IROS 2012
TRO (submitted)

Non-penetration Constraints for Optimization-based Motion Planning

Ph. D. Candidate
Youngeun Lee
youngeunlee@ewhain.net

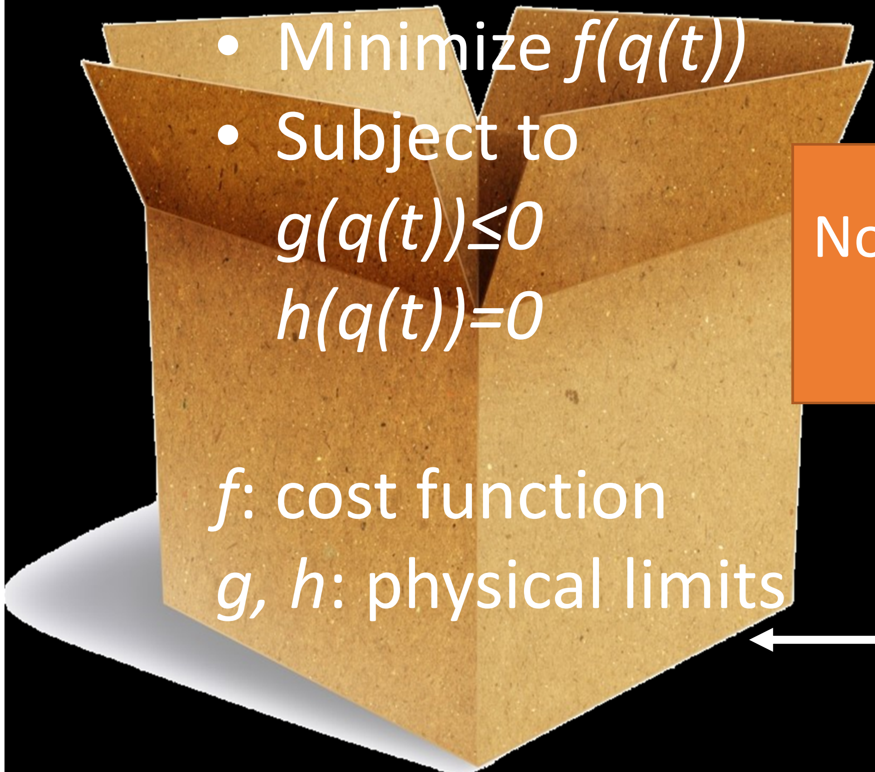
Optimization-based Motion Planning

- Finds the best object trajectories that minimize a cost function and satisfy constraints

Optimization-based Motion Planning

[Lengagne et al. IROS10]

Optimization Solver



- Minimize *f(q(t))*
- Subject to

  g(q(t))≤0

  h(q(t))=0

f: cost function

g, h: physical limits

Object Coordinates: *P*

Motion Parameterization

Object Trajectories: *q(t)*

Non-penetration Constraints

Constraints Approximation

g(q(t)), h(q(t))

Computing Extrema

Max g(q(t))

Min, Max h(q(t))

