KUKA youBot gravity compensation and trajectory teaching

Angelina Gerbolt
and
Temirlan Mussagaliyev

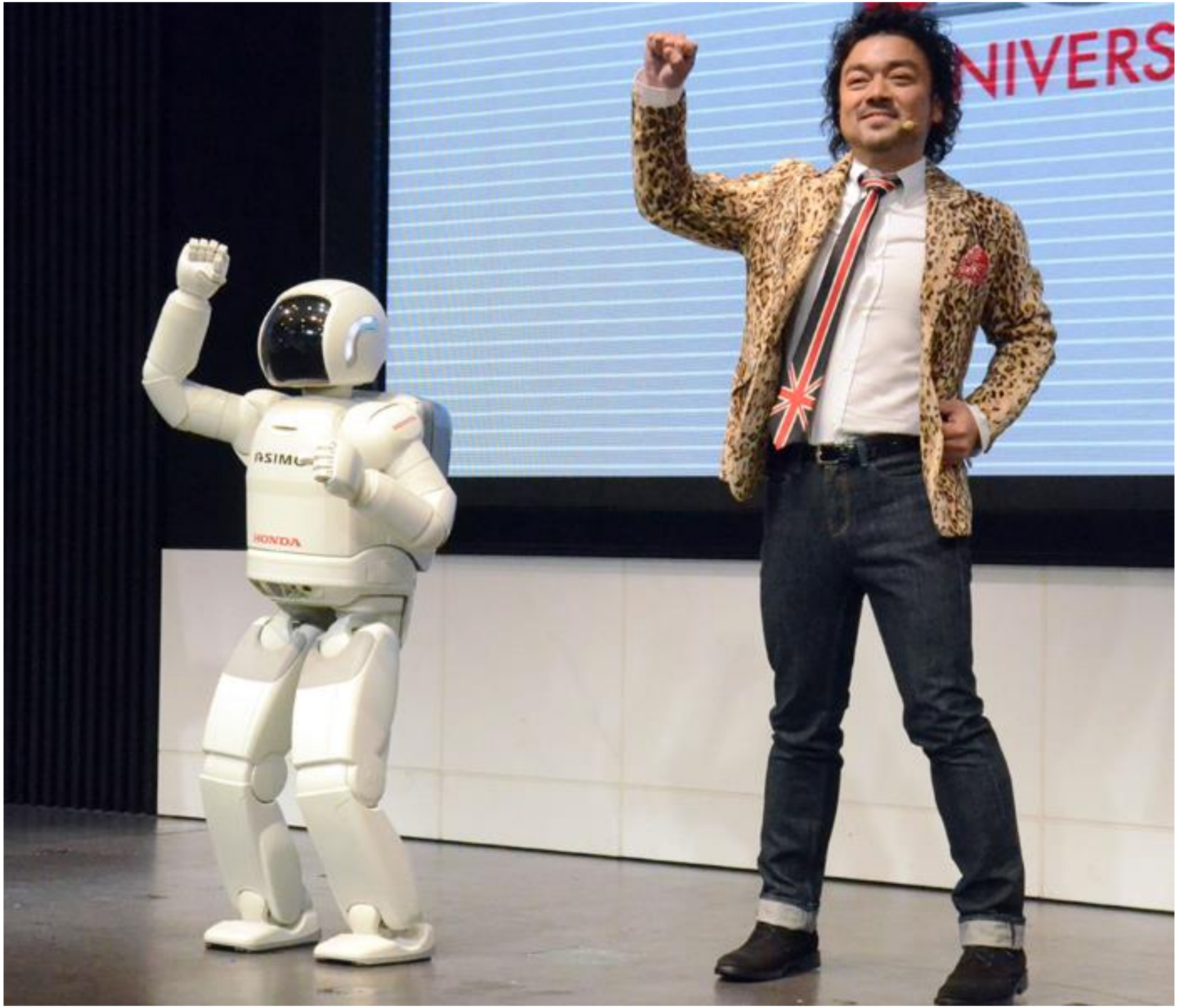
Supervisor: Dr. Almas Shintemirov
Special thanks: Tasbolat Taunyazov
Roberto Ancona

