Karlsruhe Institute of Technology

Carnegie Mellon University

High Performance Humanoid Technologies

Automated Design of Simple and Robust Manipulators for Dexterous In-Hand Manipulation Tasks using Evolutionary Strategies

Andre Meixner
supervised by Nancy Pollard and Tamim Asfour

Institute for Anthropomatics and Robotics (IAR), High Performance Humanoid Technologies (H^2T)

Karlsruhe Institute of Technology

"Dexterity means the capability of **changing** the **position and orientation** of the **manipulated object** from a given reference configuration to a different one, arbitrarily chosen **within the hand workspace**"

(Bicchi, 2000) [1]

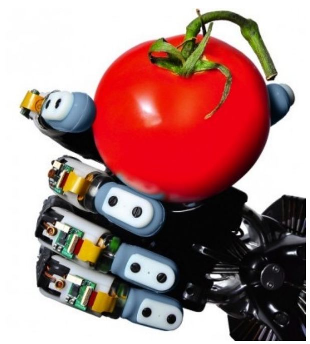
- *Advantage*
  - Cope with **limited workspace**
  - **Precise** and **efficient**
  - **Reduce power** required to accomplish a task
  - Allow **tool handling**

- Important for **general purpose** robots, **health care** and **manufacturing**, e.g.
  - Adjusting postures in-hand saves time
  - Execute complex manipulations such as humans

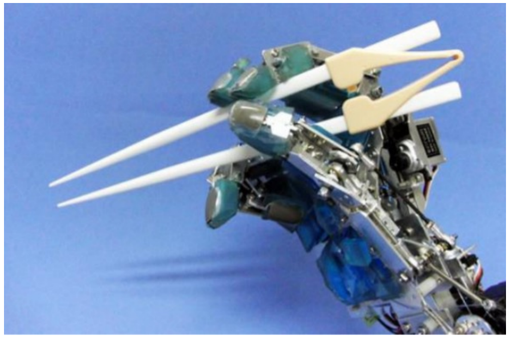
- Imitating human hands
  - Versatility/Dexterity to execute a **wide range of tasks** in **different environments**

- **Accurate sensory feedback** required to cope with uncertainties

- **Expensive** and **difficult** to build, control and maintain



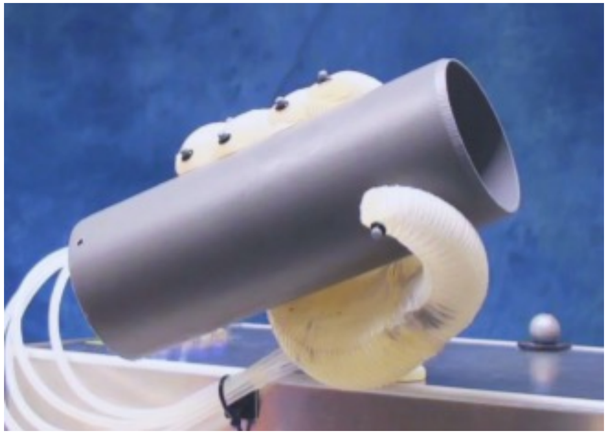
Shadow Dexterous Hand [2]



TUAT/Karlsruhe humanoid hand [3]

- Hands with **less degrees of freedom** offer **similar capabilities** when specifically designed for a set of task
  - Fewer actuators, sensors
  - Simpler control strategy

- **Design adapts to uncertainties**



RBO Hand 2 [4]

Barrett hand [5]

- **High effort** of manual design for simple manipulators
- **Automatically generate specific manipulators** based on task description

- Robotic hands often **inspired by biological structures**

- „Evolutionary robotics applies the selection, variation and heredity **principles of natural evolution** to the design of robots with embodied intelligence.“

(Doncieux et al., 2015) [7]

Goal

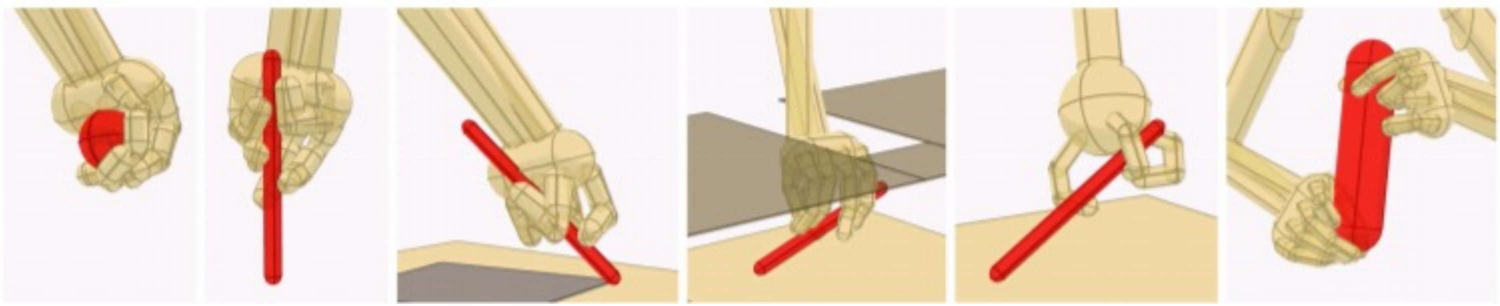
- Automatically **designing simple** and **robust** robotic hands for **desired object manipulation tasks** due to a high-level description

- Further goal
  - Creation of simple manipulators which already **incorporate robustness in their design** and do not rely on complex controllers

RELATED WORK

Contact-invariant optimization for hand manipulation [8]

- Generate a **wide variety of dexterous hand manipulations** from few high-level goals
  - E.g. grasping, picking-up objects, drawing

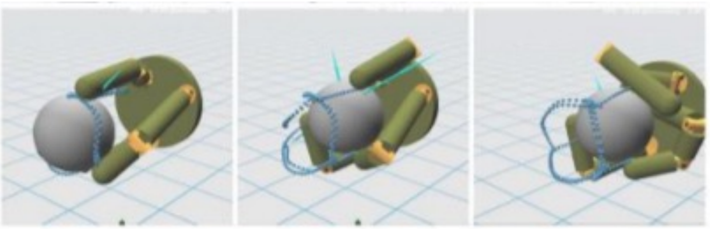


- **Generalizes to different hand morphologies** besides human hands such as three-fingered manipulators

Paper	Manipulator creation	Manipulator	Robustness	Environment	Task
Mordatch et al. (2012)	x	any	x	-	general

Automated design of manipulators for in-hand tasks [6]

- Generate **low-DOF hand designs** able execute provided manipulation task from few high-level goals
  - **Optimize contact points** and **forces** to move object to desired pose
  - Design manipulator and trajectory that **provides desired forces**

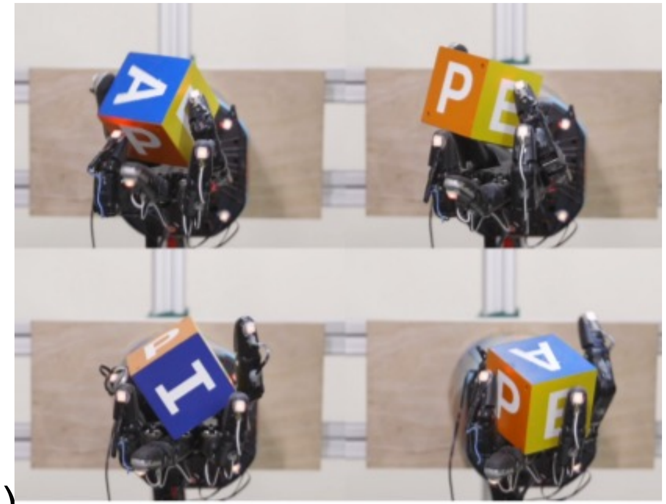


- Limitations
  - **No planning under uncertainty**
  - Required forces are interpolated and trajectory approximated
  - Fixed set of finger contacts

Paper	Manipulator creation	Manipulator	Robustness	Environment	Task
Hazard et al. (2018)	✓	any	x	-	general
Mordatch et al. (2012)	x	any	x	-	general

Learning dexterous in-hand manipulation [9]

- Applies reinforcement learning to learn **robust dexterous in-hand manipulation policy** in **simulated environment**
- Application of large-scaled distributed **domain randomization** to generate simulation that **transfer to physical robot**
  - Physical parameters (e.g. object size, mass)
  - randomizing visual appearence of the scene
  - perturbing forces

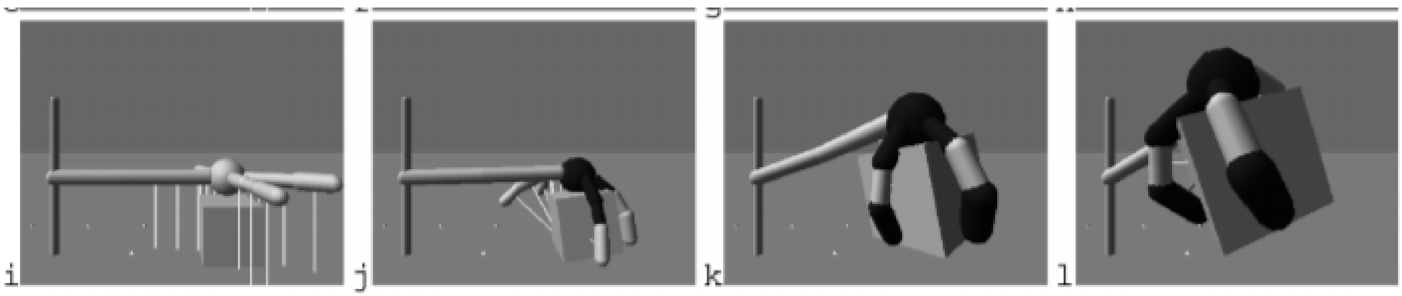


Paper	Manipulator creation	Manipulator	Robustness	Environment	Task
OpenAI et al. (2018)	x	ShadowHand	✓	Real-world	specific
Hazard et al. (2018)	✓	any	x	-	general
Mordatch et al. (2012)	x	any	x	-	general

