# Predictive Planning with Self-Explored Push Dynamics

Lars Henning Kayser

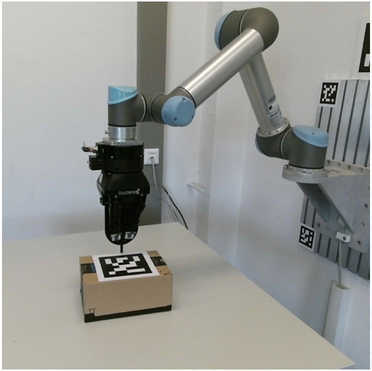
University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
**Technical Aspects of Multimodal Systems**

October 30, 2018

# Outline

# Introduction

**Goal:** Push objects on a surface to a goal position, while avoiding collisions and local optima.



**Approach:** Learn a forward dynamic model and generate paths with a sampling-based motion planner.

Considered pushes are. . .

1. *quasi-static* (inertia is neglected due to low velocity).
2. executed with a single contact point.
3. applied to rigid objects.
4. planar (objects don't roll over).

# Push Mechanics

A theoretical model is given by Mason et al [4].

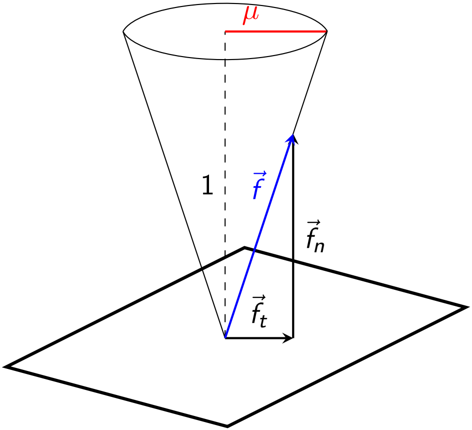
Essential factors are:

- Support force distribution of the object
- Support friction during the push
- Contact force and friction between pusher and object

Friction forces are determined by *friction coefficients*.

# Friction Cone

Model for the friction of a single point on a surface:




Friction coefficient $\mu$ determines friction force $\vec{f}$ which is composed of normal force $\vec{f}_n$ and tangential force $\vec{f}_t$.

# Friction Coefficients

They depend on materials, but also on surface structure, heat, humidity...

**Static Coefficient**:

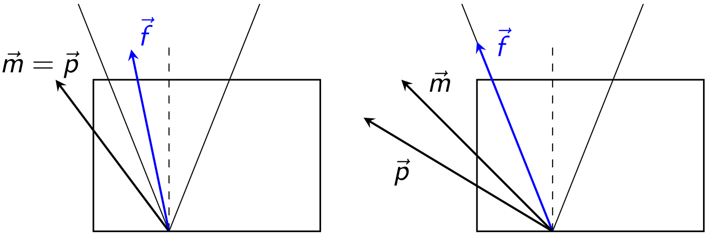
- Factor for motionless friction
- Limits the possible lateral force before sliding occurs

**Dynamic Coefficient**:

- Factor for friction during sliding

# Push Force

The friction cone also applies for the friction of the push contact.




- If $\vec{f}$ is within the cone, the motions of contact point $\vec{m}$ and pusher $\vec{p}$ align
- Otherwise, the pusher slides

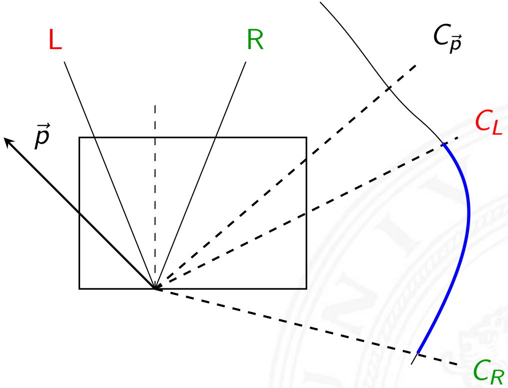
# Object Translation

**Linear translations** occur, if the push force points directly through the *Center of Friction*.

**Non-Linear translations** can only be solved numerically as part of rotations.

# Object Rotation

Rotations are described by their instantaneous Center of Rotation.