NAZARBAYEV UNIVERSITY

Outline:

- Introduction
- KUKA youBot arm
- Applications
  - Gravity compensation
  - Trajectory teaching
  - KUKA youBot API
- Ethernet connection between KUKA youBot and a laptop
  - Qt program
- Video demonstration
- Summary

NAZARBAYEV UNIVERSITY


ASIMO
HONDA

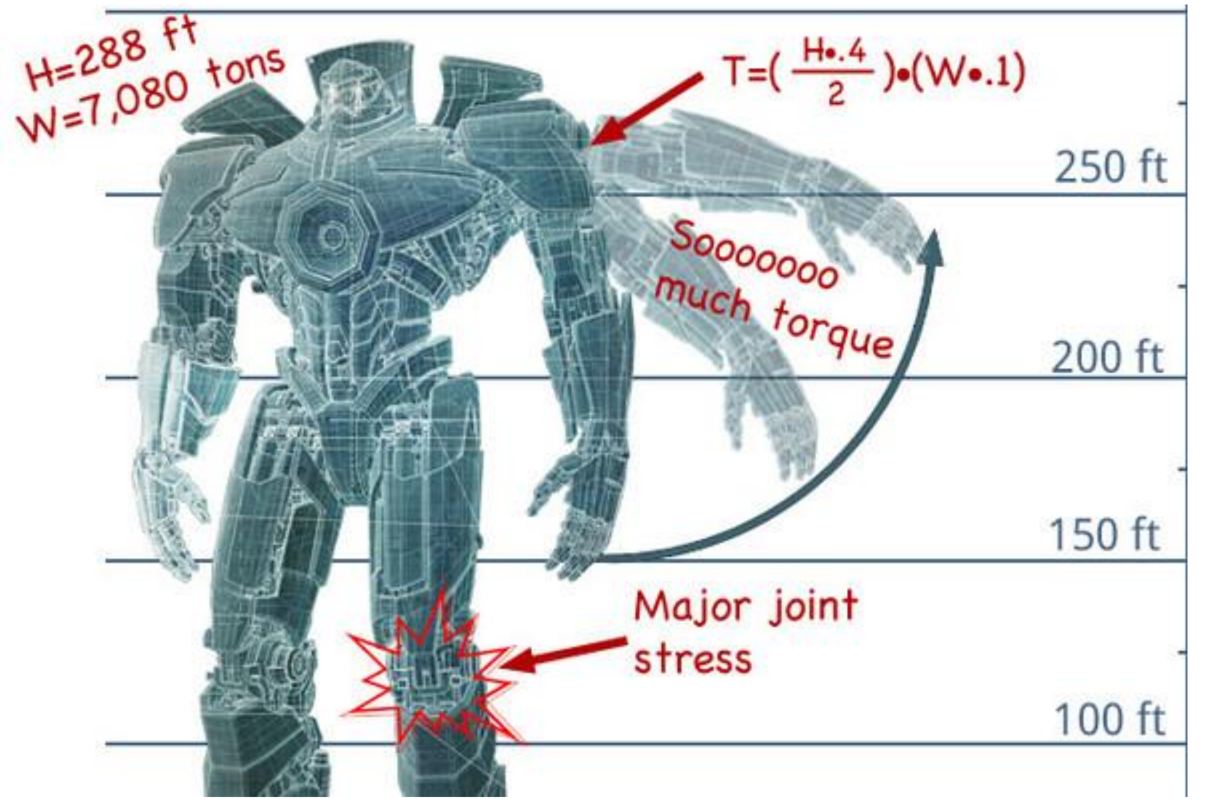
Introduction

- Robot control design is still a field of intensive study among robot constructors and research centers.

