Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

A Hybrid Genetic Swarm Algorithm for Interactive Inverse Kinematics

Sebastian Starke

Master Thesis
Colloquium

TAMS, WTM
Department of Informatics
University of Hamburg

21.06.2016

Contents

Introduction and Motivation

Problem Statement

How to adjust a set of joints in order to move an end effector to reach a Cartesian configuration of position and/or orientation?

Kinematics

„Kinema" = „Movement / Motion"

→ Field of classical mechanics

→ Motion of rigid bodies by position, velocity, acceleration

→ **No** consideration of physical dynamics (mass, force, torque, ...)

Introduction and Motivation

Applications

Robotics

→ Grasping and Object Manipulation

→ Bi-Pedal and Multi-Pedal Walking

→ Human Interaction

→ Manufacturing

Games Industry

→ Believable characters

→ Realistic motion

→ Dynamic and flexible animations

Film Industry

→ Motion Tracking

Introduction and Motivation

Challenges

→ Zero up to infinite solutions

→ Geometric complexity

→ High dimensionality

→ Suboptimal extrema and singularities

→ Joint constraints and types

→ Solution quality

→ *Accuracy* versus *Computation Time*

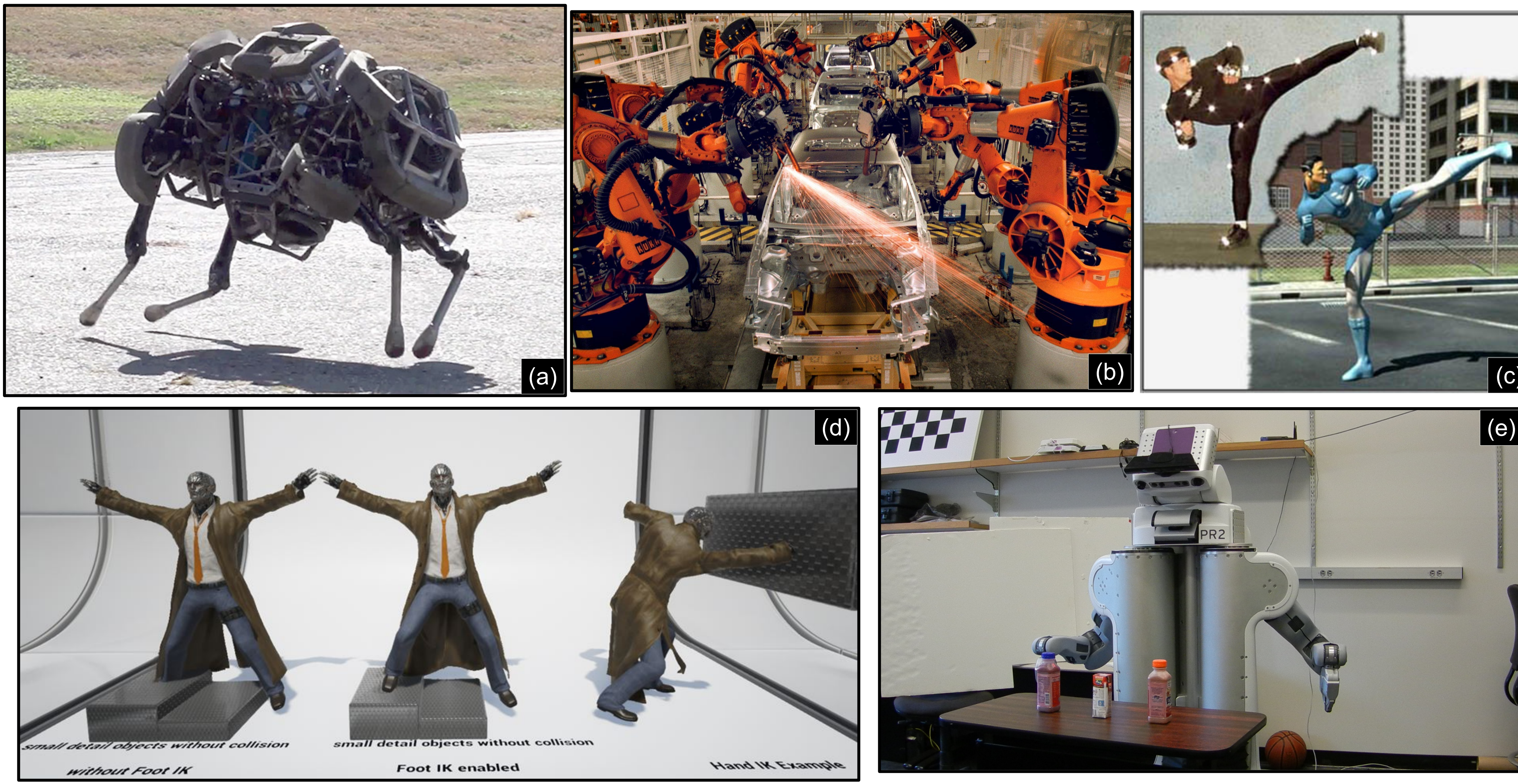
→ *Robustness* and *Reliability*

→ Displacement between solutions

→ Self-Collision

...

Introduction and Motivation



Introduction and Motivation

Major goals of this thesis

A universal IK solver for arbitrary kinematic chains

- Real-time capability for interactive frame rates
- High accuracy for both position and orientation
- Flexible, few parameters and easy-to-use

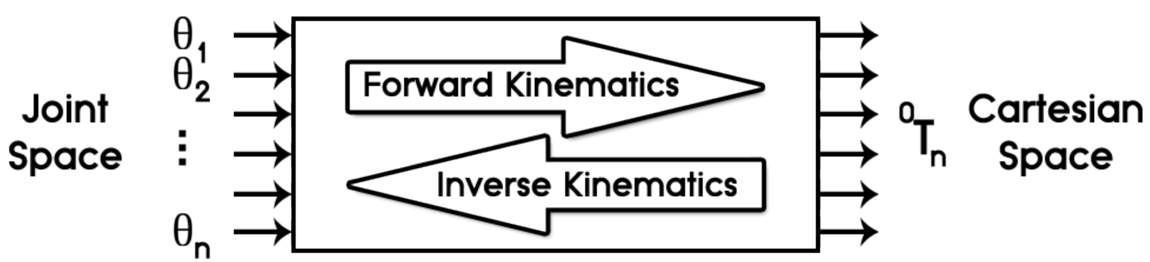
Novel algorithmic improvements for biologically-inspired optimization strategies

- Modular extensions applicable for various problems
- Higher adaptivity in exploitation and exploration
- Biologically-plausible evolutionary concepts

Problem Formalization

X → *Cartesian configuration of position and/or orientation*
θ → *Joint variable configuration*

Forward Kinematics (FK)	*Inverse Kinematics (IK)*
$$\mathcal{X} = f(\theta)$$	$$\theta = f^{-1}(\mathcal{X})$$
→ *Straightforward computation*	→ *Highly non-trivial*
→ *Unique solution*	→ *Complexity scales rapidly*
→ *Only requires kinematic specifications and joint values*	→ *Analytical versus Numerical*



Problem Formalization

Algorithmic Methodology

Analytical		**Numerical**
Extremely fast and exact	⟷	Slower computation and only approximative
Solution is always found	⟷	Solution may not always be found
Not generally available	⟷	Applicable to various kinematic models
Only derivable for simple and specific kinematic geometry	⟷	Only requires knowledge of the FK equations

Problem Formalization

Y → Cartesian target of position and/or orientation
μΘ → Weighted joint variable change

Numerical IK Update

$$\theta' = \theta + \mu\Theta \qquad d(f(\theta'), \mathcal{Y}) < d(f(\theta), \mathcal{Y})$$

Pose Distance

$$d_p = w_t \hat{d}_t + w_r d_r$$

(Rebalanced) Translational Distance

$$\hat{d}_t = \frac{\pi d_t}{\sqrt{l\Delta}} \qquad d_t = ||p_1 - p_2||$$

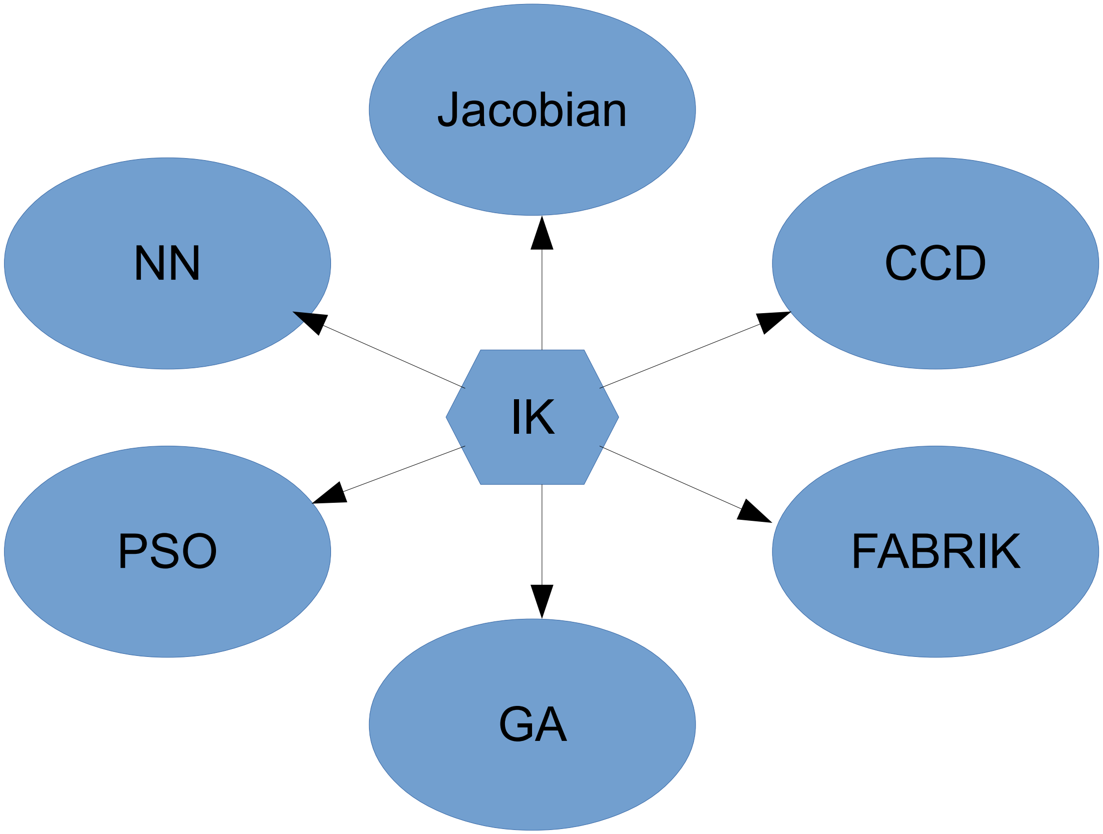
Rotational Distance

$$d_r = q_1 \cdot q_2$$

l → Length of the kinematic chain
Δ → Distance from base to end effector

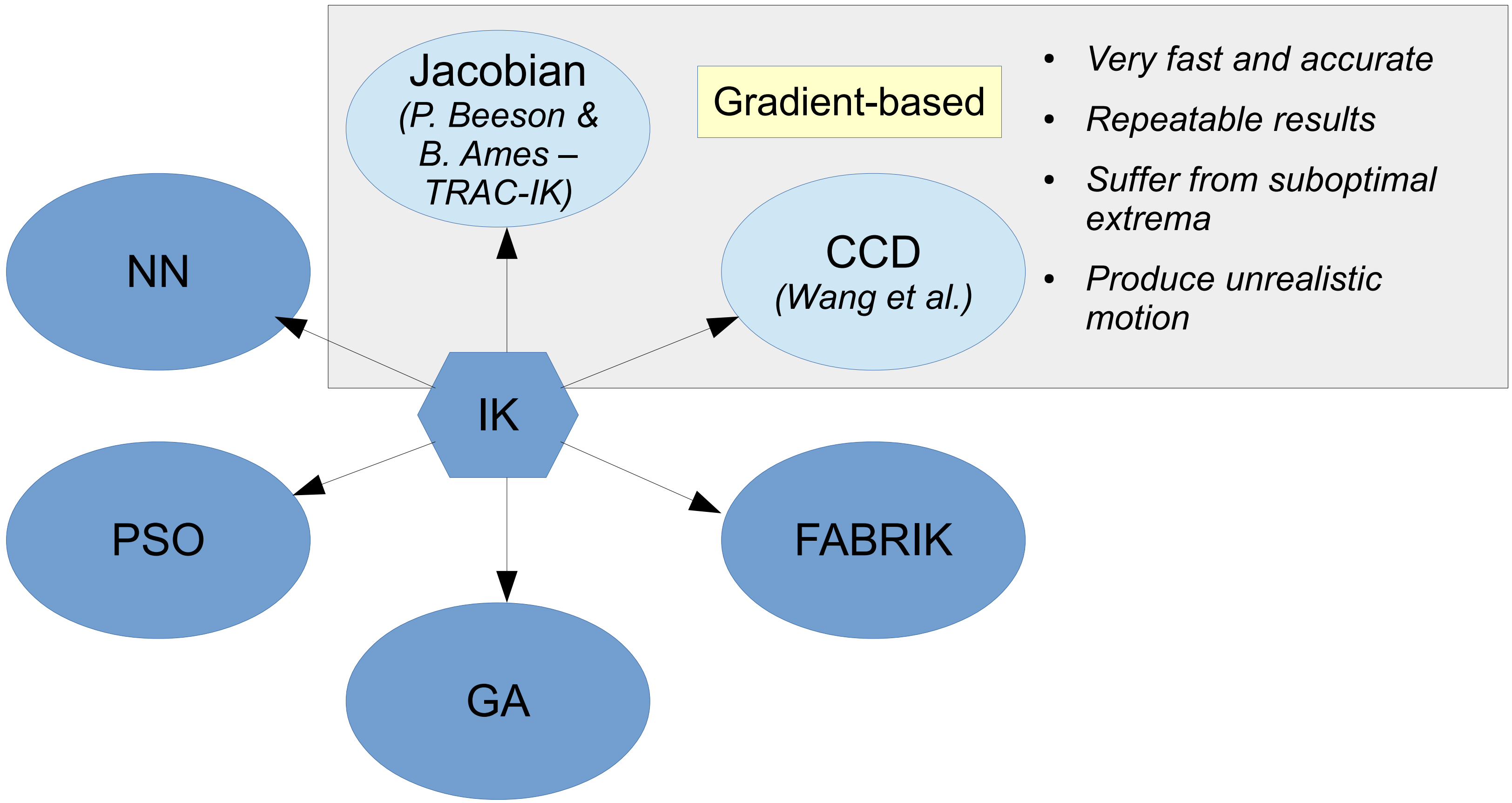
Related Work

→ IK researched over decades
→ Very many different approaches with focus on numerical



