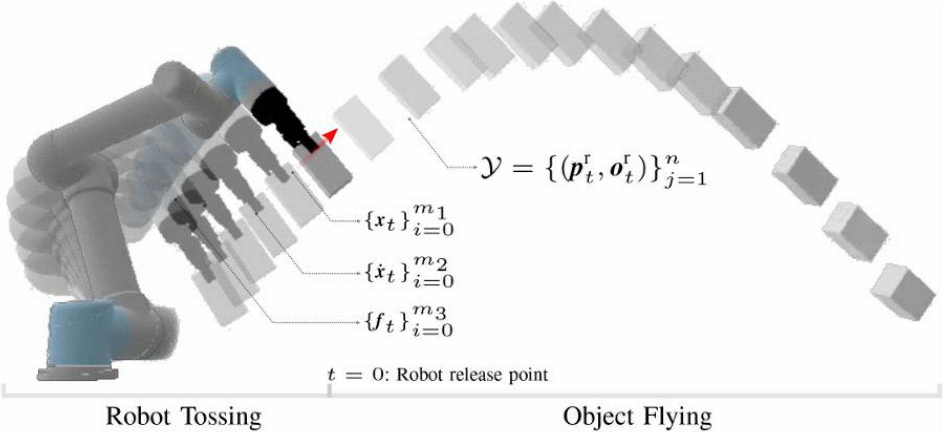
Evaluation of Robotic Systems for Prehensile Throwing Task



Picture: L. Chen et al., IEEE TRO 2024

Lukas Sommerhalder

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

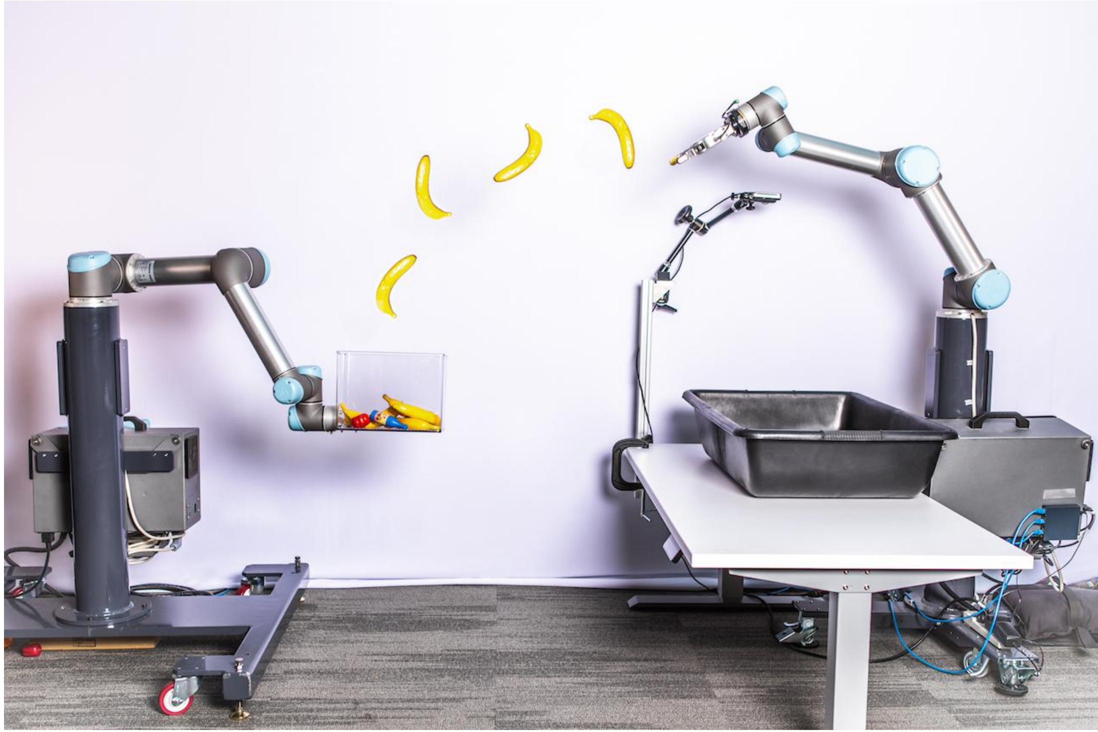
27.01.2026

Agenda

1 Introduction
2 Related Work
3 Delta Robot
4 6-DoF Robotic Arm
5 Questions & Discussion

1

Introduction



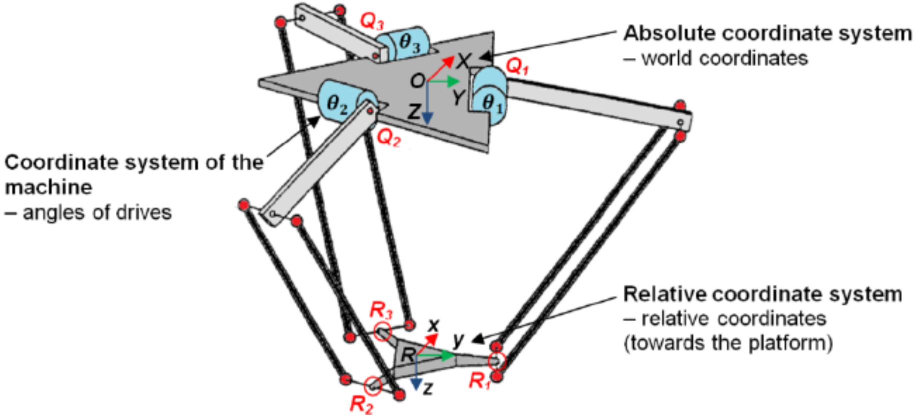
Picture: A. Zeng et al., RSS 2019

Goals

- Provide answers/insights to the following:
  - Prehensile Throwing vs. Non-Prehensile Throwing
  - Current Research
  - Prehensile Throwing with Delta Robot
  - Prehensile Throwing with Robotic Arm