- Our project focuses on 2 problems in robot control: **gravity compensation** and **teaching by demonstration.**

- **Gravity compensation** is inherent and natural for humans, but may be quite hard to implement for many-DOFs robots.

- **Trajectory teaching** (teaching by demonstration) can be more precise and intuitive than a programmable trajectory planning and can be used in many industrial applications.



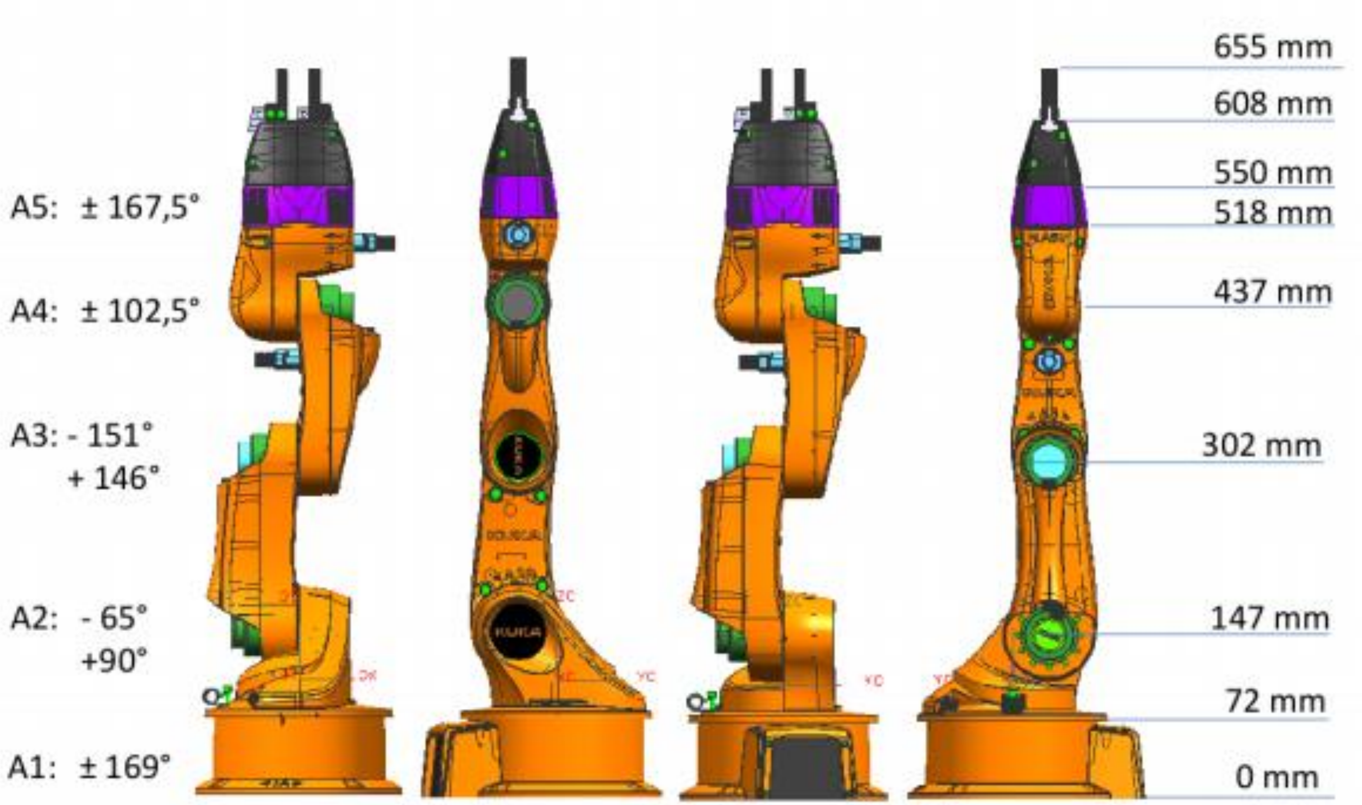
Engineers estimated that in case the robot from the "Pacific Rim" movie had actually existed, it would collapse under gravity forces due to the wrong body proportions.

