2014 IEEE Int. Conf. on Robotics and Automation

Workshop

MATLAB/Simulink for Robotics Education and Research

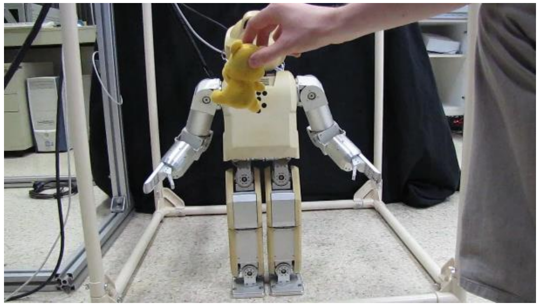
ROBOT CONTROL USING MATLAB/SIMULINK

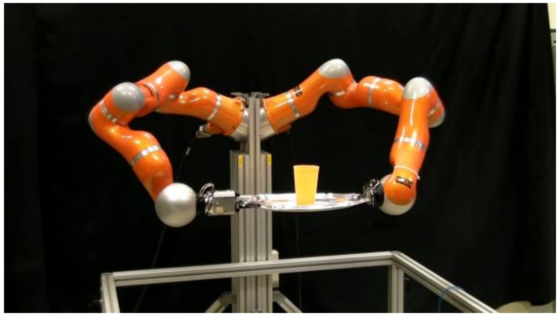
Leon Žlajpah

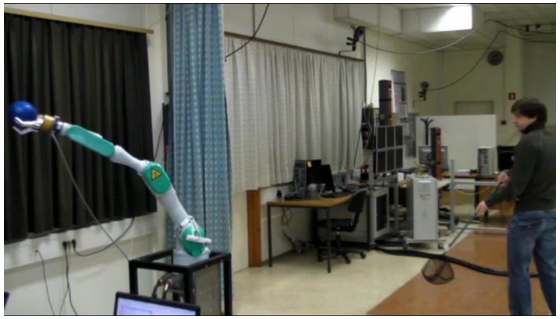
Institut
"Jožef Stefan"
Ljubljana, Slovenija

Dept. of Automation, Biocybernetics and Robotics at JSI

- The research focus is in the development of new technologies for robotic systems including the following fields:
- development of advanced control strategies for robots (including modeling and simulation)
- low level reflexive control, compliancy and redundancy
- physical interaction with humans and the environment
- transfer and generalization of interactive skills from human to robot
- cooperating robot systems
- robot motion synthesis through learning
- sensory systems (vision, force)
- automation of industrial processes







Outline

- The role of simulation in robotics
- Requirements for the simulation environment
- **Planar Manipulators Toolbox** (overview)
- The concept of our new environment **RobotBlockset**
- Multi-arm robots
- Interface for Kuka LWR robot
- Simulink and external dynamic engines
- Conclusions

The role of simulation

- Simulation allows us to study the structure, characteristics and the function of a system at different levels of details each posing different requirements for the simulation tools. As the complexity of the system under investigation increases the role of the simulation becomes more and more important.
- The simulation tools can certainly enhance design, development and operation of robotic systems.
- Augmenting the simulation with visualization tools and interfaces, one can simulate the operation of the robotic systems in a very realistic way.

Requirements for the simulation environment

Simulation tools for robotic systems differ depending on which aspect of the robot research they support. In most simulation tools the focus is on the motion of the robot manipulator in different environments.

Requirements for use in research and teaching:

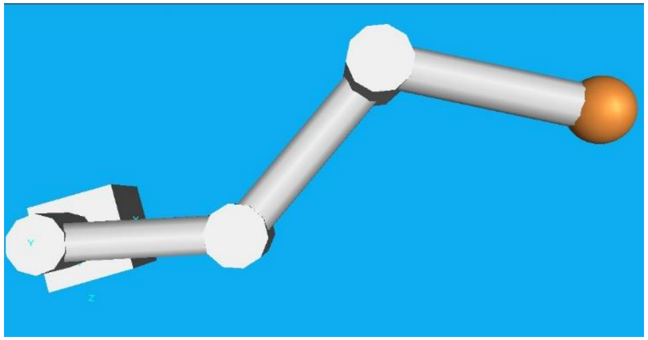
- **openness** (easy access to the system, open architecture, easy integration of modules)
- **reconfigurability** (quick modifications and verification of effects)
- **ease of use** (many users, different expertise, easy testing on real robots, hidden implementation details, unified tools and environment)

