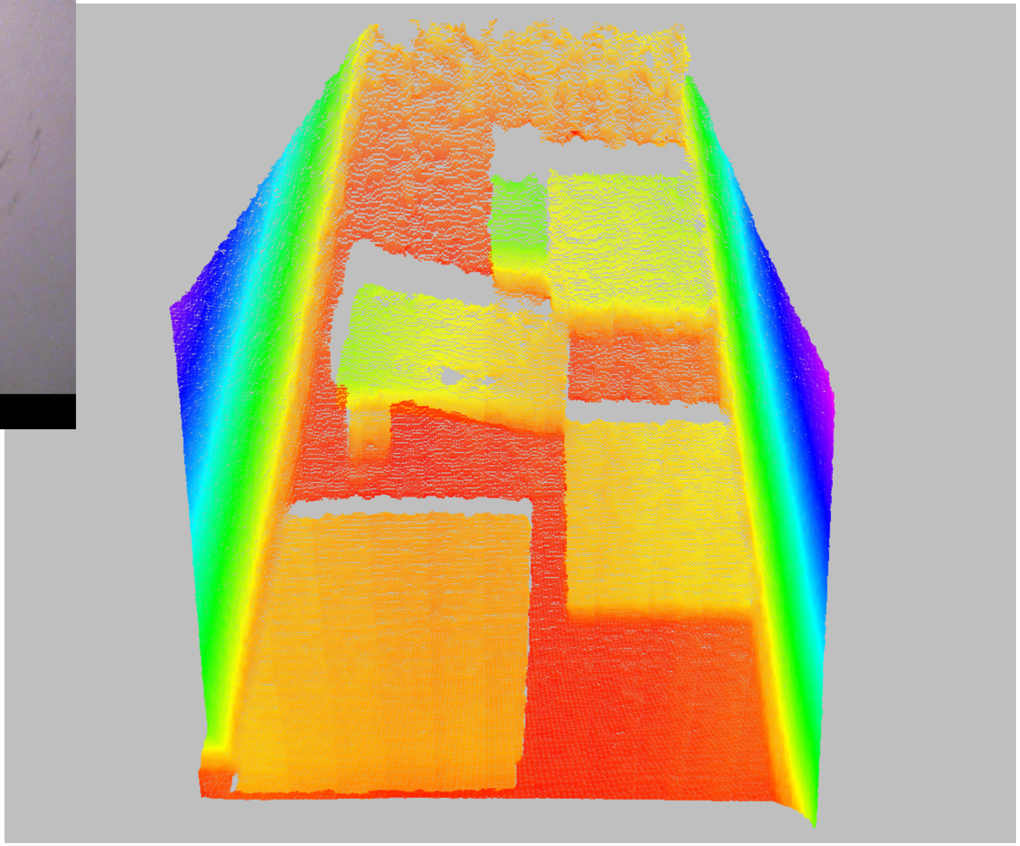
Sample based Footstep Planning

- **Requirements:**

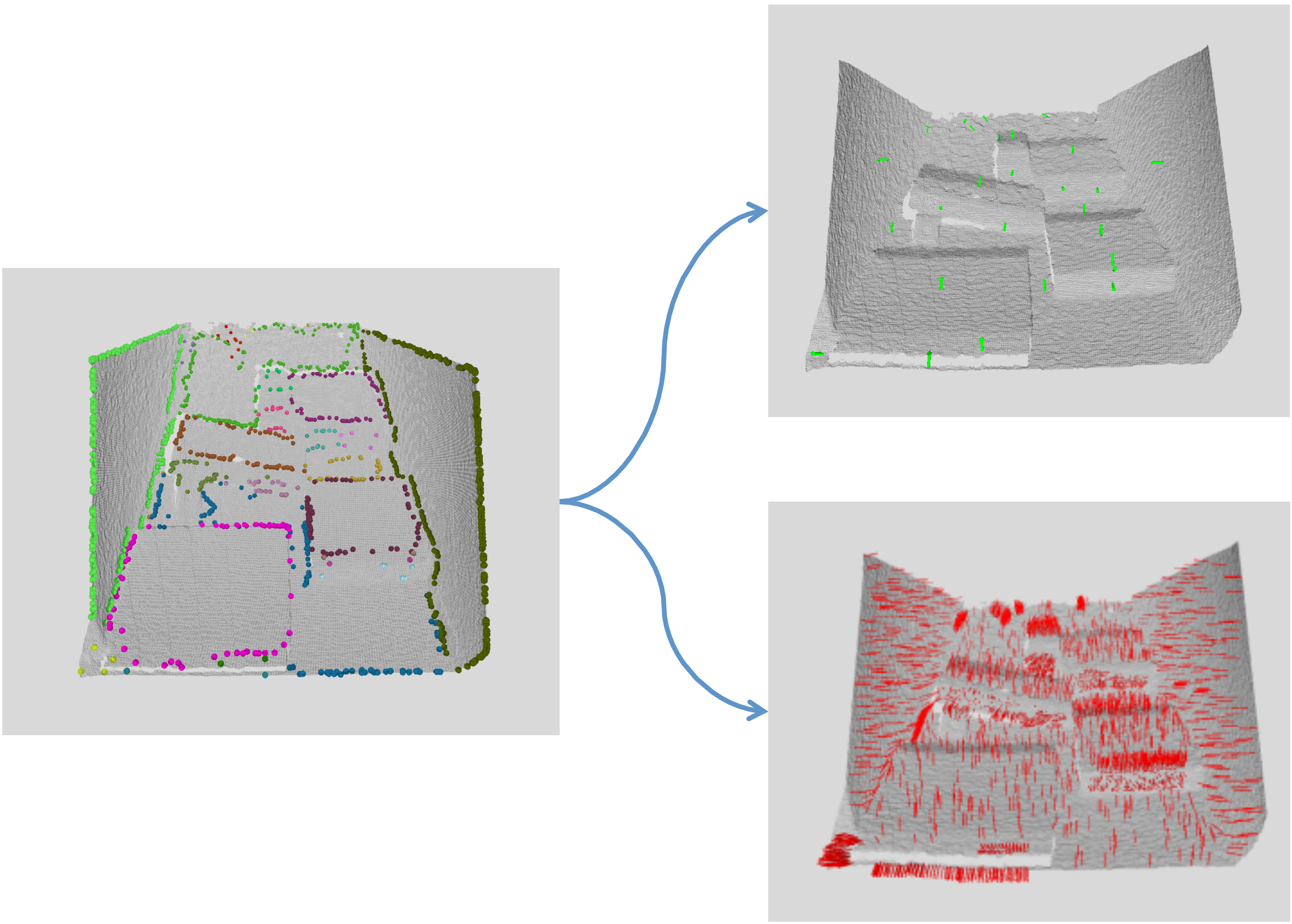
- Adaptability to different robots (URDF);
- Online steps computation;
- Adaptability to uneven and irregular scenarios;
- Intrinsically static stable set of footsteps.



