Multitouch Robot Control

Bachelor Thesis

Merlin Steuer

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Technical Aspects of Multimodal Systems

24. April 2018

Outline

Outline (cont.)

Merlin Steuer

- Part-time student
- B.Sc. Informatik since WiSe13
- 2steuer@informatik.uni-hamburg.de
- Questions? Make some noise!

Motivation & Objective

- Robots get more and more ubiquitous
- Robots enter domestic space[5]
- Interfaces have to be easy and intuitive

$\Rightarrow$ A simple, intuitive remote control interface to complex robots shall be developed.

- Multitouch gestures well known
- Android tablet computer

Grasp Synergies

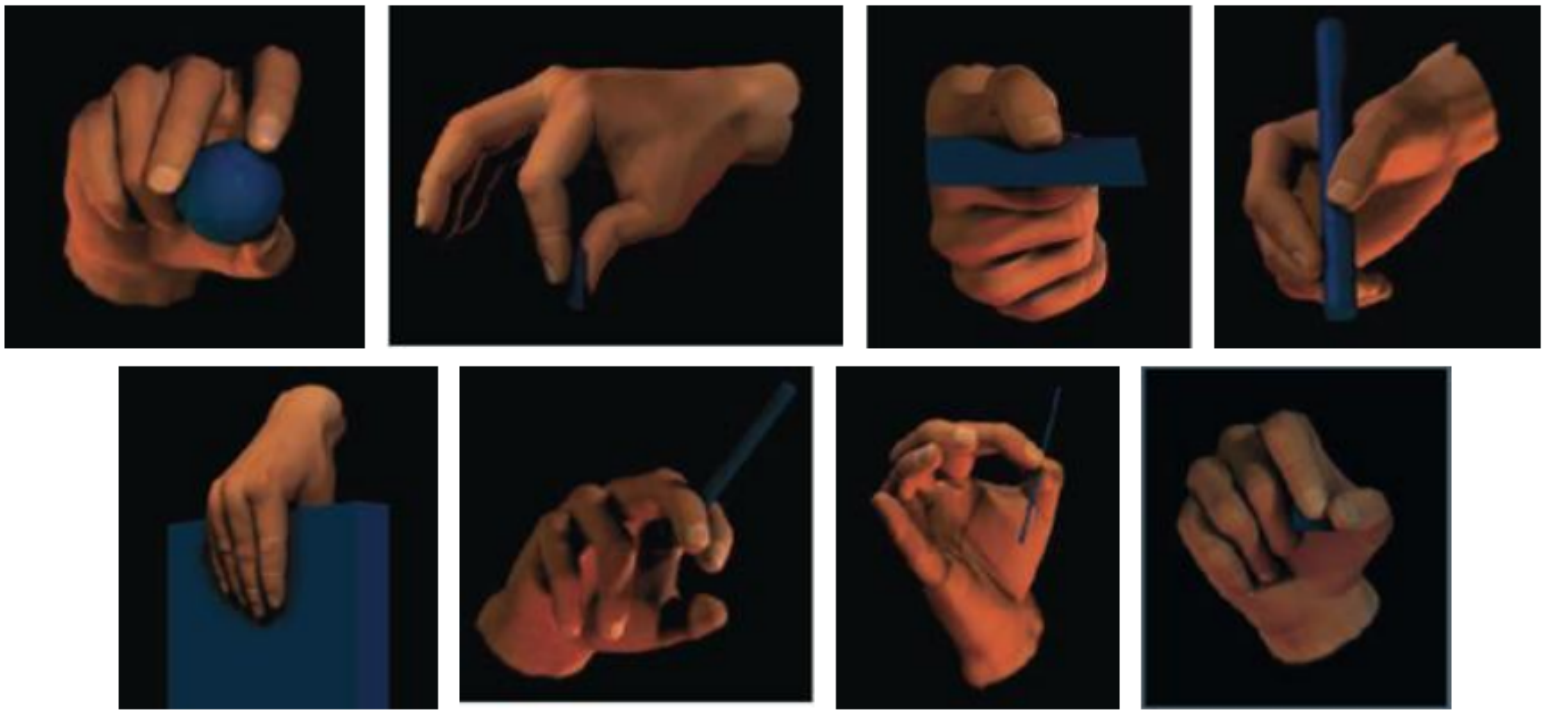
- Research at the TAMS group [2]
- PCA on human grasp postures
- Eigenvectors in matrix S (called *Synergy* here)
- Offset s_0 = mean joint values

$$\theta = s_0 + S\alpha \tag{1}$$

- $\alpha \in \mathbb{R}^{21}$ input amplitudes
- $\theta \in \mathbb{R}^{21}$ joint angles of the hand

$\Rightarrow$ Map properties of gestures to α (Absolute and relative)

Grasp Synergies



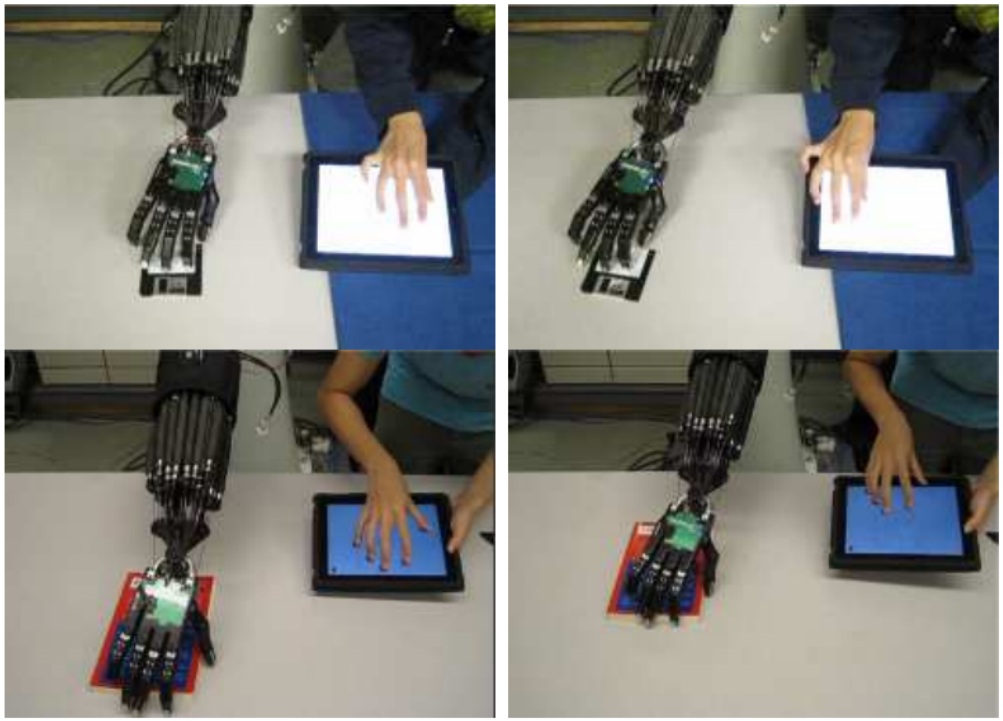
Grasp synergies recorded by Bernardino et al.[2]

Direct Fingertip Mapping

- Research done by Toh et al. [13]
- Grasp actions $\rightarrow$ fingertip movements in a plane

$\Rightarrow$ Map fingertips on touch screen to a plane in $3D$ space

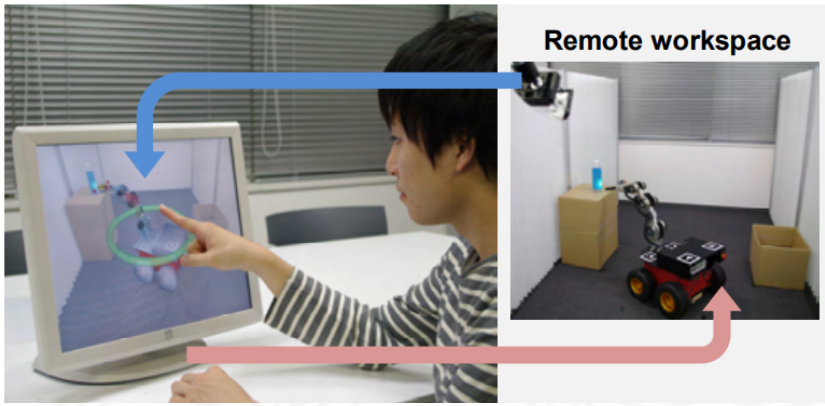
Direct Fingertip Mapping



Examples from Toh et al.[13]