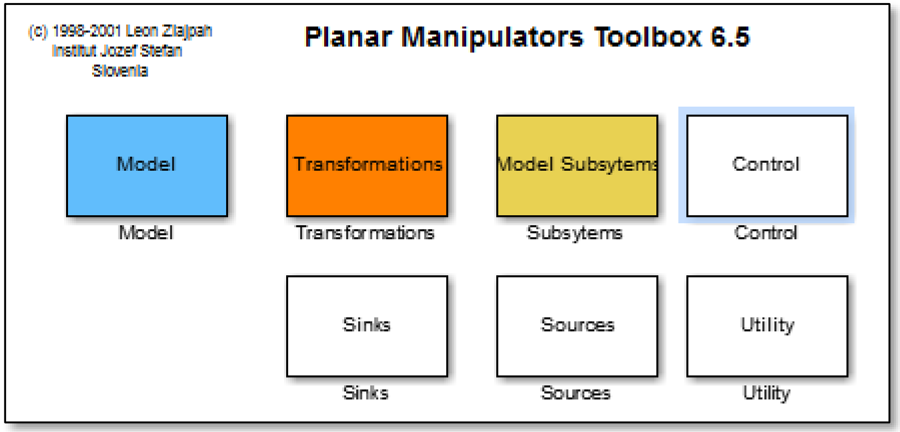
Augmenting the simulation with visualization tools and interfaces, one can simulate the operation of the robotic systems in a very realistic way.

Planar Manipulators Toolbox - PMT

The **Planar Manipulators Toolbox** is based on a kinematic and dynamic model of a planar manipulator with revolute joints.

- PMT consists of several M-files for the calculation of the model and other functions for planar manipulators which can not be created using the standard ones.
- The user has to write only some model specific functions.
- PMT enables the real-time simulation and the online robot control, i.e. Manipulator in the loop simulation, using MATLAB functions and Simulink blocks.
- For the use with Real-Time Workshop main functions have been rewritten into C MEX S-functions versions.





Planar Manipulators Toolbox – MATLAB functions

Calculation of the direct kinematics the function

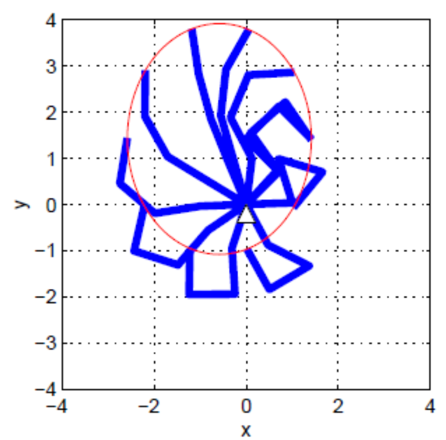
```
> nj=4;
> L=[1 1 1 1];
> q=[0.5;1;-1;-1]*pi/2;
> qd=[1;1;1;1];
> qdd=[1;0;-1;2];
> [x,J,Jd]=kinmodel(q,qd,L);
> xd=J*qd;
> xdd=J*qdd+Jd*qd;
> disp([x xd xdd])

1.4142   -1.4142   -15.5563
1.4142    4.2426     2.8284
```

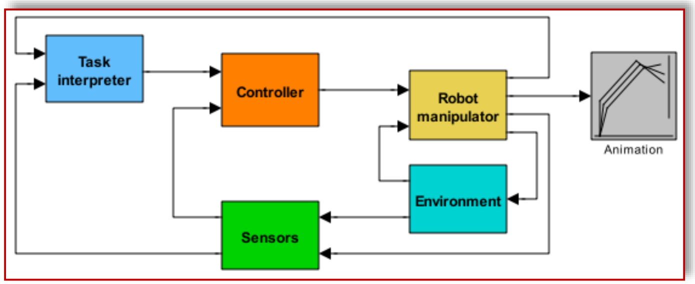
```
Different
[Man,Grad]=idxman(q,L);
[Grav,Grad]=idxgrav(q,L,Lc,m,ml,W);
```