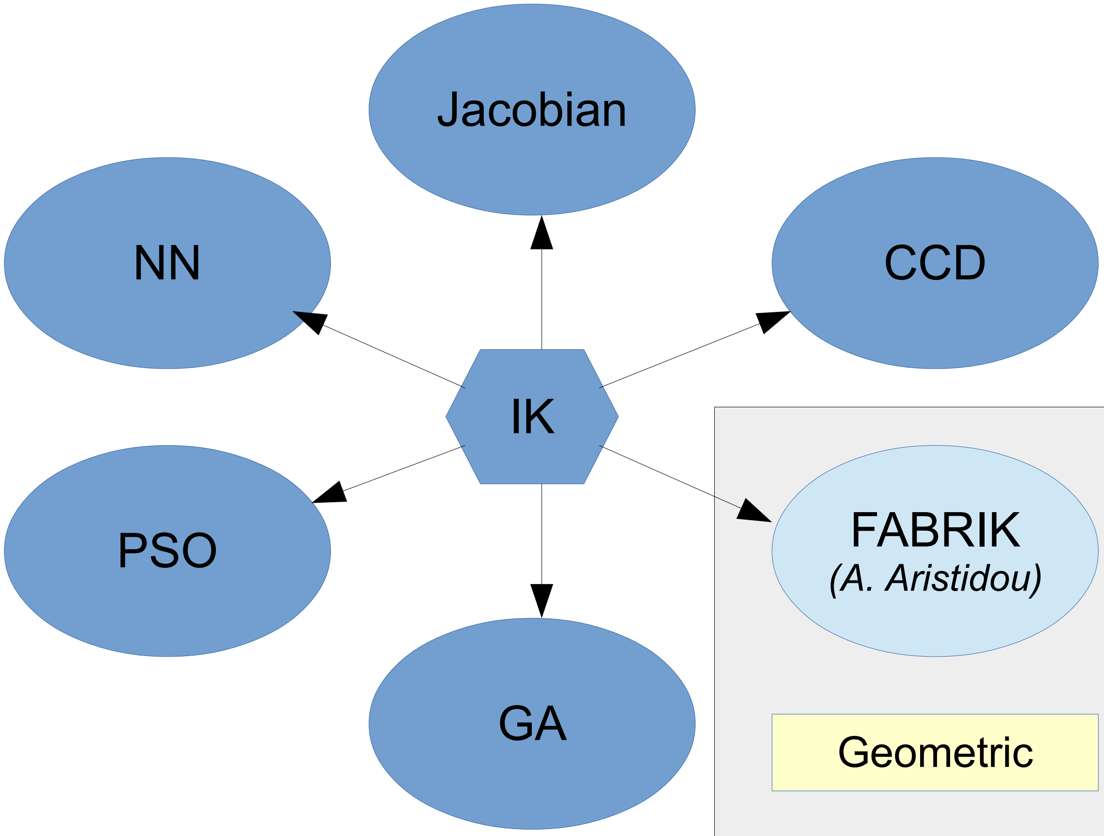
Related Work

→ IK researched over decades
→ Very many different approaches with focus on numerical



Related Work

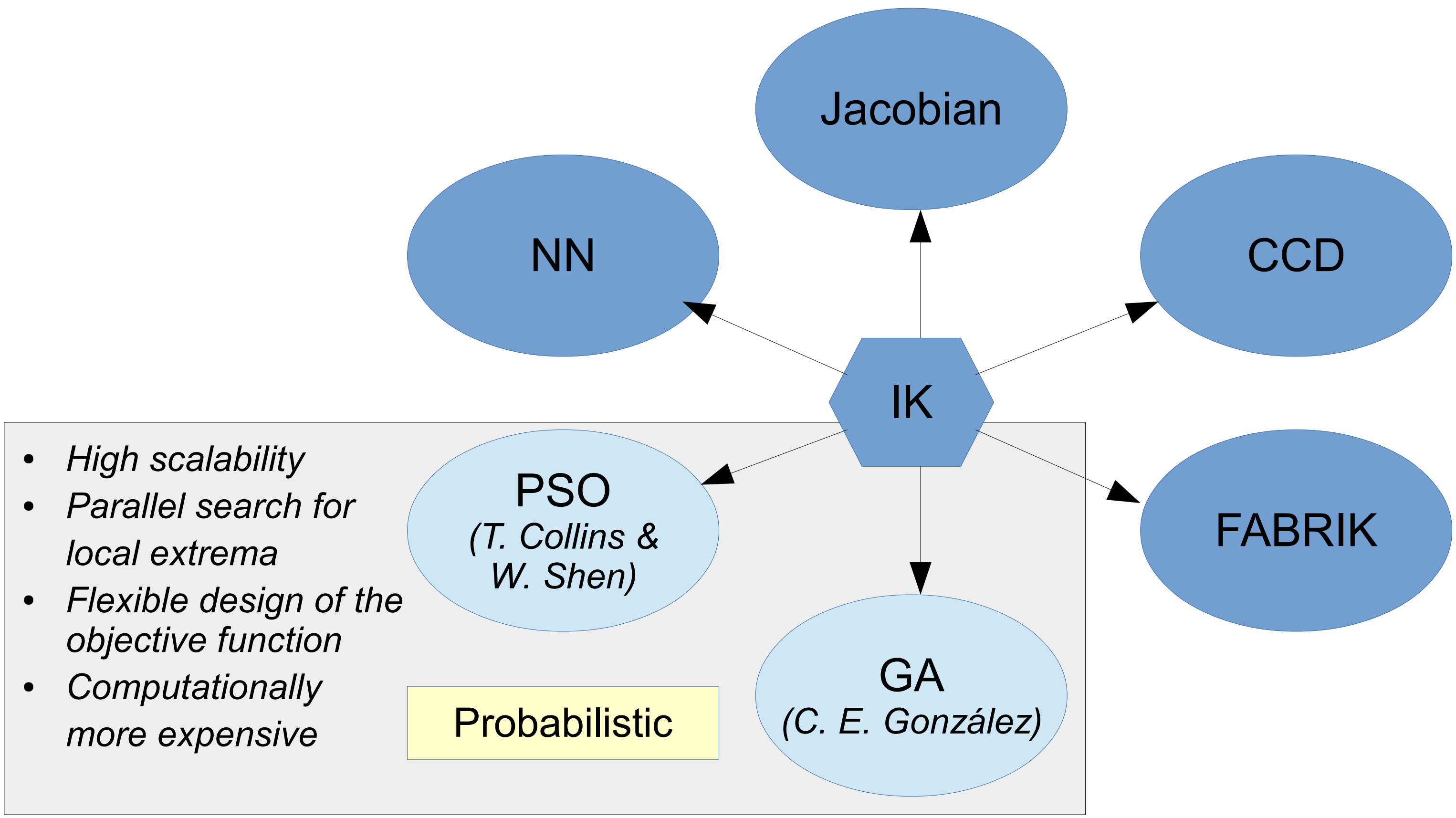
→ IK researched over decades
→ Very many different approaches with focus on numerical



- *„Forward and Backward Reaching IK“*
- *Specifically designed for character animation and motion tracking*
- *Does not operate in joint space*
- *Predominantly rotational movement*

Related Work

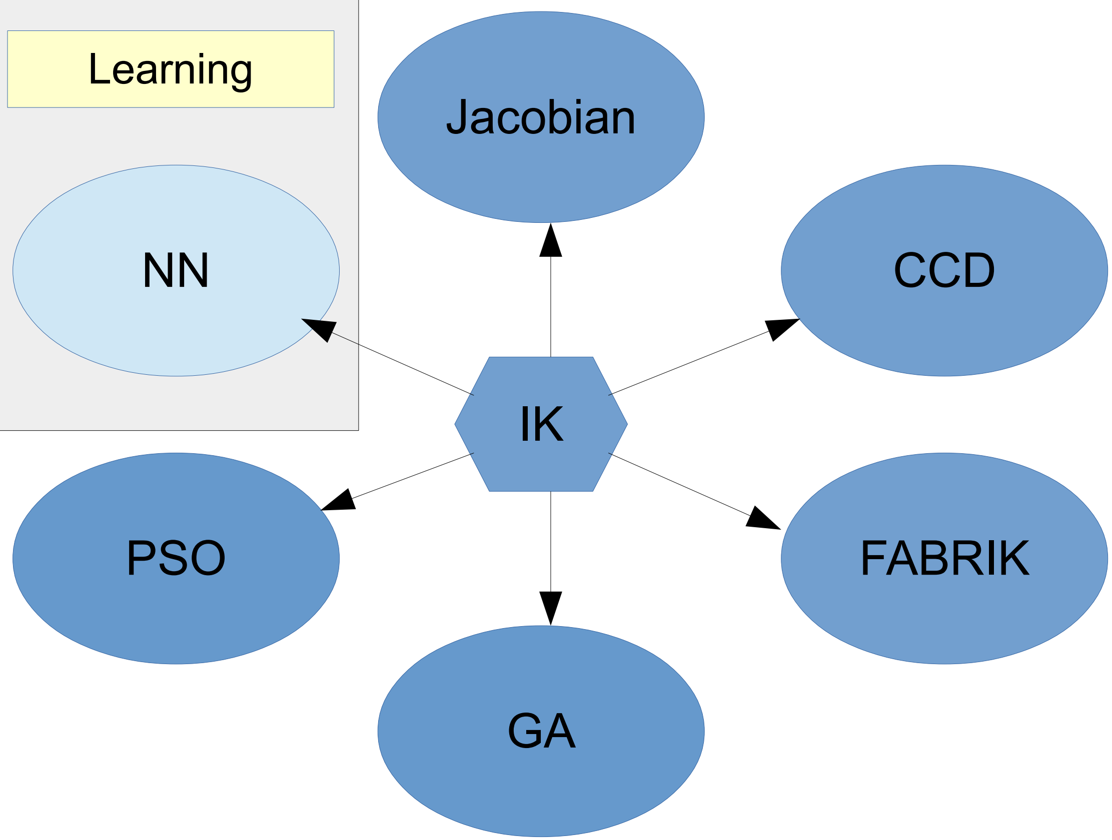
→ IK researched over decades
→ Very many different approaches with focus on numerical



Related Work

→ IK researched over decades
→ Very many different approaches with focus on numerical

- *Low accuracy*
- *Struggle with full posture objectives*
- *Only for low degree of freedom*
- *Choice of training samples remains very difficult*

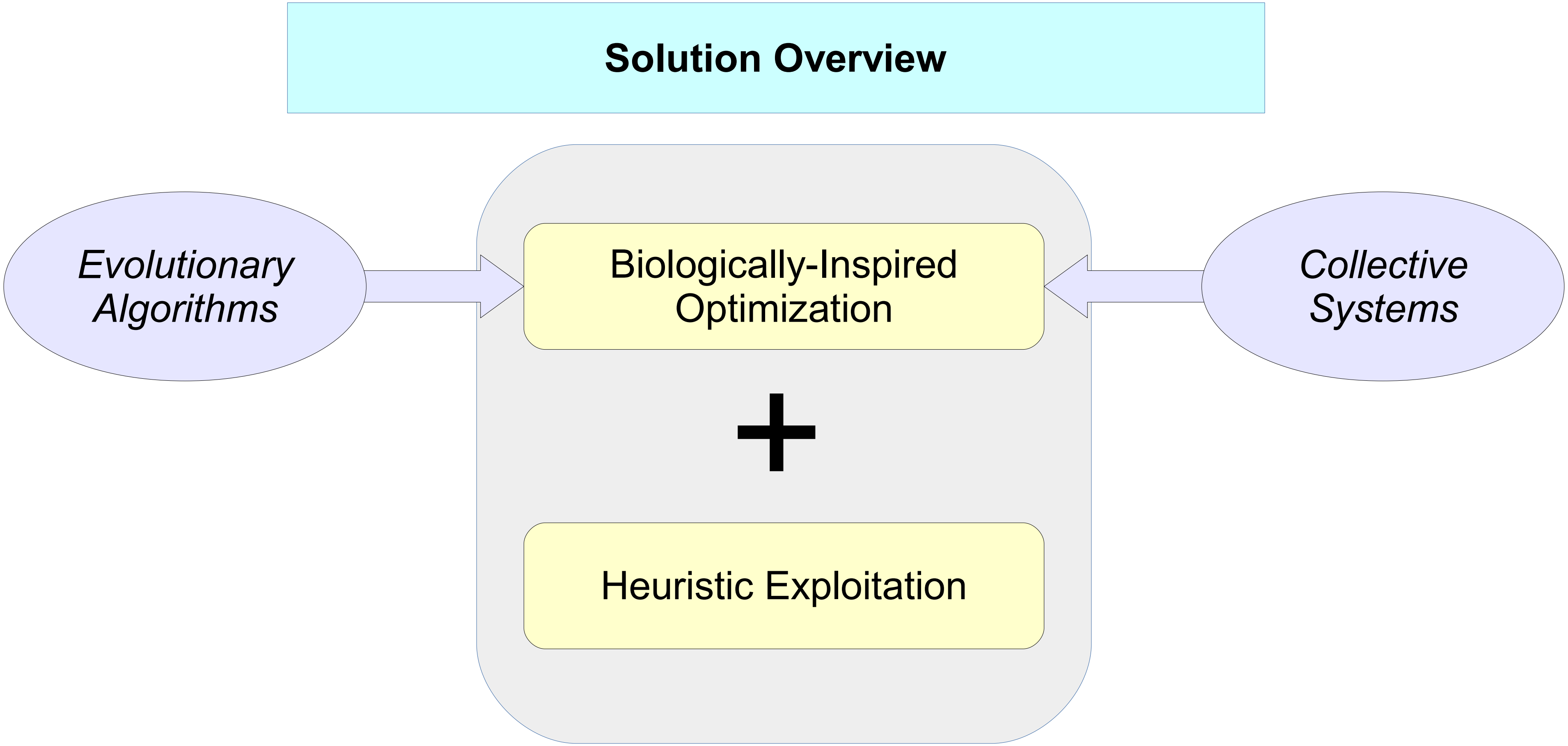


Algorithmic Approach

Design Objectives

Success	*A solution that is existent shall be found.*
Accuracy	*The solution shall be as precise as possible.* *→ approximately or less than 1mm*
Time	*The solution shall be found as fast as possible.* *→ few ms*
Displacement	*The distance between consecutive solutions shall be as minimal as possible.*
Flexibility	*The algorithm maintains high robustness, scalability and convergency even for greatly varying kinematic structure.*