Applications

- For research purposes
- Rehabilitation robotics, prosthetics
- Kinesthetic programming/ Teaching by demonstration:
    - industrial operations
    - cooking
    - writing
    - drawing, etc.

KUKA youBot arm

- 5 revolute joints
- 5 DOFs
- 2 finger gripper
- Controlled by on-board PC in real time



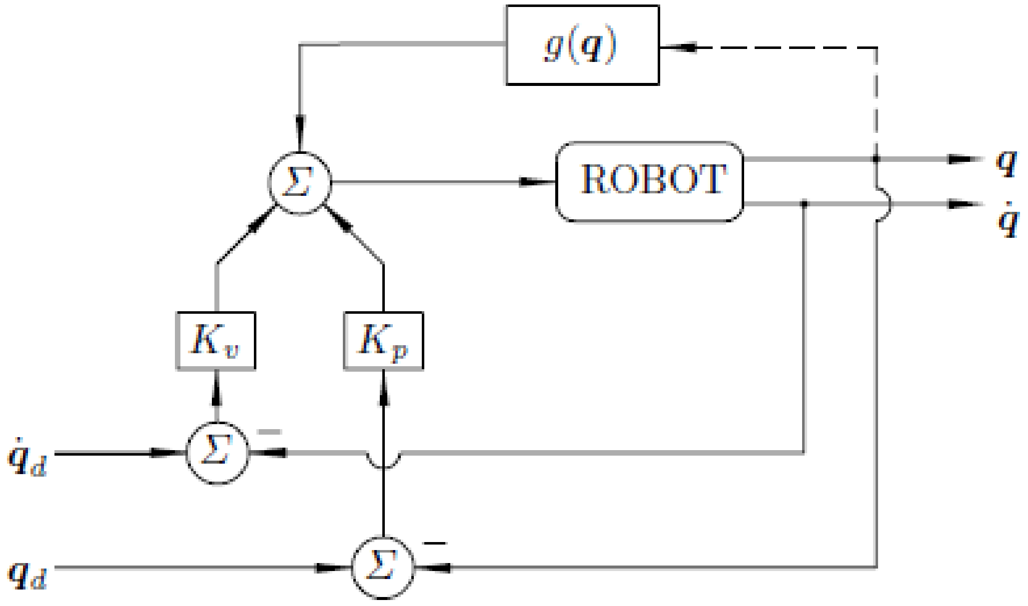
Theory: Gravity compensation with PD control

The general dynamics of a serial n-link rigid robot can be written as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F(\dot{q}) = \tau$$

Where **q, q,˙ q¨** - n × 1 vectors of the joint displacement, velocity, and acceleration;
τ - the n × 1 vector of input torques;
M(q) - the n × n manipulator inertia matrix;
C(q, q˙) - the n × n matrix of centripetal and Coriolis torques;
g(q) - the n × 1 vector of gravitational torques
F(q) - the n × 1 vector for the friction torques