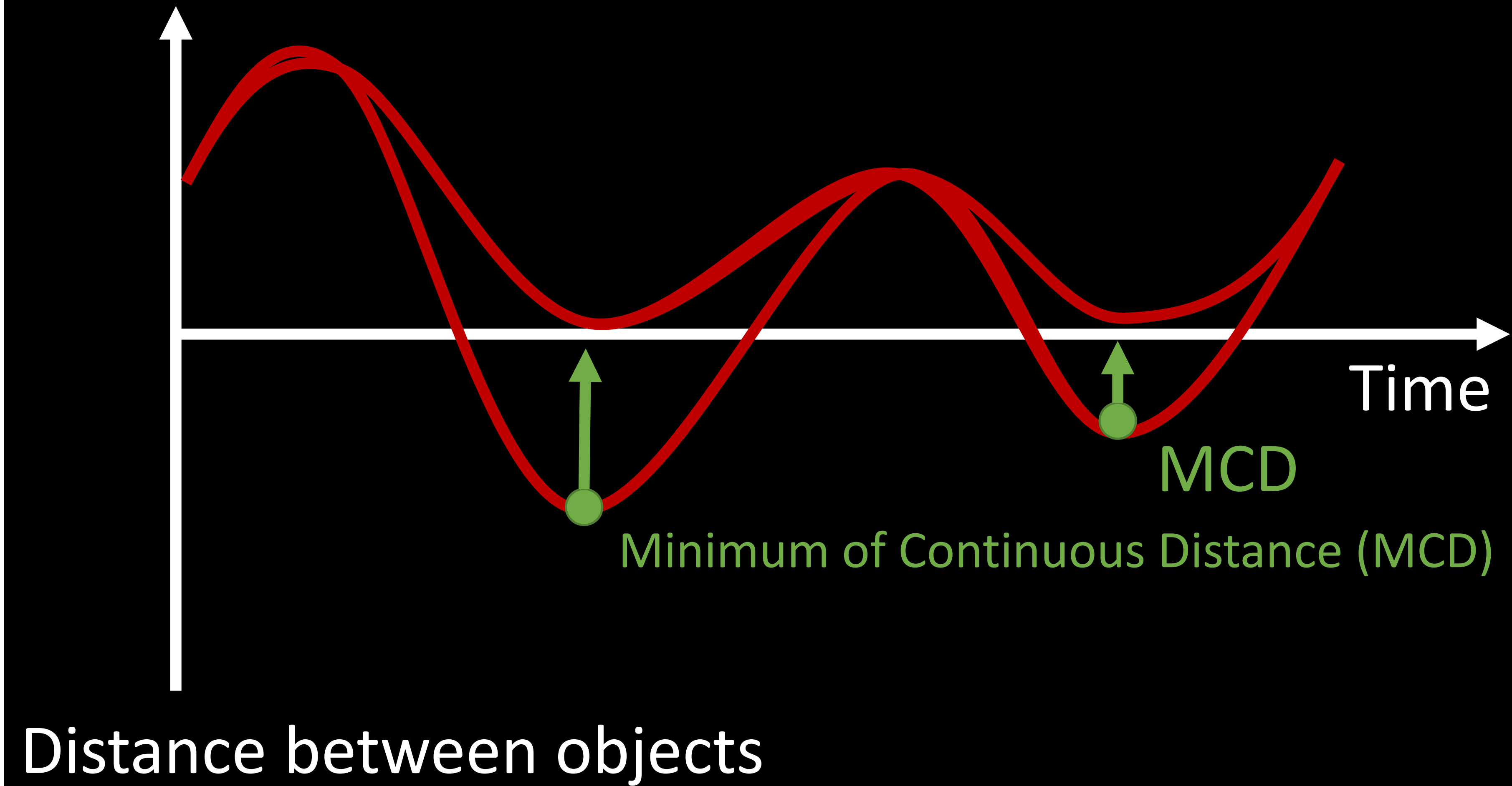
Non-penetration Constraints

- Collision avoidance
- Non-penetration constraint

$$\delta\big(A(t), B(t)\big) - \varepsilon \geq 0 \quad \forall t \in [0, T]$$

δ(·): distance function
A, B: objects

Non-penetration Constraints



Main Contributions

- Adding the non-penetration constraints into optimization-based motion planning
- Evaluates the minimum of continuous distance between pairs of objects
  - Takes a fraction of a millisecond for capsule-shaped objects
  - Takes 10~100 msec for general polygon objects consisting of tens of thousands of triangles
- Performs various simulation and experiments with HRP-2