Motivation » **Related Work** » Approach » Evaluation » Conclusion

The utility of evolving simulated robot morphology increase with task complexity for object manipulation [10]

- **Evolutionary strategy** to **evolve hand design** and **control policy** for **specific** object manipulation task (grasp, lift, perception)



- Optimization on **six different domains**
  - Objects vary in shape and size

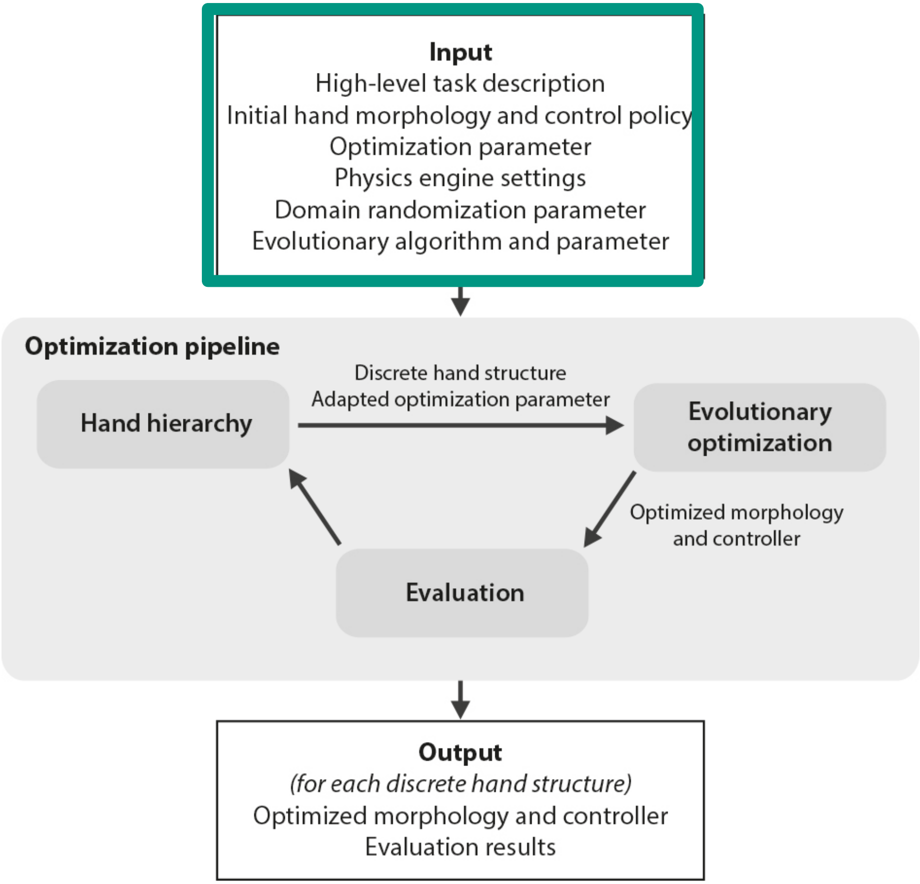
Paper	Manipulator creation	Manipulator	Robustness	Environment	Task
Bongard (2010)	✓	specific	✓	Phys. Simulation	specific
OpenAI et al. (2018)	x	ShadowHand	✓	Real-world	specific
Hazard et al. (2018)	✓	any	x	-	general
Mordatch et al. (2012)	x	any	x	-	general

My Approach

- **Co-evolve hand morphology** and **controller**
  - Standard evolutionary algorithms (CMA-ES, MO-CMA)
  - In **physics simulation**
  - Simulatenously on different world states using **Domain Randomization**

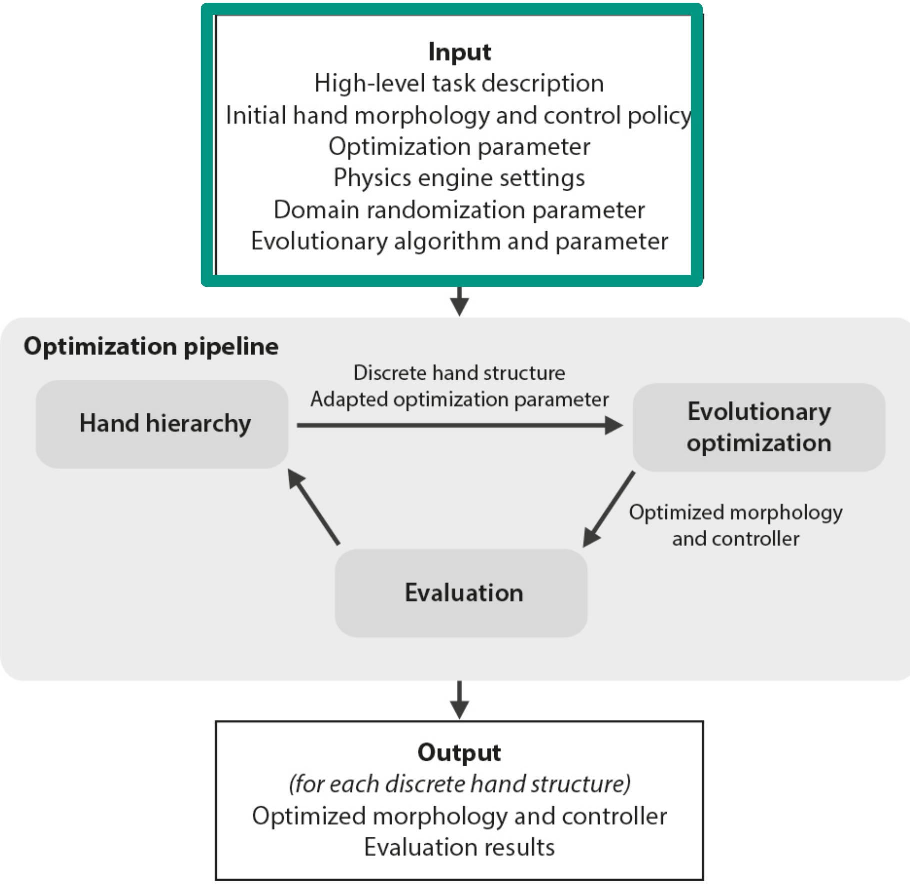
Paper	Manipulator creation	Manipulator	Robustness	Environment	Task
My approach	✓	any low-DOF	✓	Phys. Simulation	general
Bongard (2010)	✓	specific	✓	Phys. Simulation	specific
OpenAI et al. (2018)	x	ShadowHand	✓	Real-world	specific
Hazard et al. (2018)	✓	any low-DOF	x	-	general
Mordatch et al. (2012)	x	any	x	-	general

Optimization pipeline - Input



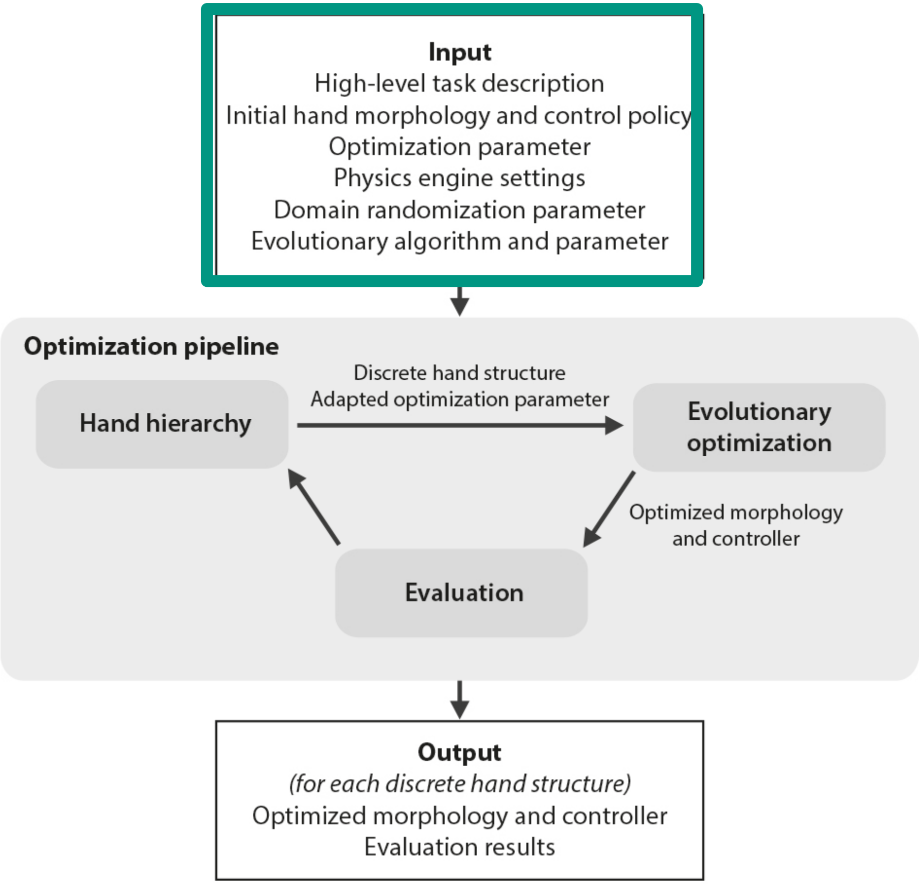
- Task description
  - **Object poses/Trajectory**
- Initial hand morphology
  - E.g. Two-fingered hand
- Control policy
  - Simple Torque control
  - PD Controller