Theory: Gravity compensation with PD control



Block-diagram: PD control with gravity compensation

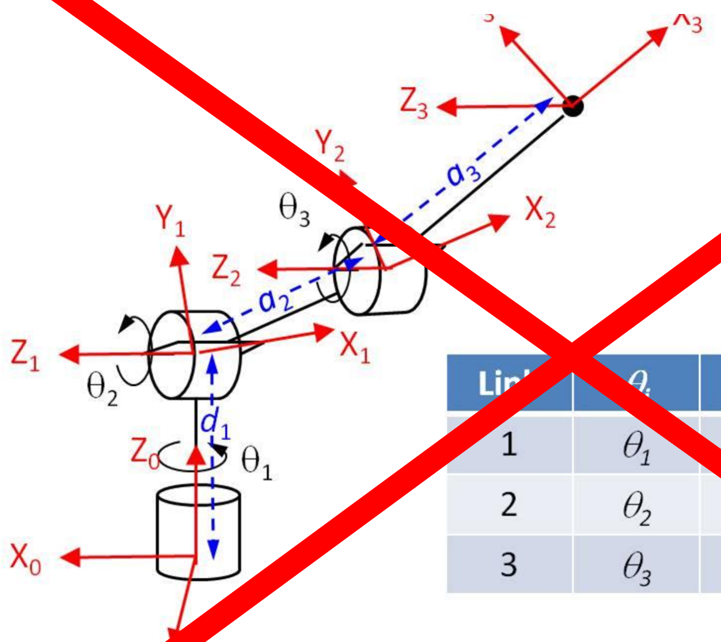
$$\tau = K_p \tilde{q} + K_d \dot{\tilde{q}} + g(q)$$

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = K_p \tilde{q} + K_d \dot{\tilde{q}} + g(q)$$

Deriving thy gravity term g(q): DH table

Using Denavit-Hartenberg parameters (DH-table) method and a matrix solution of the inverse kinematics for the robots with more than 3 DOFs is mathematically tedious and error prone.

Other method for deriving the gravity term is needed.



Link	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	90°
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

DH table method

Deriving the gravity term g(q): Lagrange dynamic equations

Consider the robot manipulator with n links

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q}).$$

where

$\boldsymbol{q} = [q_1, \cdots, q_n]^T$

$\mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ • a robot's kinetic energy

$\mathcal{U}(\boldsymbol{q})$ • a robot's potential energy

$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ • The Lagrangian

The torque for each joint is then calculated by this formula:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{q}_i}\right] - \frac{\partial \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial q_i} = \tau_i, \qquad i = 1, \cdots, n$$

Lagrange dynamic equations

Now when we have torque, we can regroup the Lagrange equation's components and find the vector with gravity forces - g(q).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + F(\dot{q}) = \tau$$

$$\tau \begin{pmatrix} 4\times 1 \\ vector \end{pmatrix} = K_p \begin{pmatrix} 4\times 4 \\ positive \\ diagonal \\ matrix \end{pmatrix} \tilde{q} + K_d \begin{pmatrix} 4\times 4 \\ positive \\ diagonal \\ matrix \end{pmatrix} \dot{\tilde{q}} + g \begin{pmatrix} 4\times 1 \\ vector \end{pmatrix}$$

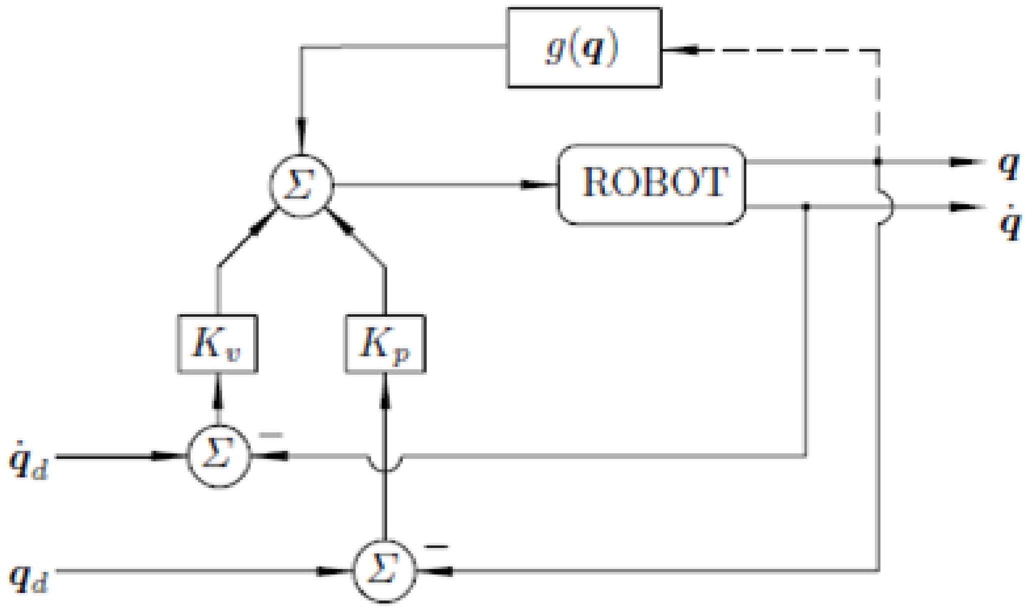
We discarded the last joint for simplicity.