Challenge

- Signed distance
- Time dependence
- Non-linear motion

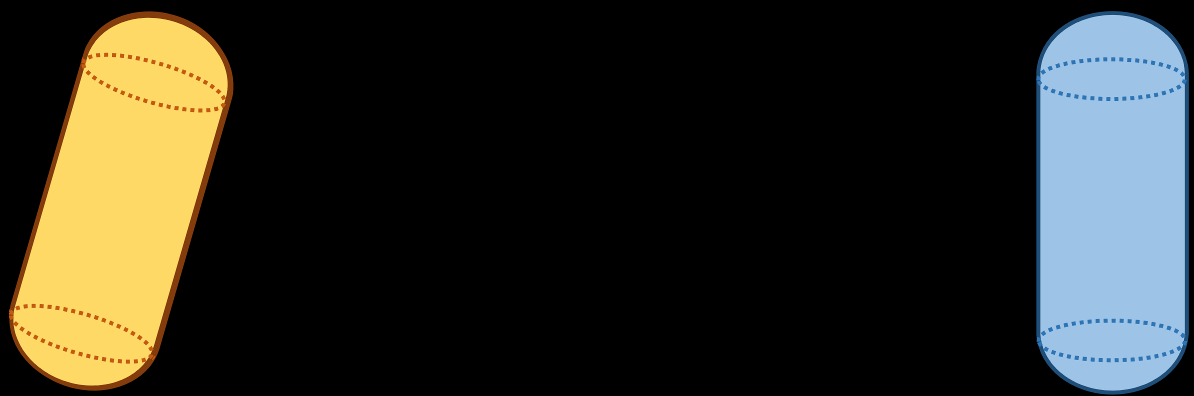
Capsule Shaped Objects

MCD Computation

- Golden Section Search
    - Fast
    - May miss the MCD

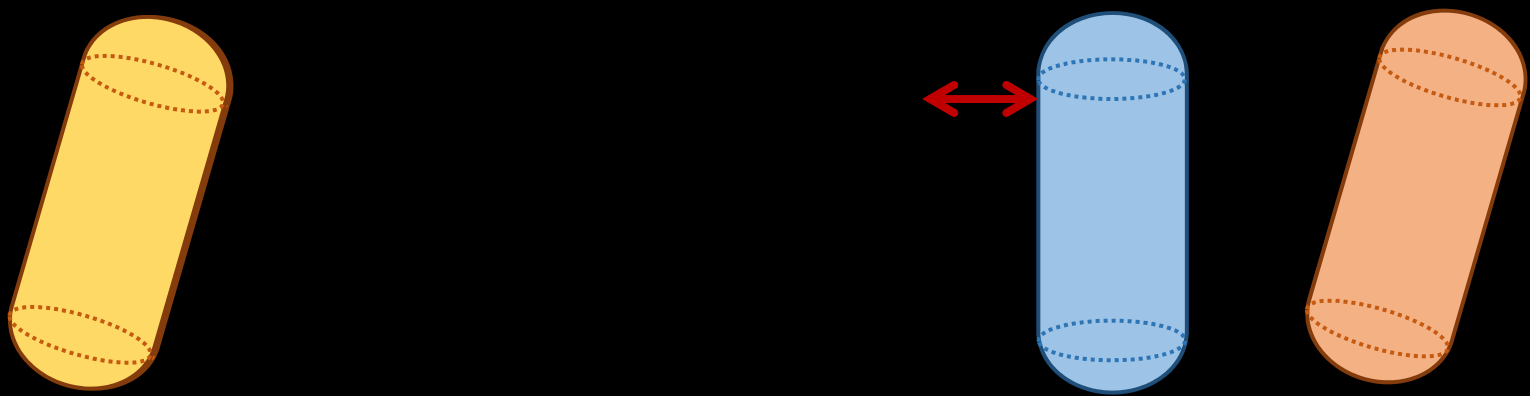
MCD Computation

- Golden Section Search
- Search Based on Conservative Advancement
    - CA computes time at $\delta(A,B)=0$ [Tang *et al.* ICRA 09]