Algorithmic Approach



Algorithmic Approach

Genetic Algorithms

- Developed by *J. F. Holland*
- Inspired by the theory of **natural evolution** as formulated by *C. Darwin*
- *„Survival of the fittest“* and *„Diversity drives change“*
- Group of individuals within a population that evolves over many generations
- **Selection**, **Recombination** and **Mutation**

Particle Swarm Optimization

- Developed by *J. Kennedy and R. Eberhart*
- Inspired by social emerging behaviour of **bird flocks** and **schools of fish**
- Rather simple organisms („particles“) collectively solve a complex problem
- **Velocity** and **direction** update according to success of **neighbouring** particles

Similarities

→ Search space exploration by a **group of organisms**

→ Solution quality determined by **fitness function**

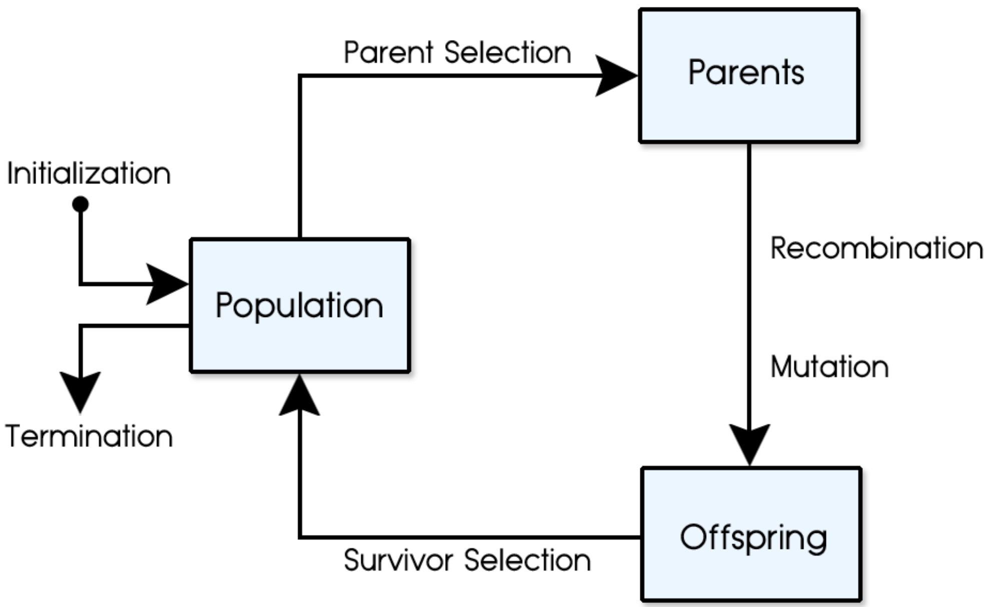
→ **Simultaneous search** for multiple local extrema

→ High **robustness** and **scalability** as well as effectiveness for **multi-objective optimization**

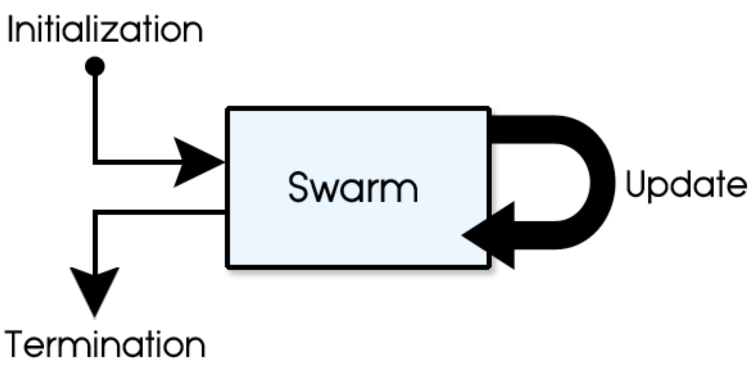
A.E. Eiben and J. E. Smith – *Introduction to Evolutionary Computing*, Springer, 2003
D. Floreano and C. Mattiussi – Bio-Inspired Artificial Intelligence, MIT Press, 2008

Algorithmic Approach

Genetic Algorithms

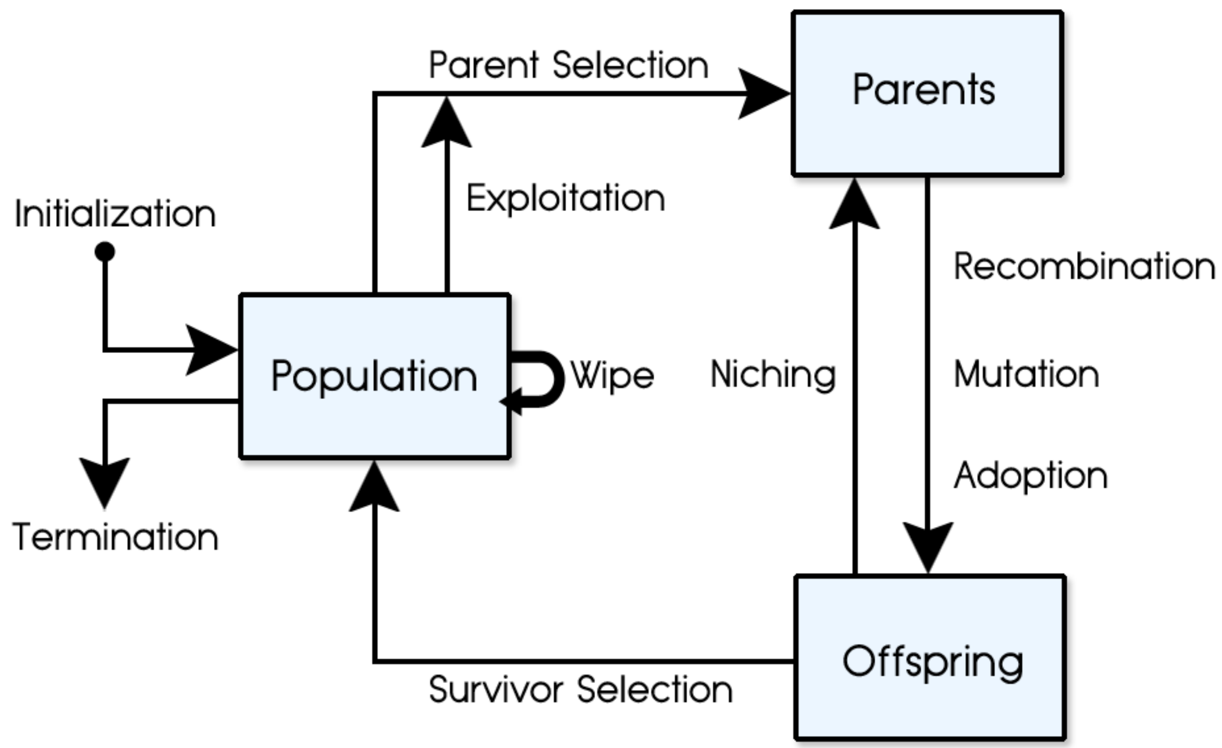


Particle Swarm Optimization



Algorithmic Approach

Hybrid Genetic Swarm Algorithm



Algorithmic Approach

Encoding

Genotype *x* → n-dimensional joint variable configuration

$$x = \left(x_1 \mid x_2 \mid x_3 \mid \ldots \mid x_{n-1} \mid x_n \right)$$

→ Independent of joint types (revolute, prismatic, ...)
→ Joint limits directly incorporated (clipping)
→ Allows algebraic vector calculations

Algorithmic Approach

Fitness Function

Idea

Use randomized weight *w* for multi-objective optimization

→ Models **dynamic environment**
→ Determines individuals that are ***„most responsive to change“***
→ Biologically plausible

Phenotype *X* → Cartesian configuration obtained by forward kinematics function *f*

$$\mathcal{X} = f(x)$$

Fitness function Ω measures fitness π under evolutionary target *Y*

$$\pi = \Omega(\mathcal{X}) \qquad \Omega : \begin{cases} d_t(\mathcal{X}, \mathcal{Y}) & Position \\ d_r(\mathcal{X}, \mathcal{Y}) & Orientation \\ w\hat{d}_t(\mathcal{X}, \mathcal{Y}) + (1-w) d_r(\mathcal{X}, \mathcal{Y}) & Pose \end{cases}$$

Algorithmic Approach

Parent Selection

Rank-based parent selection from mating pool Γ

$$\mathcal{S}_{\mathcal{P}} : \mathcal{P}_{\{1,2\}} \leftarrow p(\Gamma_i) = \frac{\gamma - i + 1}{\sum_{i=1}^{\gamma} i}$$

→ Independent of fitness value distribution
→ Sensitive to local extrema
→ Scales well with population size
→ No parameters required

Algorithmic Approach

Recombination

Idea

Let offspring dive a little deeper into the direction that caused improvement within their parents

→ *Heuristic evolutionary gradients for continuous evolution dynamics*

→ ***Evolutionary gradient*** = *amount of change during* ***mutation*** *and* ***adoption*** *(see later)*

$$\mathcal{R} : x_i = \underbrace{w_i x_i^{\mathcal{P}_1} + (1 - w_i) x_i^{\mathcal{P}_2}} + \underbrace{r_{\mathcal{P}_1} g_i^{\mathcal{P}_1} + r_{\mathcal{P}_2} g_i^{\mathcal{P}_2}} \qquad \underbrace{\mathcal{R}_r = 1.0}$$

Randomly interpolated genotype of parent chromosomes

Randomly weighted swarm dynamics from evolutionary gradient *g*

Constant recombination rate

Algorithmic Approach

Mutation

Motivation

Fixed mutation rate and strength not suitable for arbitrary kinematic models

Idea

→ Let the population itself determine the required amount of **exploitation** and **exploration** by an **extinction rate** that **controls mutation** independent from the problem dimensionality

$$\mathcal{M} : x_i \leftarrow x_i + w_i \mathcal{M}_\beta \qquad \mathcal{M}_r = \mathcal{M}_\alpha \frac{1}{n} \qquad \xi^{\Psi_i} = \frac{\pi_i + \pi_{min}(\frac{i-1}{\psi-1} - 1)}{\pi_{max}}$$

Mutation Strength

- Can grow arbitrarily small to exploit
- Exploration limited to the domain of the search space dimension

$$\mathcal{M}_\beta = \xi^{\mathcal{P}_\varnothing} \Delta n_i$$

Mutation Rate

- Normalized between [1/n; 1.0]
- Adaptive to arbitrary dimensionality

$$\mathcal{M}_\alpha = \xi^{\mathcal{P}_\varnothing}(n-1) + 1$$

Extinction Factor

- Normalized between [0.0; 1.0]
- Measures relative quality of an individual within population
- Controls whole mutation

Algorithmic Approach

Adoption

Idea

Let offspring **adopt characteristics** of
parents and most successful performing prototypes

→ Natural behaviour over lifetime
→ Dynamic search space exploration
→ Very similar to PSO

$$\mathcal{A} : x_i \leftarrow x_i + \underbrace{w_i r_{\mathcal{P}} \left(\frac{x_i^{\mathcal{P}_1} + x_i^{\mathcal{P}_2}}{2} - x_i \right)}_{\text{Averaged gradient to parents}} + \underbrace{(1 - w_i) r_* (x_i^* - x_i)}_{\text{Gradient to best individual}}$$

Averaged gradient to parents

Gradient to best individual

Randomly interpolated

Algorithmic Approach

Niching

Pre-Selection

→ Immediately remove any parent whose fitness is worse than its offspring

Goals

→ Encourages to keep track of multiple local extrema
→ Avoid premature convergency

Sebastian Starke
9starke@informatik.uni-hamburg.de

Algorithmic Approach

Survivor Selection

Age-based with elitism

→ Merge all elites and newly generated offspring

Goals

→ Let the fittest individuals *(elites)* survive in order not to lose the current evolutionary progress

Algorithmic Approach

Exploitation

Motivation

Increase convergency by further improvement of already good solutions

Approach

Iterative cyclic exploitation of **elite** individual genotypes

1. *Randomly modify each gene by current fitness value into both domain directions*
2. *Take modification that scored improvement*
3. *Calculate averaged fitness value*

Sebastian Starke
9starke@informatik.uni-hamburg.de

Algorithmic Approach

Initialization

Biased population

1 individual → currently assigned joint variable configuration

All others → randomly generated chromosomes

Termination

Determine whether desired accuracy in

position and **orientation**

is satisfied by the solution

Wipe

Problem

- Many local extrema within search space
- Escape from dead-end paths in good extrema might take too long
- Restart might be more efficient

Approach

- Perform wipe *(restart)* of population if solution

→ could not be improved within last generation
and
→ can not be improved by exploitation

Algorithmic Approach

Solution Filtering

Motivation

Real-time interactive applications → best solution so far is required

Problem

Randomized weights might replace elites by constant dynamics

Solution

Determine joint variable solution by equally weighted objective function

→ Remember best solution in history
→ Let evolution continue approximation underneath

Experiments and Results

Environmental Setup

Laptop – ASUS G751JY

Intel Core i7-4720HQ (2.6 – 3.4 GHz)

