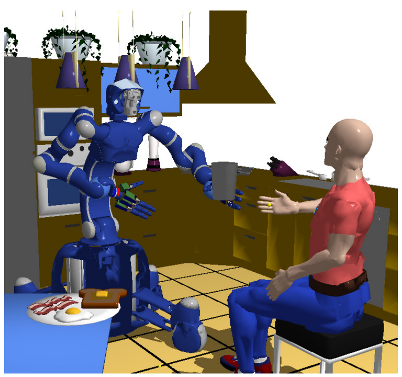
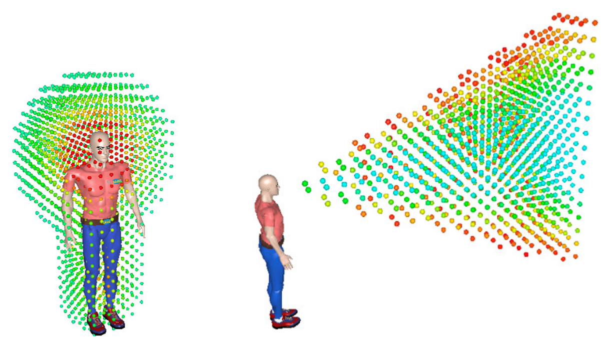
Planning human-aware motions using a sampling-based costmap planner

Team #1 Seminar Presentation

Andrew Hundt





(a) Distance costmap (b) Visibility costmap

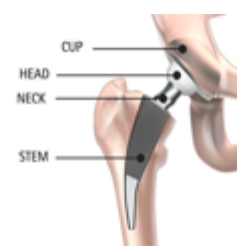
Robone Project Overview

Long term Project Goal – Develop a next generation total hip replacement surgical assistance device that performs bone milling during surgery.

Existing orthopedic surgical devices aren't perfect and require the patient to be fixed to the operating table, an invasive and time consuming process.

A next generation system will make real time position adjustments using a device such as an optical tracker so fixation is no longer necessary.

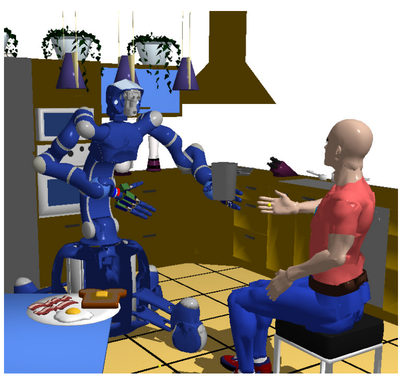
The goal behind these improvements is to enable faster and less invasive surgery with this type of computer assisted tool.

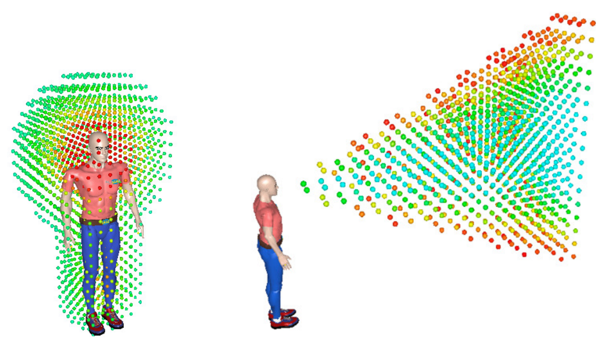


Today's Papers

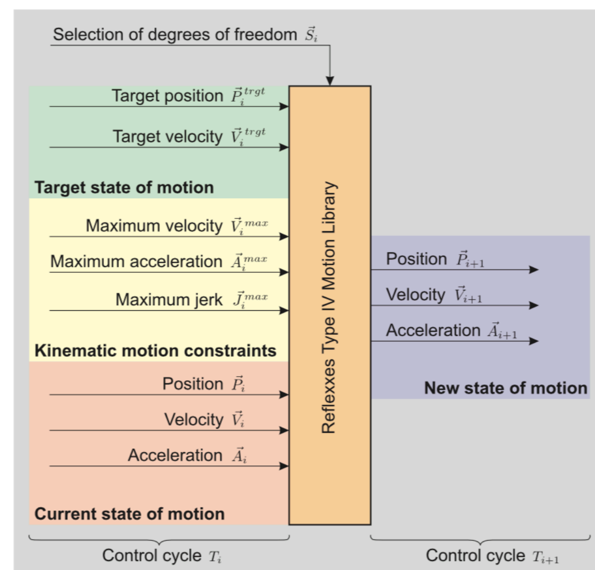
Mainprice, Jim et al. "Planning human-aware motions using a sampling-based costmap planner." *Robotics and Automation (ICRA), 2011 IEEE International Conference on* 9 May. 2011: 5012-5017.

Kroger T. Opening the door to new sensor-based robot applications;the reflexxes motion libraries. *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. 2011:1-4.





(a) Distance costmap (b) Visibility costmap



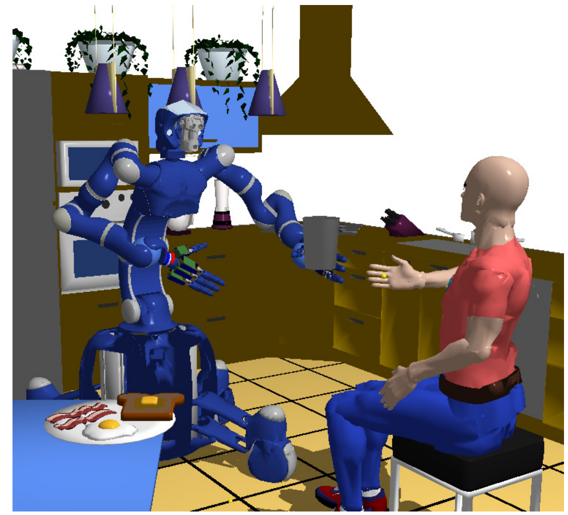
Mainprice, Jim et al. Motivation

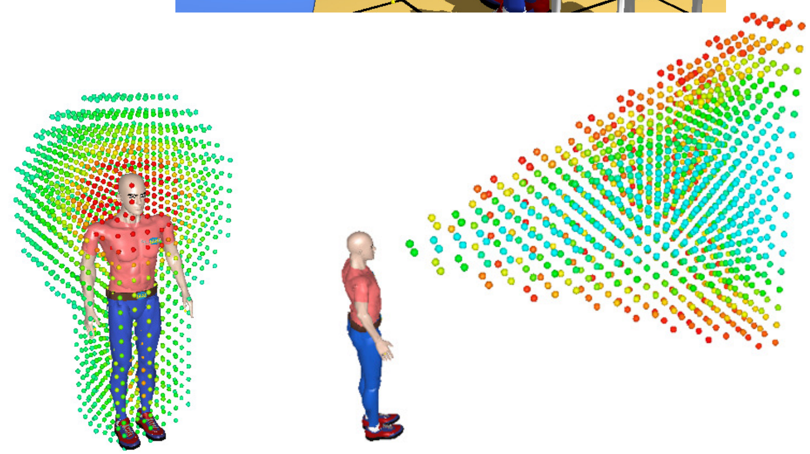
- The robone device will be used in an operating room and necessarily interact with surgeons controlling the machine and individuals undergoing surgery.

- Metrics and planning algorithms that account for the Human-Robot Interaction (HRI) experience are one potential approach for constraining motion of the redundant 7 degree of freedom KUKA iiwa arm.

Mainprice et al. Introduction

- This paper introduces a randomized cost based planner that incorporates costs based on human situational awareness.
- Relies on their own T-RRT algorithm with additional extensions for human costs and path smoothing.