Other Approaches

Controlling a robotic car with an Android phone, research by Akupati[1]



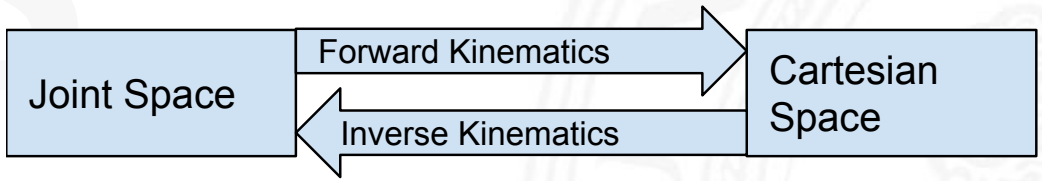
TouchMe! Controlling a robot with touch actions on an AR display. Research by Hashimoto et al.[8]

ROS

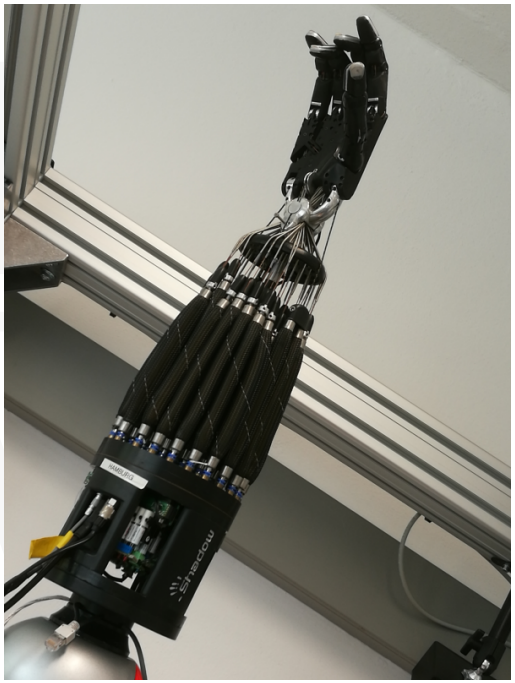
- Robot Operating System
- Software Framework to develop robots
- Communication Framework
- Publisher/Subscriber Pattern
- Services

Inverse Kinematics

- Joint Space θ
- Cartesian Space x
- Forward Kinematics: $x = f(\theta)$
- Inverse Kinematics: $\theta = f^{-1}(x)$



Shadow C5 Hand



The Shadow C5 Robotic Hand

- by *The Shadow Robot Company* [3]
- 24 DOF
- Pneumatic muscles
- Designed after human forearm

Kuka LWR 4+

The Kuka LWR robotic arm

- by *KUKA Roboter GmbH* [6]
- 7 DOF
- Integration into ROS over FRI (*Fast Research Interface*)[7]

Samsung Galaxy Tab S3

The used Android tablet

- by *Samsung* [10]
- 10 inch screen
- 2048x1536 px
- 2.15GHz Quad-Core CPU
- *rosjava* / *rosandroid*

BioIK

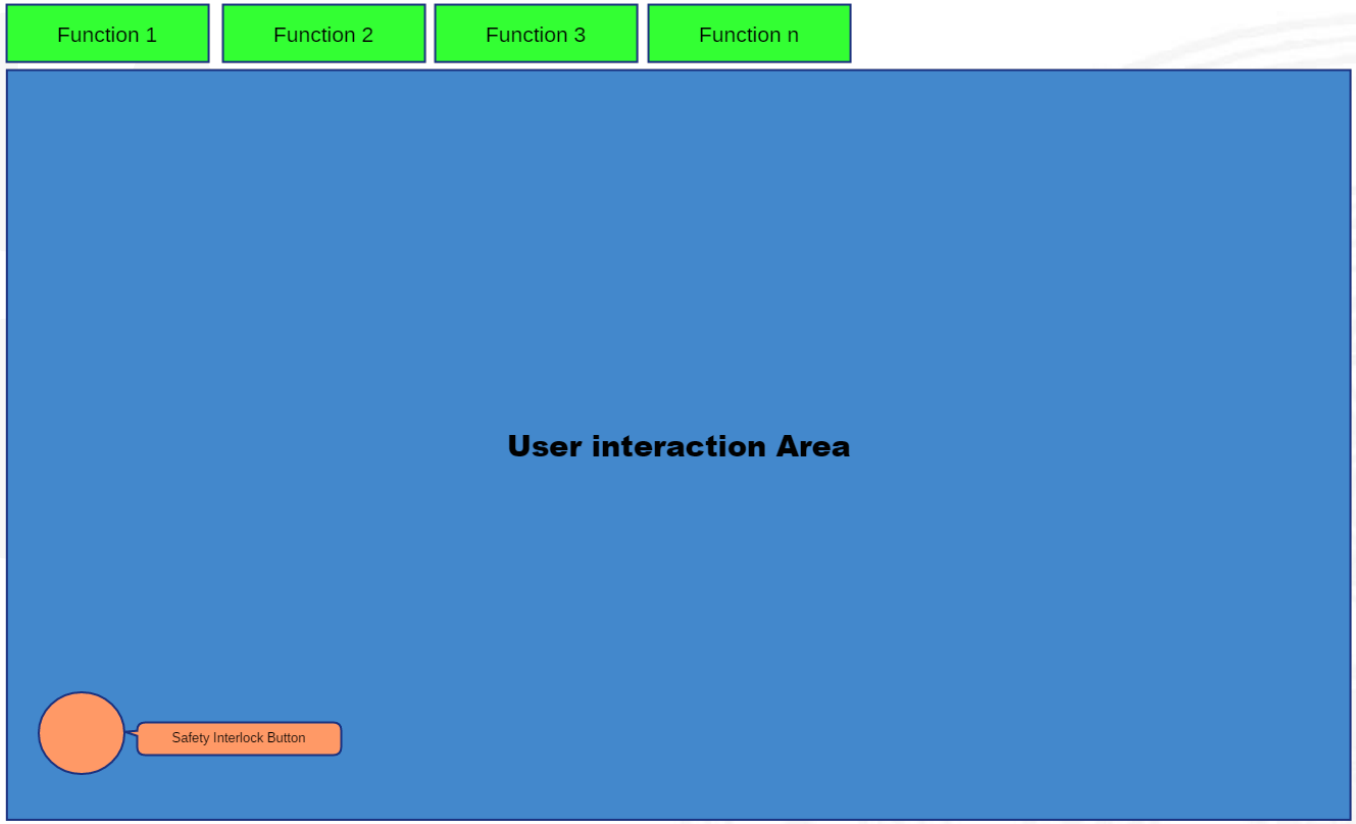
- Evolutionary, multi-goal algorithm for inverse kinematics
- Developed at the TAMS group [11, 12]
- Integrated into ROS by Philipp Ruppel [9]
- Philipp also provided the BioIK ROS service[1]

[1] Found at Philipp's GOGS

User Interface

- Most area reserved for touch interaction
- Functionality interchangeable by tab pages
- Safety interlock button
- **Interlock button does not replace hardware safety measures!**