Optimization pipeline - Input



- Optimization parameter
  - Intervals of **possible values for morphology or controller**
  - Morphology
    - Finger segment lengths
    - Finger position on palm
    - Joint axis
    - Initial joint Angle
    - Joint Limits
    - Friction coefficient
  - Controller
    - Torques OR
    - Target joint angles, P and D coefficients

Optimization pipeline - Input



- Domain randomization
  - Deviations to object
    - **Position, rotation, friction, mass, scaling**
  - Amount of domains
- Evolutionary algorithm/param.
  - MO-CMA or CMA-ES
  - Population size
  - Iteration number

Optimization pipeline – Hand hierarchy

Input
High-level task description
Initial hand morphology and control policy
Optimization parameter
Physics engine settings
Domain randomization parameter
Evolutionary algorithm and parameter

Optimization pipeline

Hand hierarchy

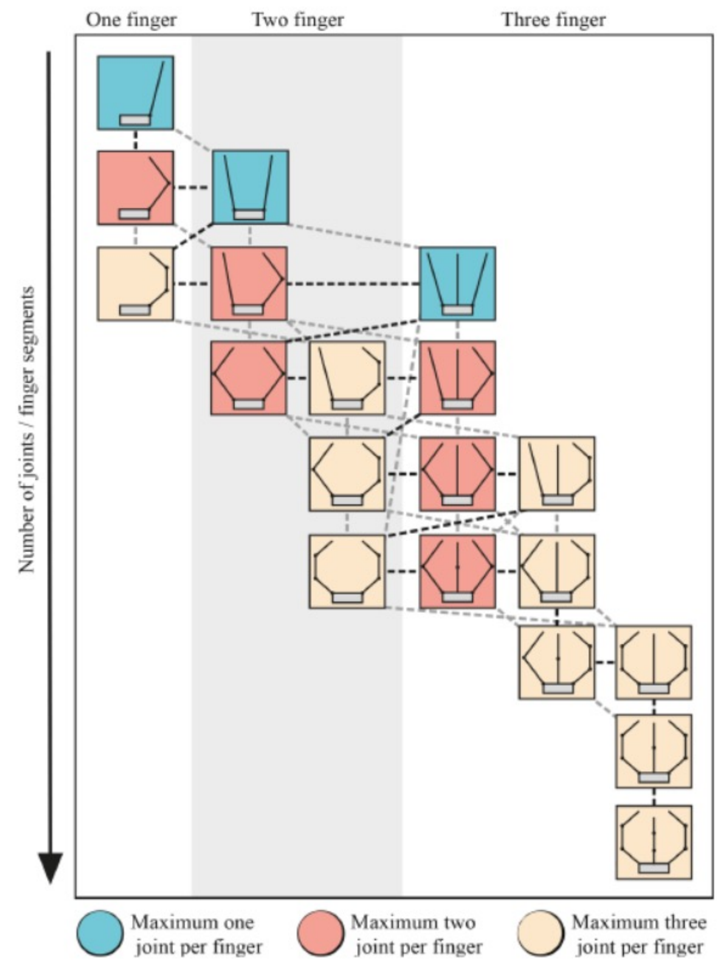
Discrete hand structure
Adapted optimization parameter

Evolutionary optimization

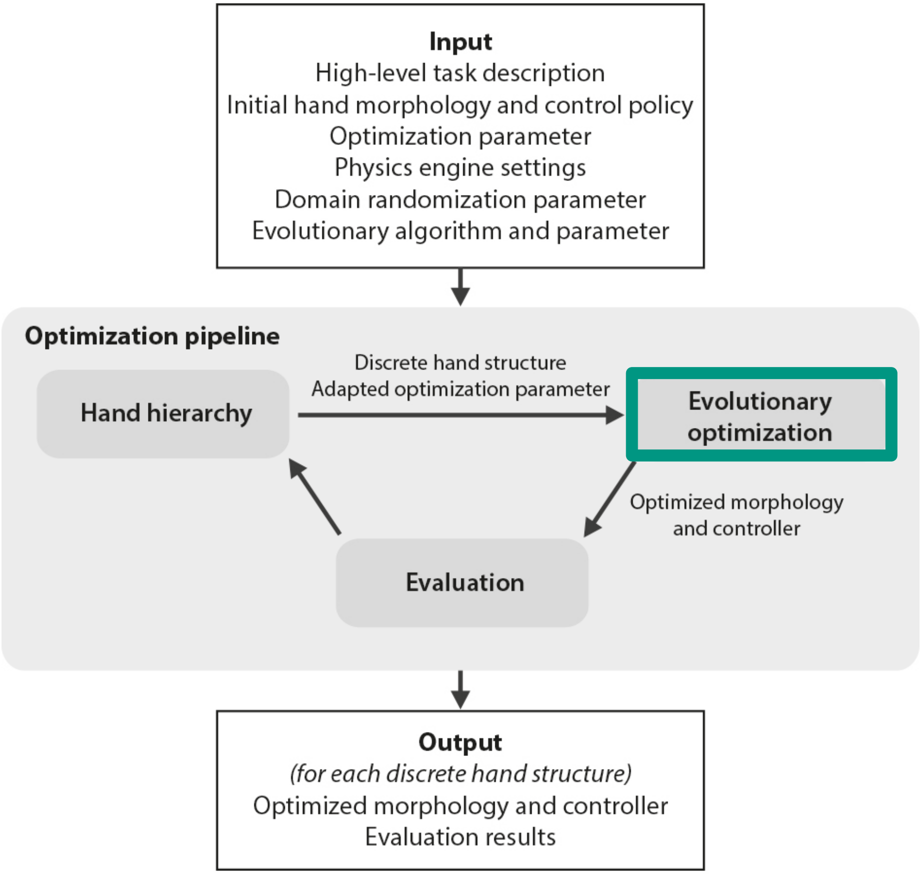
Optimized morphology and controller

Evaluation

Output
(for each discrete hand structure)
Optimized morphology and controller
Evaluation results