From point clouds to almost planar regions

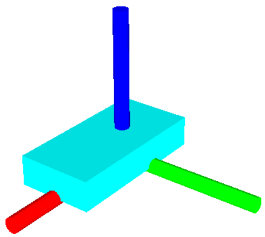


From point clouds to almost planar regions

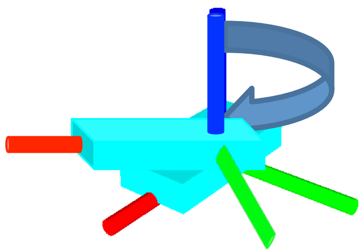


Points &
Normals

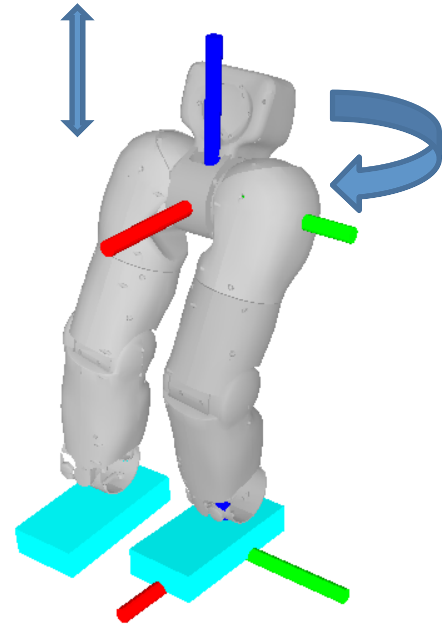
Workspace dimension



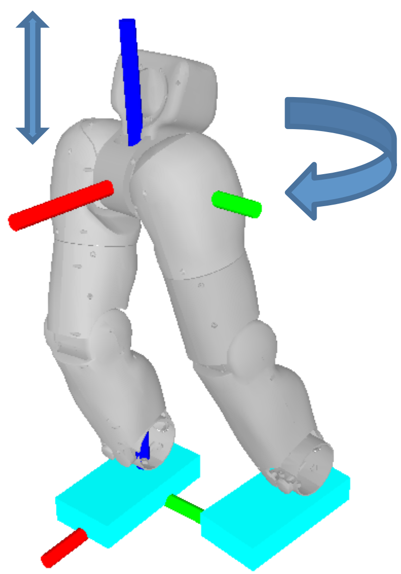
Points with normals
R^5



Swing foot frames
R^6

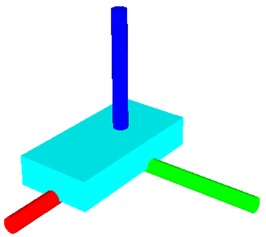


Waist on stance
R^8

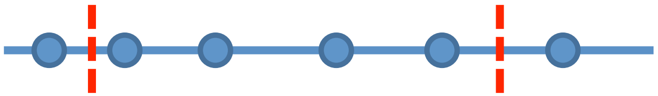


Waist on swing
R^{10}

The ‘Reduce-and-Expand’ approach



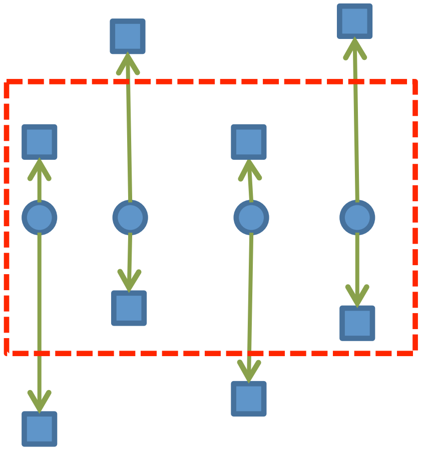
Points with normals
R^5

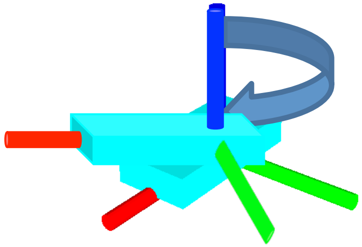


Checks:

- Tilt
- Region of interest