Delta Robots

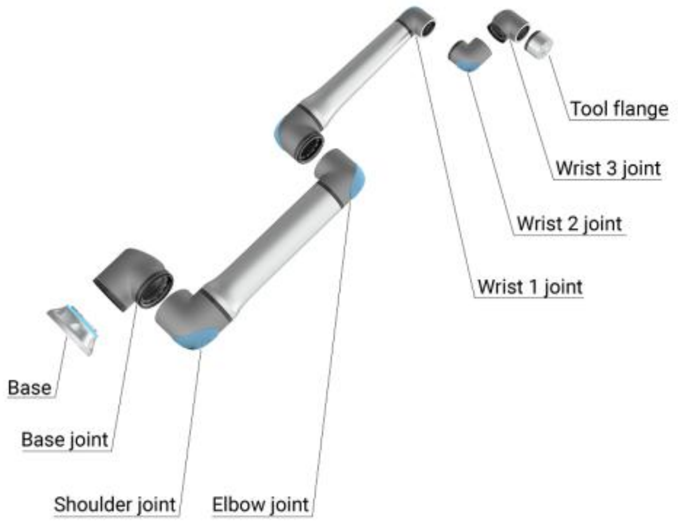
Picture: https://www.abb.com/global/en/areas/robotics/products/robots/delta-robots/irb-360



Picture: M. Opl et al., DELTA - Robot with Parallel Kinematics, Springer 2014

Articulated Robotic Arms

Picture: TAMS UR5 Setup, UHH 2025



Picture: https://www.universal-robots.com/manuals/EN/HTML/SW5_20/Content/prod-usr-man/complianceUR20/comp-introduction/comp-preface.htm

Non-Prehensile Throwing

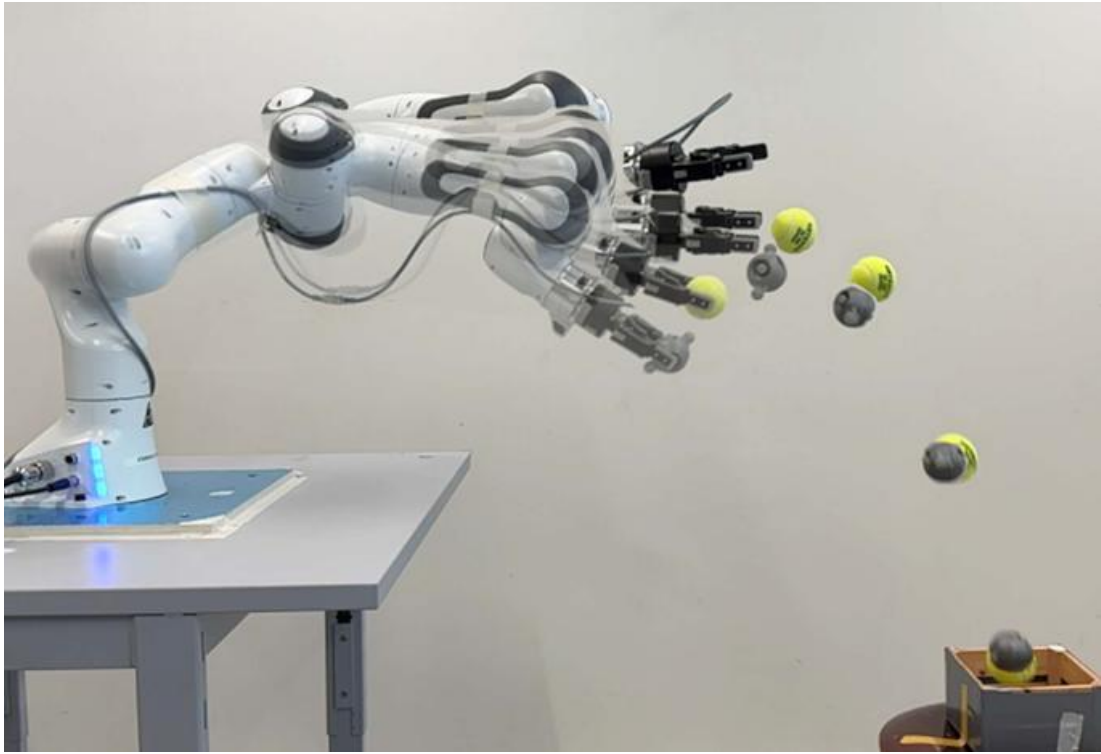
- Object motion is controlled without grasping.
- Examples include juggling, pitching, and batting.
- Typically limited to certain object geometries (spheres or flat objects).
- Relies on dynamic interaction rather than secure holding.
- Simpler release than grasp-based throwing due to instantaneous release.
- While release is simpler, to keep the object in place non-prehensile throwing can require:
  - More accurate dynamic modelling
  - Careful trajectory and contact force planning

Prehensile Throwing

- Object motion is controlled using an active grasp.
- Examples include pick-and-throw for material sorting.
- Applicable to a wide range of object shapes and sizes.
- Requires stable grasping and precise release timing.
- Complex to implement due to gripper dynamics, and release delay.

2

Related Work



Picture: Y. Liu and A. Billard, IEEE TRO 2024

Tossingbot (A. Zeng et al, Tossingbot: Learning to throw arbitrary objects with redsidual physics, RSS 2019)

GRASPING AND THROWING PERFORMANCE IN REAL (MEAN %)

	Grasping		Throwing	
Method	Seen	Unseen	Seen	Unseen
Human-baseline	–	–	–	80.1±10.8
Regression-PoP	83.4	75.6	54.2	52.0
Physics-only	85.7	76.4	61.3	58.5
Residual-physics	86.9	73.2	**84.7**	**82.3**

- Training via self-supervision through trial-and-error.
- Beyond tossing it learns robust grasps for throwing.
- Adapts to objects of varying shapes and inertia.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

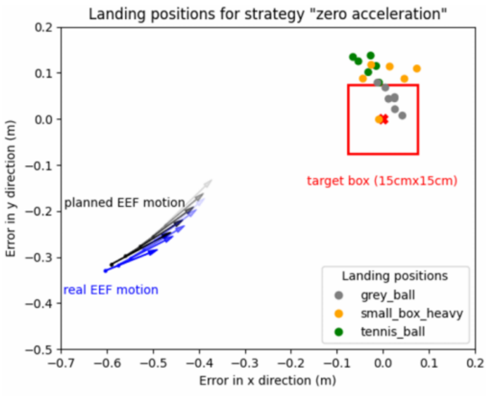
Throwing objects into a moving basket while avoiding obstacles. (H. Kasaei and M. Kasaei, ICRA 2023)