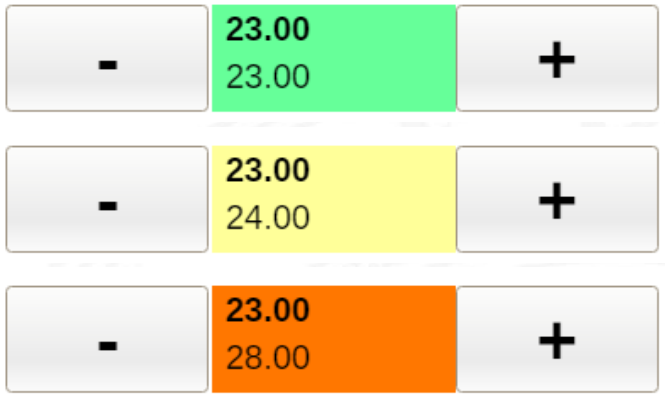
User Interface



General screen layout

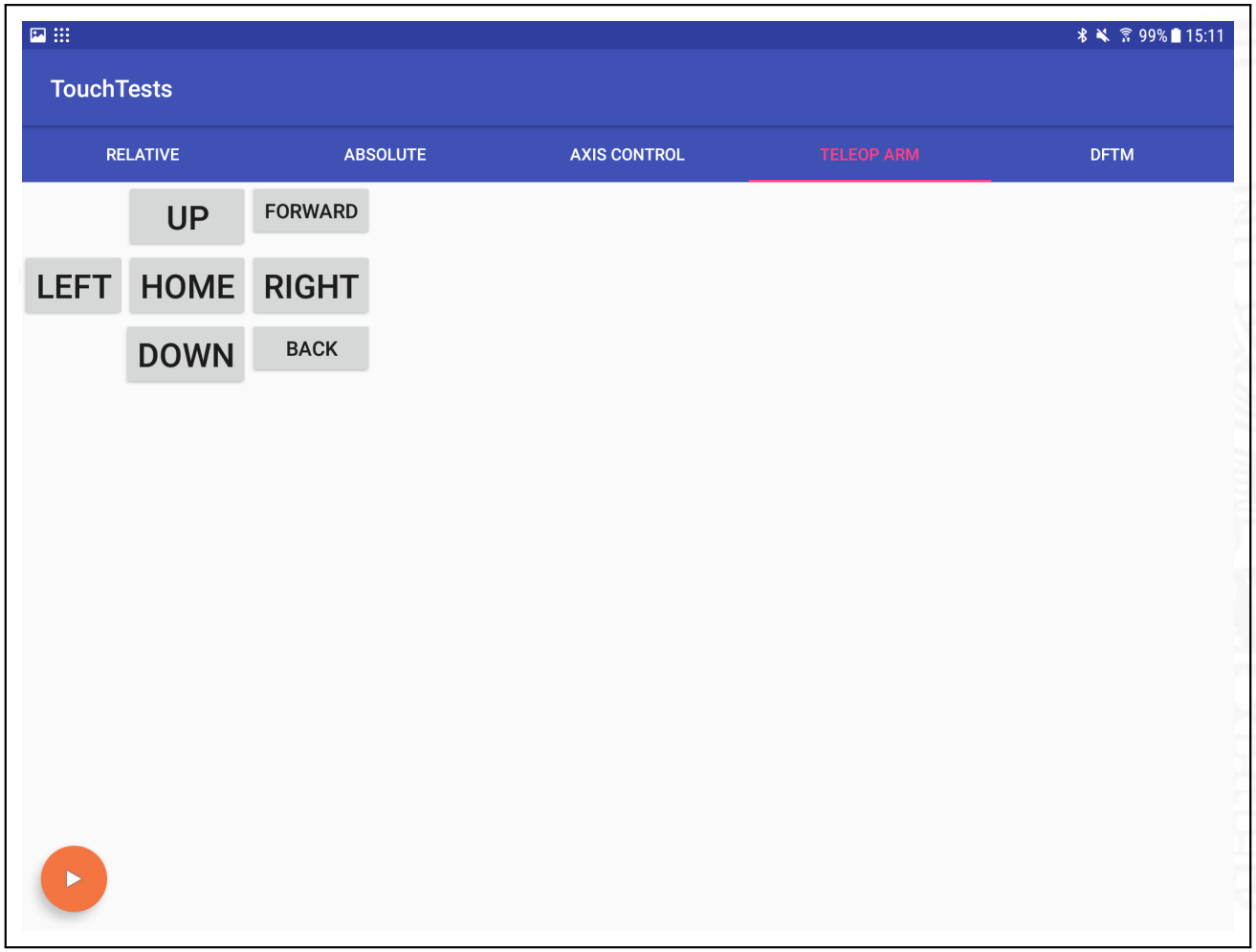
Axis Control Page

- One control for every joint
- +/- buttons to set joint angle
- Display of target value and current value
- Coloured indication of angle error



Axis Control Page

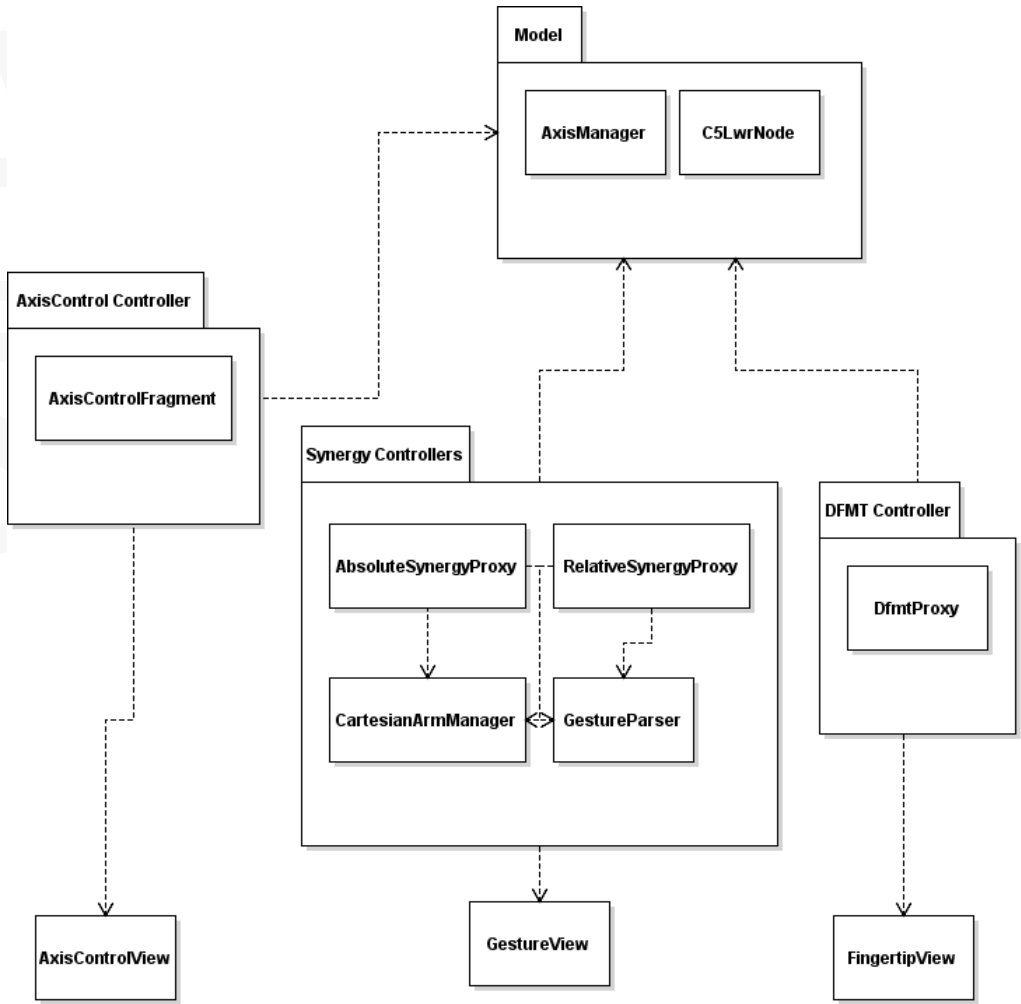
Arm Teleop


99% 15:11
TouchTests
RELATIVE
ABSOLUTE
AXIS CONTROL
TELEOP ARM
DFTM
UP
FORWARD
LEFT
HOME
RIGHT
DOWN
BACK

Software Architecture

- *Model-View-Controller* [4]
- Model: *AxisManager*, *C5LwrNode*
- View: **View*
- Controller: **Proxy*, **Fragment*, *CartesianArmManager*

Model-View-Controller



AxisManager

- *Model* $\Rightarrow$ responsible for all joint data and movement
- Enable / Disable (Safety Interlock)
- Joint Movements
- Target Values
- Maximum Speed

C5LwrNode

- Interface to ROS
- Receives joint information
- Sends joint information from AxisManager
- Methods to call the BioIK service

GraspSynergy

- $\theta = s_0 + S\alpha$
- s_0 and S provided as data files
- Functionality to load data files and calculate joint angles in *GraspSynergy* class (provided by Norman)
- Amplitudes range from -50 to 50

```
1  class GraspSynergy {
2    public void parseMatlabSynergyMean(InputStream is);
3    public void parseMatlabSynergyVecs(InputStream is);
4
5    public double[] toJoints(double[] amplitudes);
6  }
```