- Feet collision

The 'Reduce-and-Expand' approach

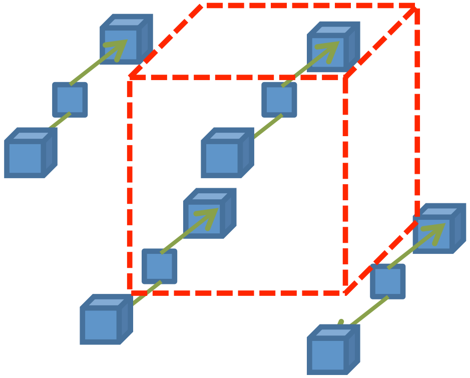


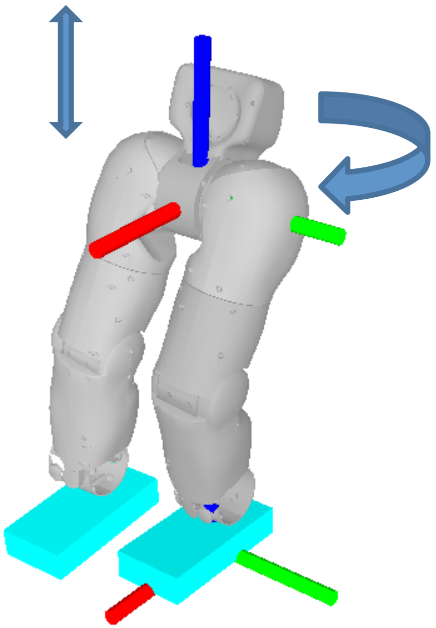


Swing foot frames
R^6

Kinematic feasibility check

The 'Reduce-and-Expand' approach

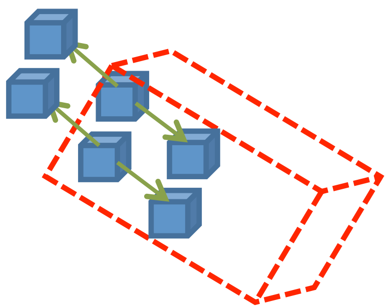


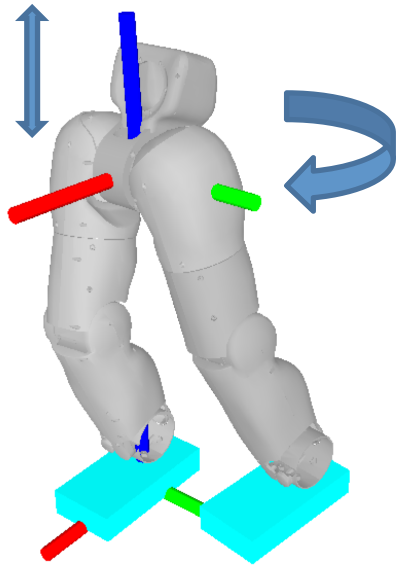


Waist on stance
R^8

Remove samples If (while waist on the stance foot) the swing foot on the sample in R^5 is not stable

The ‘Reduce-and-Expand’ approach





Waist on swing
R^{10}

Remove samples if (while waist on the swing foot) stance foot on the sample in R^5 is not stable

Optimal footstep selection

Algorithm 1: Reduce-and-Expand algorithm.