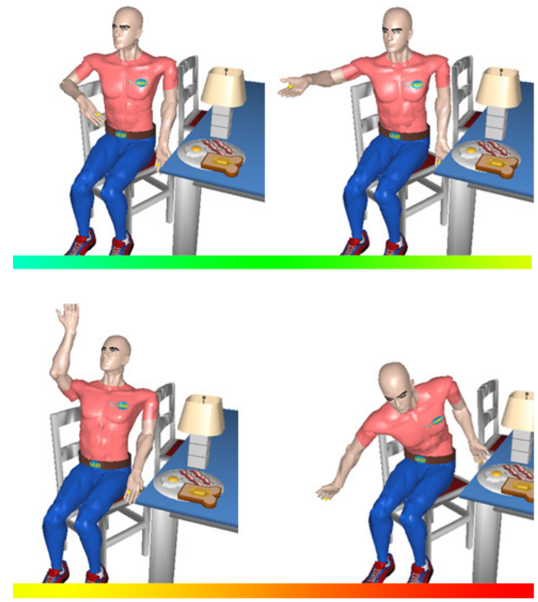


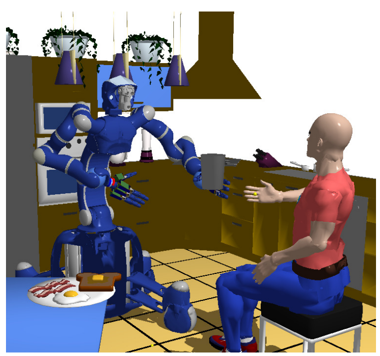


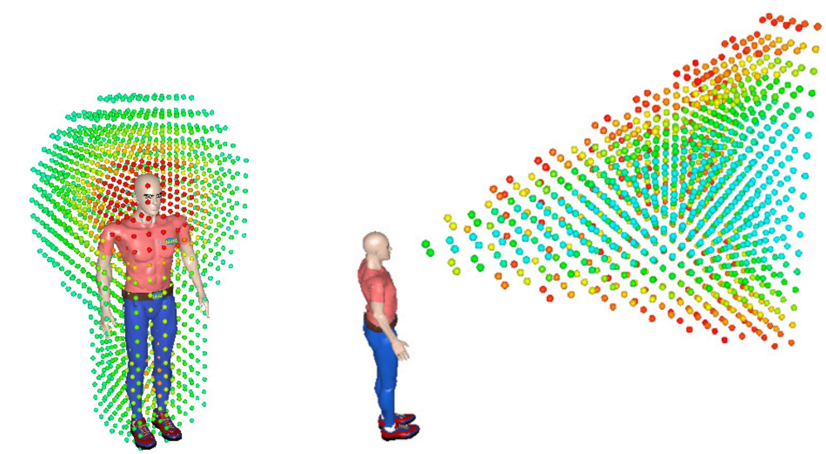
(a) Distance costmap (b) Visibility costmap

Mainprice et al. Introduction

- Cost map includes
  1. Comfort metrics for item handoff to a human
  2. Distance based safety constraints
  3. Visibility costmap







(a) Distance costmap (b) Visibility costmap

Mainprice et al. Item handoff Constraints

1. Distance of human joint angles q from a resting posture.

$$f_1 = \sum_{i=1}^{DOF} w_i (q_i - q_i^N)^2$$

2. Potential energy of human arm using forearm height, distance from resting posture Δz and an estimation of arm mass.

$$f_2 = \sum_{i=1}^{2} (m_i g)^2 (\Delta z_i)^2$$

3. Configurations close to joint limits are penalized.

$$f_3 = \sum_{i=1}^{DOF} \gamma_i \Delta q_i^2$$

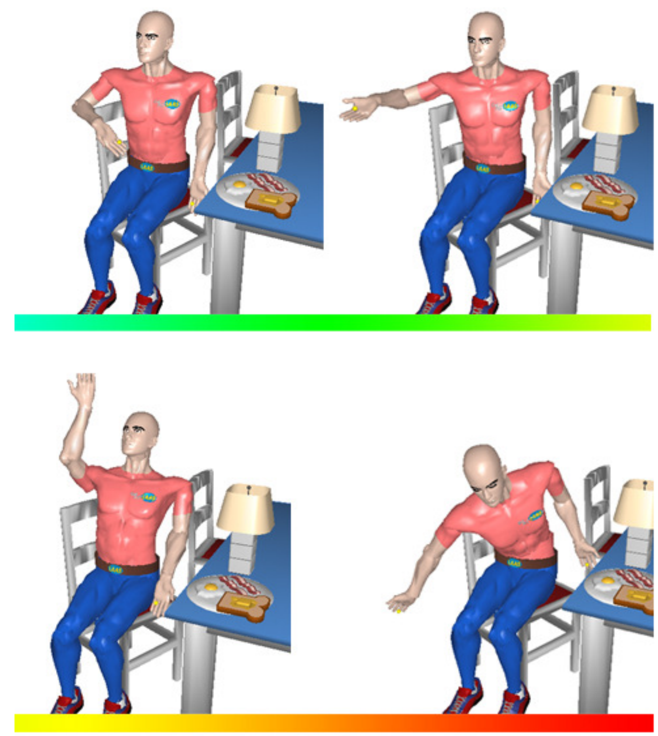


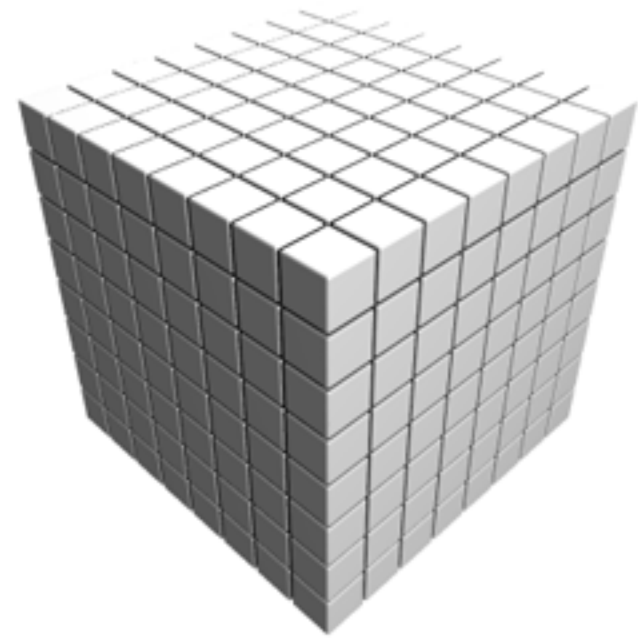
Fig. 3. Arm comfort: Four poses that vary from comfortable and natural on the upper left corner to uncomfortable and uneasy postures on the lower right corner, the color gradient expresses the corresponding cost function value.

Combined Costmap

Constraints are combined with a weighted sum in a 3D costmap.

- h – human model posture
- x – Point in workspace

$$c(h,x) = \sum_{i=1}^{N} w_i c_i(h,x),$$



Ultimately, cost is defined in the robot's configuration space.