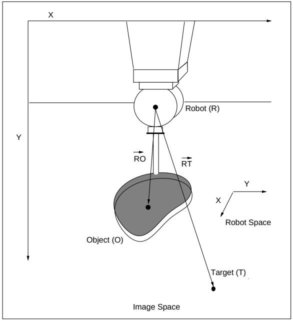

A plot of rotation centers can be constructed numerically.
The solution lies on the perpendicular to the push force.

# Method Limitations

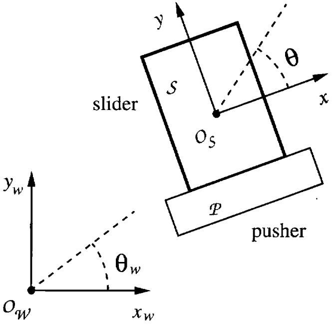
- Exact friction coefficients are unmeasurable in practice
- Methods only apply to pushes at surface level
- Inaccuracies in execution or setup calibration

$\rightarrow$ Many approaches learn friction models implicitly.

# Space Restriction



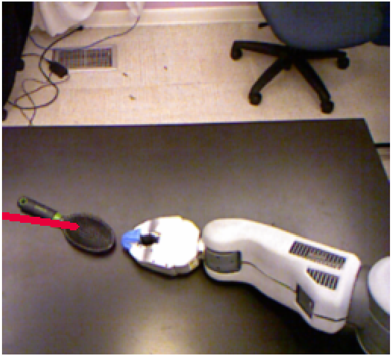

1993 - Salganicoff et al [6]

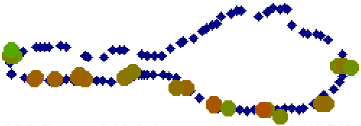



1996 - Lynch et al [3]

# Shape Functions

2013 - Hermans et al [2]

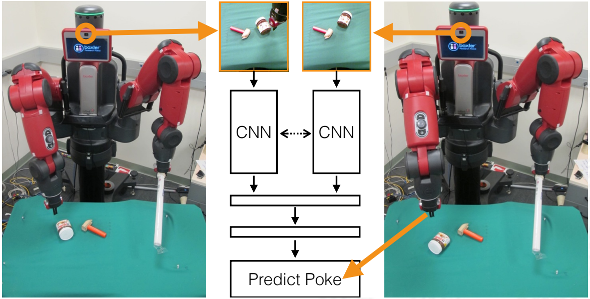




- Learning of effects and stability based on contact point
- Dynamics of unknown objects can be predicted by their shape

# End-to-End Learning

## 2016 - Agrawal et al [1]




- Deep learning of forward and inverse dynamic models
- 100k pokes executed with Baxter
- Greedy approach to reach goal state

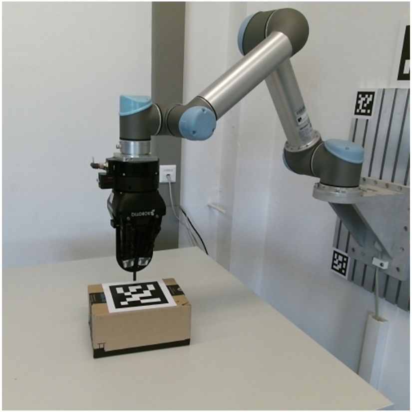
# Approach

*Learn a forward push model and use it for predictive planning.*

## Challenges

1. Autonomously explore pushes to collect samples
2. Generate forward push models
3. Implement a suitable planning strategy
4. Execute push plans

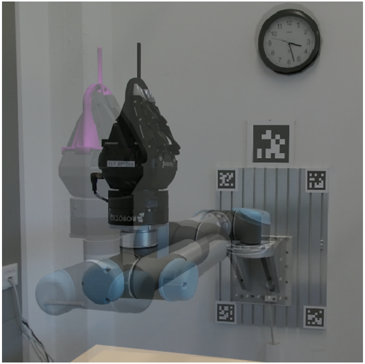
# Setup




- UR5 with Robotiq 3-finger adaptive gripper
- 3d-printed pusher tool
- Kinect2 and AprilTags2 for object localization