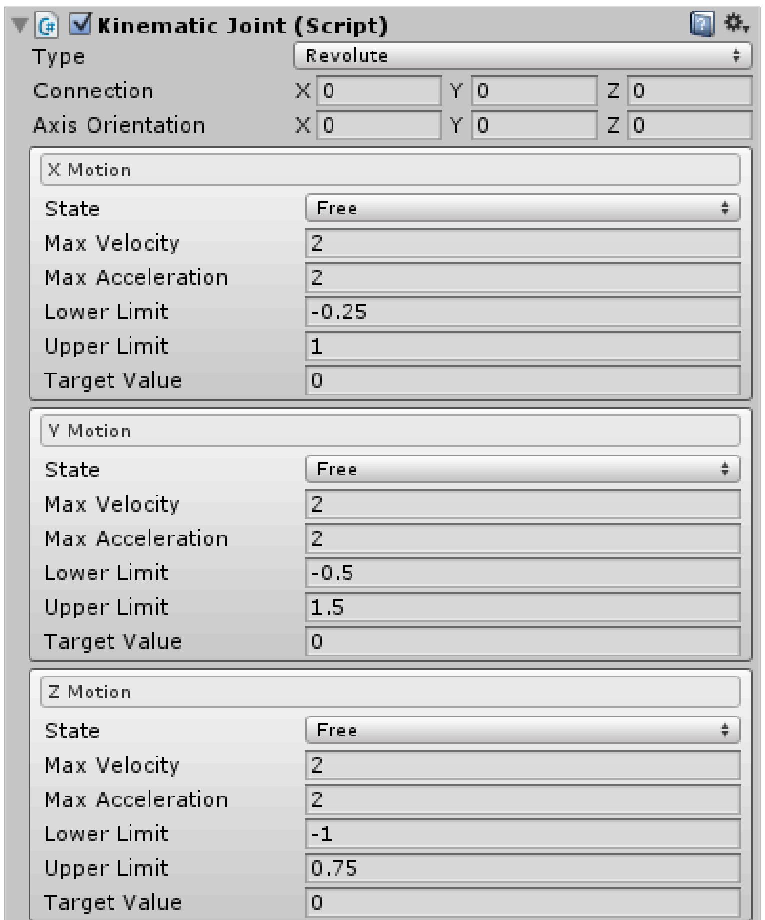
(Code running at single core implementation)

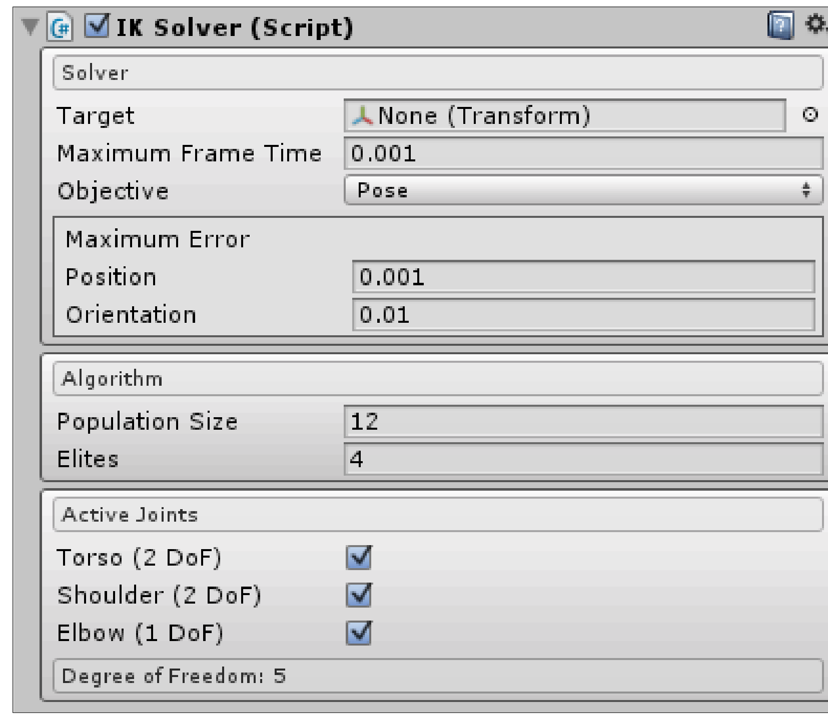
unity + **URDF Importer** + **Kinematic Joint** + **IK Solver**

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Experiments and Results

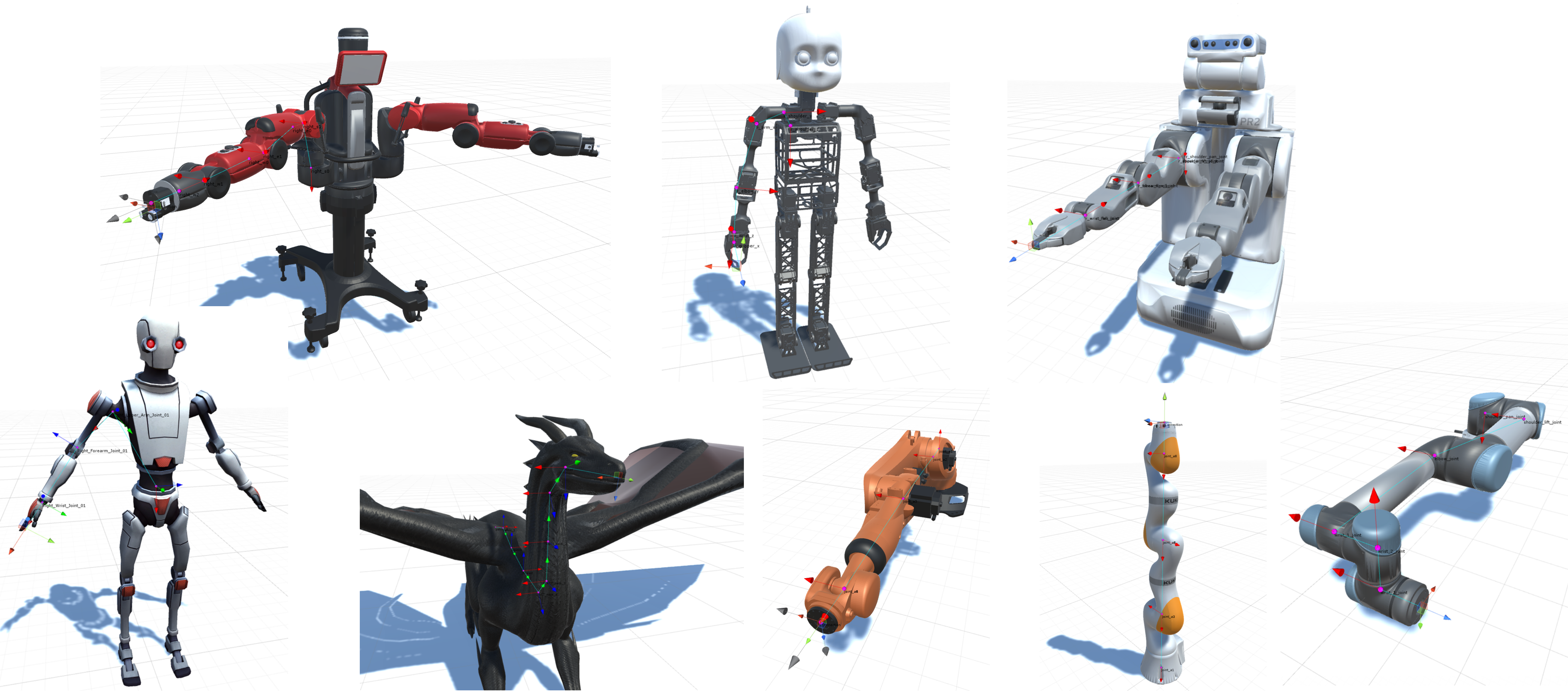
Environmental Setup





Experiments and Results

Environmental Setup



Experiments and Results

– Live Demos in Unity –

1 - https://www.youtube.com/watch?v=dRCY848mSLI
2 - https://www.youtube.com/watch?v=DZFeU_WZlhI
3 - https://www.youtube.com/watch?v=8-kw7RsuD6A
4 - https://www.youtube.com/watch?v=OXtGbrl7qUQ
5 - https://www.youtube.com/watch?v=Gu0CBf18Zf0
6 - https://www.youtube.com/watch?v=a9QPXud-j0Q

Experiments and Results

1. Parameters

→ *Balancing of population size and elites*

→ *UR5 Arm*

2. Selective Study

→ *All versus Nothing*

→ *Selectively taking away improvements*

→ *UR5 Arm*

3. Performance Study

→ *Success, Accuracy, Time, Displacement, Flexibility*

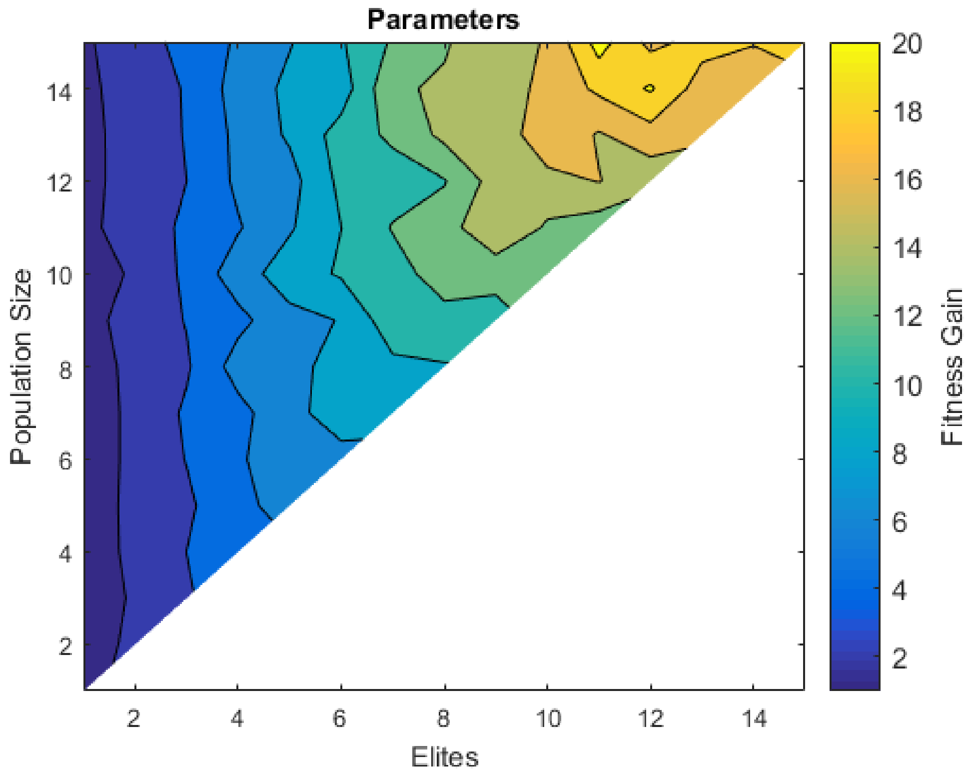
→ *8 Models + 3 high dimensional chains*

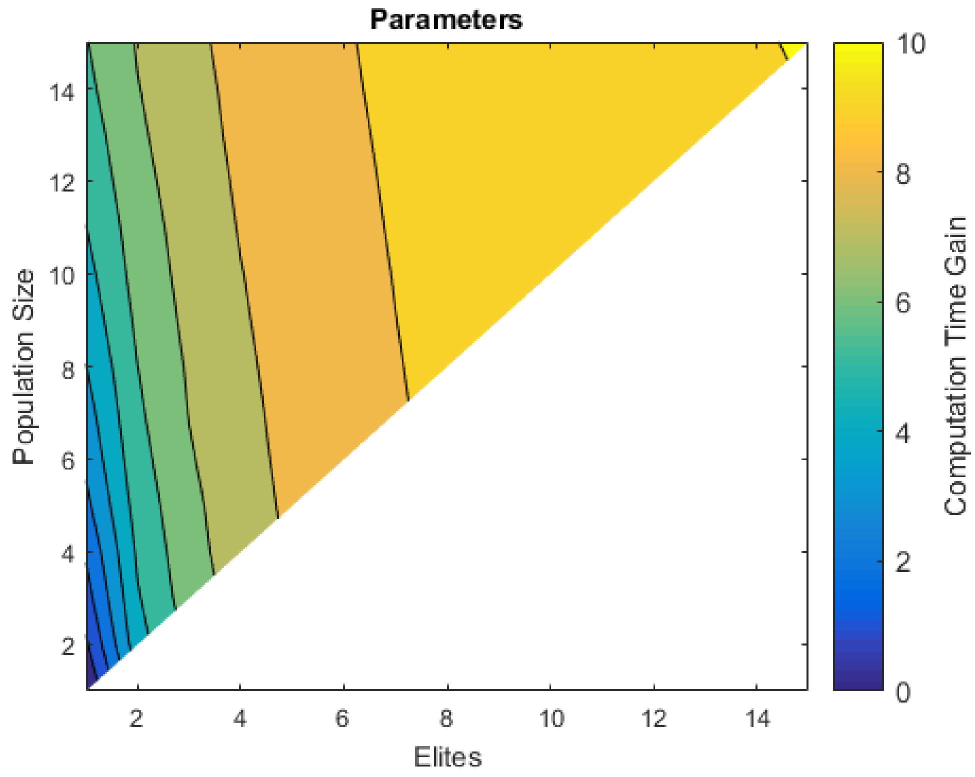
4. Data Comparison

All samples generated as **random poses** or **continual trajectories** of joint space configurations