- FK – robot forward kinematics function
- q – robot configuration

$$c(h,q) = \sum_{i=1}^{N} w_i c_i(h, FK(q)),$$

Algorithm 1: Transition-based RRT

input : the configuration space CS;
the cost function $c : CS \rightarrow \mathbb{R}_{+}^{*}$;
the root q_{init} and the goal q_{goal};

output : the tree $\mathcal{T}$;

begin

$\mathcal{T} \leftarrow$ `InitTree`(q_{init});

while not `StopCondition`($\mathcal{T}$, q_{goal}) **do**

$q_{\text{rand}} \leftarrow$ `SampleConf`(CS) ;

$q_{\text{near}} \leftarrow$ `NearestNeighbor`($q_{\text{rand}}, \mathcal{T}$);

$q_{\text{new}} \leftarrow$ `Extend`($\mathcal{T}, q_{\text{rand}}, q_{\text{near}}$);

if $q_{new} \neq NULL$

and `TransitionTest`($c(q_{near}), c(q_{new}), d_{near-new}$)

and `MinExpandControl`($\mathcal{T}$, q_{near}, q_{rand}) **then**

`AddNewNode`($\mathcal{T}$, q_{new});

`AddNewEdge`($\mathcal{T}$, q_{near}, q_{new});

end

Transition Test

Filters new configurations when the cost increases compared to the parent configuration.

$$p_{ij} = \begin{cases} exp(-\frac{\Delta c^*_{ij}}{K \cdot T}) & \text{if } \Delta c^*_{ij} > 0 \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

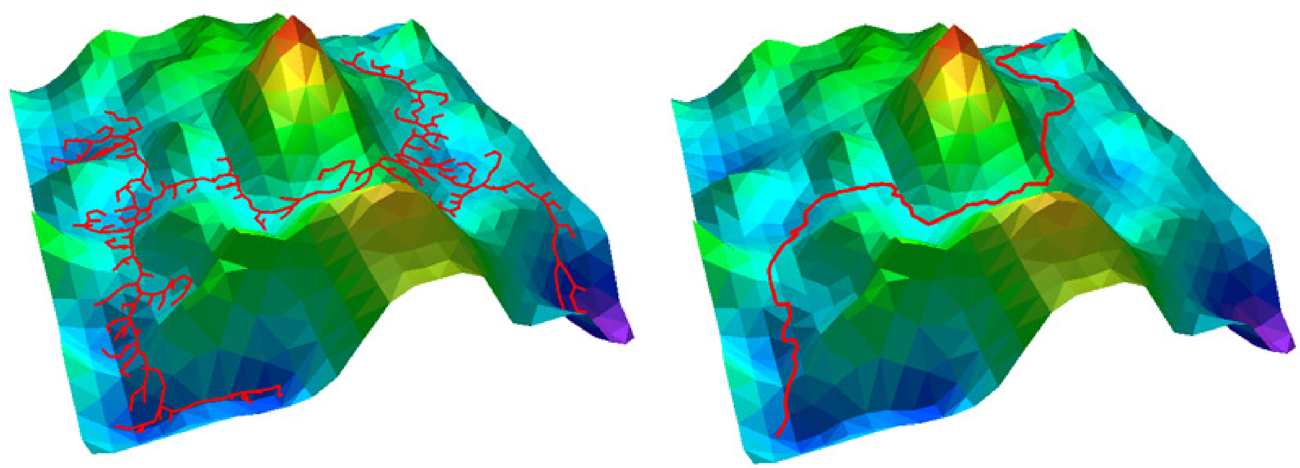


Fig. 4. T-RRT constructed on a 2D costmap (left). The transition test favors the exploration of low-cost regions, resulting in good-quality paths (right).

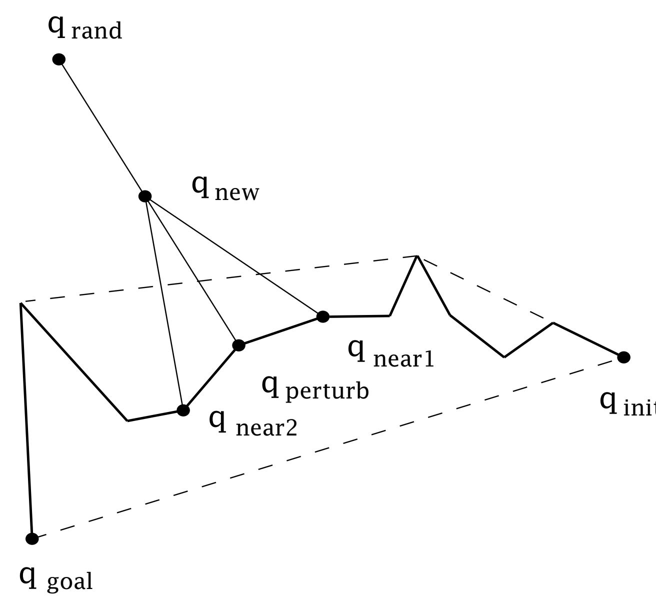
Algorithm 2: Random Cost Shortcut

```
input   : The Path 𝒫;
output  : The Path 𝒫;
begin
    while not StopCondition() do
        (q1, q2) ← 𝒫.getTwoConfig();
        ℒ𝒫 ← getSegment(q1, q2) ;
        if isValidAndLowerCost(ℒ𝒫, q1, q2) then
            𝒫.ReplacePortion(ℒ𝒫, q1, q2) ;
end
```

Algorithm 3: Random Path Perturbation

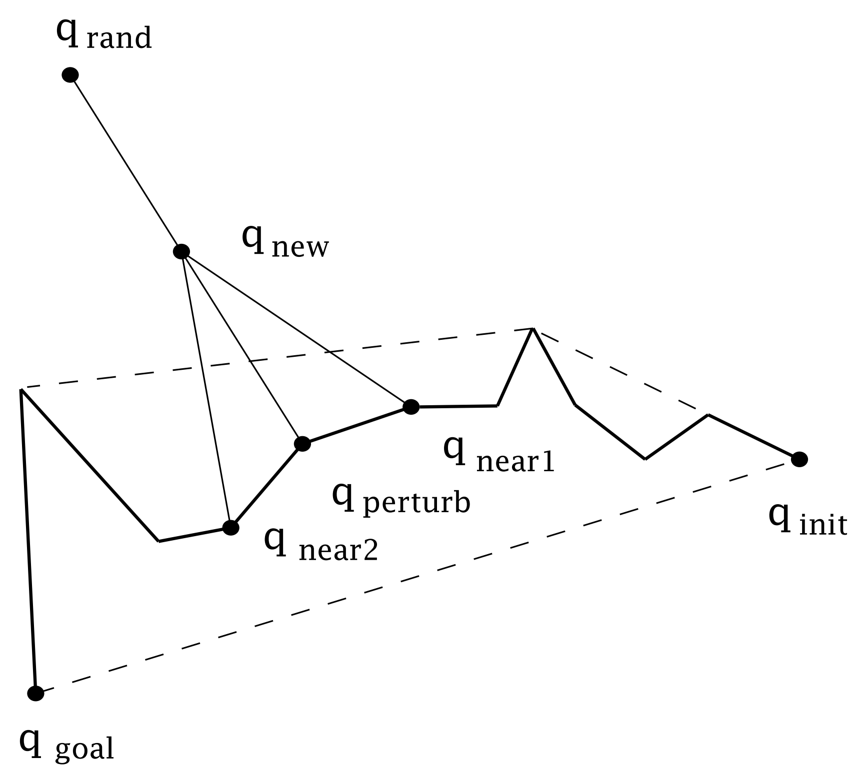
Algorithm 3: Random Path Perturbation

input : The Path $\mathcal{P}$;
output : The Path $\mathcal{P}$;
begin
  while not `StopCondition`*()* **do**
    $q_{perturb} \leftarrow \mathcal{P}$.`shootRandConfigOnPath`();
    $(q_{near1}, q_{near2}) \leftarrow \mathcal{P}$.`getNeigh` $(q_{perturb}, step)$;
    $q_{rand} \leftarrow$ `shootRandDirection`() ;
    $q_{new} \leftarrow$ `Expand`$(q_{perturb}, q_{rand}, step)$;
    $\mathcal{LP}_1 \leftarrow$`getSegment` (q_{near1}, q_{new}) ;
    $\mathcal{LP}_2 \leftarrow$`getSegment` (q_{new}, q_{near2}) ;
    $\mathcal{LP} \leftarrow \mathcal{LP}_1 + \mathcal{LP}_2$;
    if `isValidAndLowerCost`$(\mathcal{LP}, q_{near1}, q_{near2})$ **then**
      $\mathcal{P}$.`ReplacePortion` $(\mathcal{LP}, q_{near1}, q_{near2},)$;
end