# Setup Accuracy

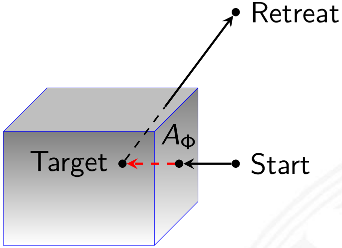
A critical factor is the camera localization.



The endeffector accuracy was increased to about 5mm by:

- upgrading to AprilTags 2
- using the mount plate for bundle detection

# Push Execution



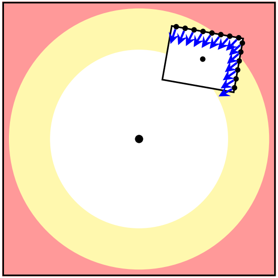

## Assumptions

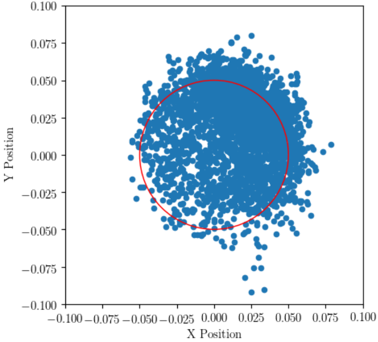
- The push movement is linear with continuous velocity
- Pusher and object stay in contact during a push
- The push movement directs the contact force
- The contact force is continuous

# Exploration

| Variable | Protocol |
|---|---|
| $A_\Phi$, $\vec{n}_\Phi$ | $U(0,1)$ * shape |
| $\vec{v}_\Phi(\beta_\Phi)$ | $U(-0.5, 0.5)$ rad |
| $d_\Phi$ | $U(0.5, 3)$ cm |

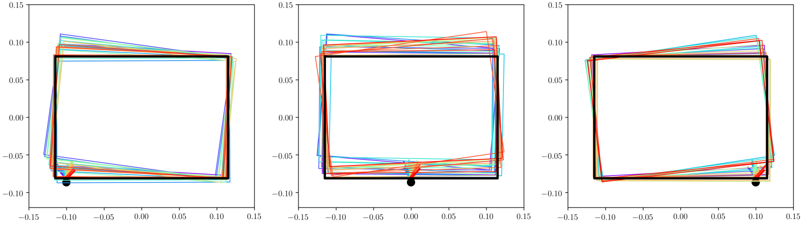
(3000 samples)



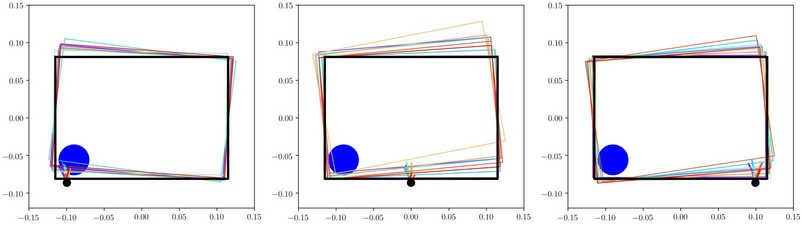



Restricted sampling keeps the object on the table.