Touch Gestures

- Gesture property $\Rightarrow$ amplitude (relative or absolute)
- Position (Center)
- Size
- Orientation

Definitions:

- $p = (p_x, p_y) \in \mathbb{R}^2$ is a **Pointer** on a touch screen
- $G = \{p_1, \ldots, p_n\}$ for pointers p_i, $1 \leq i \leq n$ and $|G| \geq 1$ is a **Gesture**

Gesture Properties

For a gesture $G = \{p_1, \ldots, p_n\}$

- Position:

$$c(G) = \frac{1}{|G|} \sum_{i=1}^{|G|} p_i \qquad p_i \in G \tag{2}$$

- Size:

$$s(G) = \frac{2}{|G|} \sum_{i=1}^{|G|} d(c(G),\, p_i) \tag{3}$$

with $d(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$ for $x, y \in \mathbb{R}^2$

Gesture Properties

For a gesture $G = \{p_1, \ldots, p_n\}$

- *Thumb Pointer*:

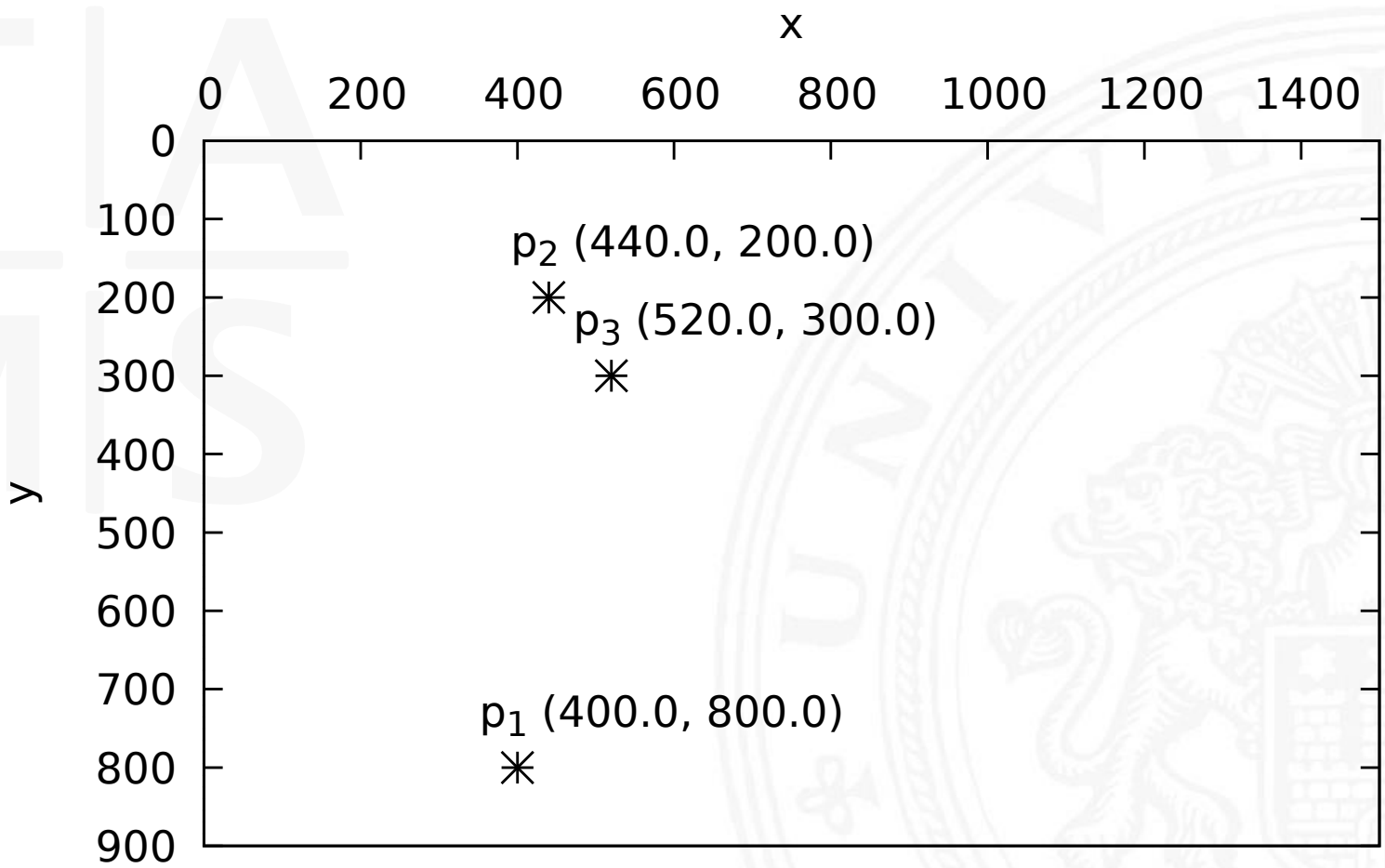
$$th(G) = \begin{cases} p_1 & |G| = 1 \\ p_n \text{ with } p_{n,y} = max\{p_{i,y} : p_i \in G\} & |G| = 2 \\ p_n \text{ with } d(c(G), p_n) = max\{d(c(G), p_i) : p_i \in G\} & |G| > 2 \end{cases} \tag{4}$$

- Orientation, for $v = c(G) - th(G)$, $b_y = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$:

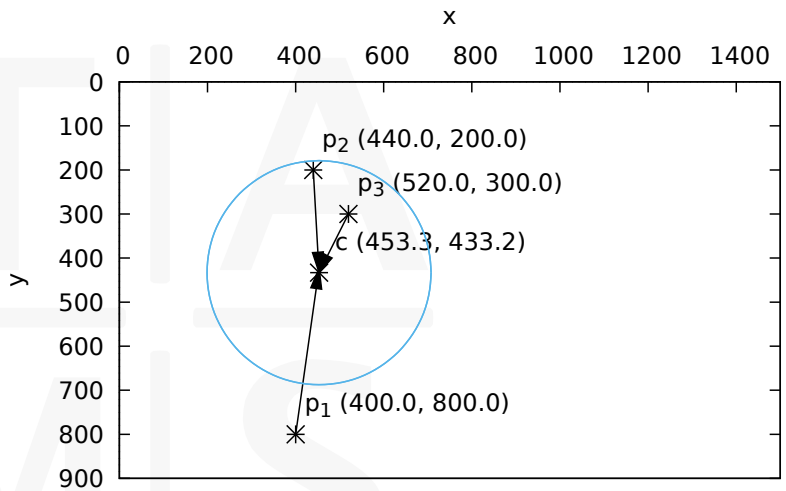
$$o(G) = \text{sign}(\det(b_y\ v)) \cdot \arccos\left(\frac{v \cdot b_y}{|v| \cdot |b_y|}\right) . \tag{5}$$

- $-\pi \leq o(G) \leq \pi$

Gesture Example



Gesture Example



- $c(G) \approx \begin{pmatrix} 453.3 \\ 433.2 \end{pmatrix}$
- $s(G) \approx 508.8$
- $o(G) \approx 0.141398 \,\widehat{\approx}\, 8.101°$

Implementation

- *Location* is a 2D vector
- *Pointer* has a *Location* and ID
- *Gesture* has multiple *Pointers*

```
1  public class Location {
2    public Location(float x, float y);
3
4    public float getX();
5    public float getY();
6
7    public Location add(Location loc);
8    public Location substract(Location loc);
9    public Location multiply(float c);
10   public Location divide(float c);
```