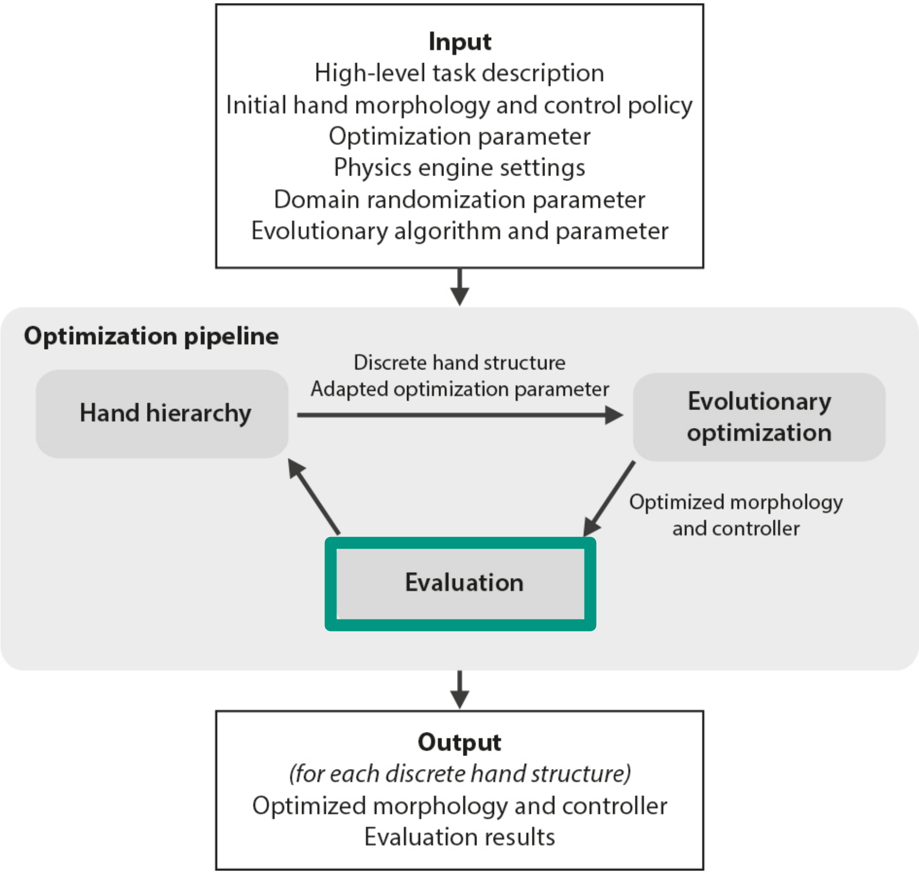


Optimization pipeline – Evo. Optimization



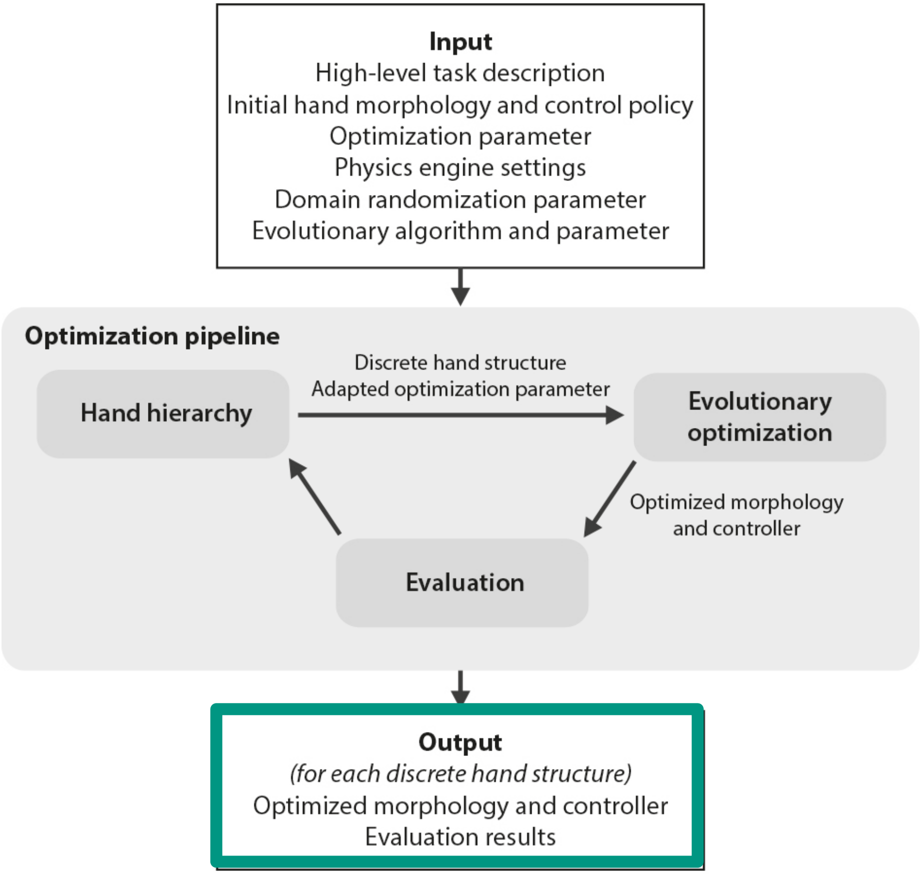
- **Minimize distance** between desired and actual position
- **Minimize angle** between desired and actual orientation
- **Averaged** on all simulated **domain randomizations**

Optimization pipeline – Evo. Optimization



- Evaluation based on X samples of domain randomization

Optimization pipeline – Output



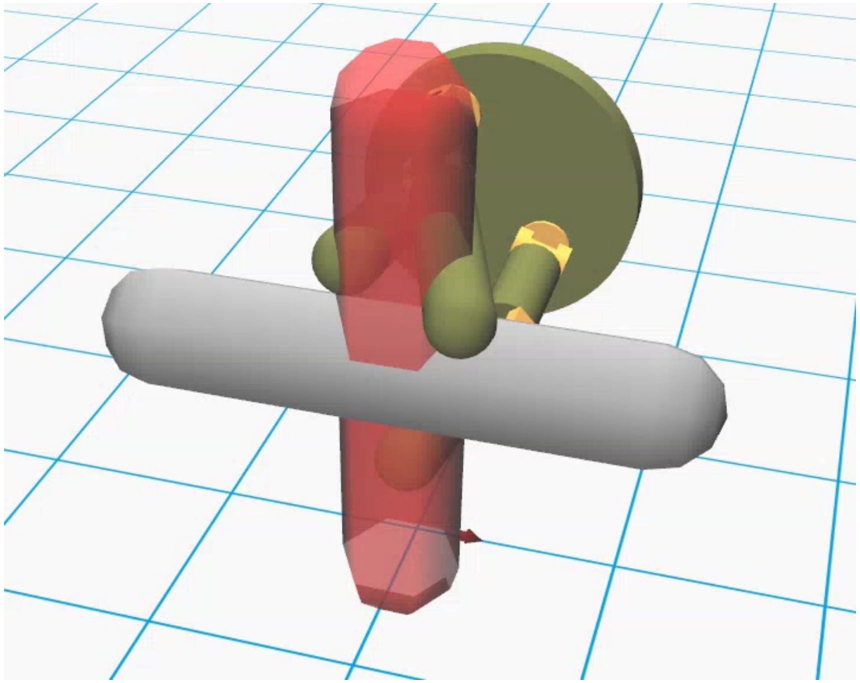
- Optimized morphology and controller
- Evaluation results

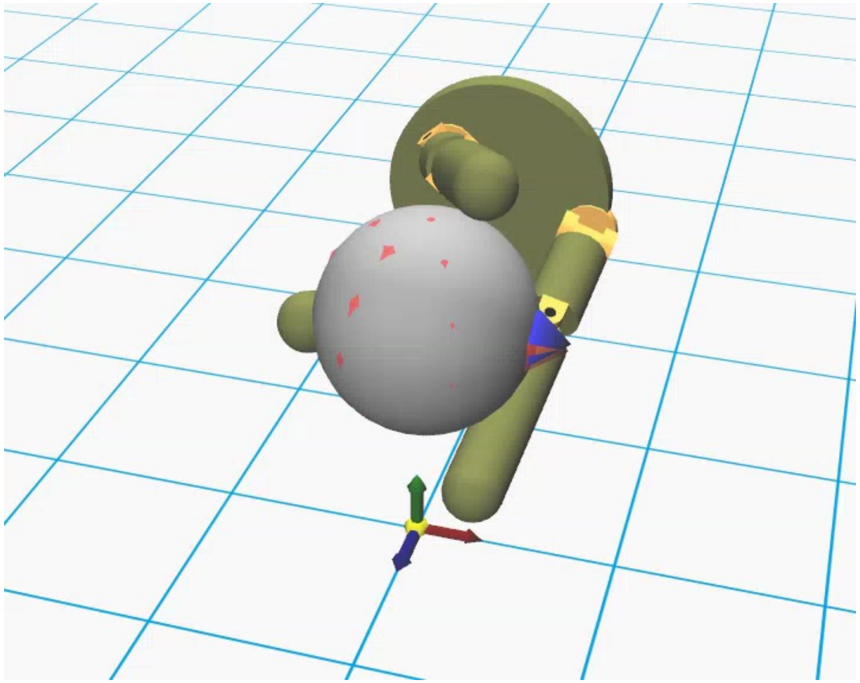
Trajectory Optimization [6]

EVALUATION

Trajectory Optimization [6]

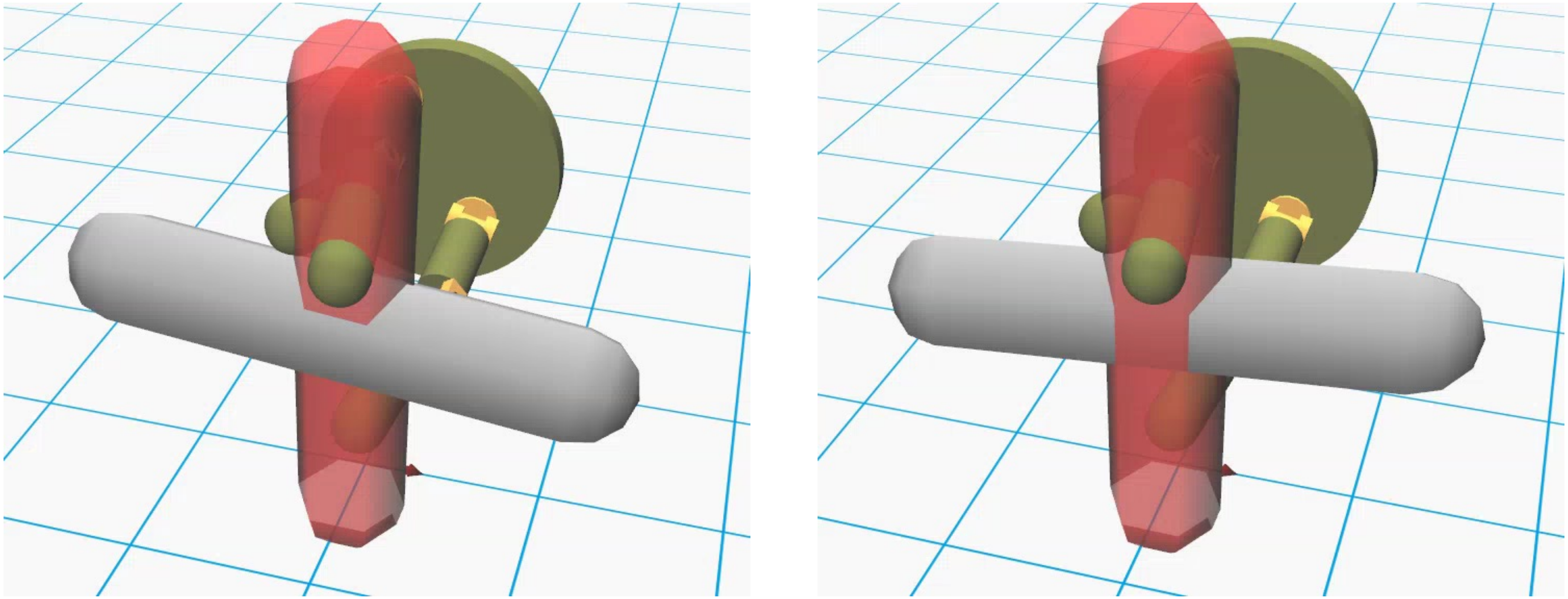
- Obtained results [6] **transfered to physics simulation** with PD Controller
- Manipulator **fail** to perform manipulation