# Sample Data



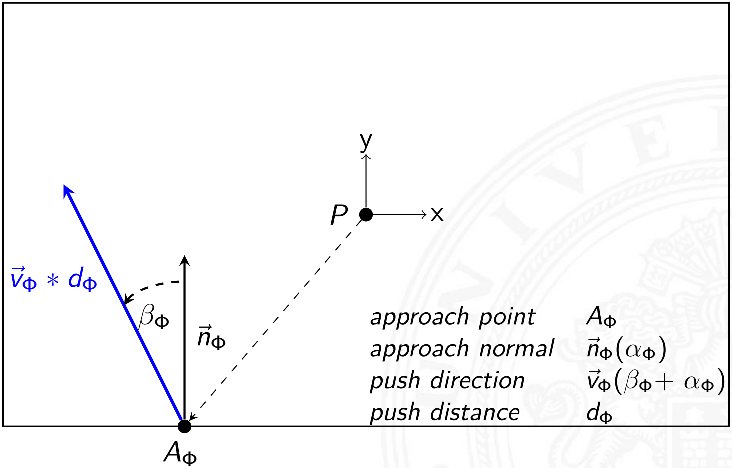

(a) Box transforms from different approach points




(b) Box transforms with weight attached to the object

# Push Specification

Pushes are defined in the object frame:



# Forward Push Model

## Learning Function

Function $p$ predicts the transformation $T$ given a push $\Phi$:

$$p : \Phi \xrightarrow{\approx} T$$

$$\equiv$$

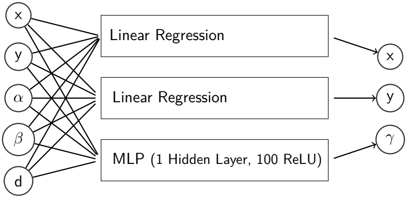
$$\langle x_\Phi, y_\Phi, \alpha_\Phi, \beta_\Phi, d_\Phi \rangle \xrightarrow{\approx} \langle x_T, y_T, \gamma_T \rangle$$

## SE(2) Loss

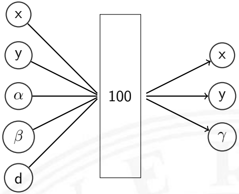
$$L_{SE(2)} = \sqrt{L_x^2 + L_y^2} + 0.5 \cdot L_\gamma$$

# Prediction Architectures

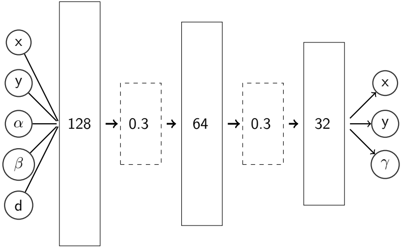
$B_0$:




$N_1$:



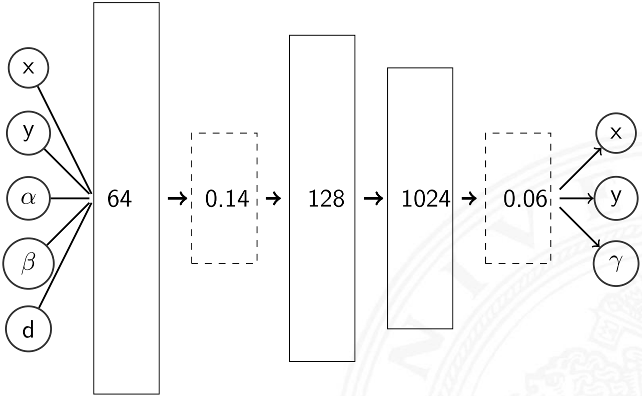

$N_2$:




## Hyperopt:

| Parameter | Domain |
|---|---|
| optimizer | *Adam, Nadam, RMSProp* |
| learning rate | 0.001 * $\log U(-0.5, 0.5)$ |
| L2 weight | 0.0007 * $\log U(-1.3, 1.3)$ |
| input activation | *linear, tanh, relu* |
| hidden layers | 1 to 4 |
| *per layer* | |
| - units | $2^{\,qU(4,10)}$ |
| - dropout | $U(0.0, 0.5)$ |
| - activation | *Linear, Tanh, ReLu* |