PLANFOOTSTEPS($\mathcal{P}, \mathcal{K}, n, \mathbf{d}$)

input : The point cloud $\mathcal{P}$ of the scene, the robot kinematics $\mathcal{K}$ from URDF, the number of steps n, the direction of motion $\mathbf{d}$, and the expansion factors $N_{5,6}$, $N_{6,8}$, and $N_{8,10}$.

output: The sequence of footsteps $\mathcal{S}^* = \{s^{(1)}, \ldots, s^{(n)}\}$

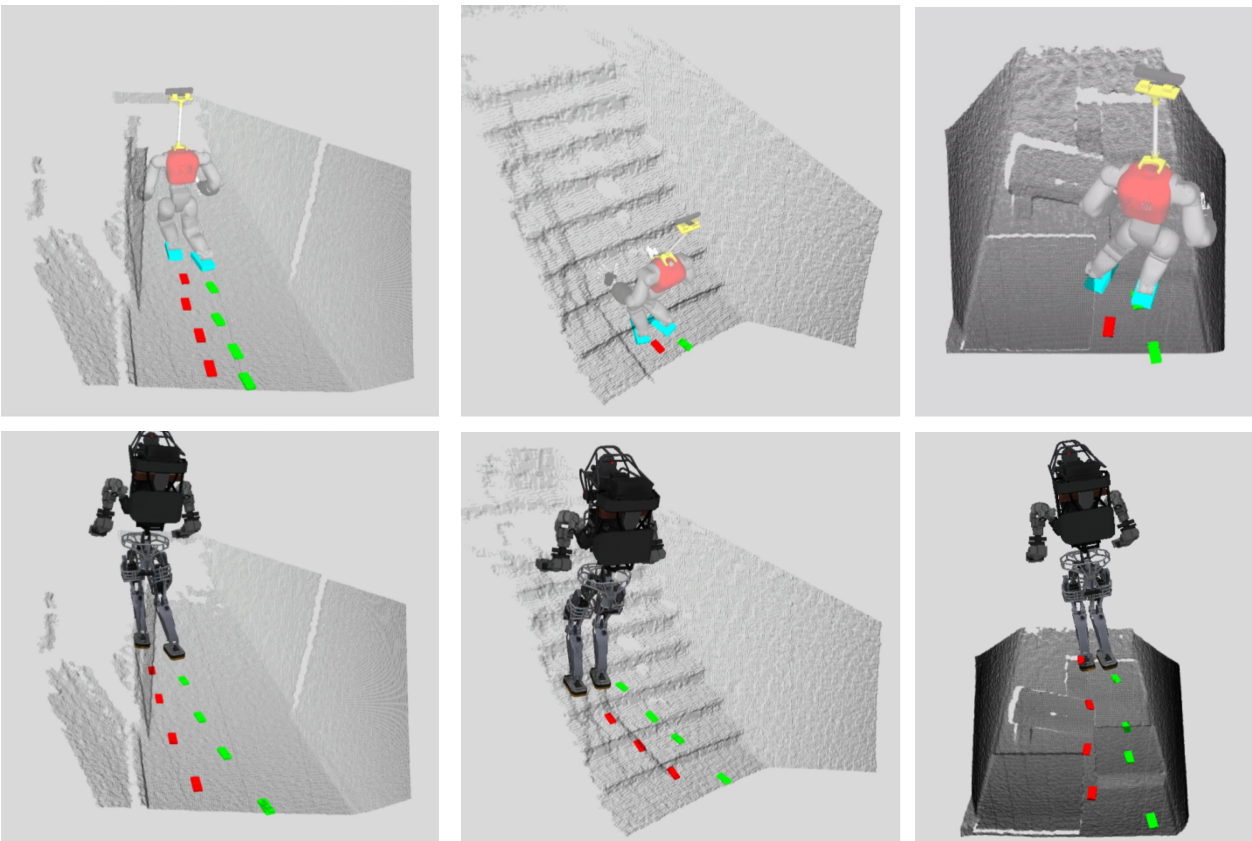
$\mathcal{C} \leftarrow$ FIND_ALMOST_PLANAR_REGIONS($\mathcal{P}$)
$\mathcal{B} \leftarrow$ EXTRACT_BORDERS($\mathcal{C}$)
$\mathcal{S}^5 \leftarrow$ SAMPLE_REGIONS ($\mathcal{C}$)
while *number_Of_Steps* $< n$ **do**
  $\mathcal{C}_G \leftarrow$ GEOMETRIC_SFM ($\mathcal{C}, \mathcal{K}$)
  $\mathcal{S}'^5 \leftarrow$ GEOMETRIC_SFM ($\mathcal{S}^5_G, \mathcal{K}$)
  $\mathcal{S}^6 \leftarrow$ EXPAND_PHASE6($N_{5,6}, \mathcal{S}'^5$)
  $\mathcal{S}^6 \leftarrow$ KINEMATIC_SFM ($\mathcal{S}^6, \mathcal{K}$)
  $\mathcal{S}^8 \leftarrow$ EXPAND_PHASE8($N_{6,8}, \mathcal{S}^6$)
  $\mathcal{S}^8 \leftarrow$ COM_SFM($\mathcal{S}^8, \mathcal{K}$)
  $\mathcal{S}^{10} \leftarrow$ EXPAND_PHASE10($N_{8,10}, \mathcal{S}^8$)
  $\mathcal{S}^{10} \leftarrow$ COM_SFM($\mathcal{S}^{10}, \mathcal{K}$)
  $\mathbf{s}^* \leftarrow$ SELECT_BEST_STEP($\mathcal{S}^{10}, \mathbf{d}$)
  $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \mathbf{s}^*$
return $\mathcal{S}^*$