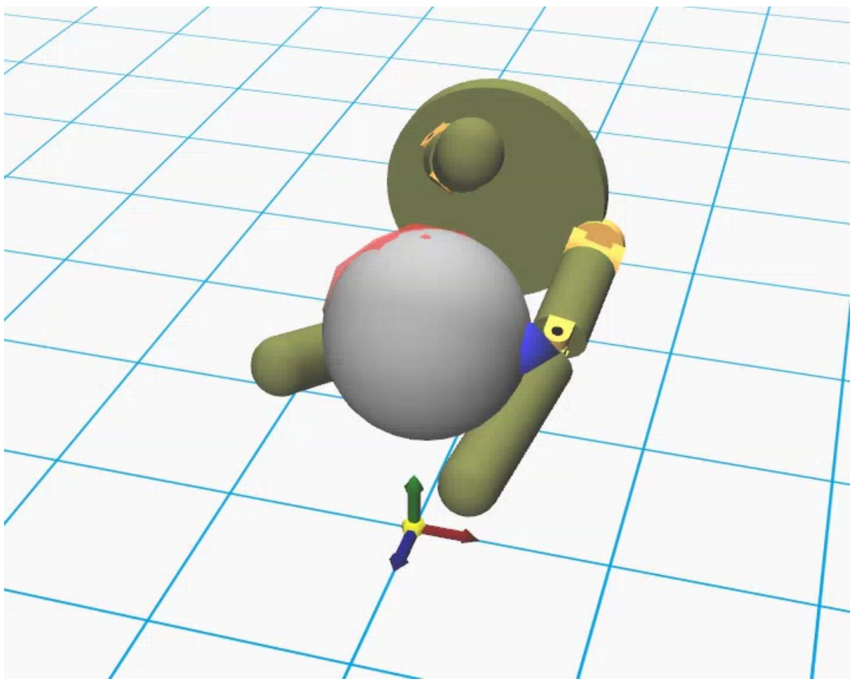
Trajectory Optimization [6]

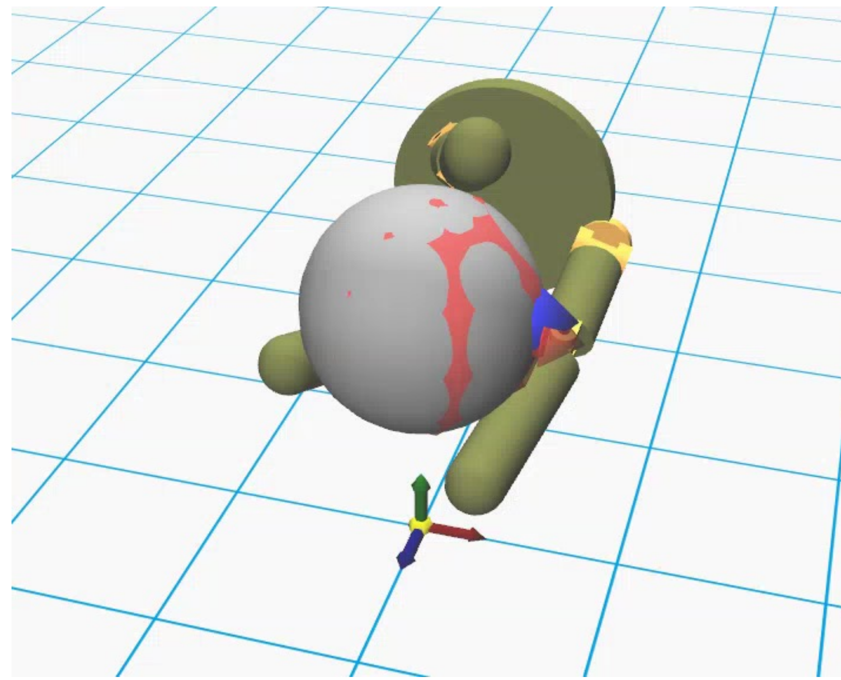
- **Further optimized** with results with evo. opt. and domain randomization
- *Task:* Rotate capsule 90° clockwise in midair
- Simulation on different world states



Trajectory Optimization [6]

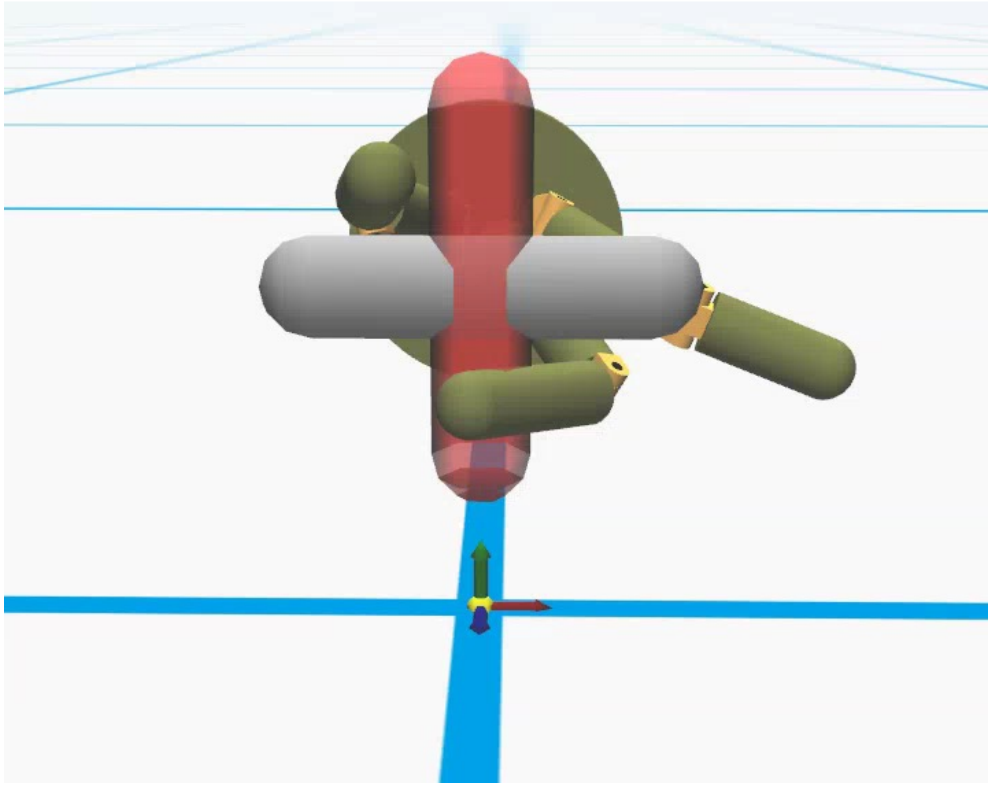
- **Further optimized** with results with evo. opt. and domain randomization
- *Task*: Rotate sphere ~180° clock- and counterlockwise in midair
- Simulation on different world states





Trajectory Optimization [6]

- Evolutionary optimization **from scratch**
- *Task:* Rotate capsule 90° clockwise in midair

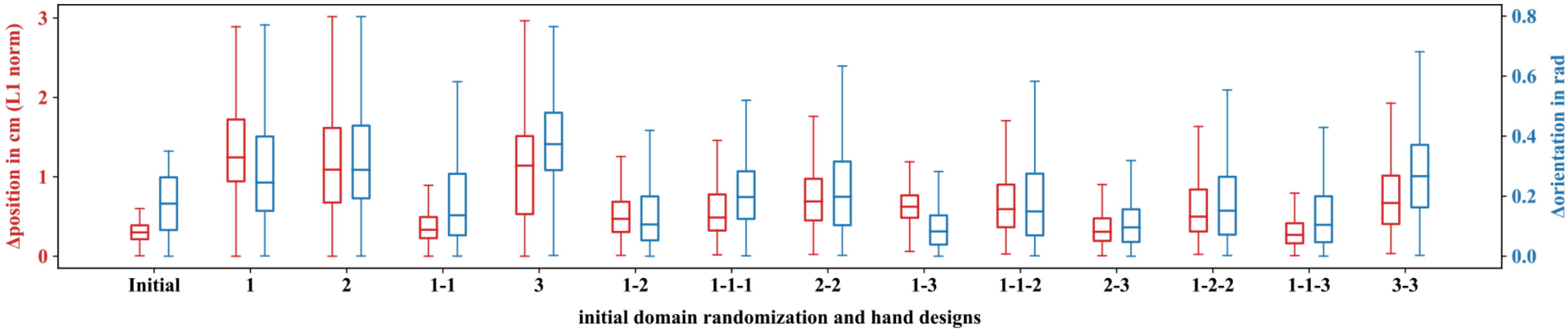


Optimization Pipeline

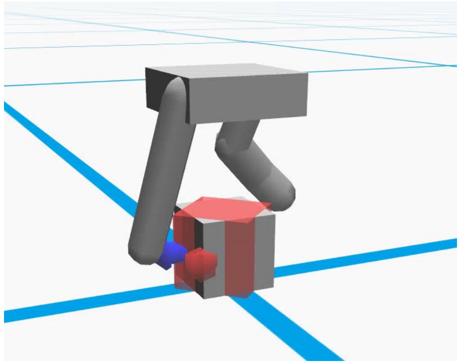
EVALUATION

Optimization Pipeline

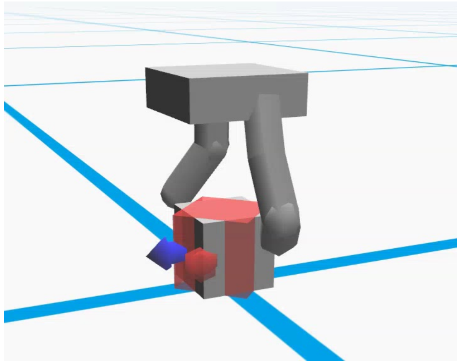
- *Task:* Rotate box 45° counterclockwise (with 1-torque control)