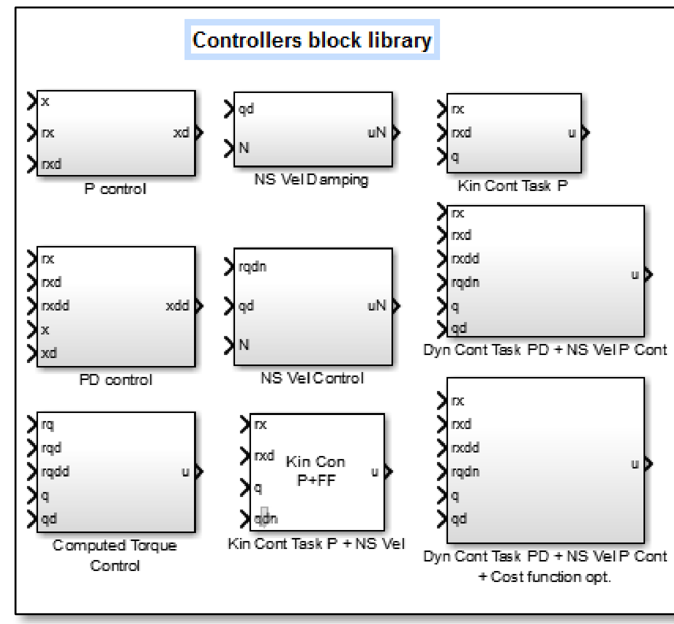
Trajectory generation (minimizing the condition number)

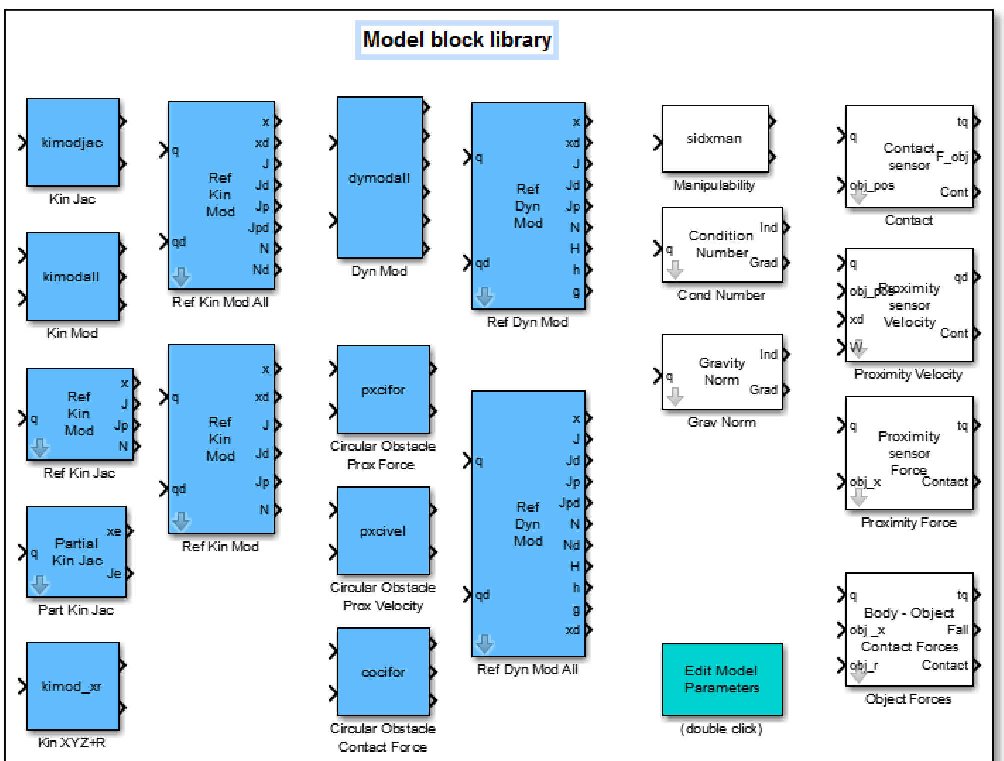
```
> nj=4;L=[1 1 1 1];
> q=[0.5;1;-1;-1]*pi/2; x0=kinmodel(q,L);
> amp=[-2;2.5];t0=5;tmax=5;tsamp=0.001;
> [xp,xv,xa,t]=refcirc(x0,amp,t0,tmax,tsamp);
> n=length(xp); qp=zeros(n,nj); qv=zeros(n,nj);
> for i=1:n
>    [x,J]=kinmodel(q,L);
>    J1=J'/(J*J'); NS=(eye(nj)-J1*J);
>    [CN,Grad]=idxcn(q,L);
>    e=xp(i,:)'-x; qd=J1*100*e-5*NS*Grad;
>    q=q+qd*tsamp; qp(i,:)=q'; qv(i,:)=qd';
> end
plotrobot(qp(1:(n-1)/10:n,:),L)
```