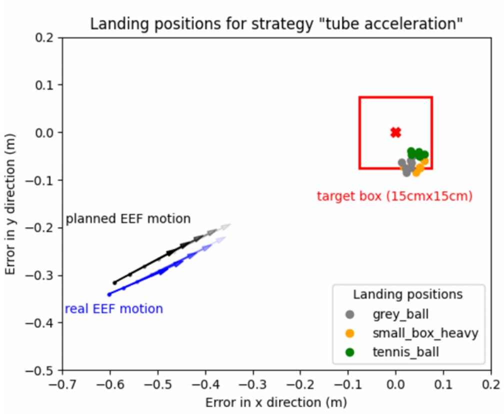
- Learning end-to-end grasping and throwing policies, similar to TossingBot but trained using reinforcement learning.
- Smaller set of thrown objects.
- Successful solutions to more challenging task variants:
  - Robust throwing with obstacles in front of the target.
  - Prediction and interception of moving targets

Picture:
https://www.youtube.com/watch?v=VmIFF__c_84

Tube Acceleration

(Yang Liu and Aude Billard, Tube acceleration: Robust dexterous throwing against release uncertainty, TRO 2024)





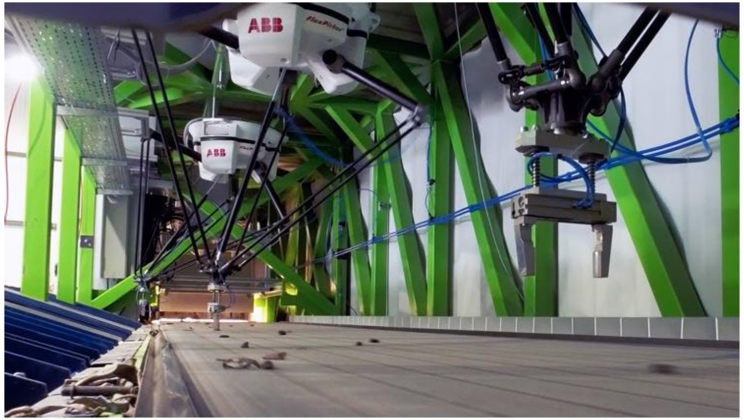
LANDING POSITION ERROR STATISTICS OF THE TWO ROBOT MOTION STRATEGIES (TUBE AND ZERO) AFTER THE NOMINAL RELEASE STATE

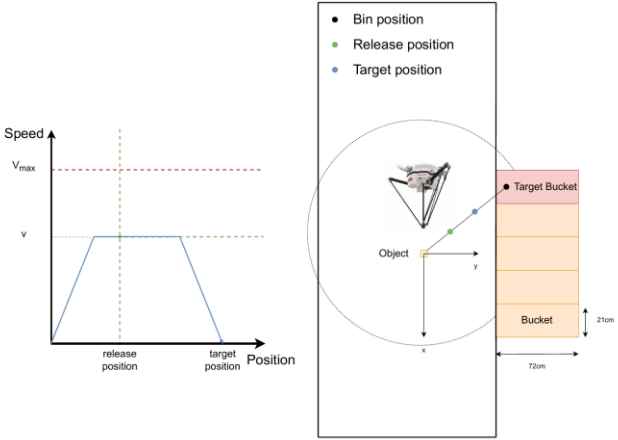
	Mean (mm)		Std. (mm)	
Object	Tube	Zero	Tube	Zero
grey_ball	74.52	**53.70**	**9.51**	15.60
small_box_heavy	**87.12**	95.30	**7.47**	44.57
tennis_ball	**66.23**	121.07	**9.55**	25.97
overall	**75.88**	88.11	**12.04**	40.74

Each strategy-object pair is repeated 5 to 6 times.
The bold means smaller landing position error Mean/Std., indicating better performance.

- Apply a constant acceleration after the gripper opens.
- Reduces scatter in the landing position of the object.
- May increase tracking error in the robotic arm trajectory

REINFORCEMENT LEARNING TO IMPROVE DELTA ROBOT THROWS FOR SORTING SCRAP METAL (Louette et al., 2024)



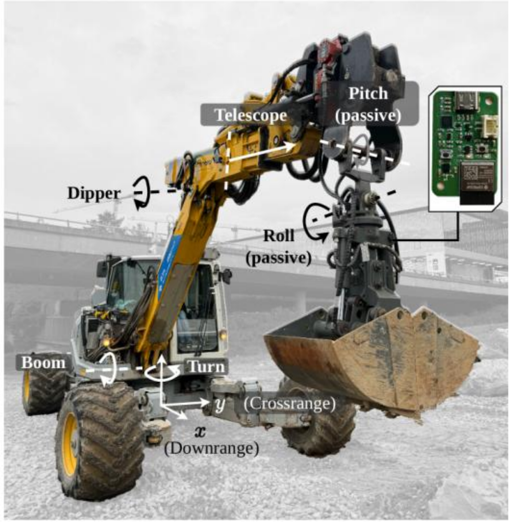


- Reinforcement learning to optimize delta robot throwing strategies.
- Evaluated in a real-world scrap-sorting scenario.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Dynamic throwing with robotic material handling machines

(L. Werner et al., IROS 2024)



- Reinforcement learning–based control of an underactuated excavation machine for dynamic throwing.
- Extended operational range beyond conventional placement.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