Different possible loss functions

- Minimum distance from pre-computed pose + preferred direction
- Minimum Energy consumption
- Maximum Mobility
- Minimum Waist heading

Simulation results

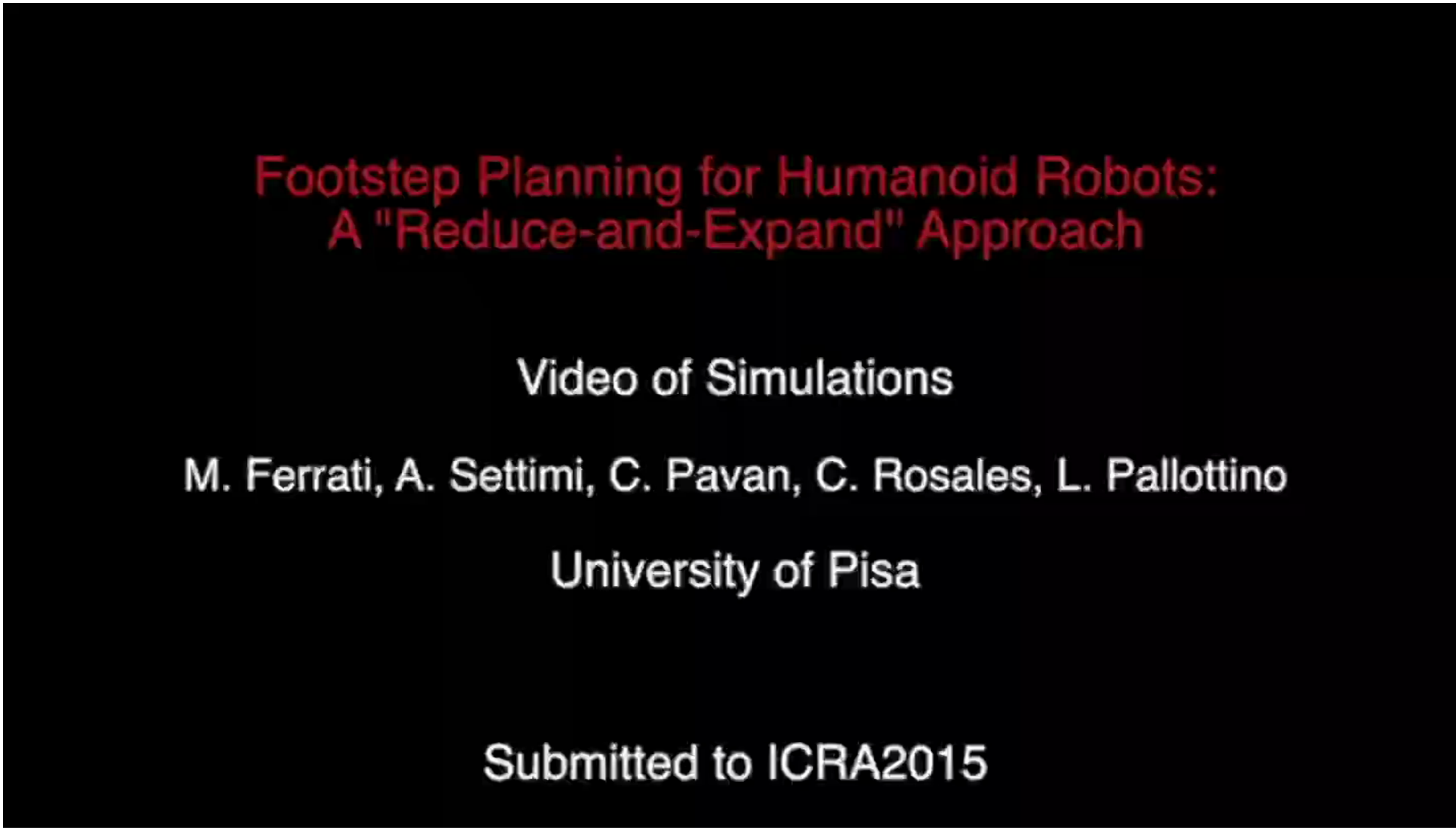
CORRIDOR STAIRS UNEVEN TERRAIN



COMAN

ATLAS

Planning results

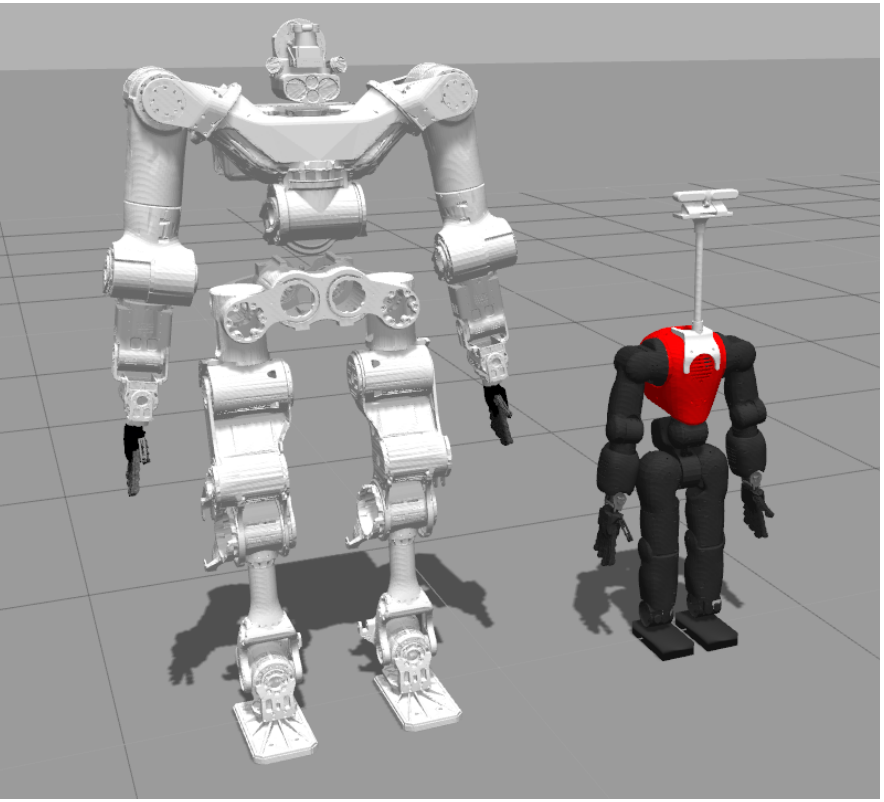


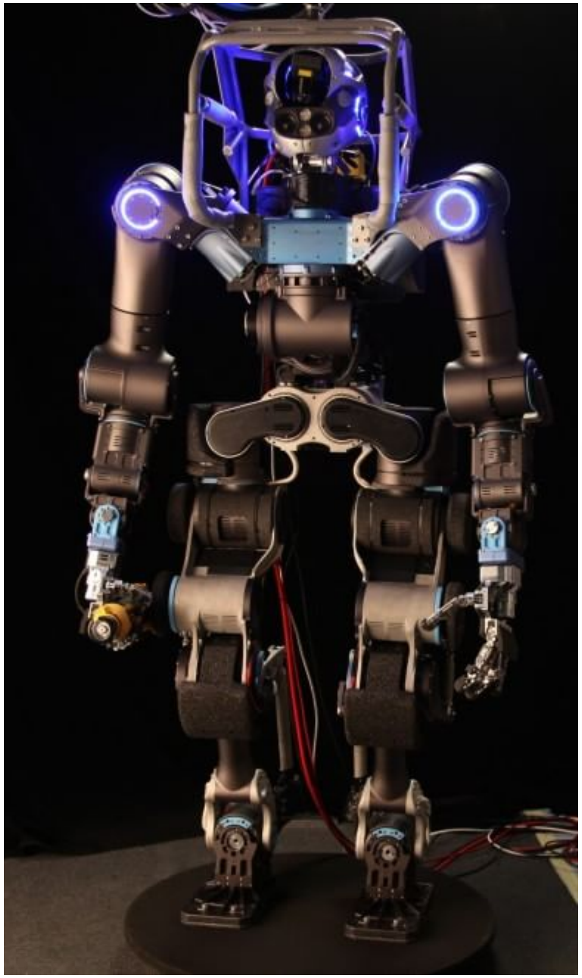
Future developments