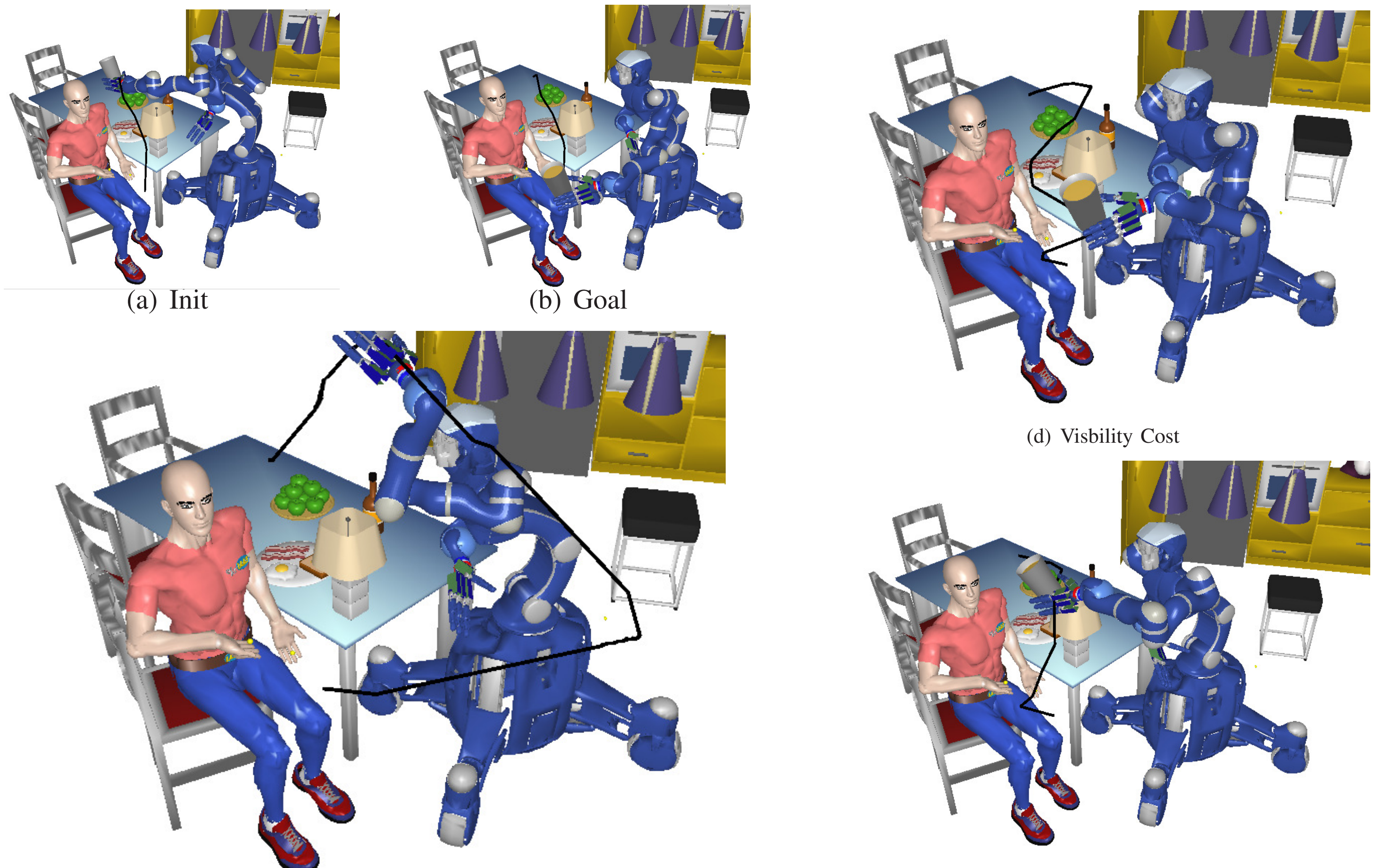


Algorithm 3: Random Path Perturbation

This path perturbation mechanism can generate positions outside the convex hull of the path, allowing better global optimization than alternatives which remain within the convex hull.