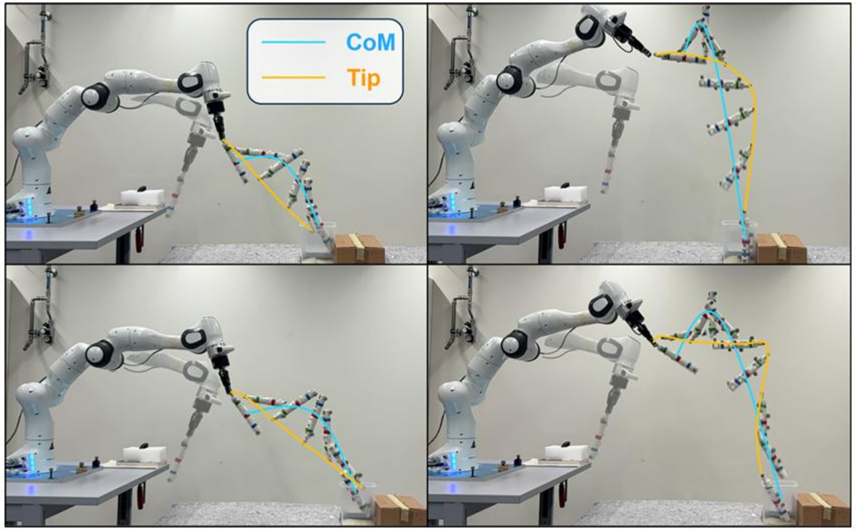
Learning to throw-flip I

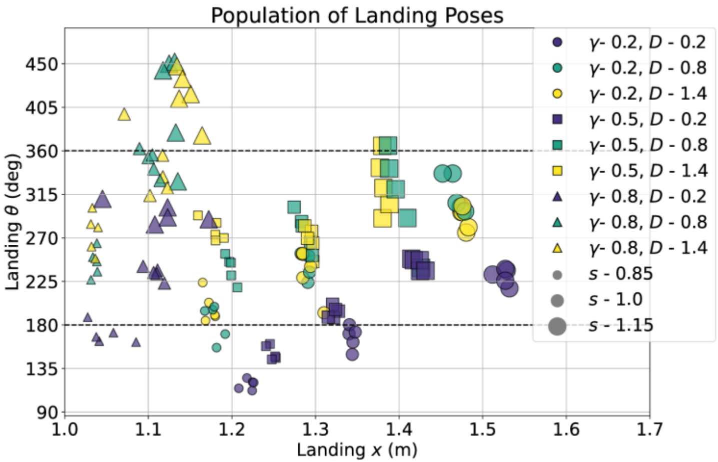
(Y. Liu and A. Billard, 2025)

- Parameterization of the throw trajectory using three variables:
  - Pitch angle
  - Speed
  - Damping
- Learning the relationship between throw parameters and the resulting:
  - Landing position
  - Final orientation
- Model-based / data-driven learning of parameter–outcome correlations
- Goal: Accurate throwing of an object to a target with a desired position and orientation

Learning to throw-flip II

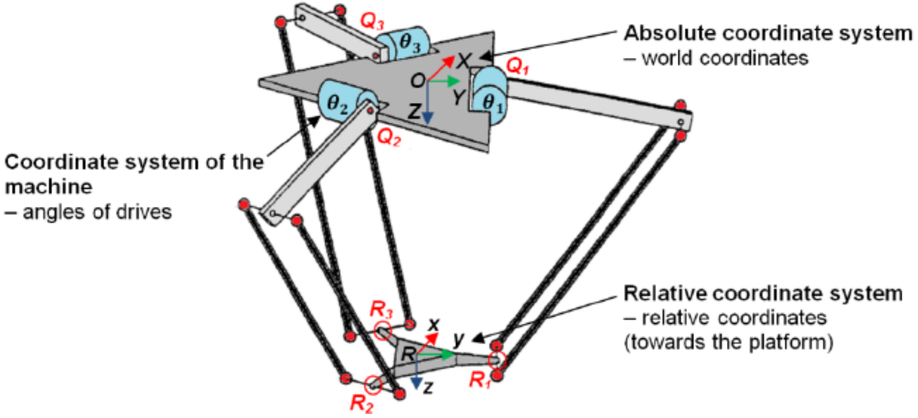
(Y. Liu and A. Billard, Learning to throw-flip, 2025)





3

Delta Robot



Picture: M. Opl et al., DELTA - Robot with Parallel Kinematics, Springer 2014

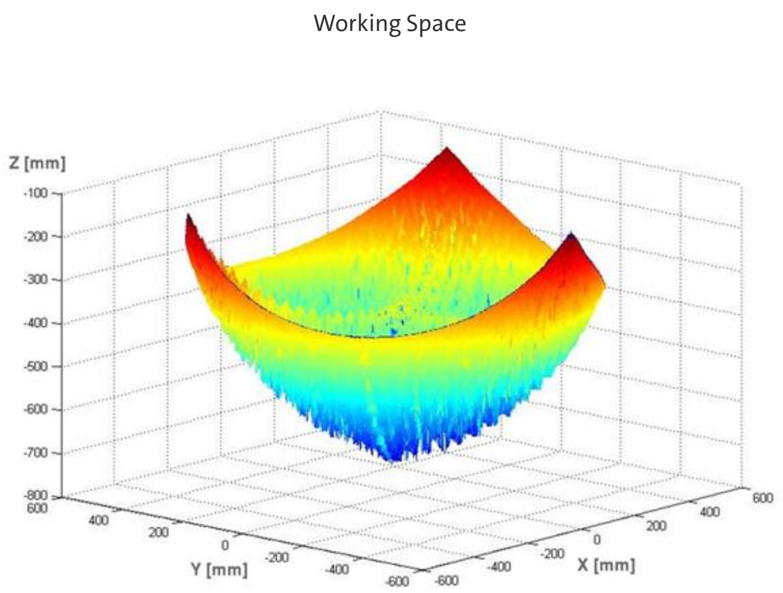
Method

- **Acceleration Phase**:
  - Straight-line motion in Cartesian space.
  - Constant acceleration in Cartesian space.
  - Variable pitch angle, increasing from 45° until a feasible target hit is achieved.
- **Release Phase**:
  - Gripper opening.
  - Horizontal velocity is kept constant.
  - Downward acceleration of 1g.
  - Duration deepends on the gripper characteristics.