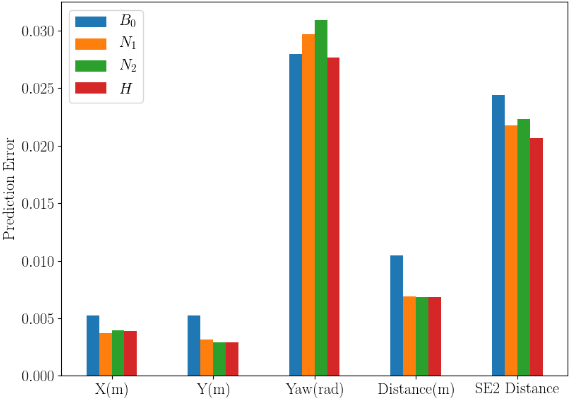
# Hyperopt Result




| | |
|---|---|
| Optimizer: | Adam |
| Learning Rate: | 0.00061 |
| l2 regularization: | 0.00021 |

# Prediction Accuracy

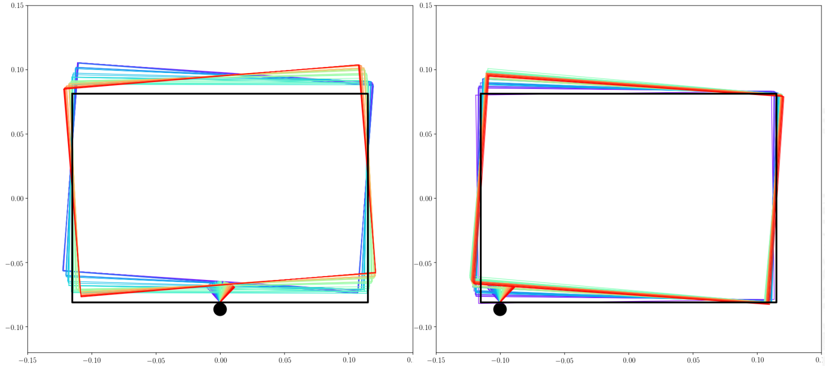
Prediction Error after 20 Epochs of 500 iterations (100 validation).




The Hyperopt-model overfits on the validation set!

# Push Prediction






Example predictions of predictor $N_2$.

# Planning Problem

*Find a sequence of pushes that produce a path of object poses from start to goal.*

This is a motion-planning problem with the following domain:

| | |
|---|---|
| State Space: | SE(2) |
| Control Space: | Pushes |

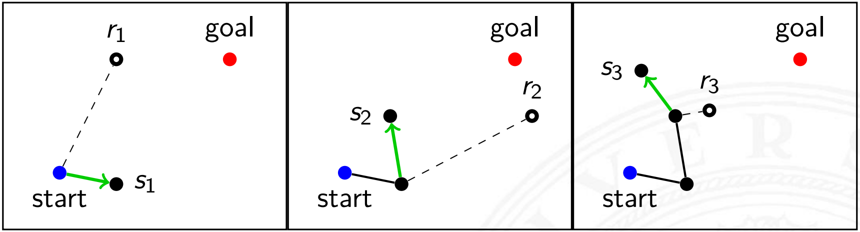
All planners are implemented using the Open Motion Planning Library (OMPL)[5].

# Sampling-based Planning (RRT)

## RRT Growth

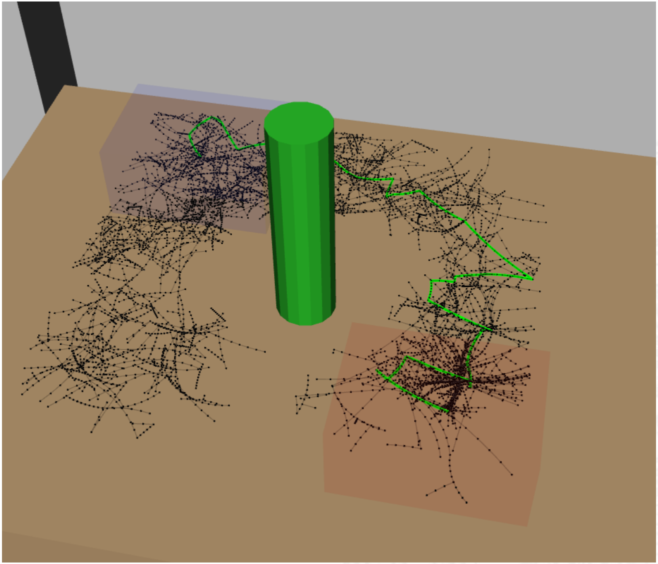
1. Sample random (goal biased) pivot state
2. Find closest existing state
3. Generate control and duration
4. Add visited states to planning tree
5. Terminate if goal state is reached