Simulated Handoff Results



(a) Init

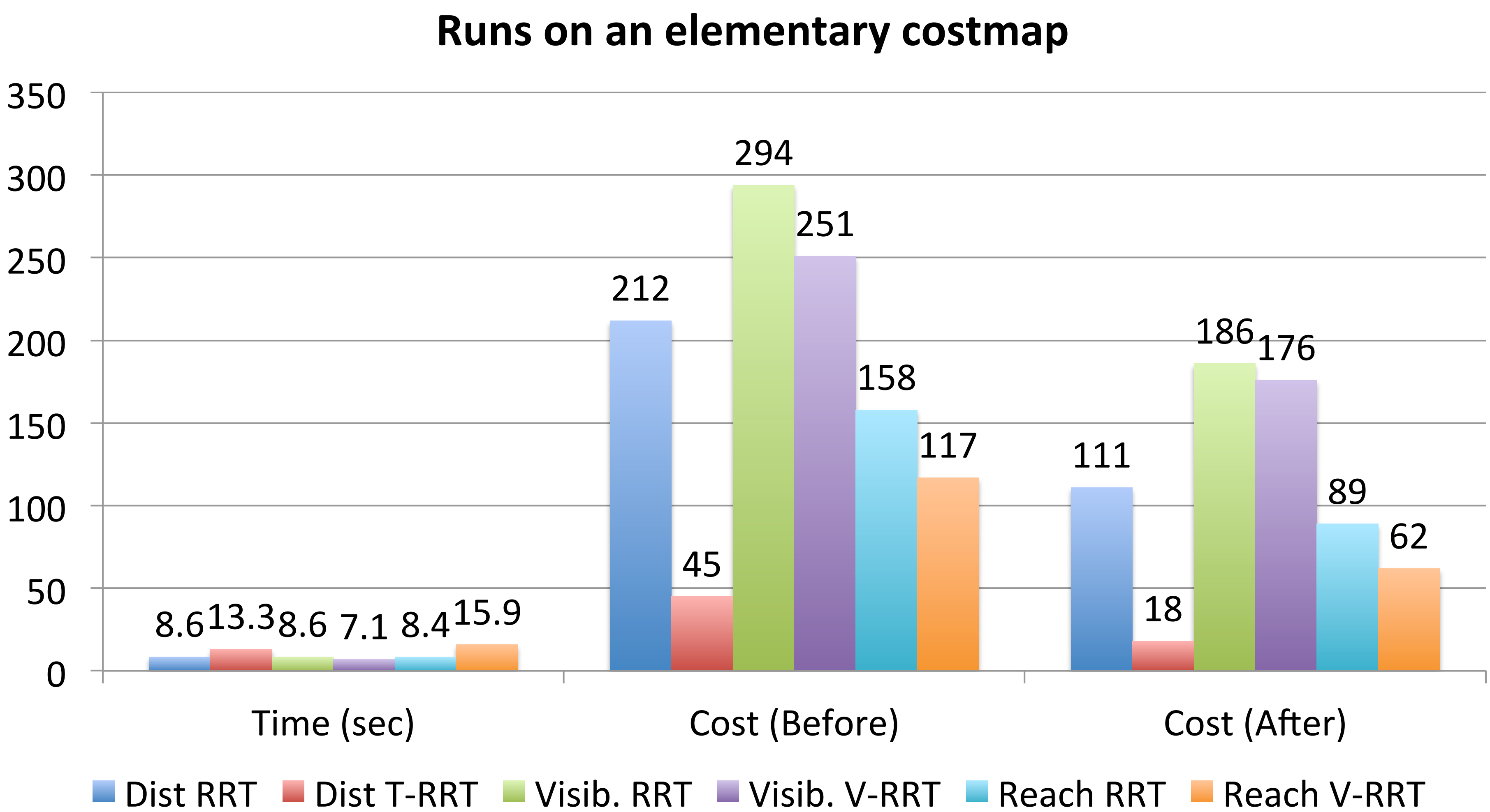
(b) Goal

(c) Distance Cost

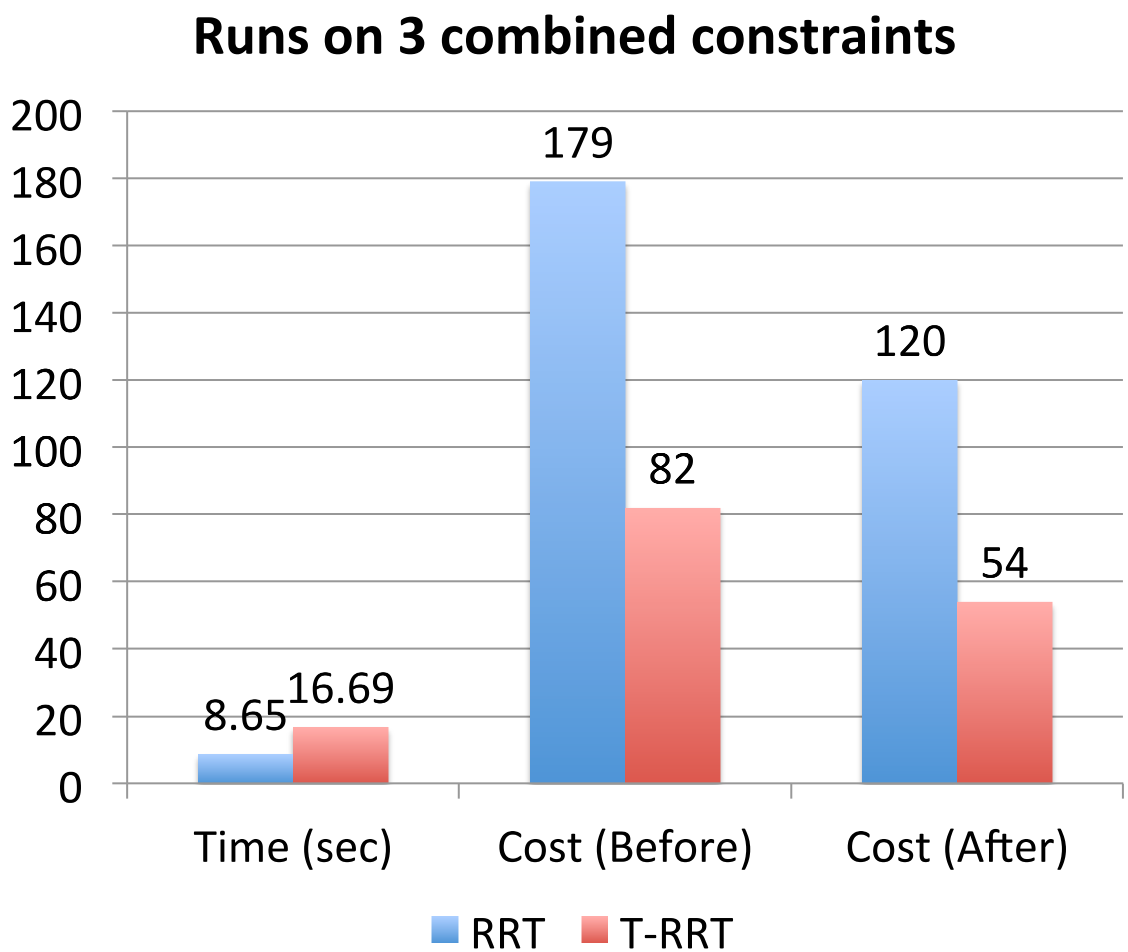
(d) Visbility Cost

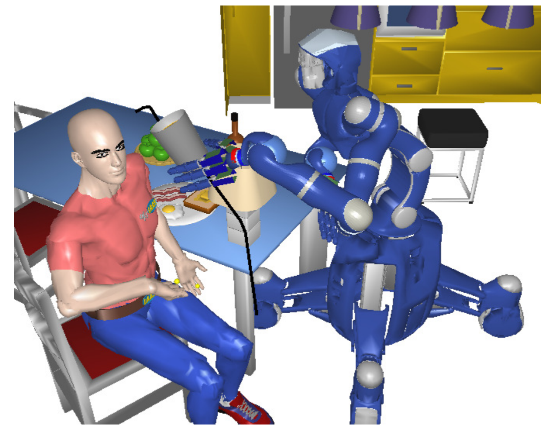
(e) Reachability Cost

Simulated Handoff Results

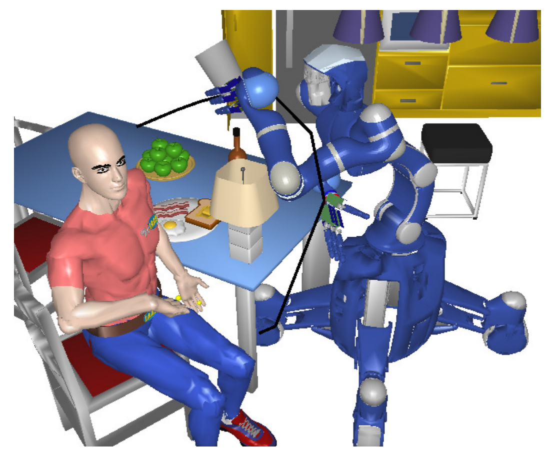


Simulated Handoff Results





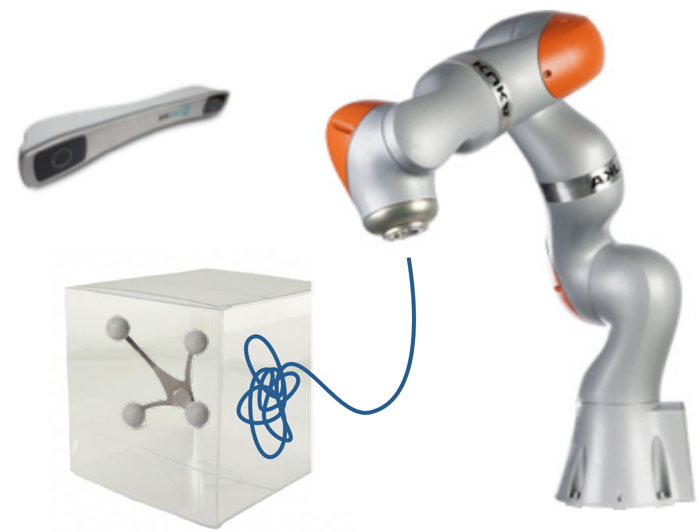
(a) RRT path



(b) Costmap path

Application to our Project

- Our end effector position is tied to the current point along the cut
- One redundant joint allows free elbow placement which could be optimized based on human interaction constraints.
- However, the runtimes of their planning algorithm are around 8 seconds, so this may not work directly in our use case.

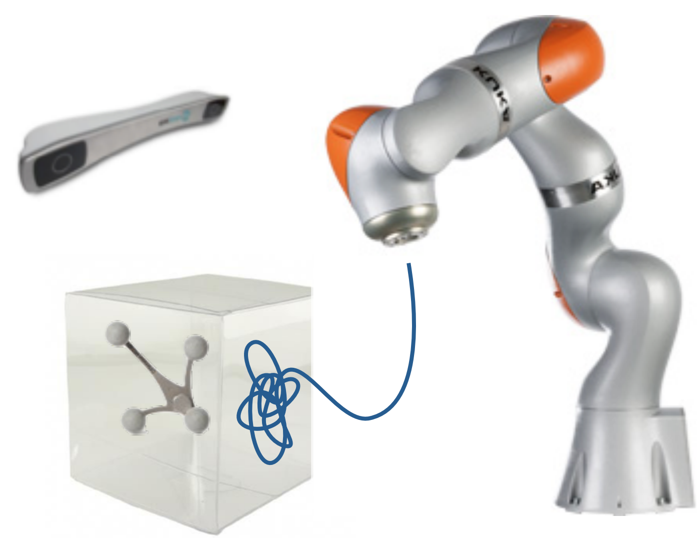


Application to our Project

- One possibility is at the start and completion of the arm's milling operation the arm must engage and disengage from the bone.