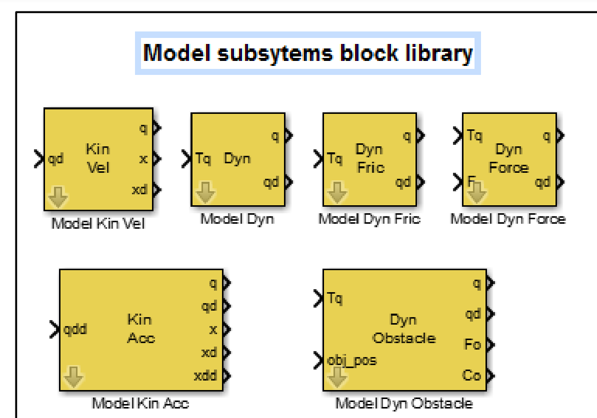


Planar Manupulators Toolbox - Simulink

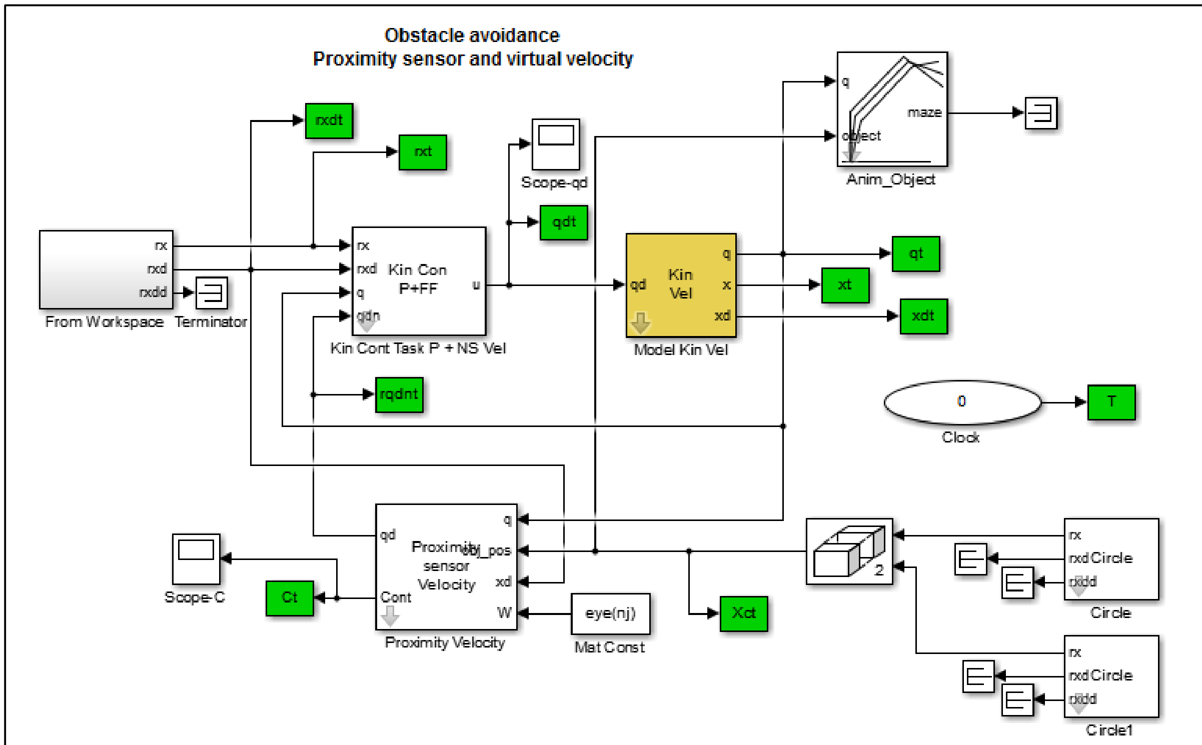


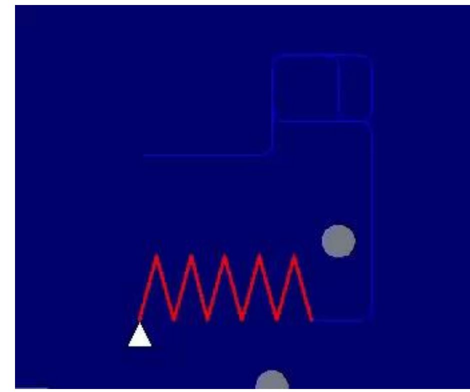






PMT – Obstacle avoidance

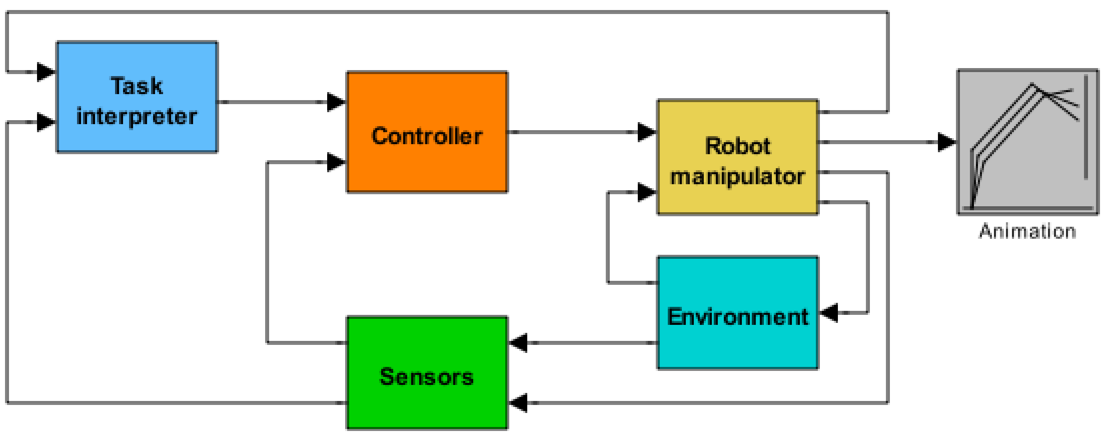