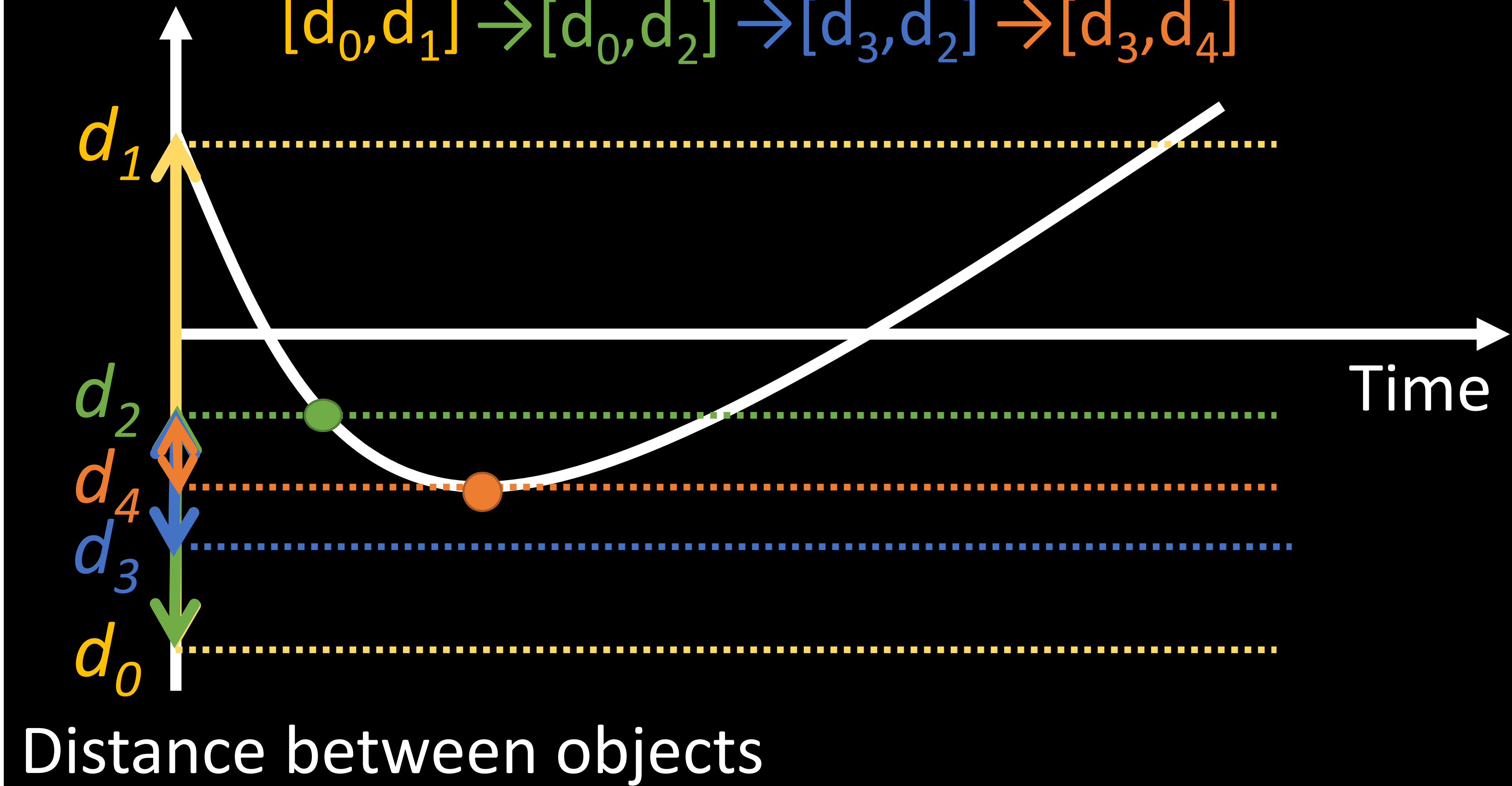
MCD Computation

- Golden Section Search
- Search Based on Conservative Advancement
  - CA computes time at *δ(A,B)=0* [Tang *et al.* ICRA 09]
  - CA also computes time at *δ(A,B)=**d***



Minimum Finding using CA

$[d_0, d_1] \rightarrow [d_0, d_2] \rightarrow [d_3, d_2] \rightarrow [d_3, d_4]$