Physical parameters of the robot's links (mass, length, inertia) that are used for calculating of g(q) vector and maximum allowed torques for each joint was found on the official youBot homepage.

Tuning the Kp and Kd gains

- Ensuring the manual guidance, the reference position should be equal to the output position ⇨
  the difference qd - q = 0 ⇨
  the position does not influence the torque ⇨ Kp can be discarded.
- Experimentally, the matrix Kd was found to be:

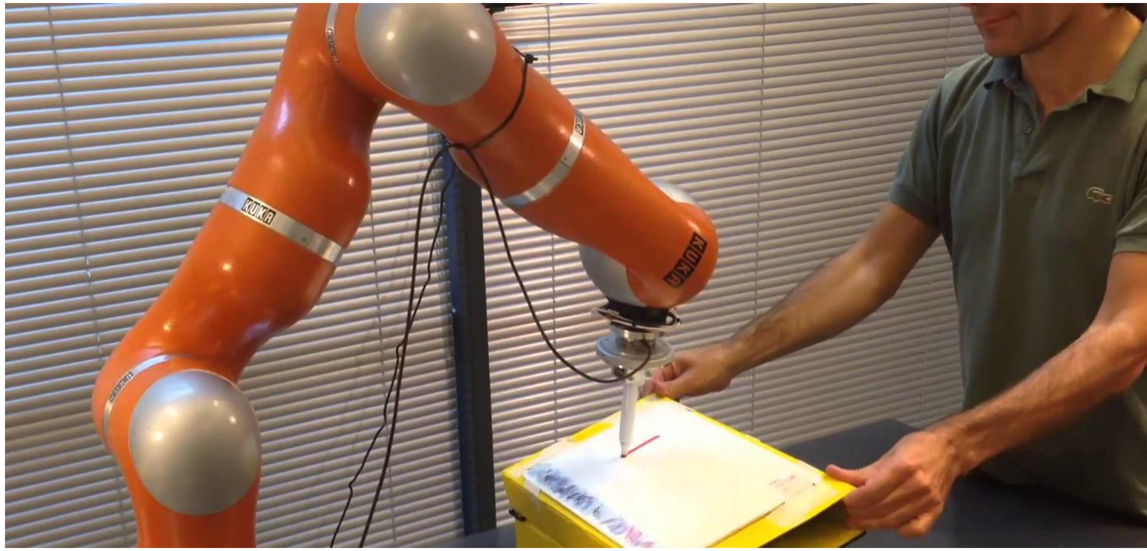
$$K_d = \begin{bmatrix} 1.27 & 0 & 0 & 0 \\ 0 & 1.12 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0.85 \end{bmatrix}$$



Trajectory Teaching

- By now, robot is programmed to remember and replicate 3 arbitrary positions, but this number can be extended to ∞.

❑ Using trajectory teaching and integrating touch and force sensors on top of the KUKA youBot's gripper, the robot can be adapted to perform such tasks as writing on the surface, rolling out the dough for pizza, etc.



KUKA youBot API

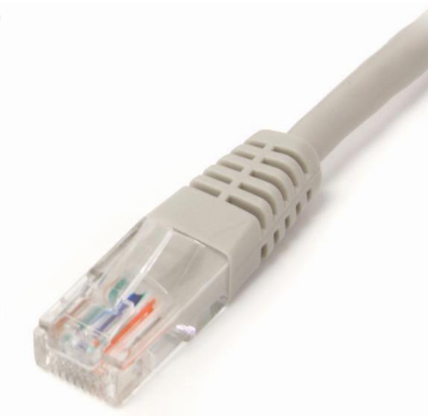
- The KUKA youBot API is the programming interface through which developer can access and control the KUKA youBot hardware.