Experiments and Results

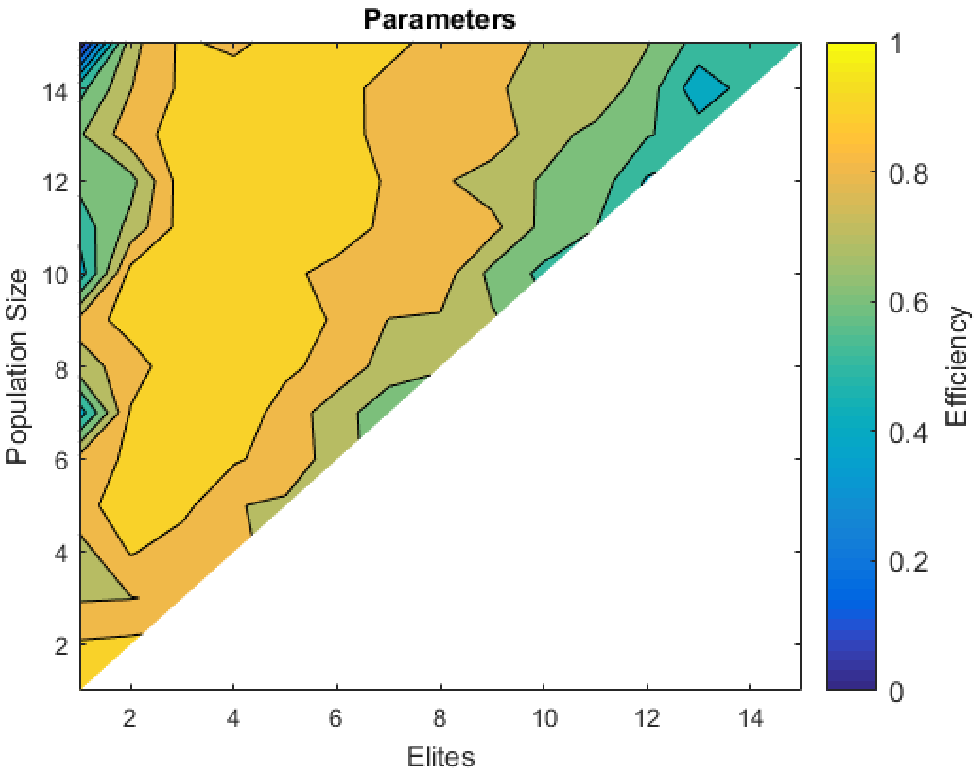
Parameters





Experiments and Results

Parameters



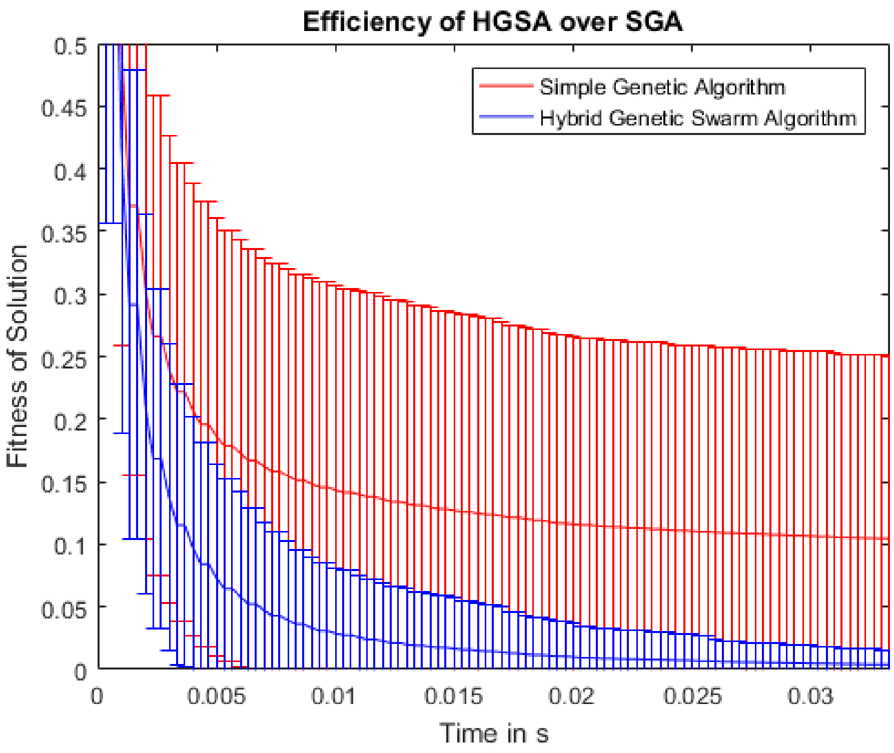
Reasonable parameter choice

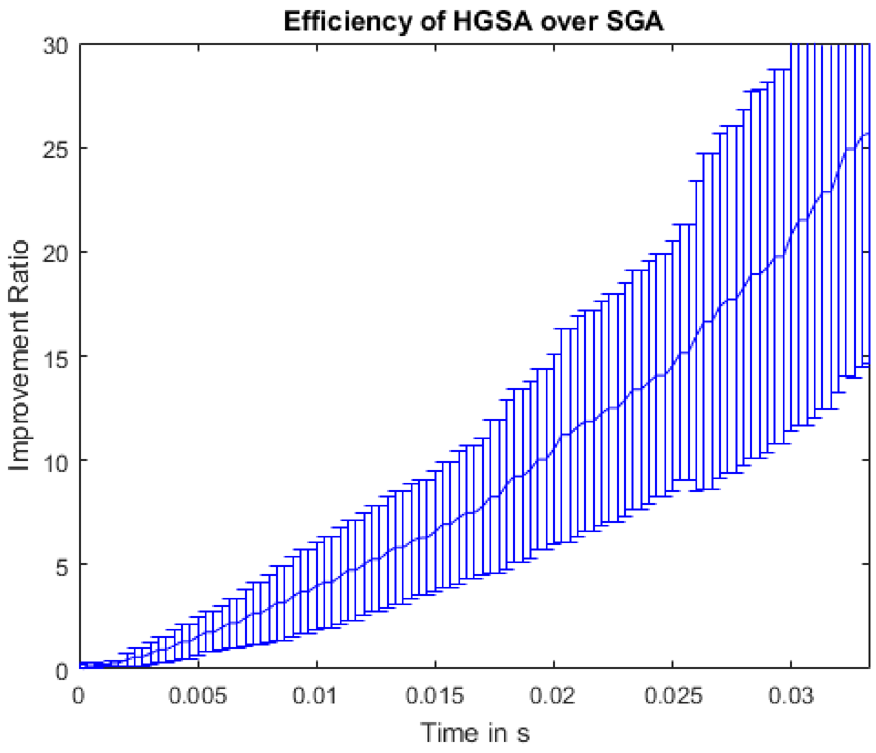
Population Size → 12

Elites → 4

Experiments and Results

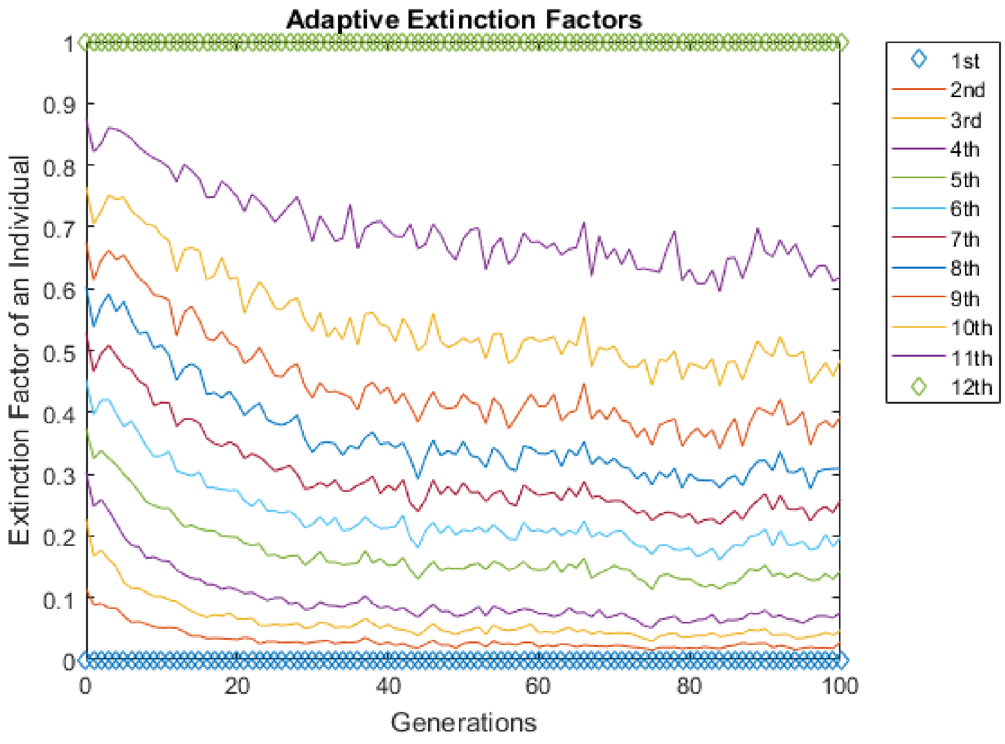
Selective Study – All versus Nothing



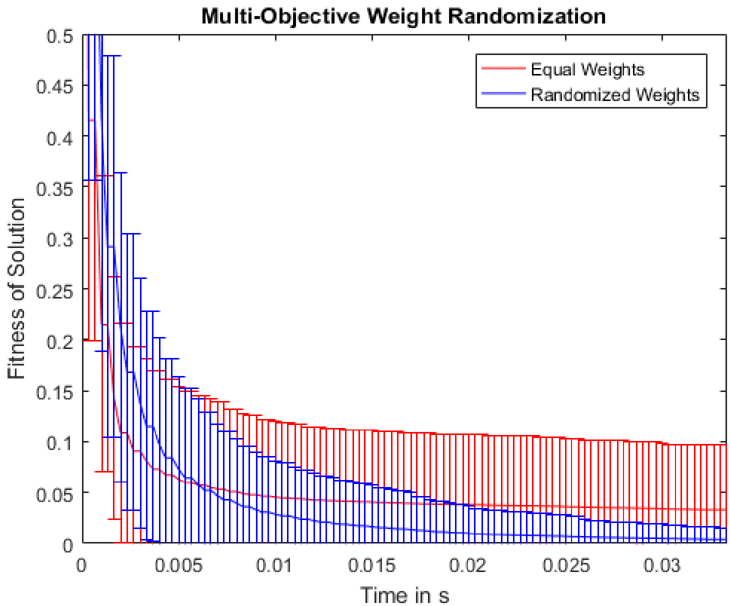


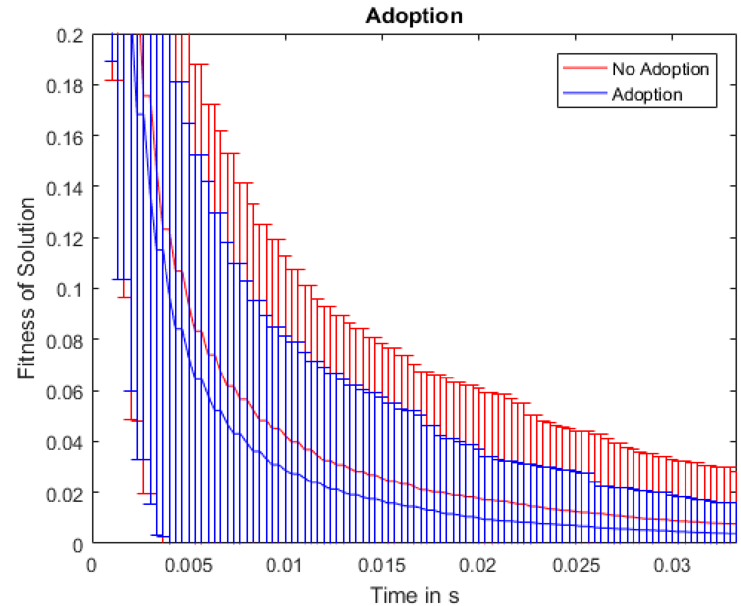
Experiments and Results

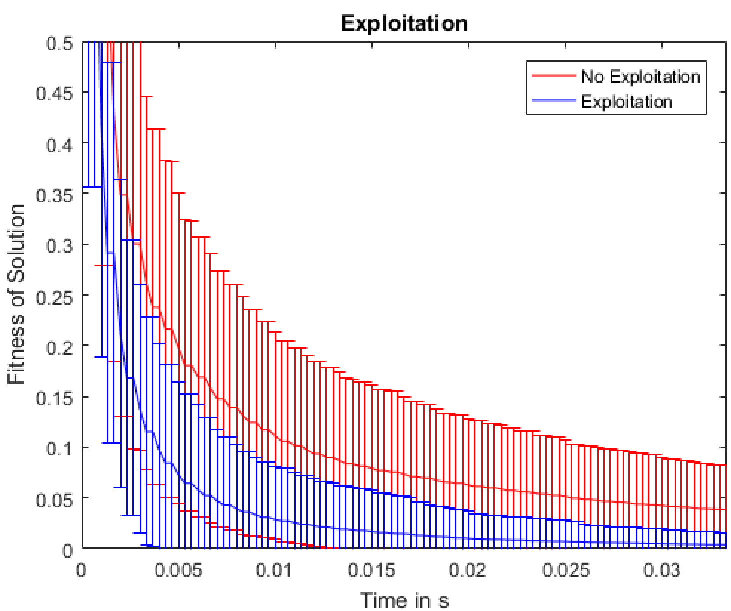
Selective Study – Extinction Factors

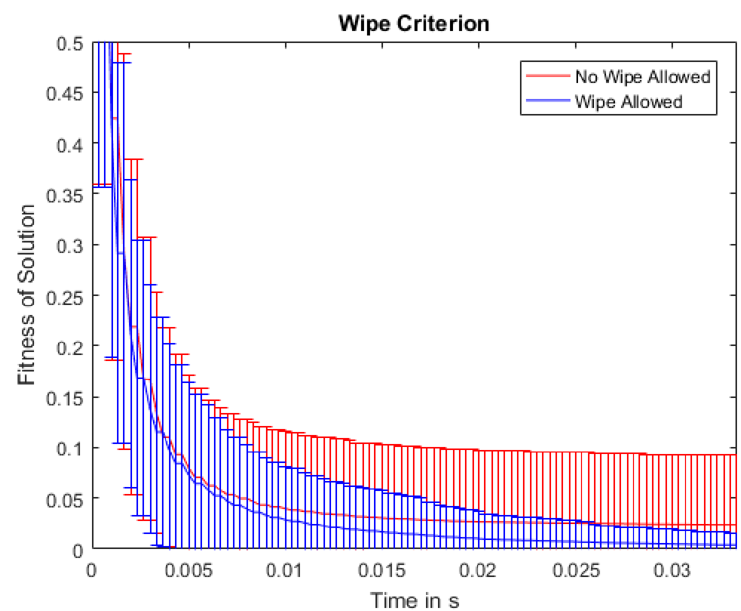


- Extinction factors **adapt** to the current evolutionary progress
- Maintains **explorative diversity**
- Ensures **local extrema sensitivity**


Multi-Objective Weight Randomization
Equal Weights
Randomized Weights
Fitness of Solution
Time in s