Simulation of 1-3 hand

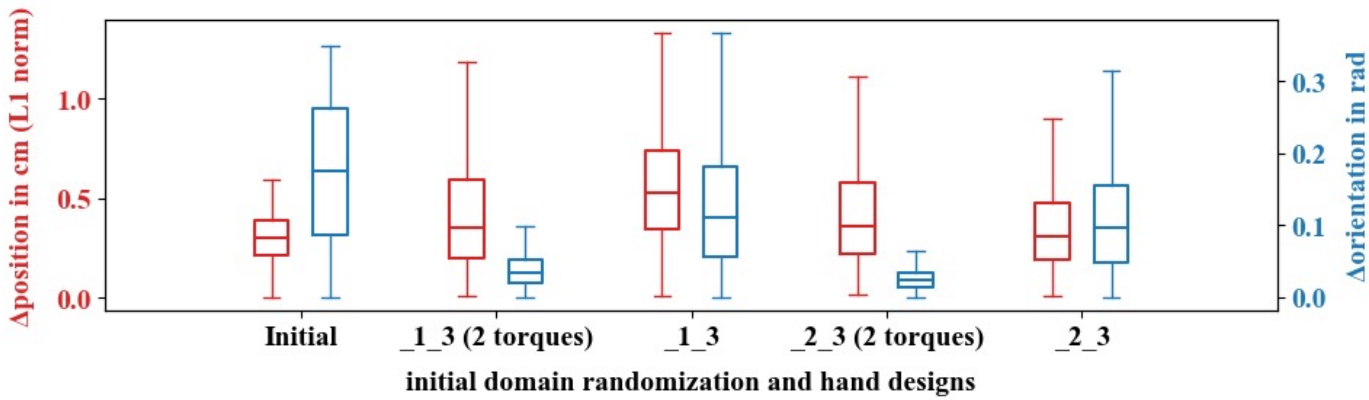


Simulation of 2-3 hand

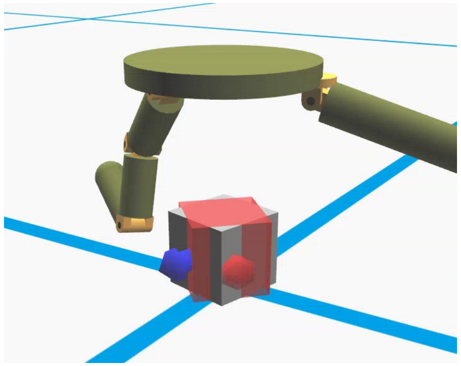


Optimization Pipeline

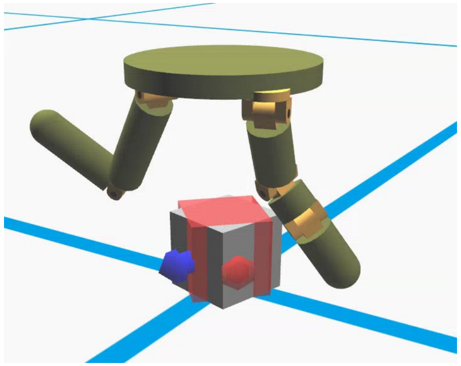
- *Task:* Rotate box 45° counterclockwise (with 2-torque control)



Simulation of 1-3 hand



Simulation of 2-3 hand

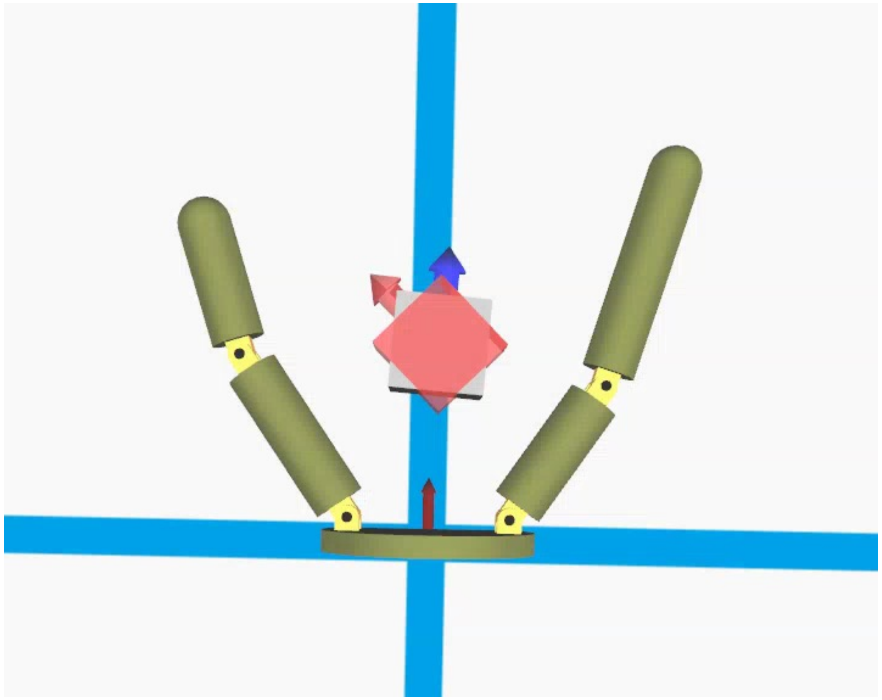


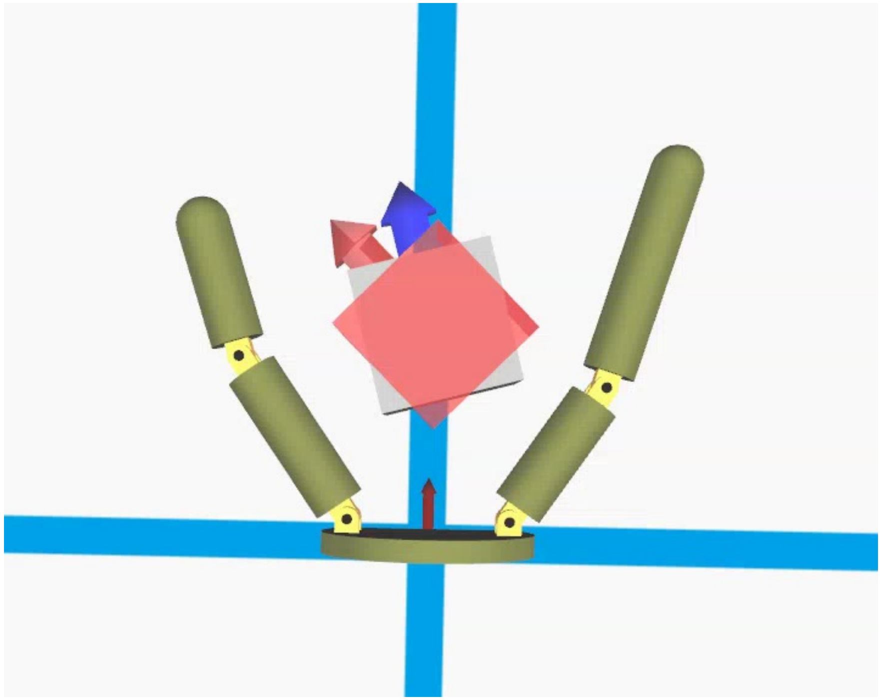
Optimizing Joint Limits

EVALUATION

Optimizing joint limits

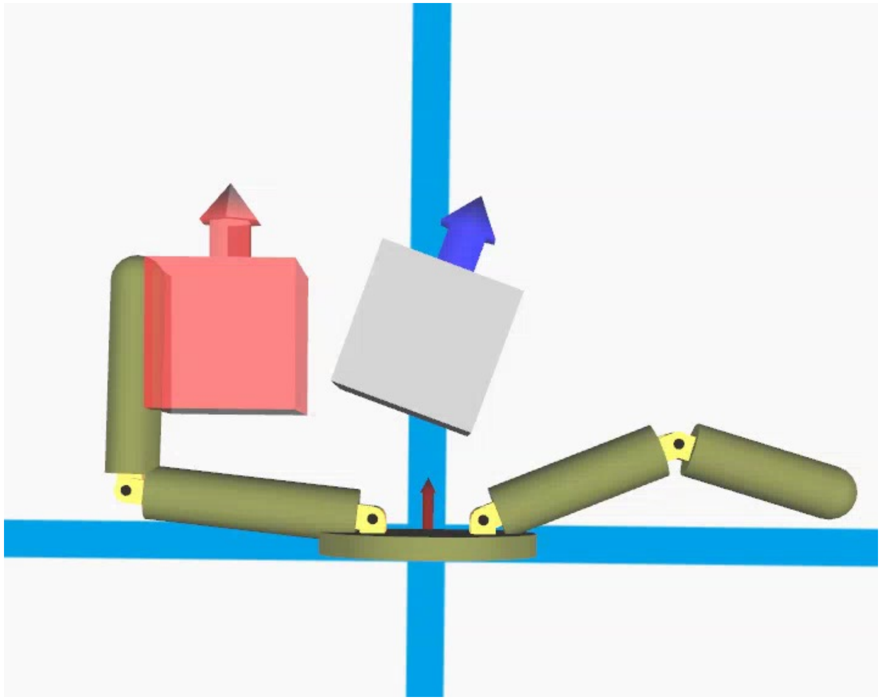
- *Task*: Rotate box 45° counterclockwise
- Optimized manipulator simulated on different world states