Future (current) works focused on integration with:

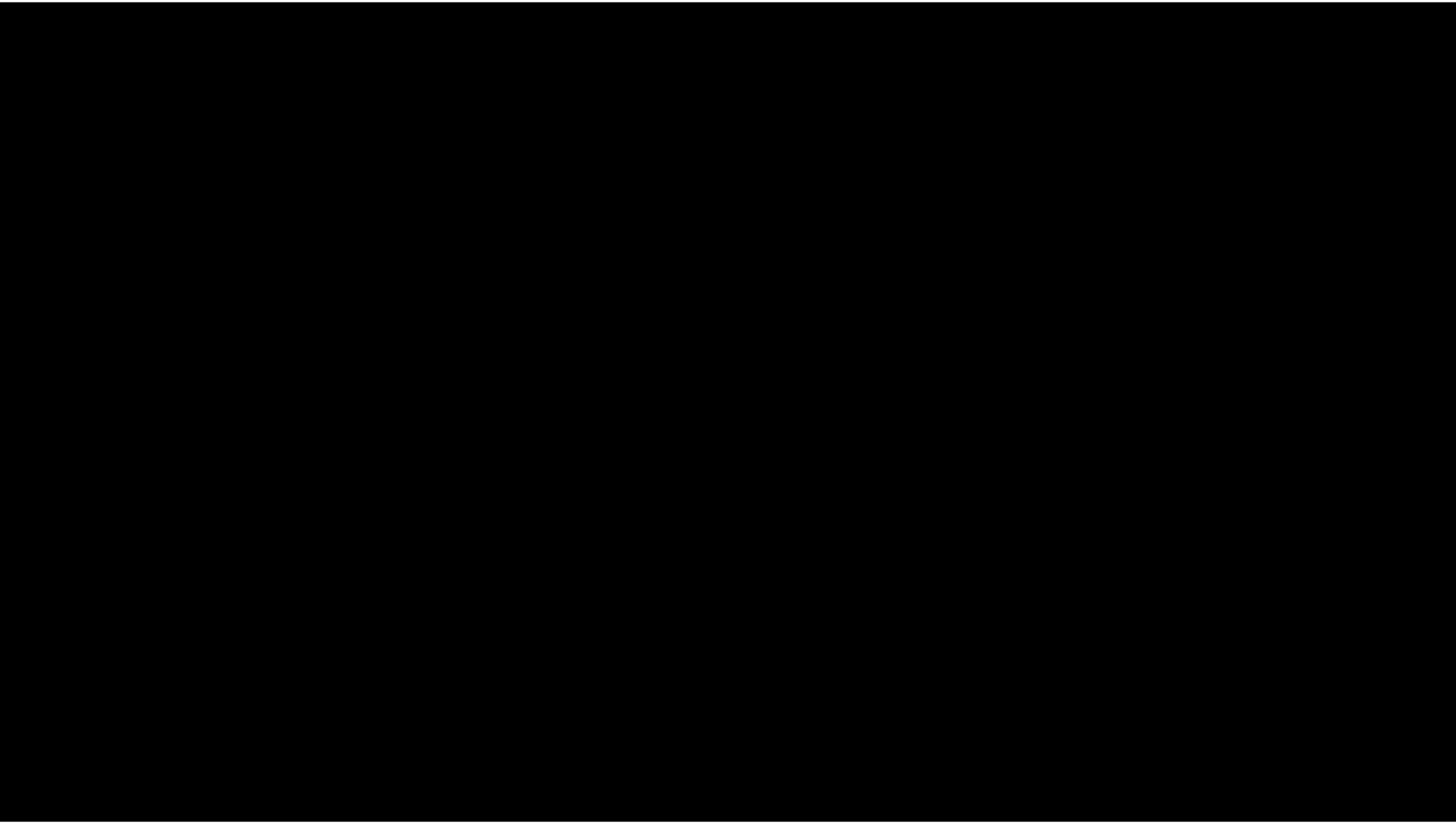
- Static walking control module
- Dynamic walking control module
- Whole body control module (extension to hands)