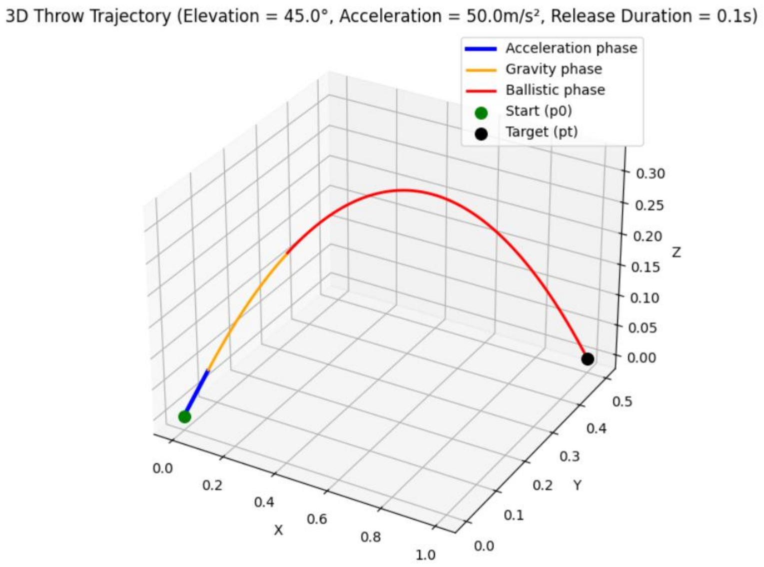
Trajectory I: Implementation

- A simple bisection search is used to determine the duration of constant acceleration, iterating over launch angles from 45° to 89° until a feasible solution is found.
- The trajectory is discretized; therefore, interpolation is applied at the first acceleration step using a reduced acceleration to exactly match the final position and velocity at the end of the acceleration phase.
- Additional constant-velocity steps are appended after the acceleration phase:
  - Horizontal velocity is held constant at the value from the final acceleration step.
  - Vertical velocity is reduced by 1 g at the end of the final acceleration step.
- Moving back to grasp next object
  - Returning to grasp the next object is not implemented, the robot simply stops.

Trajectory II: Cartesian Space

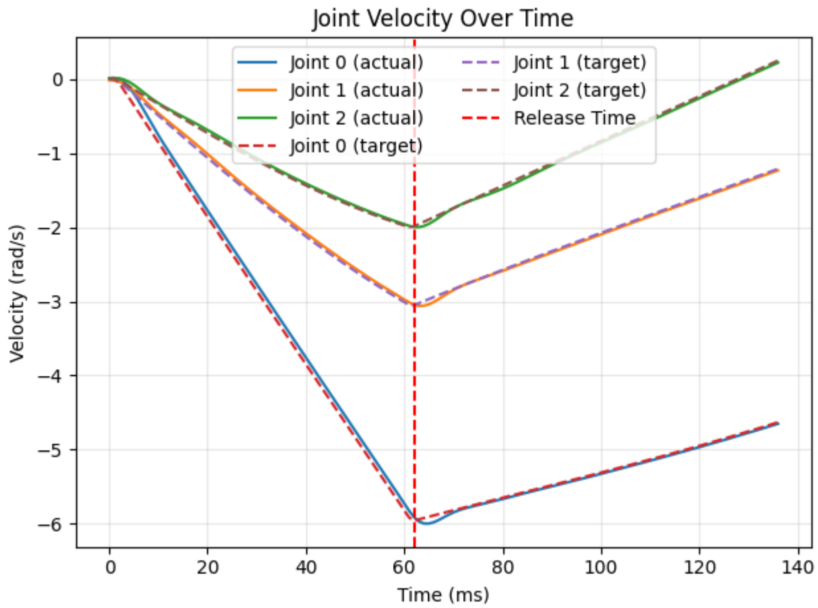


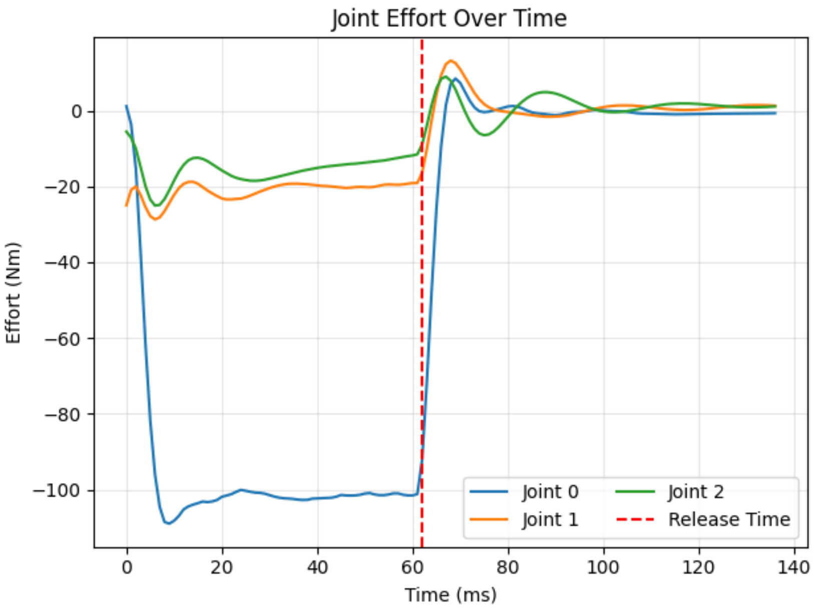
Picture: M. Opl et al., DELTA - Robot with Parallel Kinematics, Springer 2014



Picture: Example Trajectory in Cartesian Space

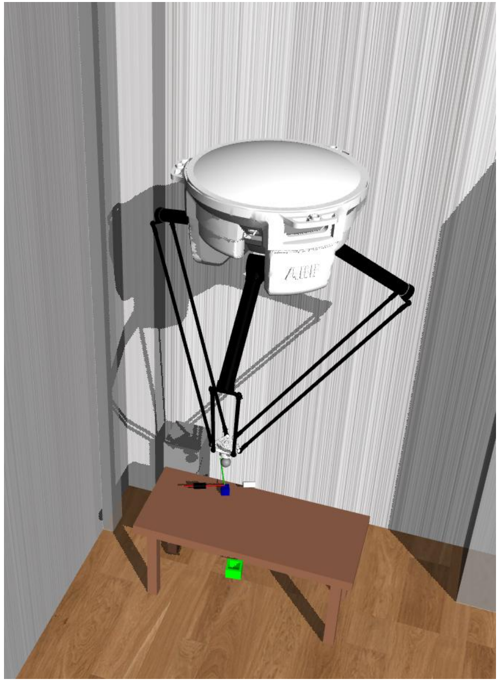
Trajectory III: Joint Space (Mujoco Simulation)