"RobotBlockset" Simulink library

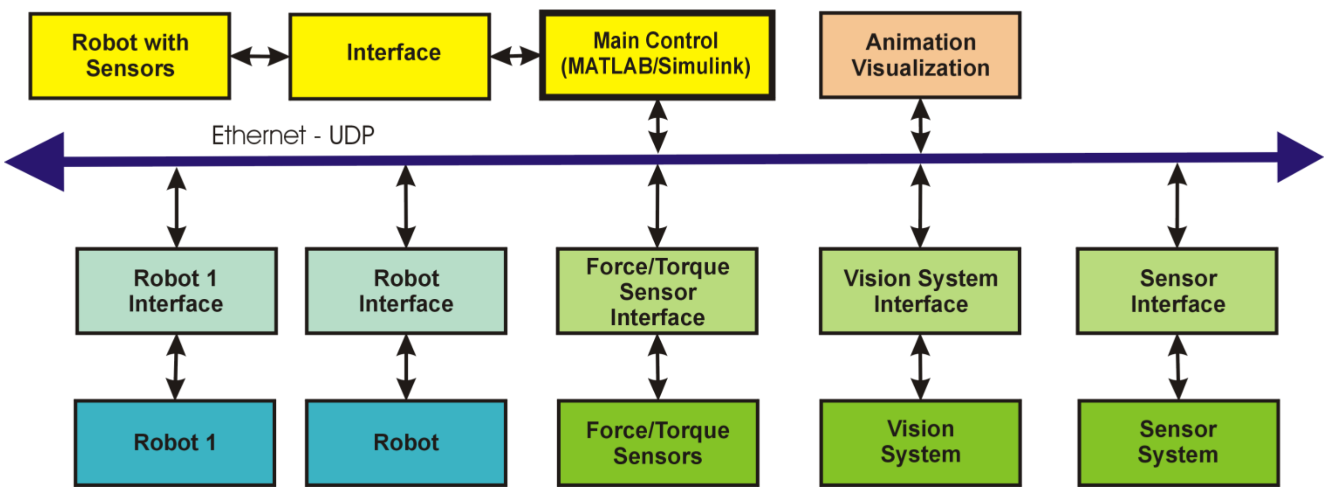
A new MATLAB/Simulink based integrated environment for the simulation and for the control design for more general robot systems.