# Random Controls



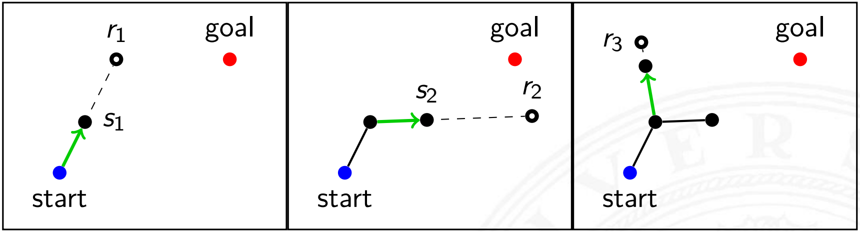

- Push controls are random and not directed towards pivot state
- Goal-biased state sampling *pulls* the tree towards the goal

# Random Controls



# Directed Controls

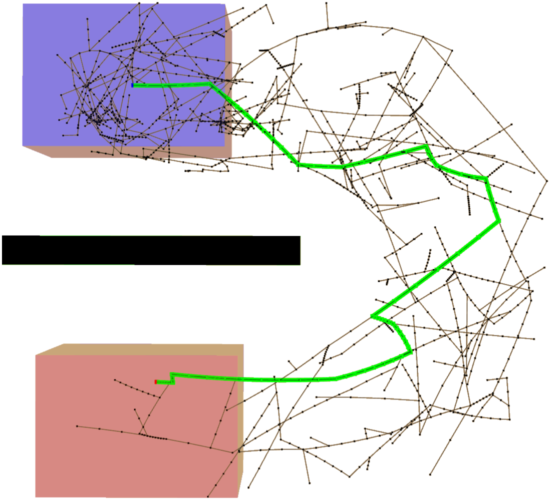




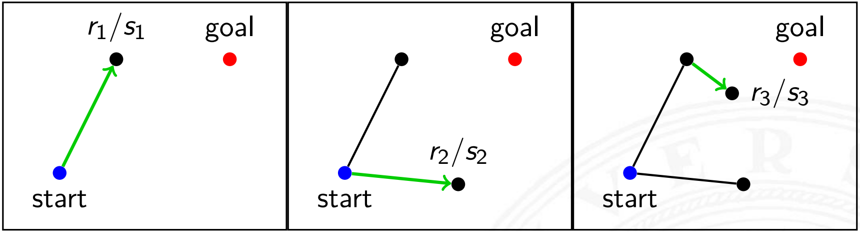

- Push controls are directed towards the pivot state
- Controls require sampling with distance minimization