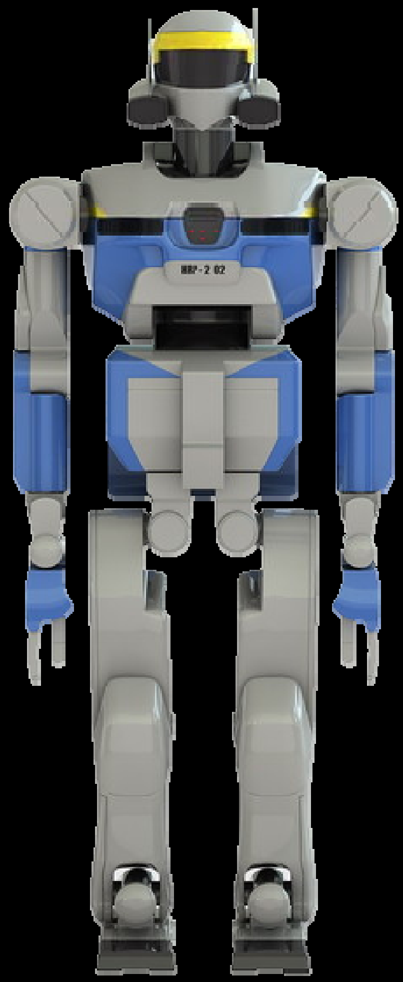
MCD Computation

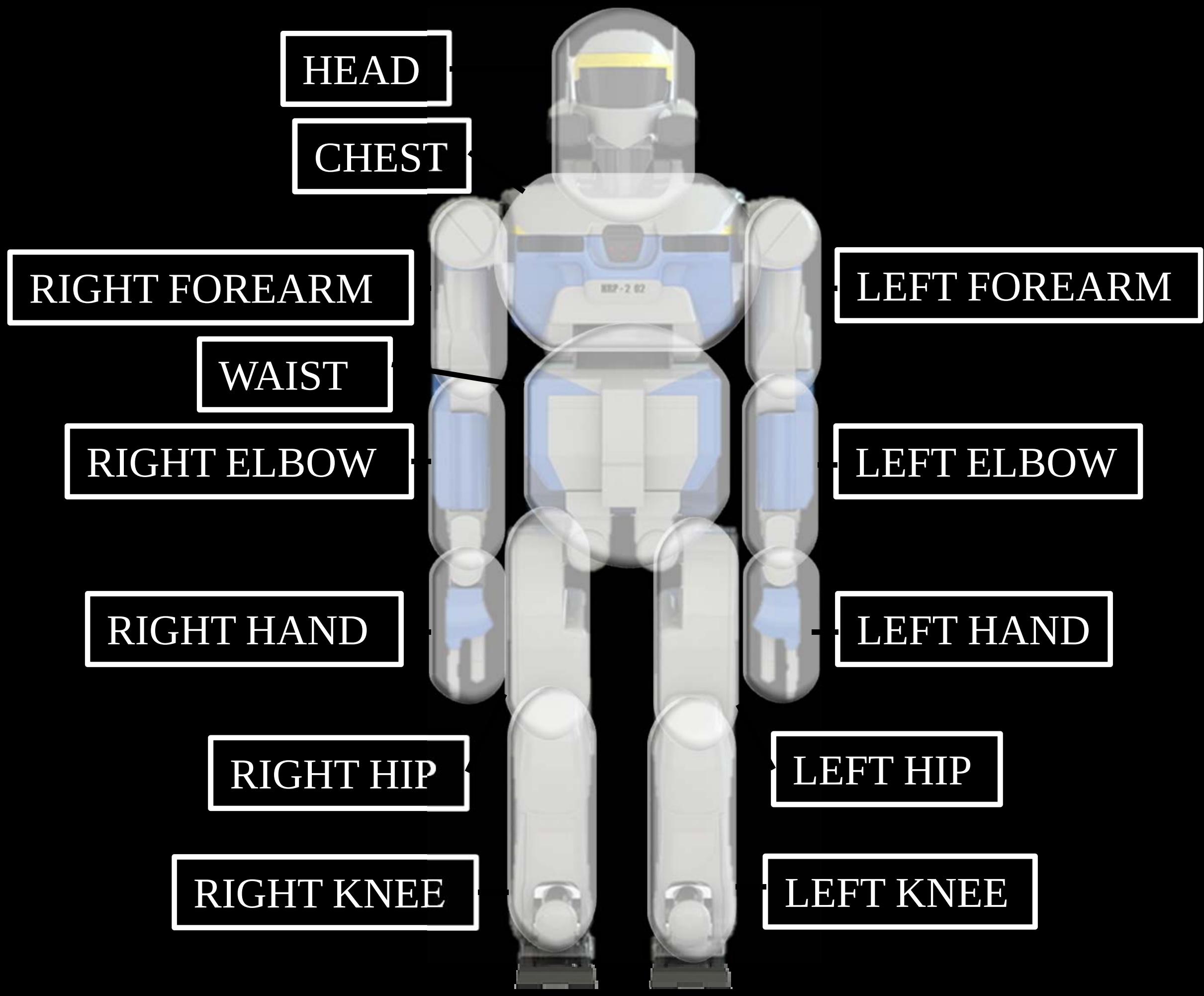
- Golden Section Search
- Search Based on Conservative Advancement
  - Guarantees the resulting interval on the codomain always contained the MCD
  - Slow