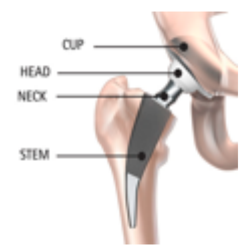
- These paths can potentially be planned at a high level using an algorithm such as this one

- Constraints can be evaluated based on comfort and safety metrics given the location and position of the surgeon, patient, etc.



Kroger, reflexxes motion libraries motivation

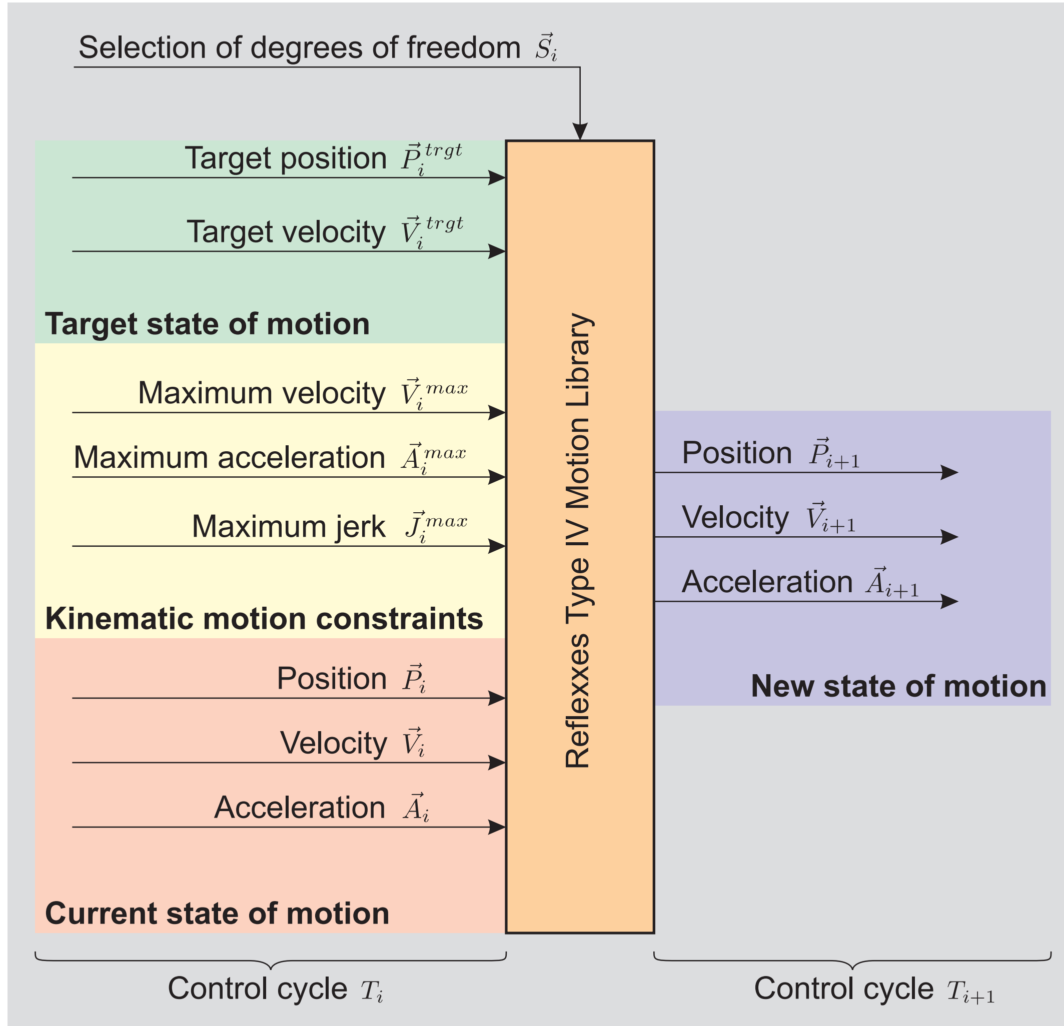
- Evaluating approaches to handling the additional degree of freedom in the KUKA iiwa arm
- Methods underlying the reflexxes motion library Type IV provide a possible option for our use case.



Reflexxes Motion Planning Library

The key to the algorithms are that they solve for the *only* minimum time solution to the planning problem.

Therefore, reflexxes finds a closed form solution even in devices with redundant degrees of freedom.

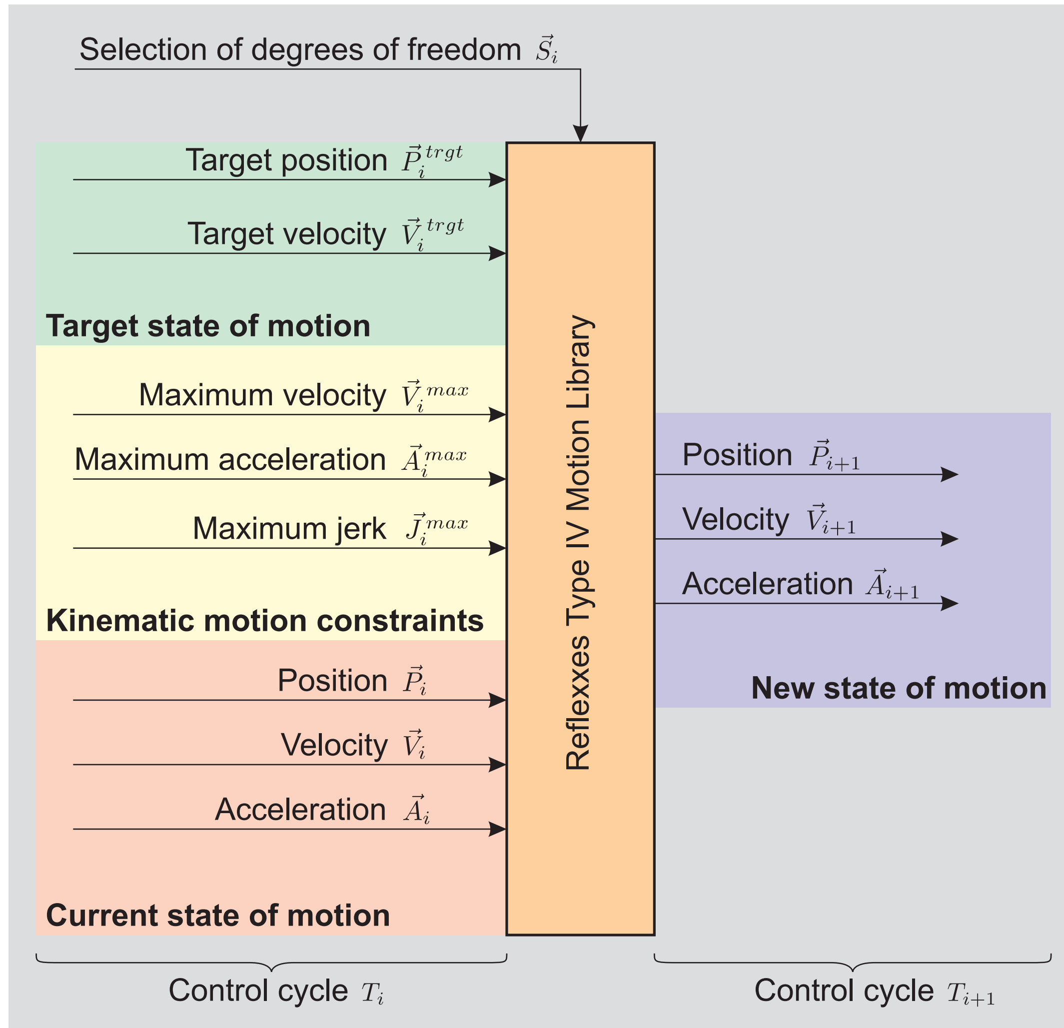


Reflexxes Motion Planning Library

Unfortunately, the algorithm solutions themselves are not presented in this paper.