Walkman and Coman Robots

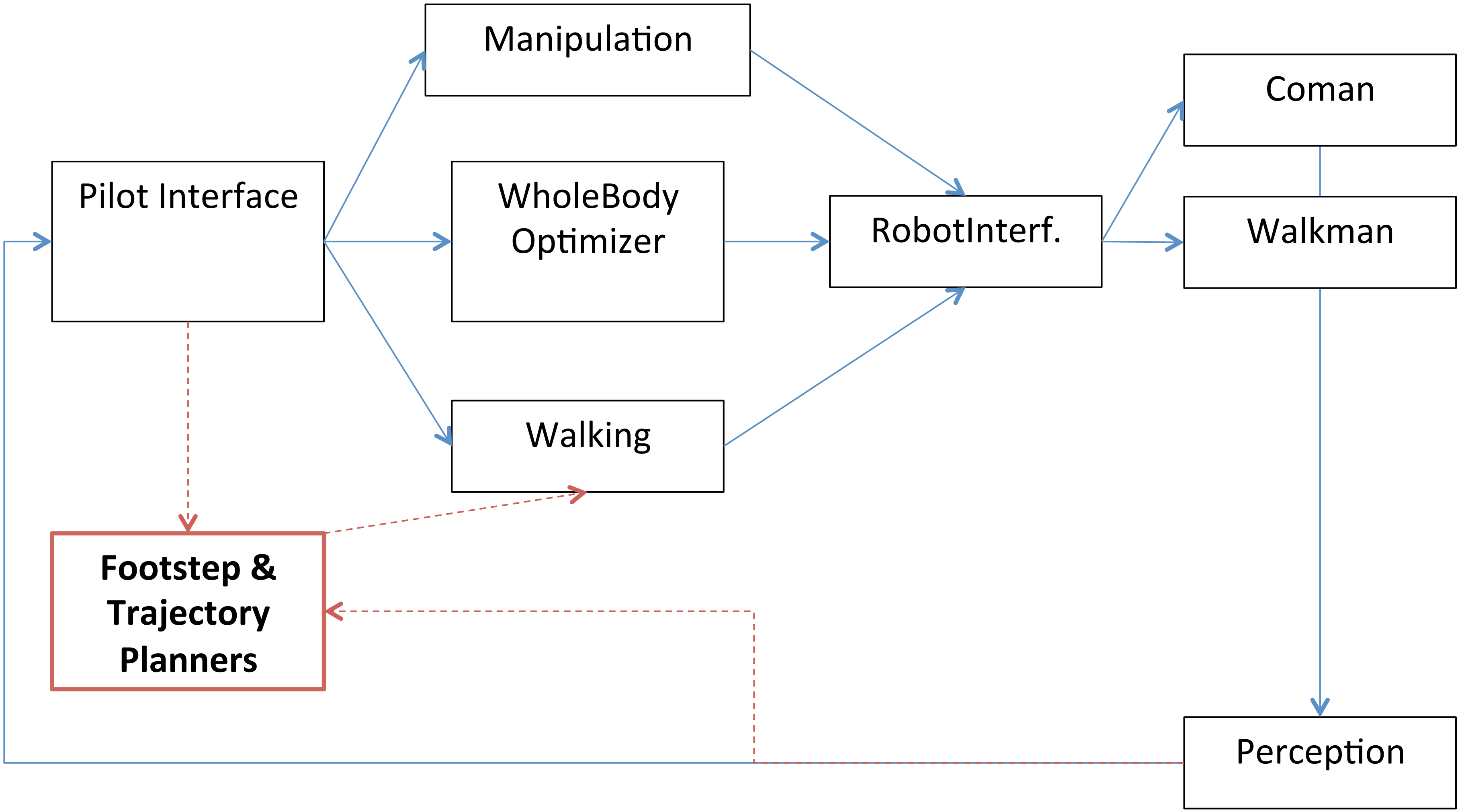




Walk-Man in action



Overall software architecture



Pilot Interface