- The robot arm is represented as 5 DOF kinematic chain, and the omnidirectional mobile platform is treated as a collection of revolute joints, where each joint consists of a motor and a gearbox.

- Has 3 main classes: **YouBotManipulator** (represents KUKA youBot arm), **YouBotBase**(omni-directional platform) and **YouBotJoint** (represents a joint either of the arm or the platform).

- Each joint can be accessed individually; Such physical values as angular position, angular and linear velocity, torque, current, etc. can be easily sensed and set.

Ethernet connection between KUKA youBot and a laptop



- Provides an easy way of remote control of KUKA youBot avoiding the usage of the external I/O devices (monitor, keyboard, mouse) and numerous cables that may restrict the mobility of the robot.

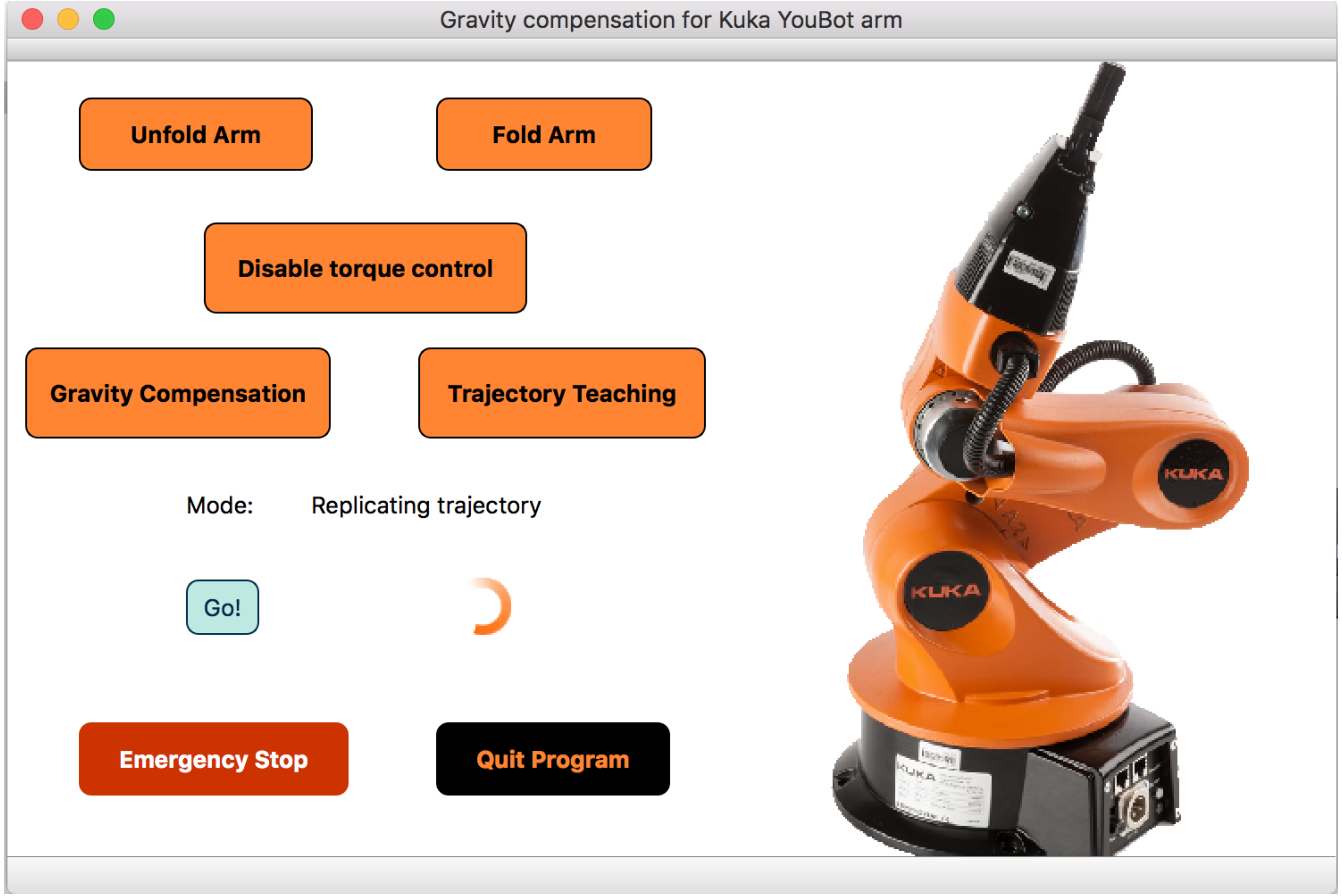
- Established using EtherCAT ports on top of the KUKA youBot mobile base and a UTP CAT 5e Patch Cable.