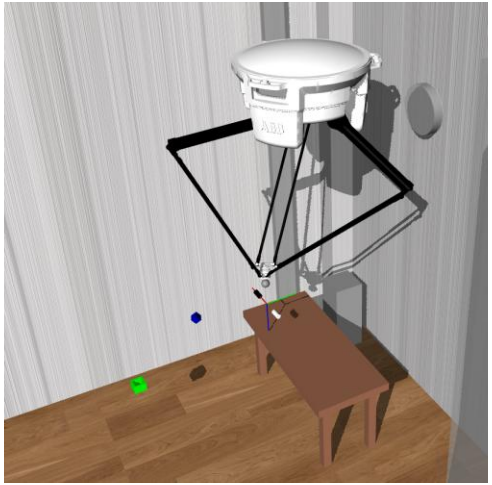


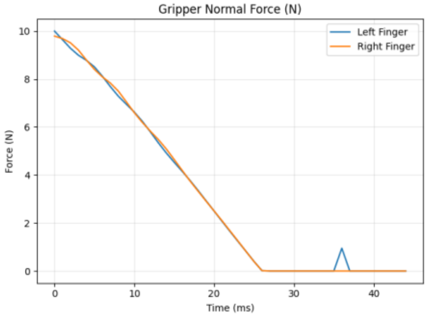


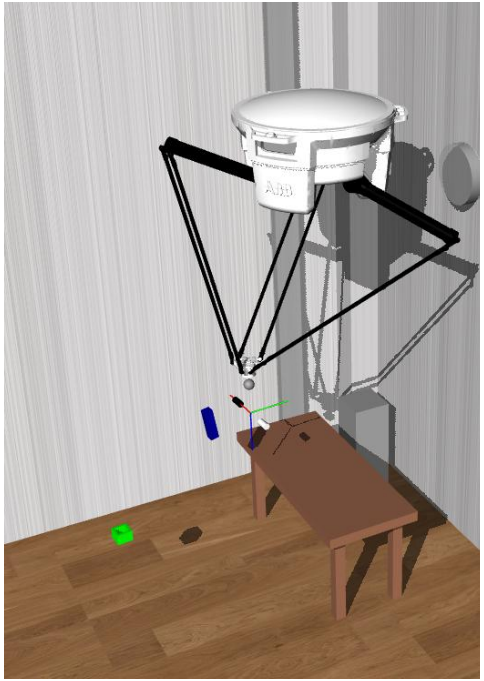
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Task Demonstration in Simulation



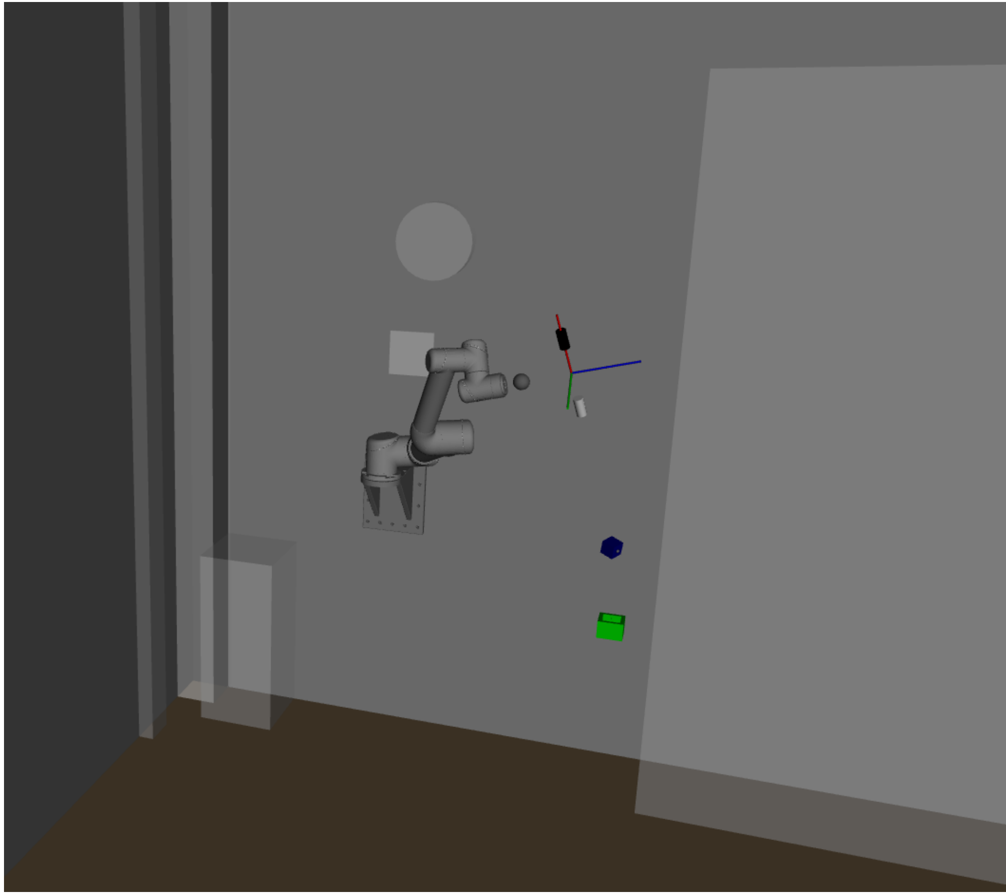






4

6-DoF Arm



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Throw Configuration

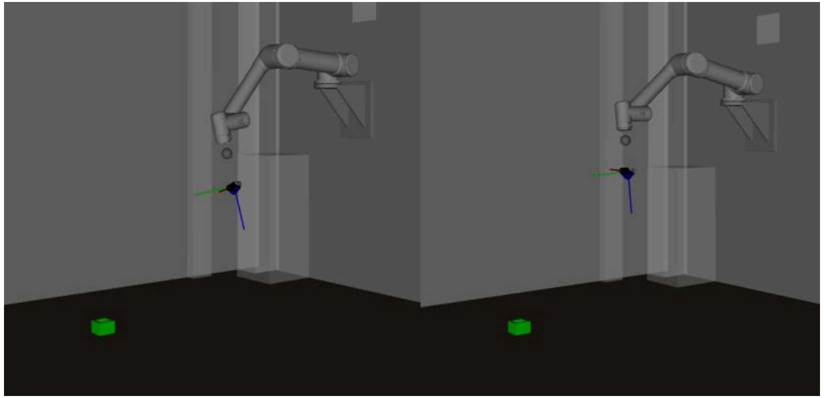
- **Objective:** Robust throwing performance while avoiding acceleration-induced tracking errors (e.g., from tube dynamics)
- **Approach:** Identify throw configurations—joint positions and velocities—that improve robustness to release-time uncertainty under constant end-effector velocity.
- **Benefit:** Enables reliable throwing without aggressive control or high-bandwidth tracking.