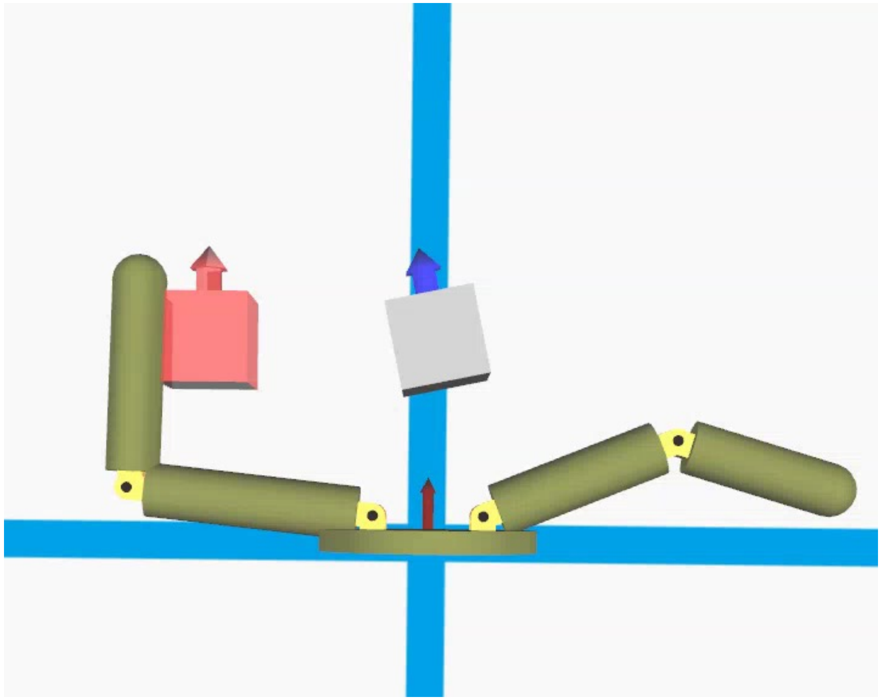




Optimizing joint limits

- *Task*: Align box at specific line in space
- Optimized manipulator simulated on different world states





CONCLUSION & OUTLOOK

Conclusion

- Introduced an **optimization pipeline** to automatically generate simple and robust manipulators for desired manipulation tasks
  - **Robust results on simple manipulation tasks**
  - Limitations
    - Performance
    - Complexity of tasks

- Demonstrated the significance of optimizing **joint limits for robustness**

- Evolutionary approach **complementary** to trajectory optimization [6]
  - **Improve robustness** of results
  - Limitations of trajectory optimization apply

Outlook

- Evolve manipulators for complex motions
  - Evolutionary strategies potentially **converge to local minima**
  - **Generate higher diversity** in population
    - Literature provides
      - Adaptions of basic evolutionary algorithms
      - Non-performance based objectives to guide the simulation
  - Dimension reduction

- Evaluate robustness of results on **real manipulators**

Thank you for your attention!

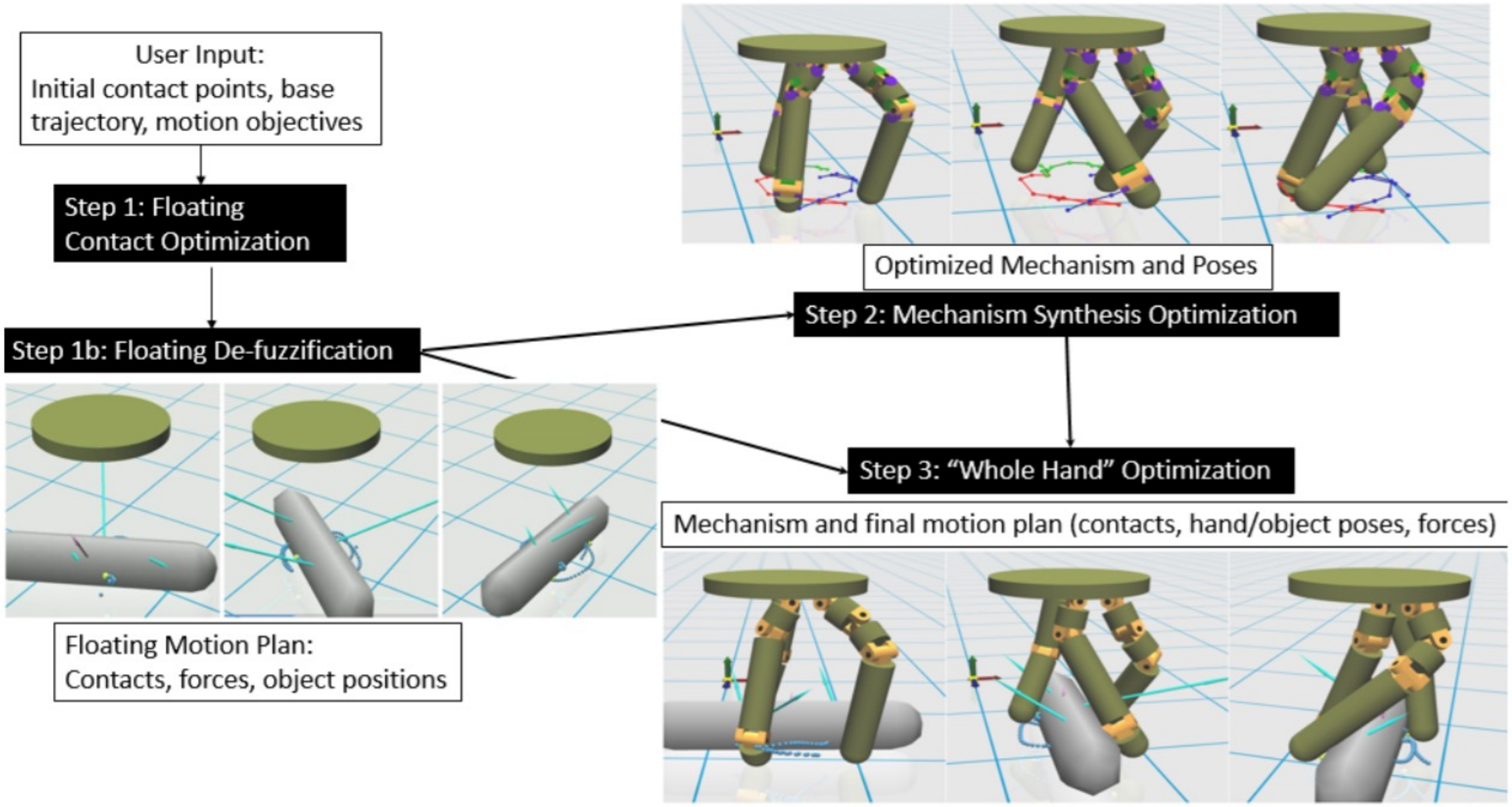
Any questions left?

References

- [1] A. Bicchi. Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. IEEE Trans. Robotics and Automation, 16:652–662, 2000.
- [2] ShadowRobot. Shadow dexterous hand. URL https://www.shadowrobot.com/products/dexterous-hand/. Accessed on 2019-08-27.
- [3] N. Fukaya, T. Asfour, R. Dillmann, and S. Toyama. Development of a five-finger dexterous hand without feedback control: The tuat/karlsruhe humanoid hand. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4533–4540, Nov 2013. doi: 10.1109/IROS.2013.6697008.
- [4] R. Deimel and O. Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. The International Journal of Robotics Research, 35(1-3):161–185, 2016. doi: 10.1177/0278364915592961. URL https://doi.org/10.1177/0278364915592961.
- [5] M. T. Mason, S. S. Srinivasa, and A. S. Vazquez. Generality and simple hands. In C. Pradalier, R. Siegwart, and G. Hirzinger, editors, Robotics Research, pages 345–361, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-19457-3.
- [6] C. Hazard, N. Pollard, and S. Coros. Automated design of manipulators for in-hand tasks. pages 1–8, 11 2018. doi: 10.1109/HUMANOIDS.2018.8624932.
- [7] S. Doncieux, N. Bredèche, J.-B. Mouret, and A. E. Eiben. Evolutionary robotics: What, why, and where to. Front. Robotics and AI, 2015, 2015.
- [8] I. Mordatch, Z. Popovi´c, and E. Todorov. Contact-invariant optimization for hand manipulation. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’12, pages 137–144, Goslar Germany, Germany, 2012. Eurographics Association. ISBN 978-3-905674-37-8. URL http://dl.acm.org/citation.cfm?id=2422356.2422377.
- [9] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. W. Pachocki, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. CoRR, abs/1808.00177, 2018. URL http://arxiv.org/abs/1808.00177.
- [10] J. Bongard. The utility of evolving simulated robot morphology increases with task complexity for object manipulation. Artificial Life, 16(3):201–223, 2010. doi: 10.1162/artl.2010.Bongard.024.