Some of the underlying algorithms, information, and implementation may be available via the book written by Thorsten Kroger, On-Line Trajectory Generation in Robotic Systems[3].

This review will focus on capabilities over algorithms in accordance with the paper.



Integration of Online Trajectory Generation (OTG)

Several new capabilities are advocated with the Reflexxes Motion Planning Libraries:

A. Unforeseen switchings of coordinate frames

B. Unforeseen switchings of control state spaces

C. Deterministic, instantaneous reactions to sensor signals

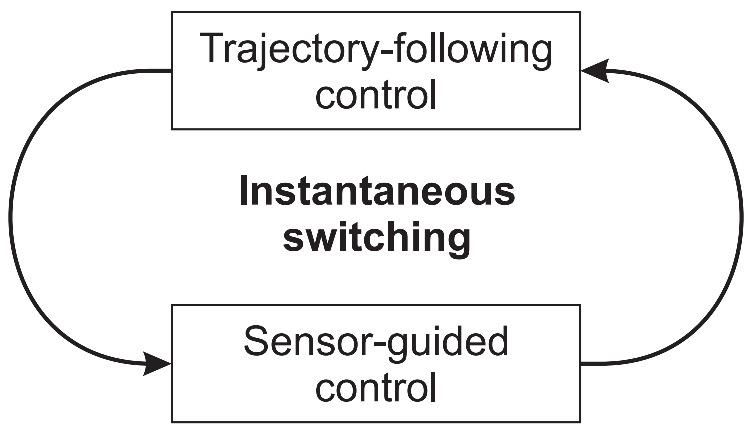
D. Safe and stable reactions to sensor failures

E. Simple visual servo control

F. Stable switched-system control

Transfer of Control

Immediately after the switching event, the transfer function of the respective controller generates command variables for lower-level controllers.



Immediately after the switching event, the **Reflexxes Motion Library** provides a feasible, continuous, and jerk-limited motion trajectory.

Transfer of Control: Reference frames

This example illustrates a switch in control frame based on a sensor event. For instance, if a camera used for visual servoing were to be blocked, control of motion would immediately need to transfer to a different device with a different frame of reference in a smooth and reliable manner.

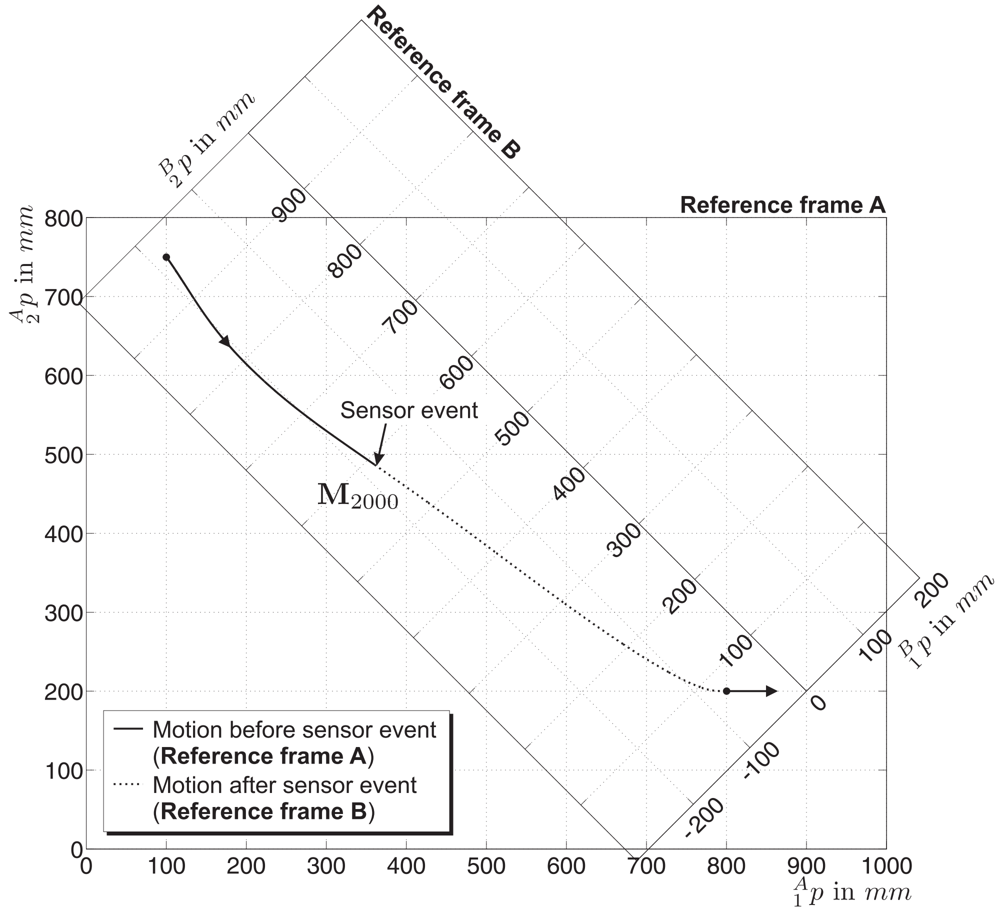


Fig. 2. XY-plot of a two-DOF *path*, whose trajectory is executed w.r.t. reference frame A (solid line). Right after the indicated sensor event (dotted line), frame B acts as reference frame for the motion controller.

Transfer of control: Control State Space

This example illustrates switching from cartesian to joint based control based on a sensor event like the previous slide.