- Connection established using TCP core internet protocol and network sockets.

- KUKA youBot serves as a server; the server code is written in C++.

- Laptop takes the role of a server; the code is written in C++, adapted to the syntax of Qt.

Qt Graphical User Interface



- Qt is a very convenient application framework for creating graphical user interfaces (GUIs).
- GUI is useful for end users who are not familiar with Linux or C++.
- Visualizes the commands and processes that KUKA youBot can execute.

Pseudocode in C++

```
initialize socket connection
while true do
        if a message "a" is received, set angles to unfolded position
        if a message "b" is received, set angles to home position
        if a message "s" is received, stop and exit the program
        if a message "q" is received,
        set angles to home position,
        close socket connection with a laptop,
        exit the program
        if a message "f" is received, set all torques to zero //disable the torque control
        if a message "g" is received,
        repeat
              algorithm with gravity compensation with PD control
             if a message "t" is received, begin trajectory teaching algorithm
                  wait for messages "x", "y", "z" to remember desired position 1, 2, 3 respectively
        until
              other messages are received
end while
```

Video demonstration - Enjoy!

Summary

- PD + g(q) controller ensures an effective gravity compensation and manual guidance for the KUKA youBot arm links.
- Teaching by demonstration can serve as an useful tool for performing some industrial and custom tasks.
- User friendly Qt GUI allows to intuitively control the robot without delving into Linux, C++ and compilation routine.

Future improvements

- Joint friction and disturbance (which represents the external load at the robot's end effector) can be introduced into PD + gravity controller in order to provide an effective gravity compensation for the KUKA youBot holding a payload.

- The controller gains may not be constant, but continuously readjusted for each iteration of the gravity routine in order to make the performance more smooth.

- The robot can be taught to perform some meaningful motions instead of the 3 arbitrary motions.

References:

- B. Keiser. T*orque Control of a KUKA youBot Arm*. Master Thesis at Robotics and Perception Group University of Zurich, 2013. Retrieved from http://rpg.ifi.uzh.ch/docs/theses/Benjamin_Keiser_Torque_Control_2013.pdf
- “KUKA youBot User Manual ”, 2012
- Manual Guidance with KUKA youBot (Torque Control) by Roberto Ancona https://www.youtube.com/watch?v=JWH5HFYnl3o
- R. Kelly, V. Santibáñez and A. Loría. *Control of Robot Manipulators in Joint Space, chapters 7-8* . Springer-Verlag London Limited 2005.
- http://www.youbot-store.com
- S. Sharma, G. K. Kraetzschmar, C. Scheurer, and R. Bischoff. Unified Closed Form Inverse Kinematics for the KUKA youBot. Proc. of ROBOTIK 2012; 7th German Conference on Robotics, pages 1 _ 6, 2012.