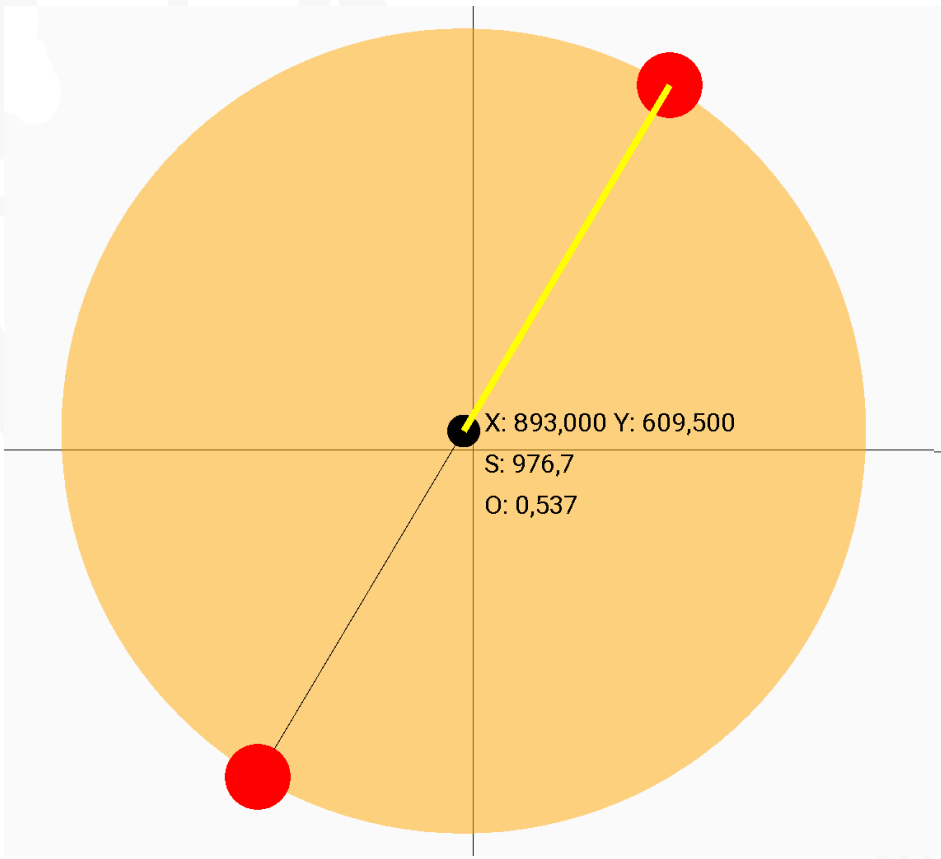
Implementation (cont.)

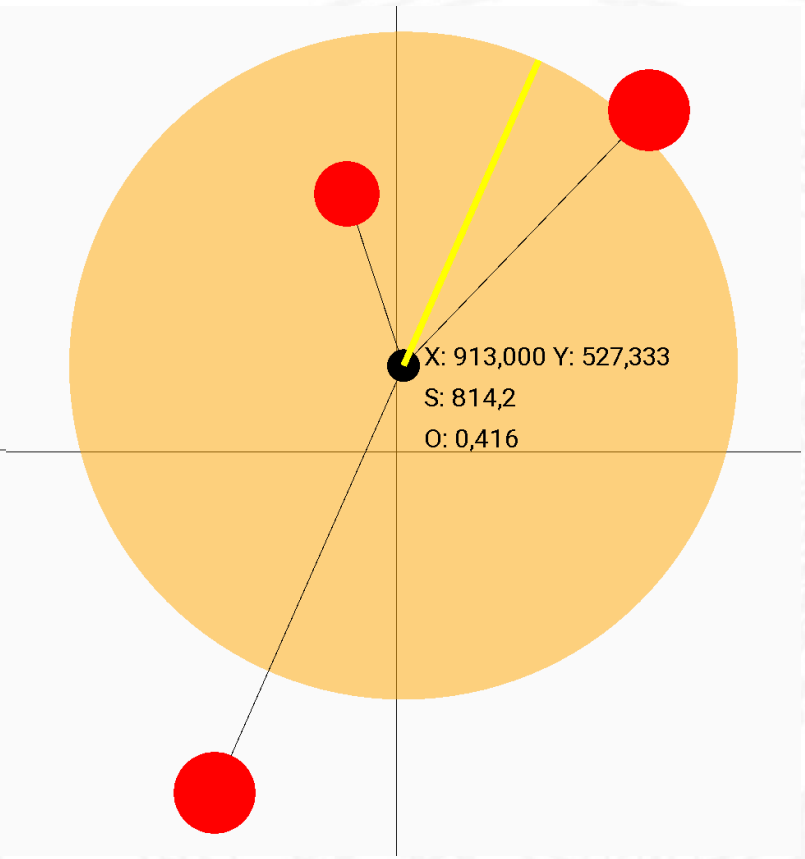
```
11
12    public double scalarProduct(Location loc);
13
14    public double getVectorLength();
15    public double distanceTo(Location loc);
16    public double getAngleTo(Location loc);
17    public Location getTurned(double angleRad);
18  }
```

Implementation (cont.)

```
1  public class Gesture {
2    public void addPointer(Pointer p);
3    public void removePointer(Pointer p);
4    public int getPointerCount();
5
6    public Location getCenter();
7    public float getSize();
8    public double getOrientation();
9  }
```

Gesture Visualization





AbsoluteSynergyProxy

- Maps gesture properties linearly ti amplitude values
- Maps gesture properties to robot palm position
- Gesture with two pointer used for amplitude control

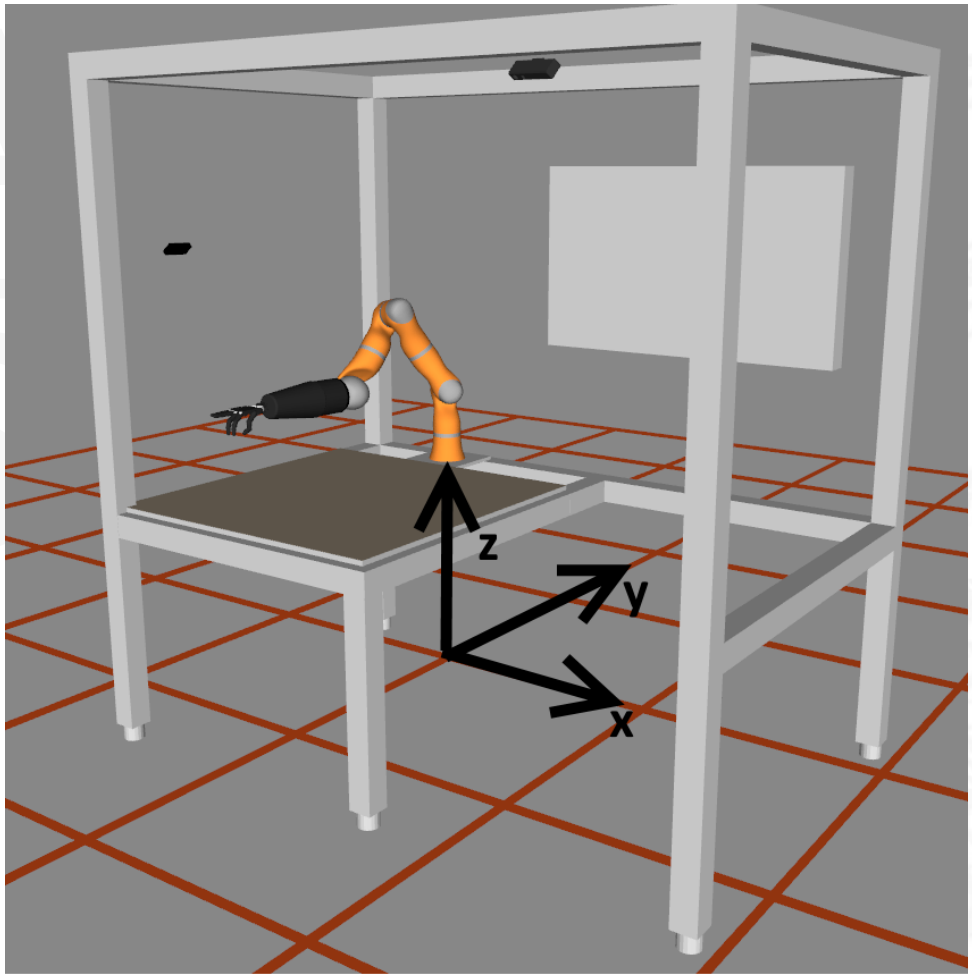
#	**Property**	**50 at**	**-50 at**
1	$s(G)$	1200	300
2	$c(G)_1$	$0.25 \cdot$ screenwidth	$0.75 \cdot$ screenwidth
3	$o(G)$	$-\frac{\pi}{2}$	$\frac{\pi}{2}$