Method I

1. Initialize a random joint configuration $\boldsymbol{q}$.
2. Compute joint velocity $\dot{\boldsymbol{q}}$ at $\boldsymbol{q}$ to hit the target.
   - If no feasible $\dot{\boldsymbol{q}}$ is found, return to Step 1
3. Compute $\dot{\boldsymbol{q}}_{\boldsymbol{rt}}$ at $\boldsymbol{q}_{\boldsymbol{rt}} = \mathrm{q} + \dot{\boldsymbol{q}} * \mathrm{rt}$, where $\boldsymbol{rt}$ is the release time.
   - If no feasible $\dot{\boldsymbol{q}}_{\boldsymbol{rt}}$ found, return to Step 1.
4. Compute mean values:
   - $\boldsymbol{q}_{\boldsymbol{mean}} = (\boldsymbol{q} + \boldsymbol{q}_{\boldsymbol{rt}}) / 2.$
   - $\dot{\boldsymbol{q}}_{\boldsymbol{mean}} = (\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}_{\boldsymbol{rt}}) / 2.$
5. Update Configuration:
   - $\boldsymbol{q} \Leftarrow \boldsymbol{q}_{\boldsymbol{mean}} - \dot{\boldsymbol{q}}_{\boldsymbol{mean}} * (\boldsymbol{rt} / 2).$
   - Evaluate Performance:
     - Sum of distances between landing position and target position.
     - Measured at $\boldsymbol{q}$ and $\boldsymbol{q}_{\boldsymbol{rt}}$ using $\dot{\boldsymbol{q}}_{\boldsymbol{mean}}$.
   - Return to Step 2.

Method II

- The heuristic from the previous slide is applied independently at grid points with 10 cm spacing around the robot to evaluate the algorithm.
- Qualitative results will be demonstrated later in simulation.

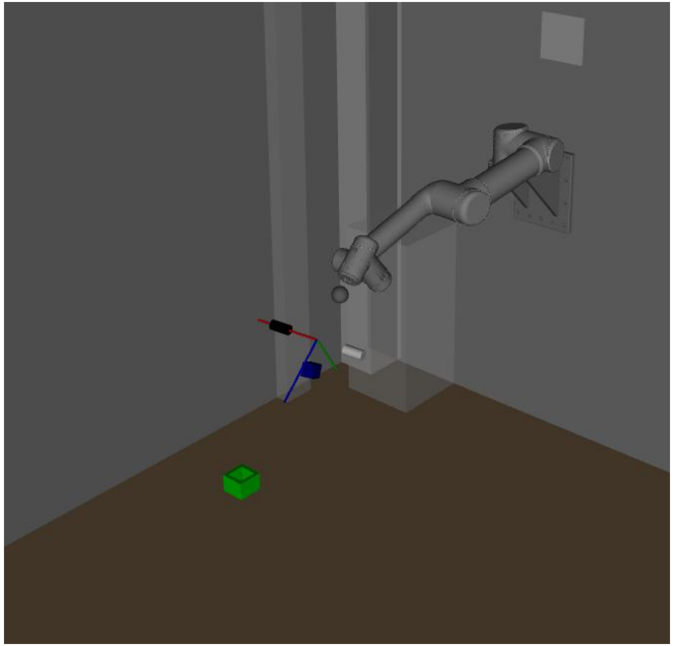


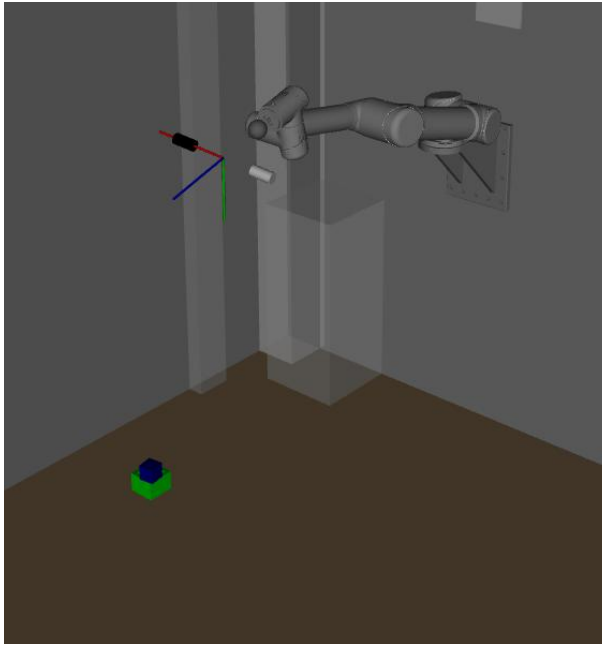
Toss to target with identical trajectories: gripper opens 30 ms before q (left), and gripper opens 70 ms after q (right); both successfully hit the target.

Ongoing Work

- Evaluate Performance metrics:
  - Distance to the target (as proposed previously).
  - Difference between $\dot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{rt}}$ to $\dot{\boldsymbol{q}}_{\boldsymbol{mean}}$.
  - ...
- Computation:
  - Evaluate the heuristic on a grid and re-run using neighbouring intersection configurations instead of random sampling.
  - Interpolate configurations between grid points.
  - Compute robust throw configurations along a line extending from the robot base and generalize via shoulder pan rotation.
  - ...
- Evaluation:
  - Compare results against related work.
  - Assess potential contributions toward improving methods such as "*Learning to Throw-Flip*" (Y. Liu & A. Billard, 2025).