MCD Computation

- Golden Section Search
- Search Based on Conservative Advancement
- Hybrid Method
  - Fast
  - Guarantees the error bound

Bounding Volumes





Benchmark 1

Non-constrained

Ours

Benchmark 2

Non-constrained

Ours

Real Robot Execution

Benchmark 3

Non-constrained

Ours

General Polygon Objects

Adaptive Subdivision

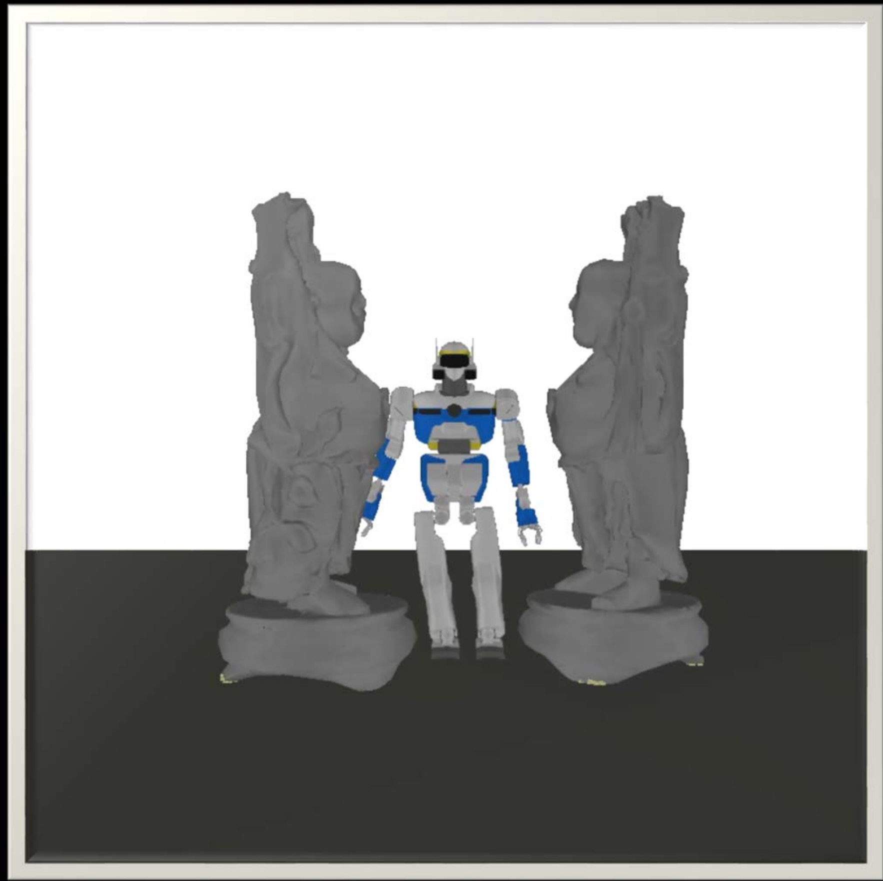
- Capsule shapes can be too conservative for complicated motion planning