Arm Control

- *CartesianArmManager*
- Accepts target positions for the C5 hand palm
- Requests IK solution for position endlessly
- When solution received: Set at *AxisManager*
- 3-Pointer-Gestures are used for arm positioning

Axis	**min**	**max**	**Prop.**	**min at**	**max at**
X	-0.2	0.4	$c(G)_1$	$0.25 \cdot$ screenwidth	$0.75 \cdot$ screenwidth
Z	1.10	1.35	$c(G)_2$	$0.75 \cdot$ screenheight	$0.25 \cdot$ screenheight

Arm Coordinate System



RelativeSynergyProxy

- Very similar to absolute mapping
- Mapping from property to Amplitude/Axis same
- Rate of change stored for every mapping
- Current value is altered

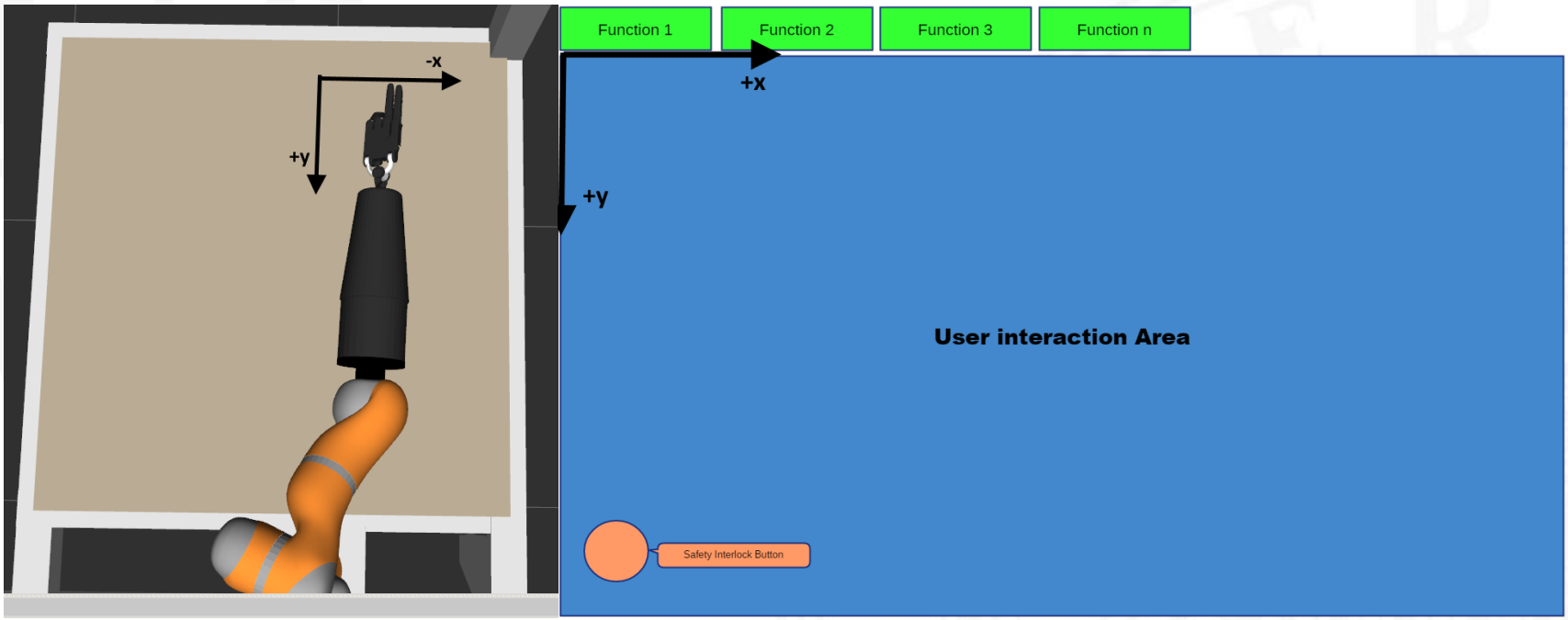
Direct Fingertip Mapping

- *DFTM*
- Mapping fingertips from touch screen to a plane in 3D space
- Relative positions of fingertips equal
- Plane in 3D space:

$$E : \vec{p} = \vec{b} + x \cdot \vec{e_1} + y \cdot \vec{e_2} \tag{6}$$

- $|\vec{e_1}| = |\vec{e_2}| = 1$ (1m)