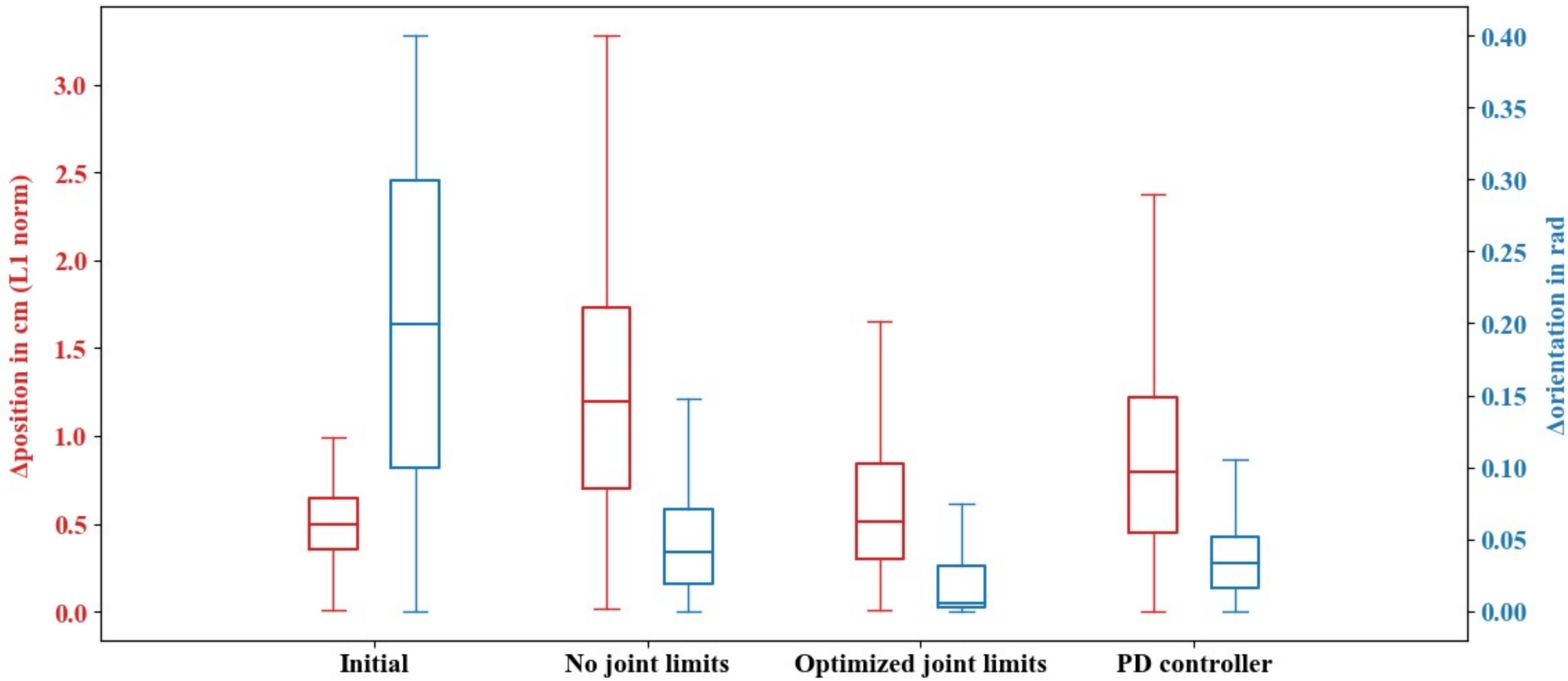
APPENDIX

Trajectory Optimization Pipeline



Optimizing joint limits

- Comparison of different evolved hand designs with PD controller or with torque policy



FOUNDATION

PD Controller with feed forward term

- Follow provided joint angle trajectory

$$\tau = \underbrace{M(\theta)\ddot{\theta}_d + C(\theta,\dot{\theta})\dot{\theta}_d + N(\theta,\dot{\theta})}_{\tau_{ff}} \underbrace{-K_p e - K_v \dot{e}}_{\tau_{fb}}$$

- Feedback control τ_{fb}
  - Reduce control error between setpoint and measurement
  - K_p, K_v gain matrices chosen to be diagonal
    - Diagonal entries P and D coefficients
- Feedforward control τ_{ff}
  - Adjust control signal according to a function of disturbances
  - Inertia matrix, Coriolis and centrifugal forces, gravity compensation

Evolutionary Robotics (ER)

- Robotic hands often **inspired by biological structures**

- „Evolutionary robotics applies the selection, variation and heredity **principles of natural evolution** to the design of robots with embodied intelligence.“

  (Doncieux et al., 2015) [7]

[ER] Genotype-to-Phenotype-Mapping

- *Genotype* **encodes** a solution as sequence of bits or numbers
- *Phenotype* corresponds to the **robotic system**
- Direct mapping
  - Mapping each parameter directly
- Indirect mapping
  - Encode robotic system as neuronal network

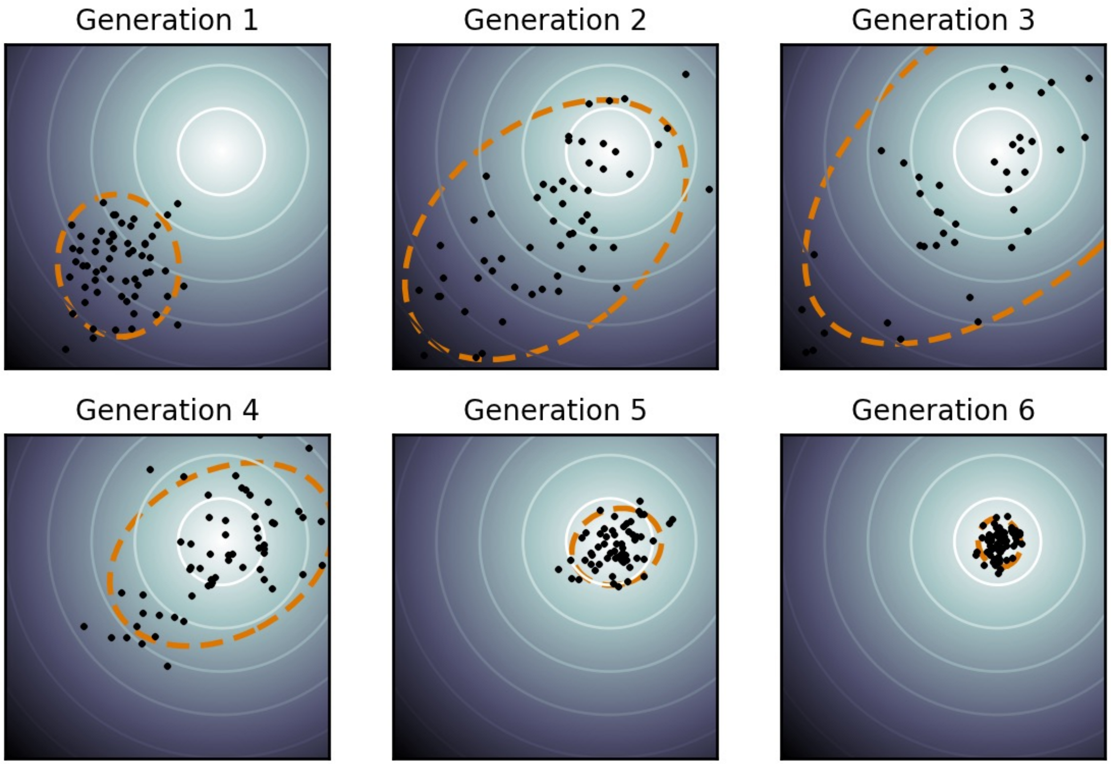
[ER] Evolutionary algorithm

- Init
  - Randomly generated genotypes form **parent population** at generation 0

- Iteration
  - **Generate** new **offspring** from parent based on **mutation**/recombination
  - Map genotype to phenotype
  - **Select individuals** from offspring and parent based on **fitness function**
    - Form new parent population for next iteration

[ER] Covariance Matrix Adaption ES

- Illustration for two-dimensional problem

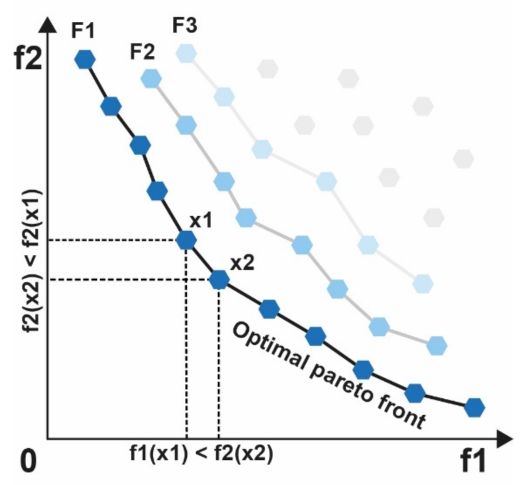


Source: https://en.wikipedia.org/wiki/CMA-ES

[ER] Multiple objectives / fitness functions

- Fitness function for CMA-ES as **weighted mean** of fitness functions

- Multi-Objective Algorithms
  - Def. Domination
    - A solution X dominates a solution Y when X is not worse than Y for all objectives but X is strictly better than Y on at least one objective
  - Sorting individuals into fronts
    - Non-Dominated Sorting
  - Ranking individuals in front
    - Diversity (Crowding distance, contribution to hypervolume, …)



- Multi-Objective CMA-ES
  - Contains n CMA-ES with one individual as population