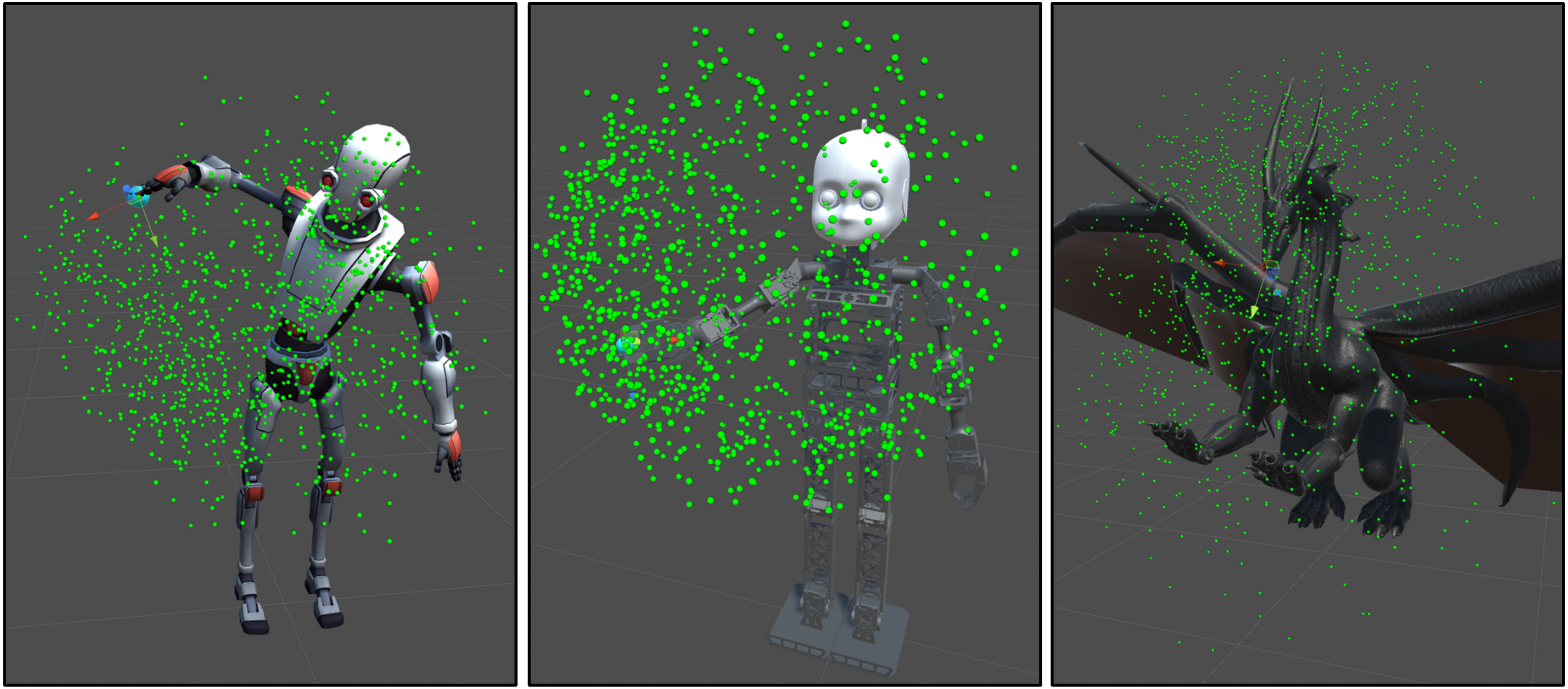

Adoption
No Adoption
Adoption
Fitness of Solution
Time in s


Exploitation
No Exploitation
Exploitation
Fitness of Solution
Time in s


Wipe Criterion
No Wipe Allowed
Wipe Allowed
Fitness of Solution
Time in s

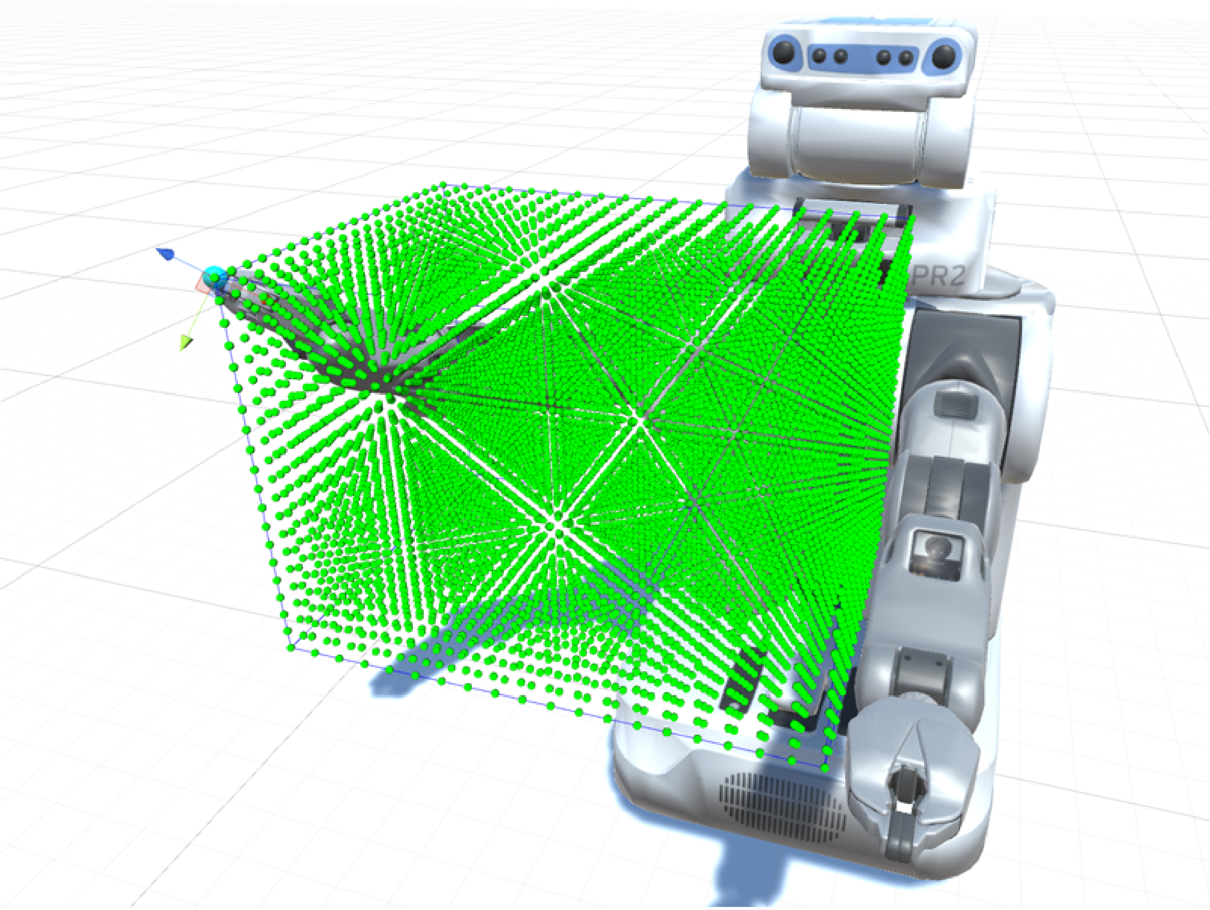
Experiments and Results

Performance Study – Success *(Pose Objective)*



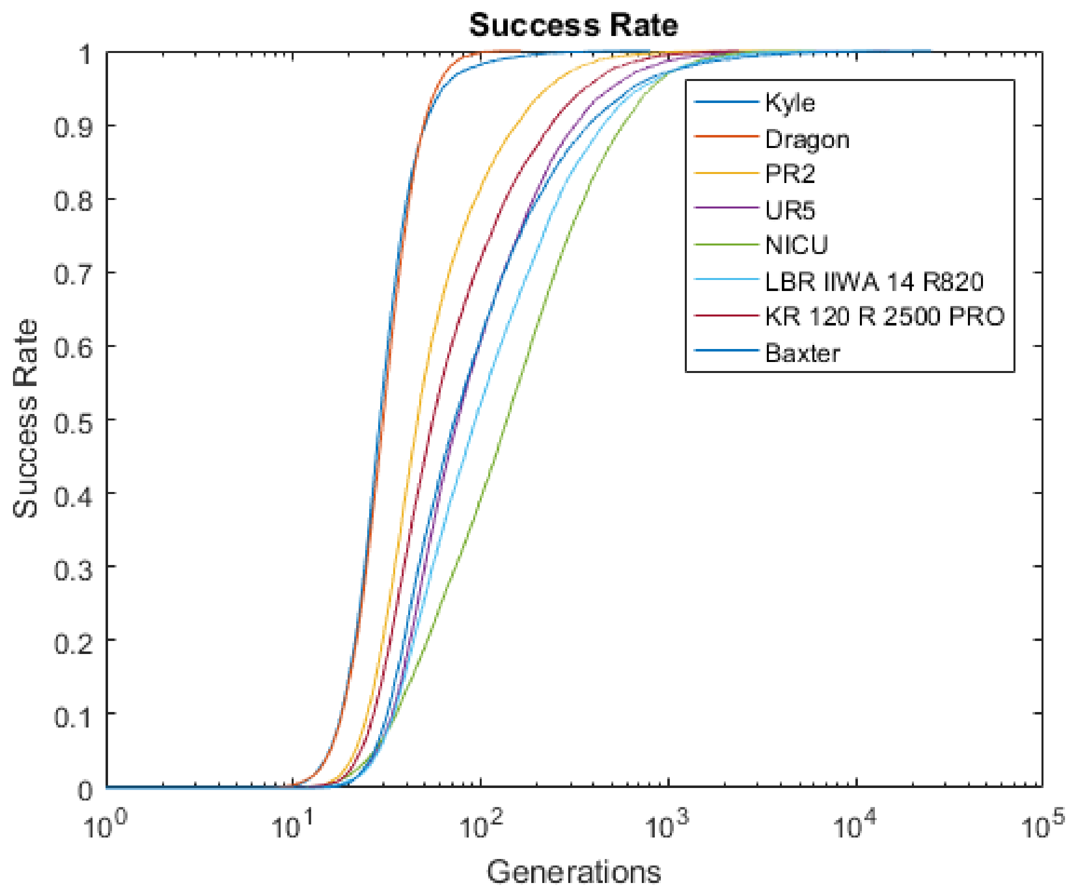
Experiments and Results

Performance Study – Success *(Position Objective)*



Experiments and Results

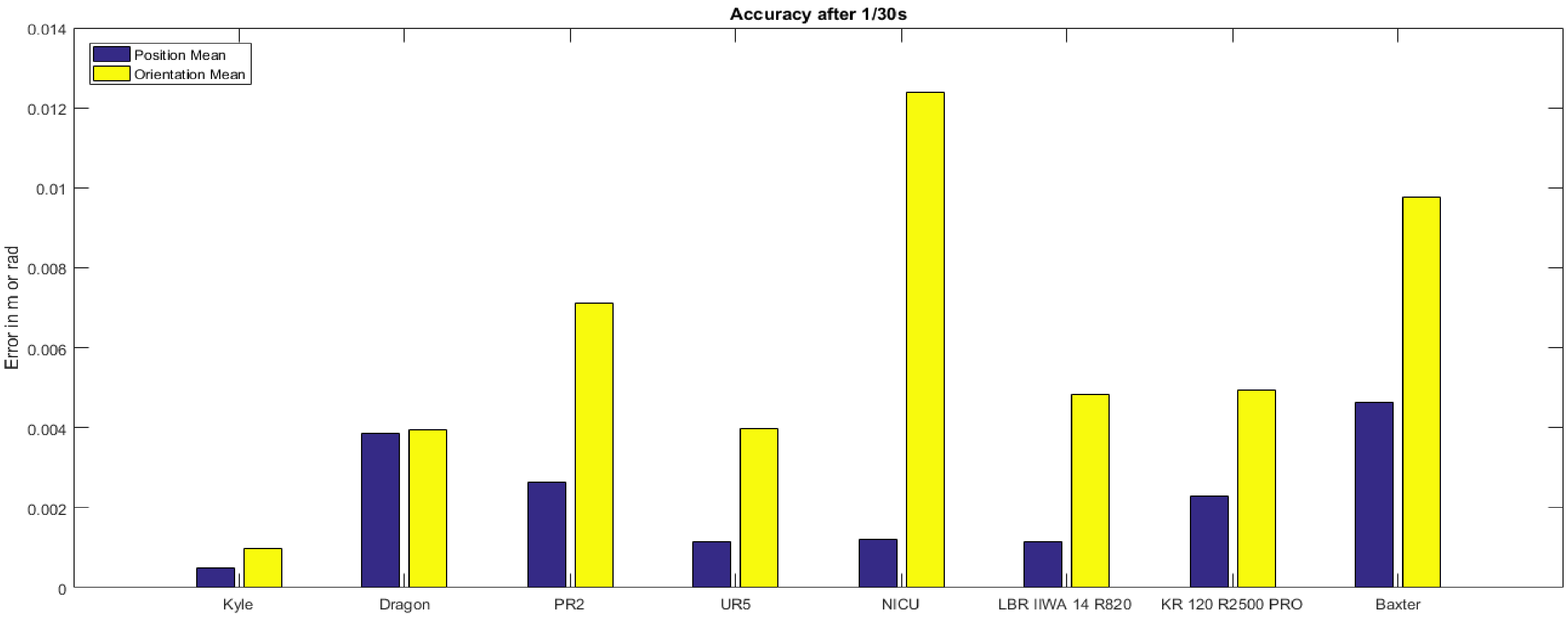
Performance Study – Success



All 10.000 random poses for 8 different models were found.