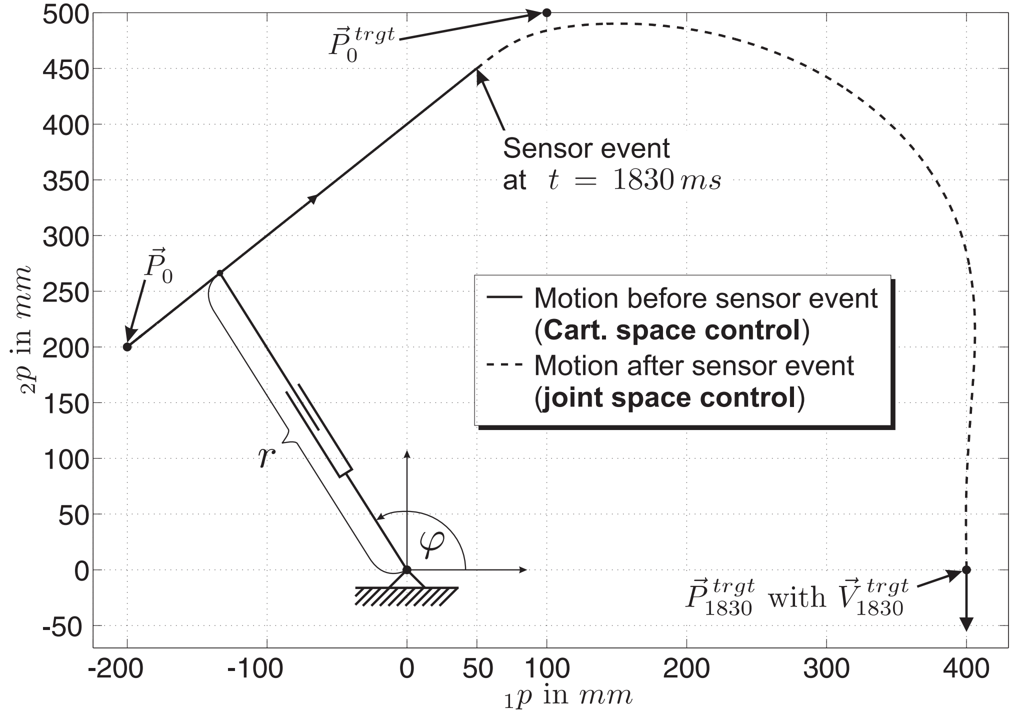
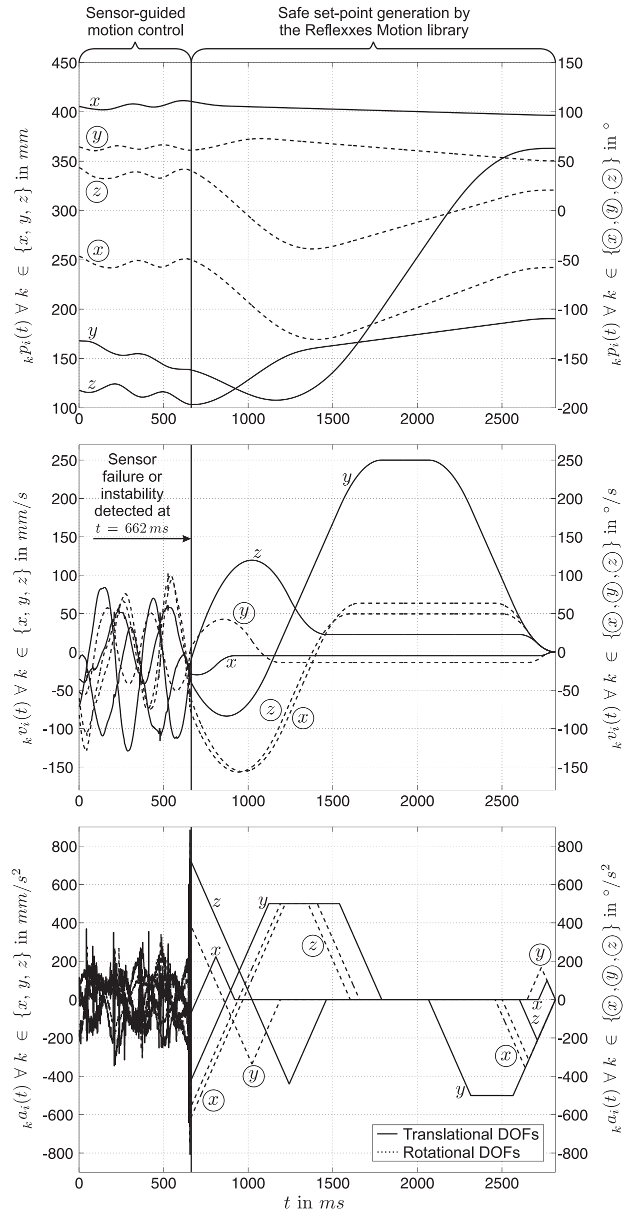


Fig. 3. An r-φ-manipulator, which executes a Cartesian motion command, that is unexpectedly interrupted at the position ${}^{Cart}\vec{p} = (50,\, 450)\,mm$ (solid line). Immediately after the interruption, the motion is controlled in joint space (dashed line) until the target state of motion has been reached.

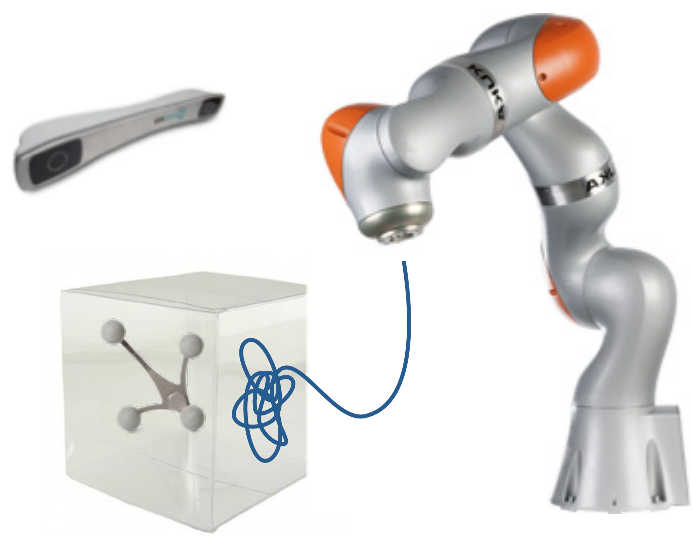
Switching between sensor-guided and trajectory-following motions



"Position, velocity, and acceleration progressions of a six-degree-of-freedom robot using a Cartesian motion controller. At t = 662ms, a sensor failure is detected, and the controller immediately switches from sensor-guided to trajectory following control in order to keep the system stable (cf. Fig. 6 and [8]). "

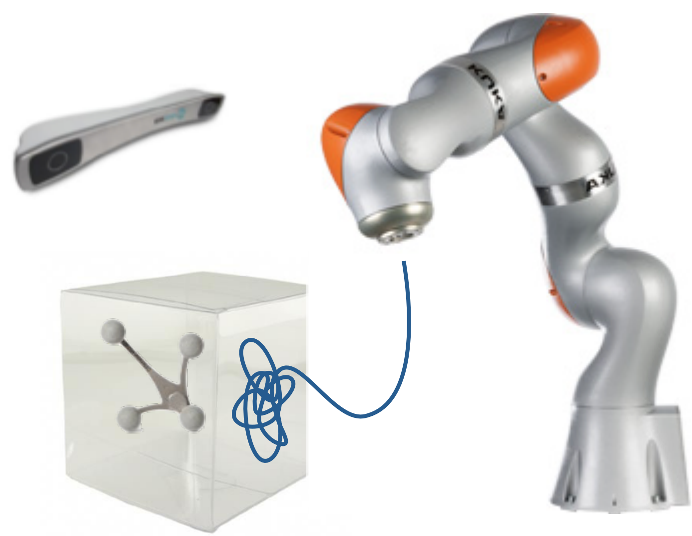
Application to our Project

- Key concerns:
  - Handling the additional degree of freedom in the KUKA iiwa arm
  - Implementing a tight control loop with an optical tracker.
- The underlying implementation of Reflexxes motion library Type IV provides a viable option for our use case



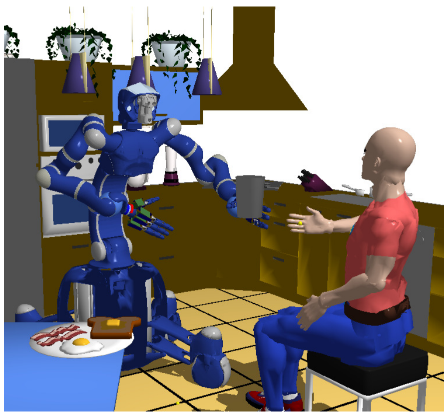
Application to our Project

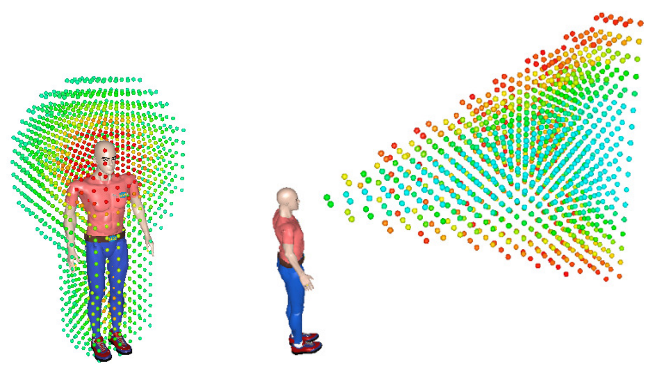
- Key concerns for our project:
  - Handling the additional degree of freedom in the KUKA iiwa arm
  - Implementing a tight control loop with an optical tracker.
- The underlying implementation of Reflexxes motion library Type IV may provide a viable option for our use case.
- We are using reflexxes via the V-REP simulation suite, which uses it directly.
- However, the library is no longer available directly because the company was sold to Google.



Questions?

- 3. Kroger T. *On-line trajectory generation in robotic systems*. First ed. Berlin, Heidelberg, Germany: Springer; 2010.





(a) Distance costmap (b) Visibility costmap