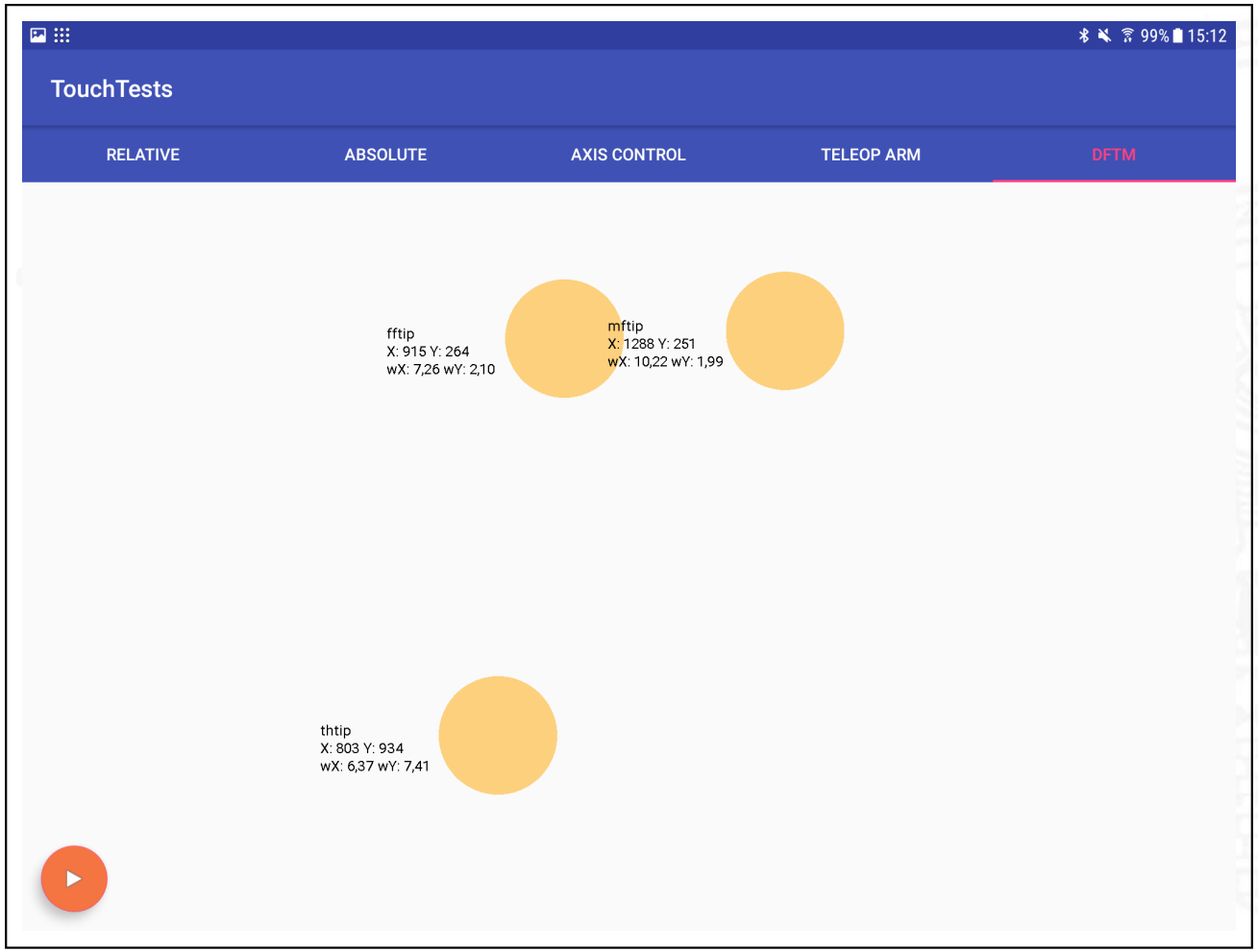
Direct Fingertip Mapping



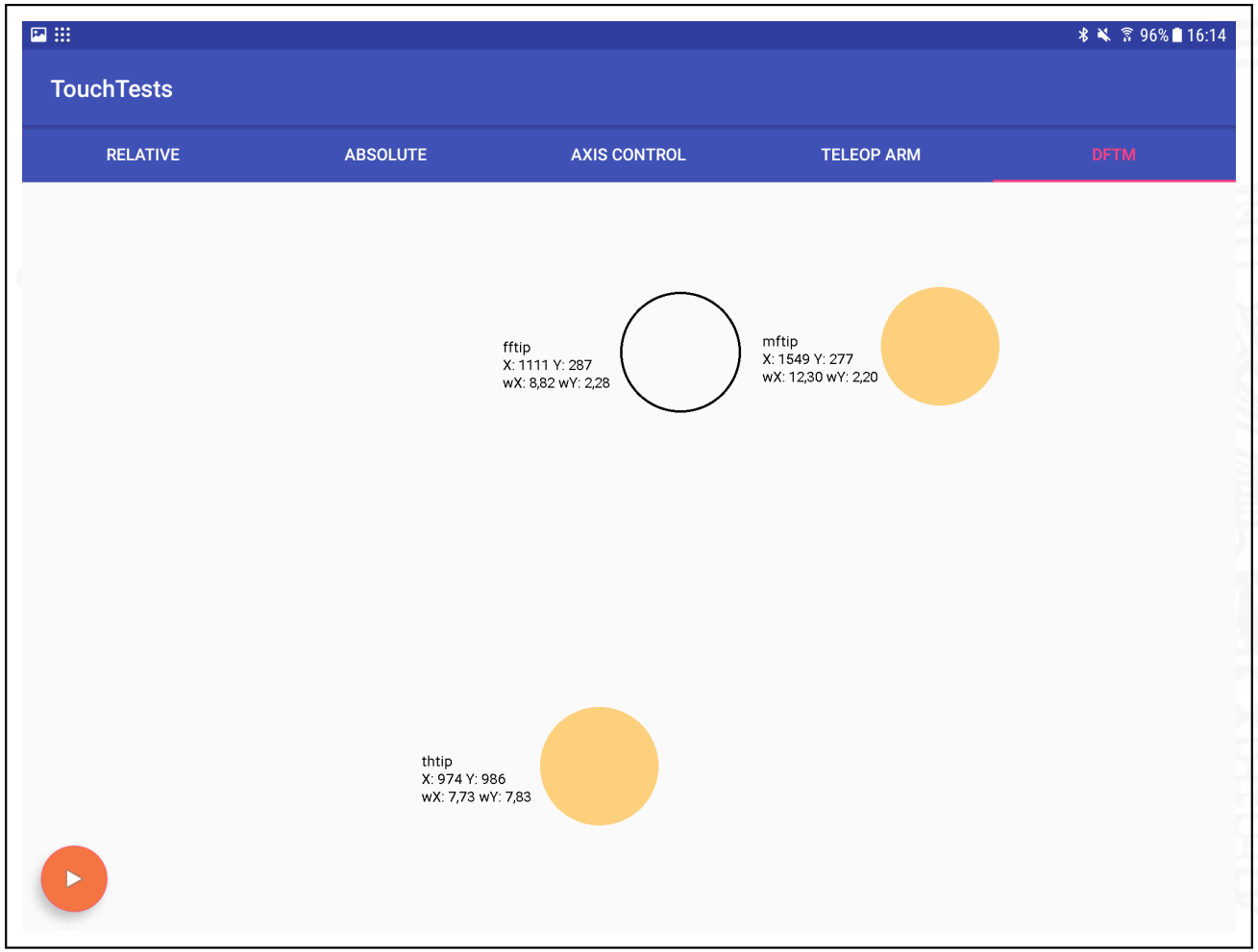
DFTM Parameters

- $\vec{b} = \begin{pmatrix} 0 \\ -1.25 \\ 1.2 \end{pmatrix}$
- $\vec{e_1} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$
- $\vec{e_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

DFTM Screen


99% 15:12
TouchTests
RELATIVE
ABSOLUTE
AXIS CONTROL
TELEOP ARM
DFTM
fftip
X: 915 Y: 264
wX: 7,26 wY: 2,10
mftip
X: 1288 Y: 251
wX: 10,22 wY: 1,99
thtip
X: 803 Y: 934
wX: 6,37 wY: 7,41

DFTM Screen


96% 16:14
TouchTests
RELATIVE
ABSOLUTE
AXIS CONTROL
TELEOP ARM
DFTM
fftip
X: 1111 Y: 287
wX: 8,82 wY: 2,28
mftip
X: 1549 Y: 277
wX: 12,30 wY: 2,20
thtip
X: 974 Y: 986
wX: 7,73 wY: 7,83

Summary

- Kuka LWR Arm + Shadow C5 Hand
- Relative Synergy Control
- Absolute Synergy Control
- Direct Fingertip Mapping Control
- on a 10” Android Tablet

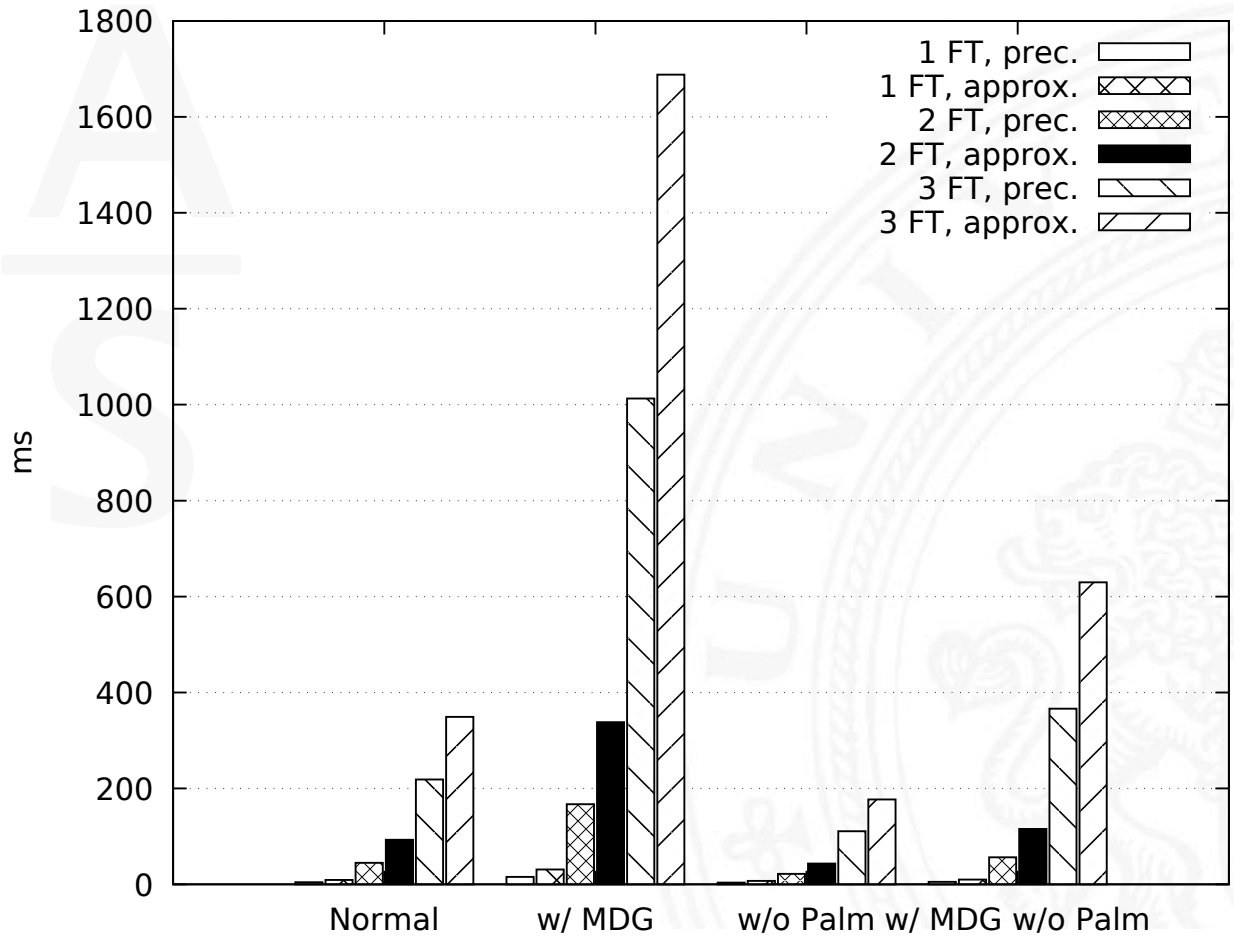
Evaluation

- Works as expected, in principle
- Synergy control is fine
- Unwanted arm movements when gesture is stationary
- BioIK seems to return a different / new solution for the same position
- Jittering even more disturbing in DFTM mode
- $\Rightarrow$ *MinimumDisplacementGoal* helps, slows down IK