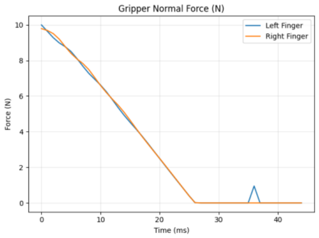
Task Demonstration in Simulation

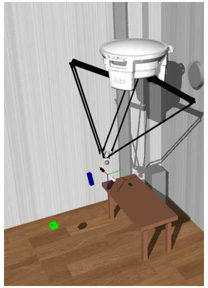


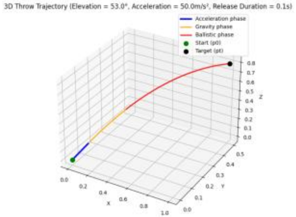


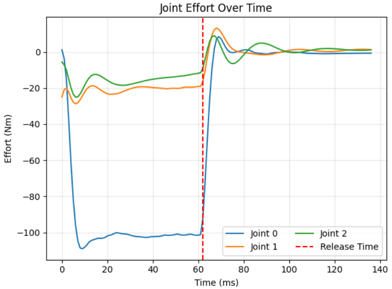
5

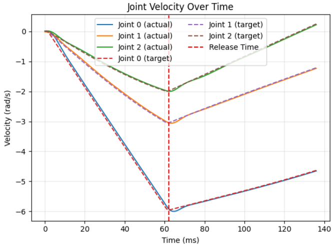
Questions & Discussion

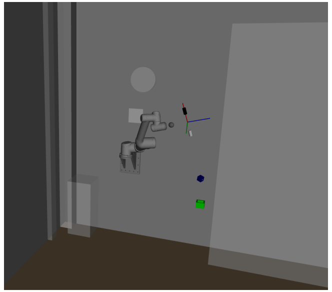


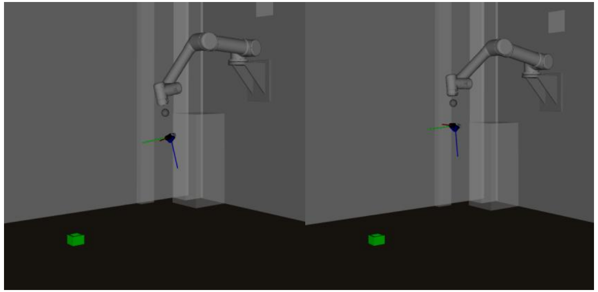












References

- Lars Berscheid and Torsten Kroeger. Jerk-limited real-time trajectory generation with arbitrary target states. In Robotics: Science and Systems XVII. Ed:: Dylan A. Shell.Robotics: Science and Systems Foundation, 2021. ISBN 978-0-9923747-7-8. https://doi.org/10.15607/RSS.2021.XVII.015.

- Lipeng Chen, Weifeng Lu, Kun Zhang, Yizheng Zhang, Longfei Zhao, and Yu Zheng: Learning to accurately measure and predict robot throwing of arbitrary objects in real time with proprioceptive sensing. IEEE Transactions on Robotics, 40:3232–3251, 2024. https://doi.org/10.1109/TRO.2024.3416009.

- Hamidreza Kasaei and Mohammadreza Kasaei. Throwing objects into a moving basket while avoiding obstacles. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 3051–3057, 2023. https://doi.org/10.1109/ICRA48891.2023.10160215.

- Seungsu Kim and Stephane Doncieux. Learning highly diverse robot throwing movements through quality diversity search. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO ’17, page 1177–1178, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349390. https://doi.org/10.1145/3067695.3082463.

- Takahiro Kizaki and Akio Namiki. Two ball juggling with high-speed hand-arm andhigh-speed vision system. Proceedings - IEEE International Conference on Roboticsand Automation, pages 1372–1377, 05 2012. https://doi.org/10.1109/ICRA.2012.6225090.

- Yang Liu and Aude Billard. Tube acceleration: Robust dexterous throwing against release uncertainty. IEEE Transactions on Robotics, 40:2831–2849, 2024. https://doi.org/10.1109/TRO.2024.3386391.

- Yang Liu, Bruno Da Costa, and Aude Billard. Learning to throw-flip, 2025. https://arxiv.org/abs/2510.10357.

- Arthur Louette, Gaspard Lambrechts, Damien Ernst, Eric Pirard, and Godefroid Dislaire. Reinforcement learning to improve delta robot throws for sorting scrap metal, 2024. https://arxiv.org/abs/2406.13453.

- Yuntao Ma, Yang Liu, Kaixian Qu, and Marco Hutter. Learning accurate whole-body throwing with high-frequency residual policy and pullback tube acceleration. 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1771–1778, 2025. https://api.semanticscholar.org/CorpusID:279464943.

- M. Opl et al. (2011). DELTA - Robot with Parallel Kinematics. In: Jabłoński, R., Březina, T. (eds) Mechatronics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23244-2_54.

- L. Rey and R. Clavel. The delta parallel robot. In C. R. Boer, L. Molinari-Tosatti, and K. S. Smith, editors, Parallel Kinematic Machines, pages 401–417, London, 1999. Springer London. ISBN 978-1-4471-0885-6. https://doi.org/10.1007/978-1-4471-0885-6_29.

- Avishai Sintov and Amir Shapiro. A stochastic dynamic motion planning algorithm for object-throwing. In 2015 IEEE International Conference on Robotics and Automation(ICRA), pages 2475–2480, 2015. https://doi.org/10.1109/ICRA.2015.7139530.

- Yinhuai Wang, Qihan Zhao, Runyi Yu, Hok Wai Tsui, Ailing Zeng, Jing Lin, Zhengyi Luo, Jiwen Yu, Xiu Li, Qifeng Chen, Jian Zhang, Lei Zhang, and PingTan. Skillmimic: Learning basketball interaction skills from demonstrations. In 2025IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages17540–17549, 2025. https://doi.org/10.1109/CVPR52734.2025.01634.

- Lennart Werner, Fang Nan, Pol Eyschen, Filippo A. Spinelli, Hongyi Yang, and Marco Hutter. Dynamic throwing with robotic material handling machines. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 98–104, 2024. https://doi.org/10.1109/IROS58592.2024.10802743.

- Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. In Robotics:Science and Systems (RSS), 2019. https://tossingbot.cs.princeton.edu/.