# Directed Controls





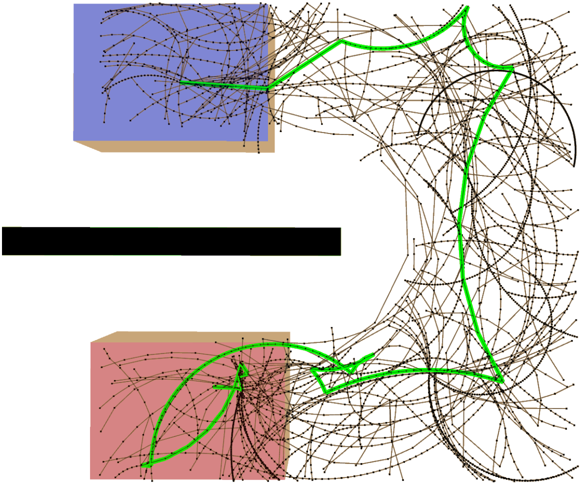
# Steered Controls



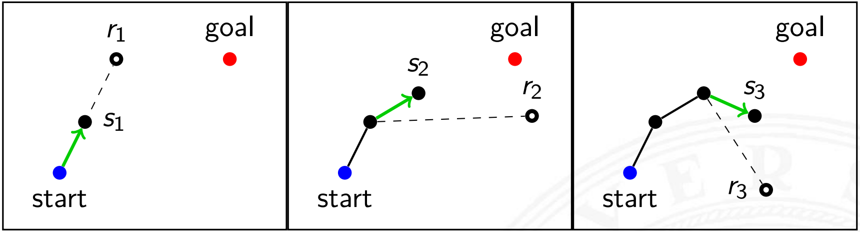

- Pushes are sampled that reach the pivot state
- This requires additional computation of the control duration
- Controls require distance minimization + repeated propagation

# Steered Controls



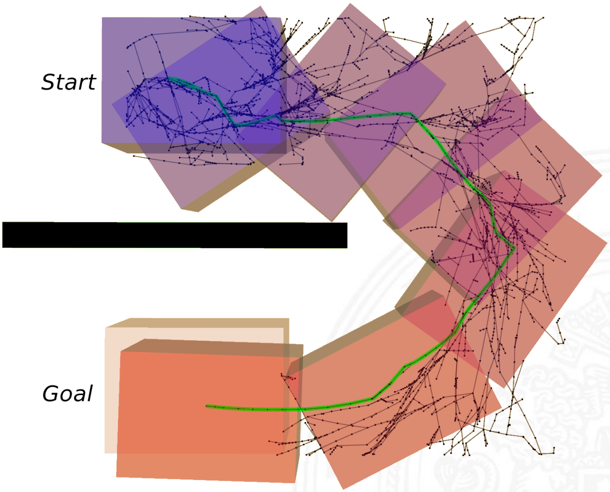


# Chained Controls



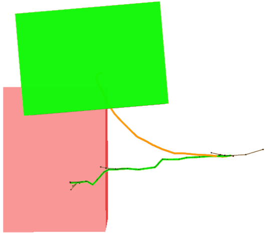

- Contact points are sampled within vicinity
- Adjacent pushes are *similar* which leads to smoother paths
- Reduced search space allows efficient distance minization sampling

# Chained Sampling



# Execution Method

Open-loop execution causes error accumulation and object deviation.



(20 pushes with an average rotation error of 4°per push)

## Model-predictive control

Execute plan as long as the object is on path, otherwise replan.

# Conclusion

## MPC Execution:

+ Feasible approach to avoid local minima
+ Can correct inaccurate predictions

- Long planning times (about 0.5s-1s)
- Collision avoidance requires safety padding
- *Only* step-wise pushes possible

# References



[1] Pulkit Agrawal et al. "Learning to Poke by Poking: Experiential Learning of Intuitive Physics". In: *CoRR* abs/1606.07419 (2016). arXiv: 1606.07419. URL: http://arxiv.org/abs/1606.07419.

[2] T. Hermans et al. "Learning Stable Pushing Locations". In: *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EPIROB*. 2013. URL: http://www.ias.tu-darmstadt.de/uploads/Team/TuckerHermans/hermans-icdl2013.pdf.

[3] Kevin M. Lynch and Matthew T. Mason. "Stable Pushing: Mechanics, Controllability, and Planning". In: *The International Journal of Robotics Research* 15.6 (Dec. 1996), pp. 533–556. DOI: 10.1177/027836499601500602.

[4] Matthew T. Mason. "Mechanics and Planning of Manipulator Pushing Operations". In: *The International Journal of Robotics Research* 5.3 (Sept. 1986), pp. 53–71. DOI: 10.1177/027836498600500303.

[5] Physical and Biological Computing Group - Department of Computer Science - Rice University. *OMPL - The Open Motion Planning Library*. 2018. URL: https://ompl.kavrakilab.org/ (visited on 07/17/2018).

[6] Marcos Salganicoff et al. "A Vision-Based Learning Method for Pushing Manipulation". In: *IRCS TECHNICAL REPORTS SERIES*. 1993.