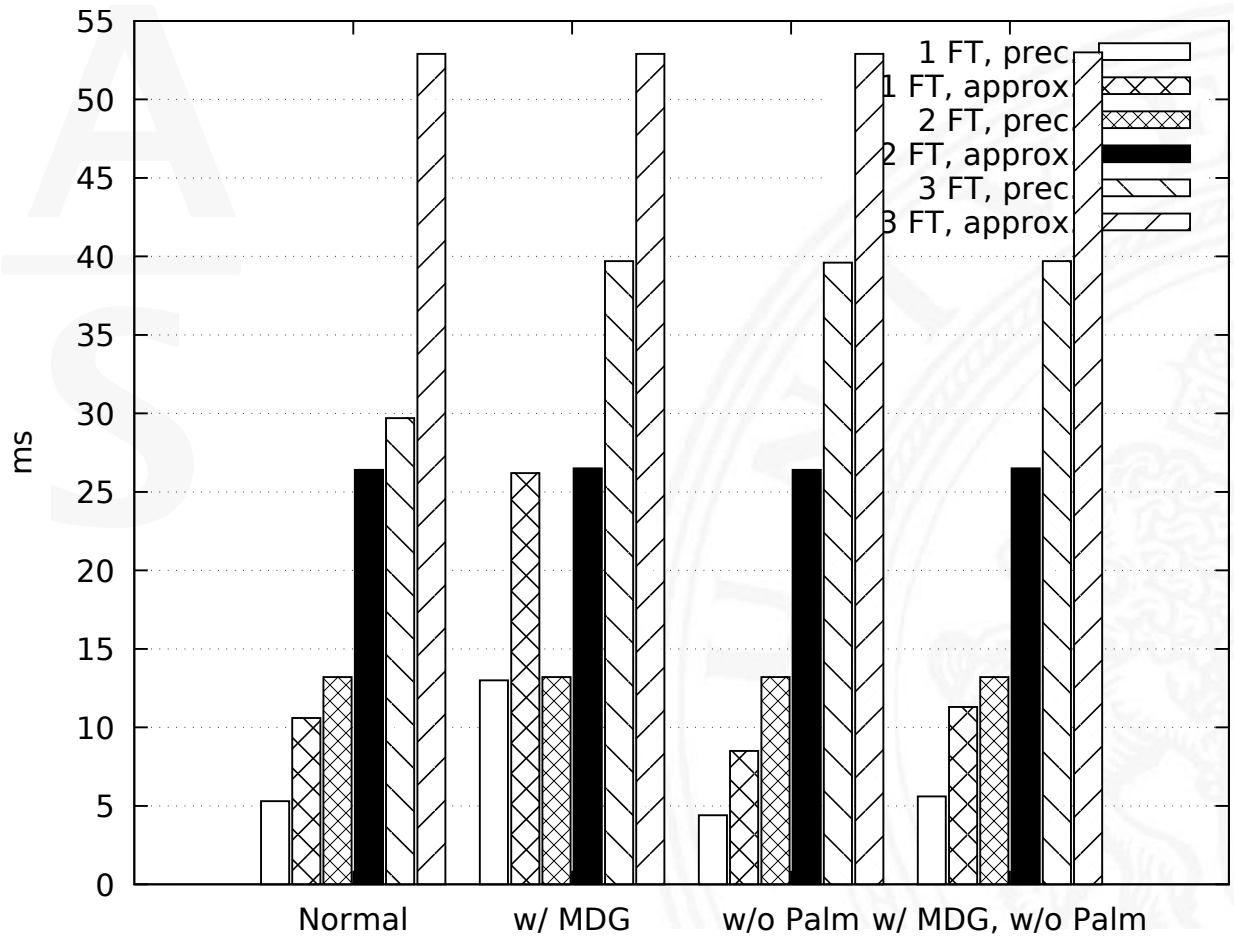
Performance

- Application overall performance is good
- Device gets warm $\Rightarrow$ constant network flow with small packages
- Service calls in *rosjava* / *rosandroid* designed to be asynchronous, but UI blocked during service calls
- Service calls to the BioIK service take long

BioIK Service Performance (1000ms timeout)



BioIK Service Performance (10ms timeout)



Conclusions

- BioIK uses the time it is given [9]
- IK frequency doesn't rise significantly in application
- The BioIK service doesn't seem to be the problem
- Implementation of *rosjava* / *rosandroid* buggy?

Future Work

- Look into performance issues
- Look into arm position jitter
- Implement user feedback (sound, visual, vibration)
- User Studies

- Implement more approaches

Thank you.

Bibliography

[1] Charan Ram Akupati. "Control of ROS Compatible Mobile Robot with Android Platform using ROSJAVA". Diploma. Universität Siegen, 2017. URL: https://www.eti.uni-siegen.de/ezls/lehre/studiendiplomarbeiten/controlofmobilerobotwithandroidplatform/control_of_mobile_robot_with_android_platform.pdf (visited on 2018-04-01).

[2] Alexandre Bernardino, Marco Henriques, Norman Hendrich, and Jianwei Zhang. "Precision grasp synergies for dexterous robotic hands". In: *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2013. DOI: 10.1109/robio.2013.6739436.

[3] Shadow Robot Company. *Shadow Dexterous Hand C5 Technical Specification*. 2008. URL: http://www.shadowrobot.com/downloads/shadow_dextrous_hand_technical_specification_C5.pdf (visited on 2018-03-05).

Bibliography (cont.)

[4] Karl Eilebrecht and Gernot Starke. *Patterns kompakt: Entwurfsmuster für effektive Software-Entwicklung*. Springer Vieweg, 2013. ISBN: 9783642347177.

[5] Jodi Forlizzi and Carl DiSalvo. "Service robots in the domestic environment". In: *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI*. ACM Press, 2006. DOI: 10.1145/1121241.1121286.

[6] KUKA Roboter GmbH. *Lightweight Robot 4+. Specification.* 2010-07-06.

[7] KUKA Roboter GmbH. *Preliminary Documentation. Fast Research Interface (FRI).* 2010-08-29.

Bibliography (cont.)

[8] Sunao Hashimoto, Akihiko Ishida, Masahiko Inami, Takeo Igarashi, and and. "TouchMe: An Augmented Reality Interface for Remote Robot Control". In: *Journal of Robotics and Mechatronics* 25.3 (2013), pp. 529–537. DOI: 10.20965/jrm.2013.p0529.

[9] Philipp Sebastian Ruppel. "Performance optimization and implementation of evolutionary inverse kinematics in ROS". MA thesis. Universität Hamburg, 2017. URL: https://tams.informatik.uni-hamburg.de/publications/2017/MSc_Philipp_Ruppel.pdf.

[10] Samsung.com. *Samsung Galaxy Tab S3 Datasheet*. 2017. URL: http://www.samsung.com/de/tablets/galaxy-tab-s3-9-7-t820/SM-T820NZKADBT/ (visited on 2018-03-05).

Bibliography (cont.)

[11] Sebastian Starke, Norman Hendrich, and Jianwei Zhang. "A memetic evolutionary algorithm for real-time articulated kinematic motion". In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017. DOI: 10.1109/cec.2017.7969605.

[12] Sebastian Starke, Norman Hendrich, Dennis Krupke, and Jianwei Zhang. "Evolutionary multi-objective inverse kinematics on highly articulated and humanoid robots". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017. DOI: 10.1109/iros.2017.8206620.

[13] Yue Peng Toh, Shan Huang, Joy Lin, Maria Bajzek, Garth Zeglin, and Nancy S. Pollard. "Dexterous telemanipulation with a multi-touch interface." In: *Humanoids*. IEEE, 2012, pp. 270–277. DOI: 10.1109/HUMANOIDS.2012.6651531.