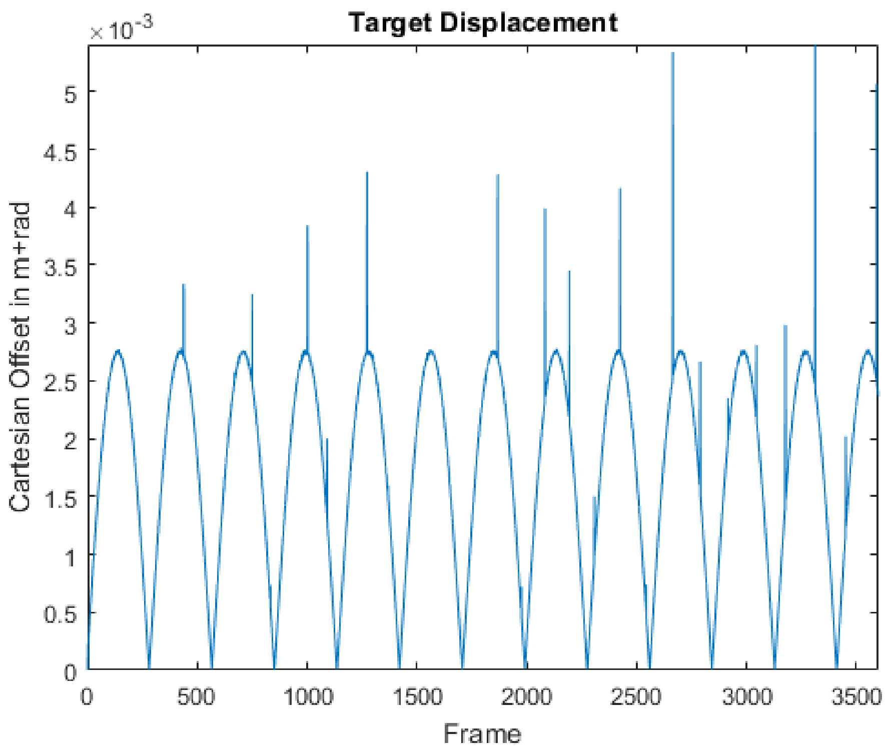
Experiments and Results

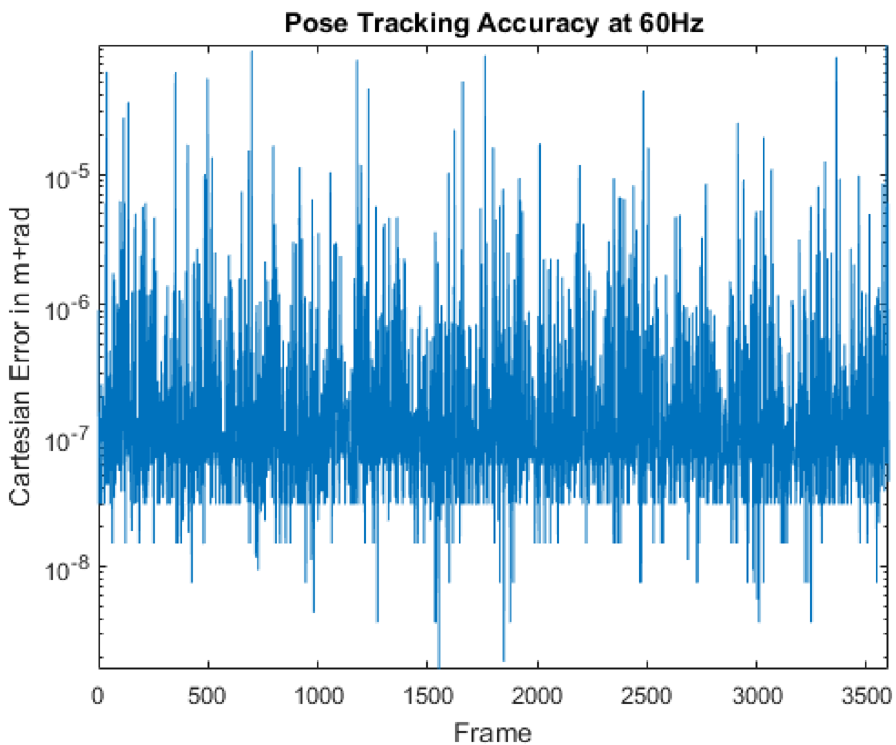
Performance Study – Accuracy



Experiments and Results

Performance Study – Accuracy

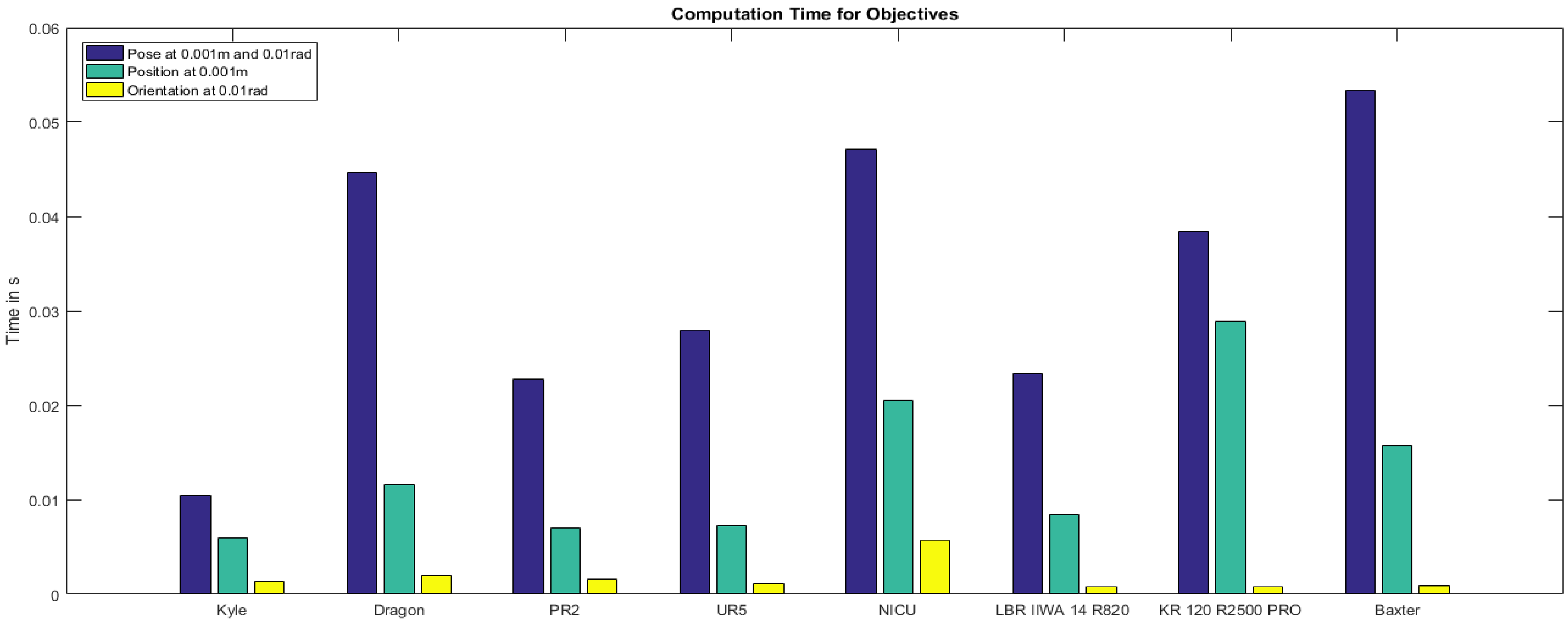




→ **Pose tracking with 10^(-6) cartesian error** *(sum of position and orientation errors)*