- CA can not be applied to general polygonal objects

- Adaptive subdivision recursively subdivides the time intervals which contain the MCD until the distance results is within the error bound

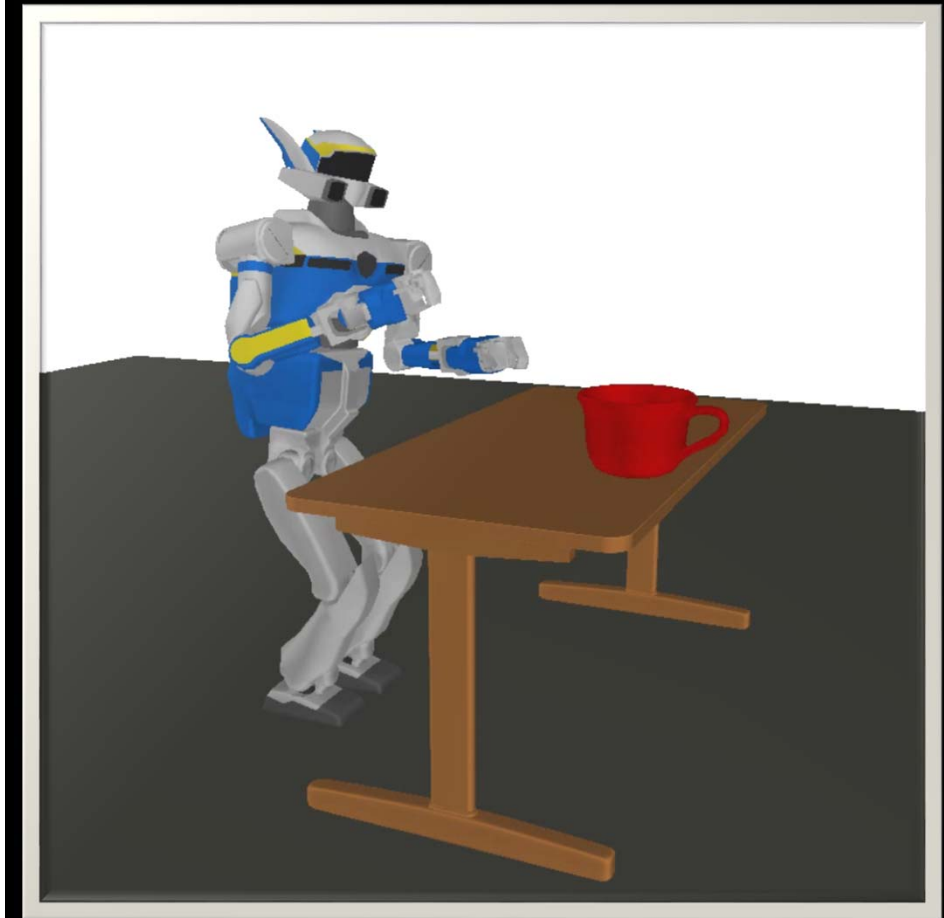
Benchmark 1

Non-constrained



Ours

Benchmark 2



Non-constrained Ours

Summary & Future Work

- Summary
  - Integrated the non-penetration constraints into optimization-based motion planning
  - Defined and computed the minimum of continuous distance

- Future work
  - Computing the continuous and stable distance
  - Applying the MCD into sampling-based motion planning

Thank you

Conclusion

- Integrated the non-penetration constraints into optimization-based motion planning
- Defined and computed the minimum of continuous function
  - Capsules: GSS, CA, and Hybrid
  - General polygon models: AS
- Performed various simulations humanoid character models

Future Work

- Improving the performance of distance computation
- Computing the continuous and stable distance
- Applying the MCD into sampling-based motion planning

Motion Planning

- Produces a continuous motion that connects a start configuration and a goal configuration

- Sampling-based planning
  - Widely used techniques
  - Mostly limited static environments