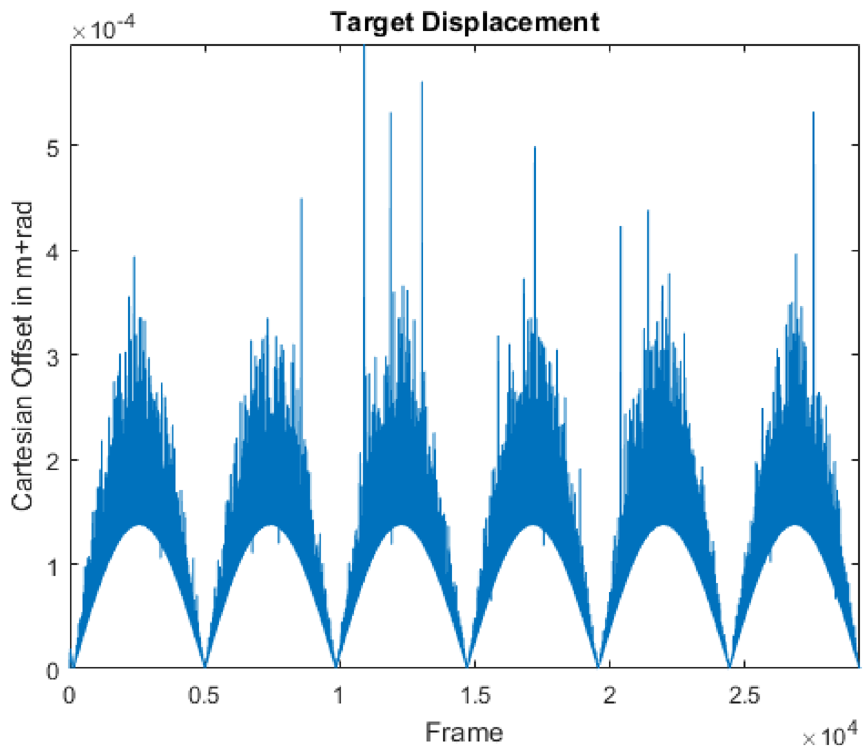
Experiments and Results

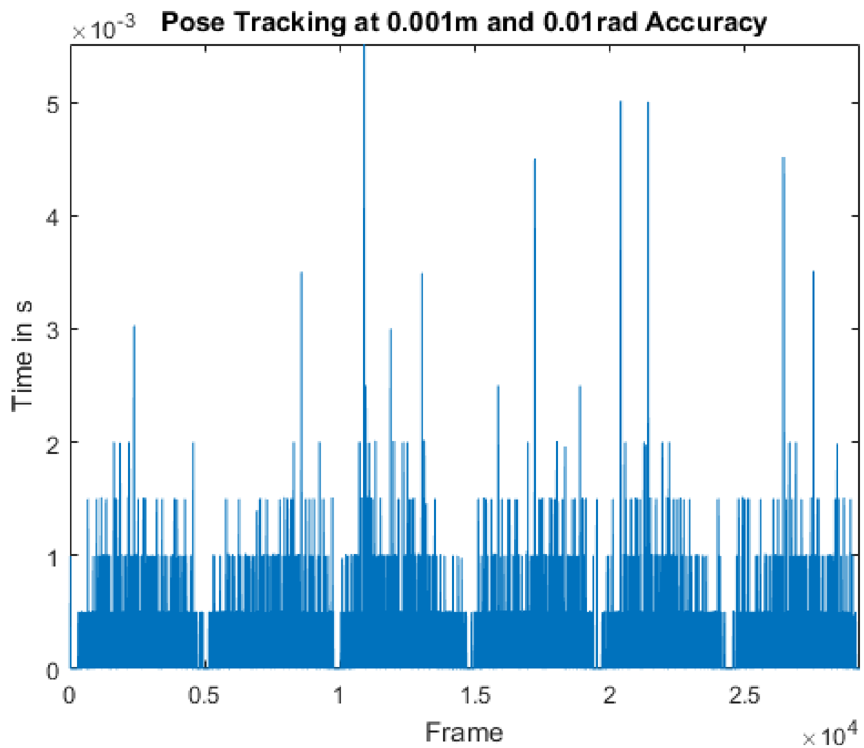
Performance Study – Time



Experiments and Results

Performance Study – Time

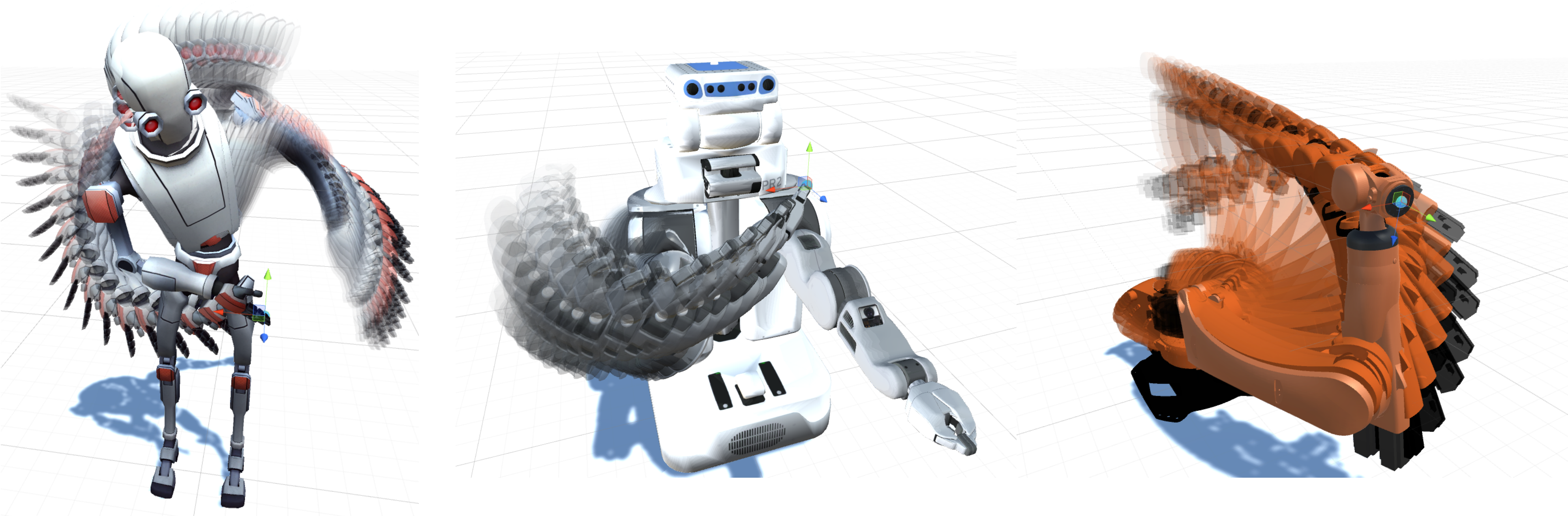




→ Accurate pose tracking at 1000-2000 Hz

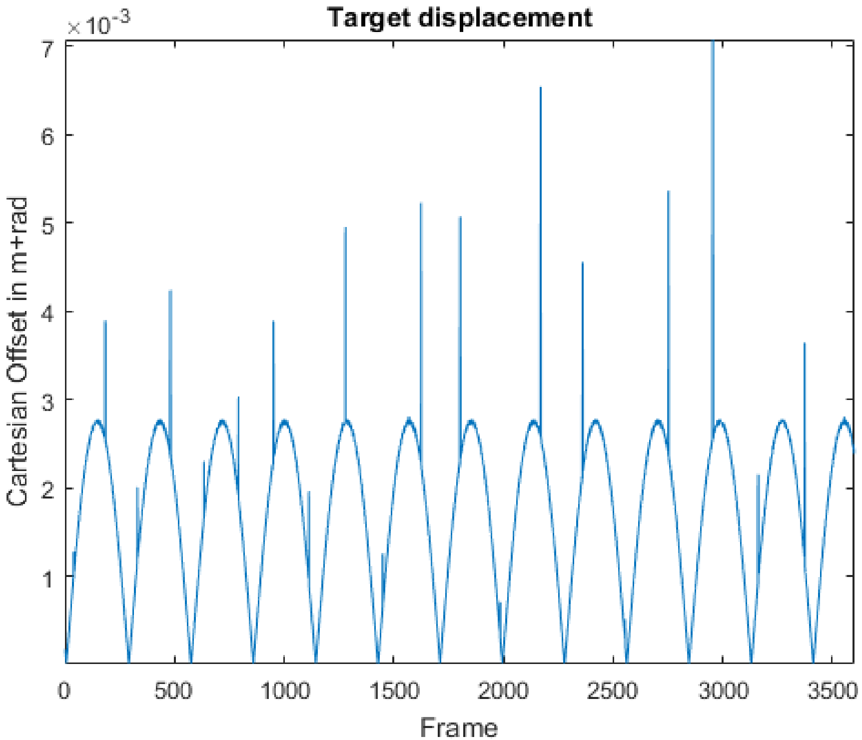
Experiments and Results

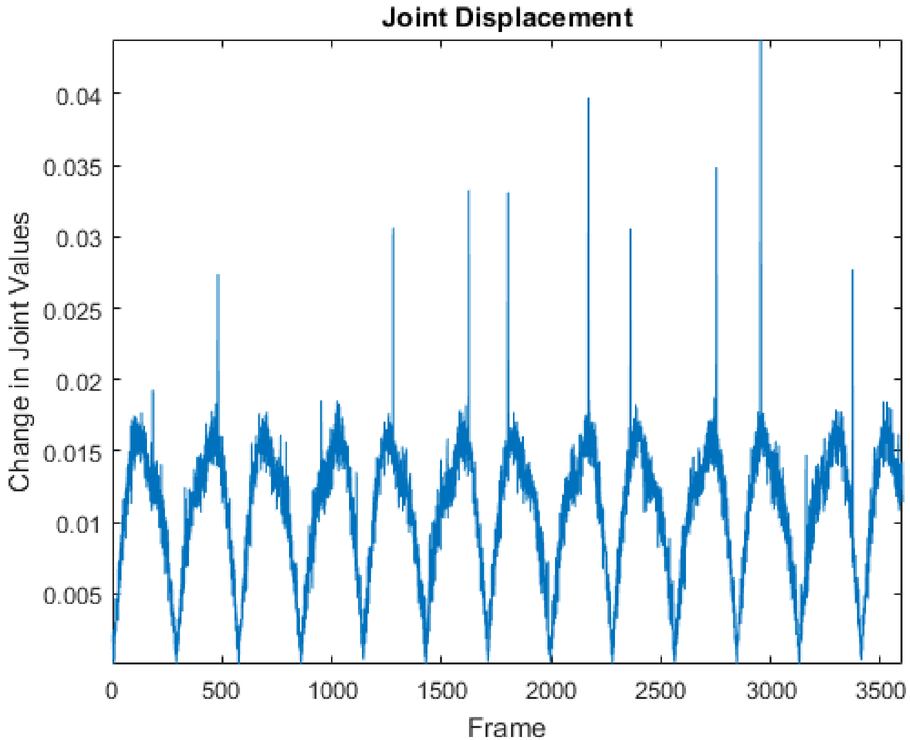
Performance Study – Displacement



Experiments and Results

Performance Study – Displacement

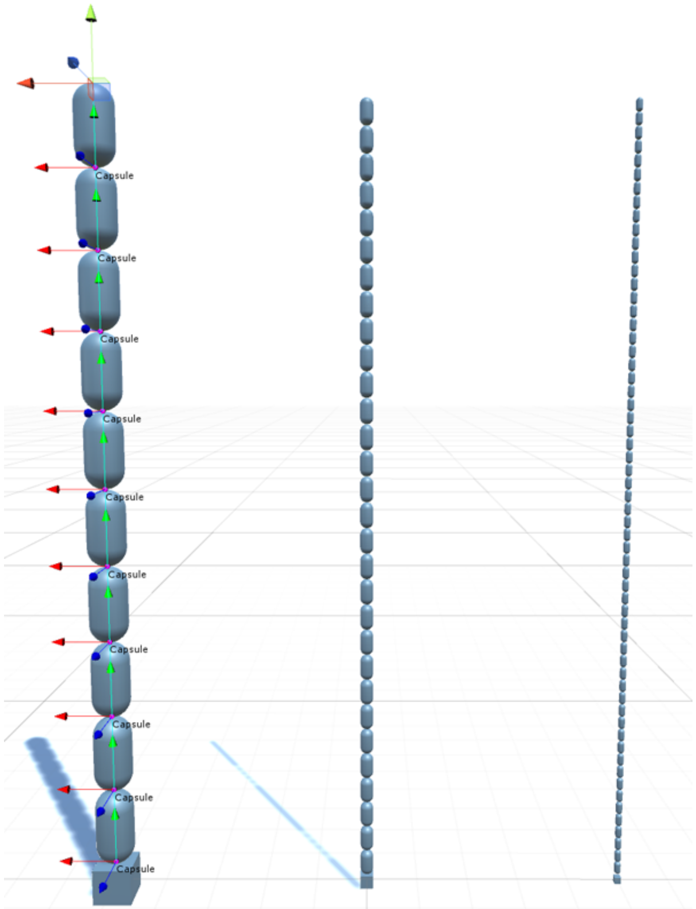




→ Responsive shape-resembling joint value change

Experiments and Results

Performance Study – Flexibility

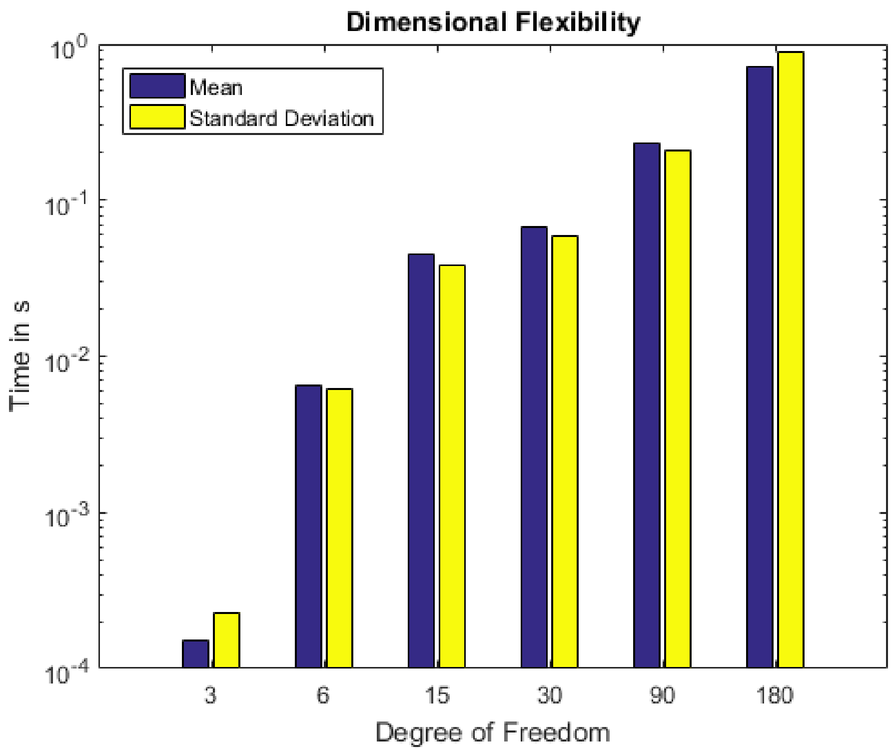


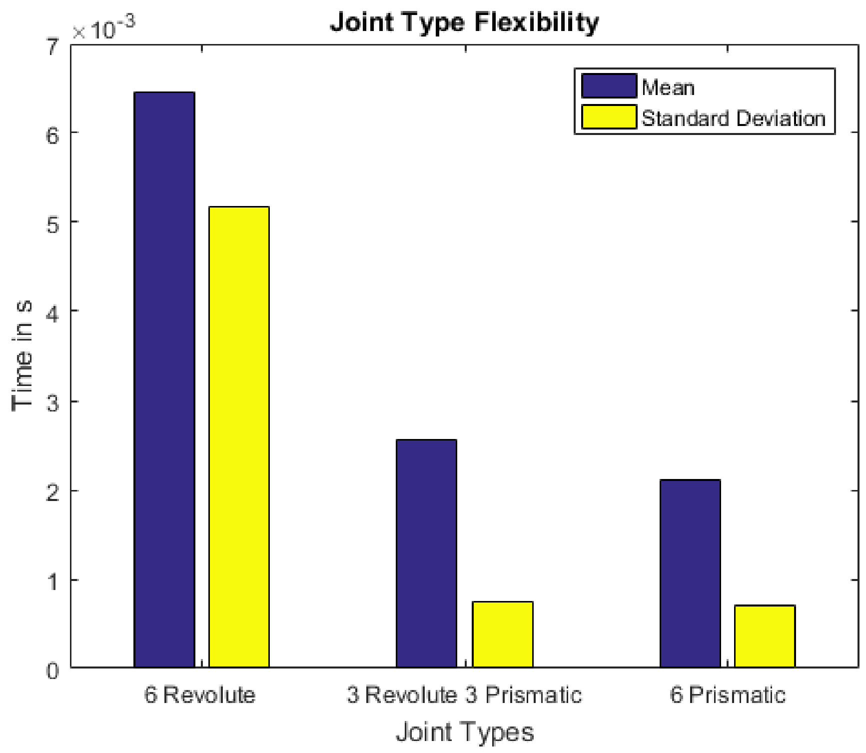
Spherical Joints

left – 30 DoF
middle – 90 DoF
right – 180 DoF

Experiments and Results

Performance Study – Flexibility





Experiments and Results

Data Comparison
Reported efficiency in literature on various robot models

	Jacobian	CCD	FABRIK	GA	PSO	NN	HGSA
Accuracy *(m | rad)*	0.00001	0.00001	0.0001	0.001	0.001	0.01	0.001
Time *(ms)*	<1 - 10	1 - 100	10-50	50 - 500	30 - 600	-	10 - 60
Robustness	Medium	Medium	Low	High	Medium	Low	High
Scalability	– –	–	+	+	+ +	– –	+ +

Experiments and Results

Data Comparison
Orocos' KDL vs TRAC-IK vs HGSA

	Orocos' KDL *(no joint limits)*		TRAC-IK *(Beeson & Ames, 2015)*		HGSA	
	Success	**Time**	**Success**	**Time**	**Success**	**Time**
PR2	83.14%	1.37ms	99.84%	0.59ms	100%	22.75ms
LBR IIWA	37.71%	3.37ms	99.63%	0.56ms	100%	23.39ms
UR5	35.88%	3.30ms	99.55%	0.42ms	100%	27.88ms
Baxter	61.07%	2.21ms	99.17%	0.60ms	100%	53.33ms

Scalability?

Joint displacement due to random restarts?

Experiments and Results

Data Comparison
PASO vs HGSA

	PASO *(Collins & Shen, 2016)*	HGSA
	Time	**Time**
30 DoF	1.57s	0.066s
90 DoF	7.46s	0.233s
180 DoF	37.03s	0.717s

Sebastian Starke
9starke@informatik.uni-hamburg.de

Conclusion

General

- *Universal IK solver for kinematic chains*
- *Arbitrary joint types, link geometry and degree of freedom*
- *Algorithm based on biologically-inspired evolutionary and collective concepts*

Algorithmic Improvements

Extinction Factors	Swarm Adoption	Heuristic Exploitation
Multi-Objective Weight Randomization	Wipe Criterion	

Results

- *100% success rate at high accuracy and real-time capability*
- *Minimal joint displacement and high scalability*
- *Maintains performance under various kinematic models*
- *Can compete with the State-of-the-Art*

Sebastian Starke
9starke@informatik.uni-hamburg.de

Future Work

Extinction Factor Study
→ Analyse loss by adaptive mutation control in contrast to model-specific optimized parameters

Parallel GPU / Multi-Core Implementation
→ >95% of time is required by fitness calculation *(FK equations)*
→ Reimplement the algorithm in C++/Python
→ Efficient parallel instead of sequential single-core fitness calculation

Multiple End Effectors
→ Currently, only serial kinematic chains are supported
→ Extend algorithm to solve for multiple end effectors simultaneously

Path Planning
→ Extend algorithm for path-planning tasks

Collision Avoidance
→ Incorporate link geometry to filter solutions that would result in collisions

Neural Learning
→ Learn mappings from previously evolved joint configurations and trajectories

Sebastian Starke
9starke@informatik.uni-hamburg.de

References

S. Starke – *A Hybrid Genetic Swarm Algorithm for Interactive Inverse Kinematics, Master Thesis, 2016*

A.E. Eiben and J. E. Smith – *Introduction to Evolutionary Computing, Springer, 2003*

D. Floreano and C. Mattiussi – *Bio-Inspired Artificial Intelligence, MIT Press, 2008*

A. Aristidou and J. Lasenby – *FABRIK: A fast, iterative solver for the Inverse Kinematics problem, volume 73, pages 243-260. Graphical Models, 2011*

T. Collins and W. Shen – *PASO: An Integrated, Scalable PSO-based Optimization Framework for Hyper-Redundant Manipulator Path Planning and Inverse Kinematics. Information Sciences Institute Technical Report, 2016*

Li-Chun Tommy Wang and Chih Cheng Chen – *A combined optimization method for solving the inverse kinematics problems of mechanical manipulators, volume 7, pages 489{499. IEEE Transactions on Robotics and Automation, 1991*

C. E. Gonzalez Uzcategui – *A Memetic Approach to the Inverse Kinematics Problem for Robotic Applications. Doctoral Thesis, Carlos III University of Madrid, 2014*

P. Beeson and B. Ames – *TRAC-IK: An open-source library for improved solving of generic inverse kinematics. In Proceedings of the IEEE RAS Humanoids Conference, Seoul, Korea, November 2015*

Online – *https://bitbucket.org/traclabs/trac_ik.git (20.06.2016)*
Online – *http://www.unity3d.com (20.06.2016)*
(a) Online – *http://spectrum.ieee.org/image/MjM4NDQzNA (20.06.2016)*
(b) Online – *http://core0.staticworld.net/images/article/2013/12/kuka-robot-arms-at-work-100155065-orig.jpg (20.06.2016)*
(c) Online – *http://www.batou.fr/wp-content/uploads/motion_capture_1.jpg (20.06.2016)*
(d) Online – *http://www.batou.fr/wp-content/uploads/motion_capture_1.jpg (20.06.2016)*
(e) Online – *https://sensor.cs.washington.edu/robotbci/images/PR2.jpg (20.06.2016)*