A crucial feature for development of robot control systems is that the user can easy switch between using model or a real system in the simulation loop.

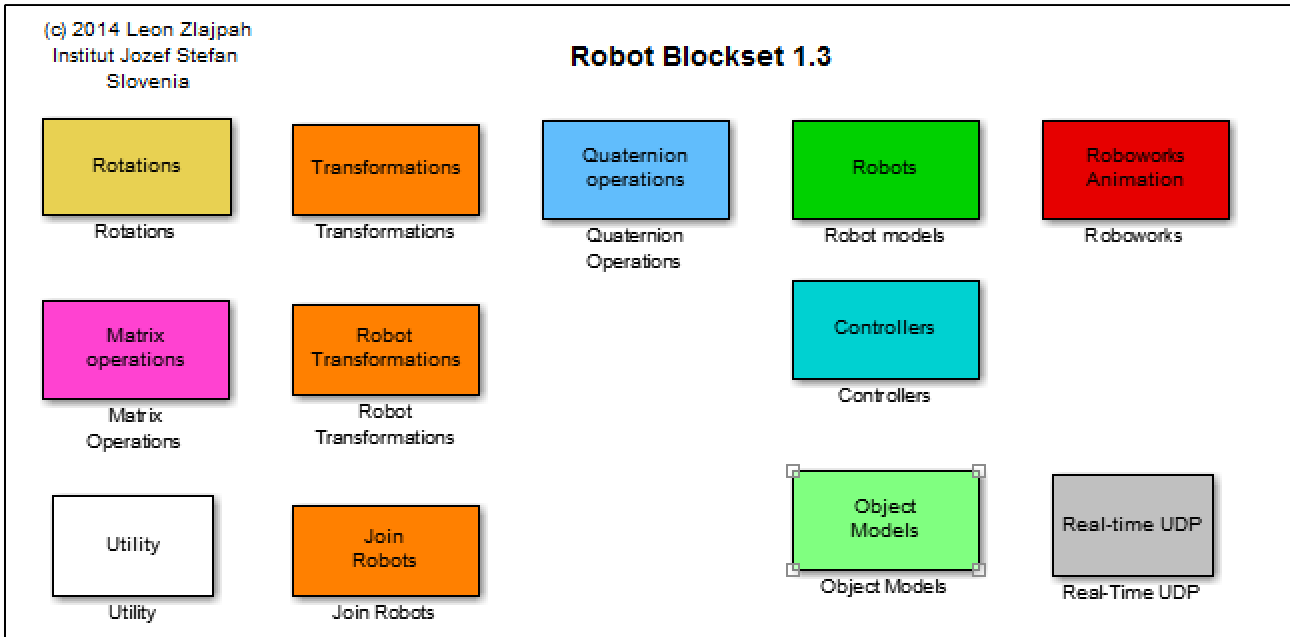


The concept

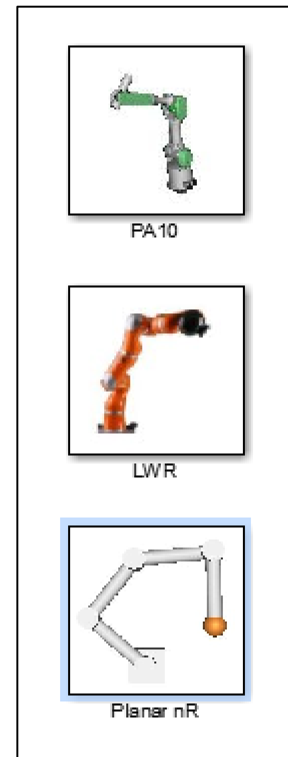
- to simulate the kinematics and dynamics of arbitrary chosen kinematic chain describing different manipulators,
- to enable integration of different sensor systems like vision and force sensors,
- to enable simulation of scenarios for complex robot tasks,
- to include the model the robots' environments,
- to visualize the robots and their environment,
- to enable integration of real robots in the simulation loop and
- to allow distributed computation over more computers.



“RobotBlockset” Simulink library

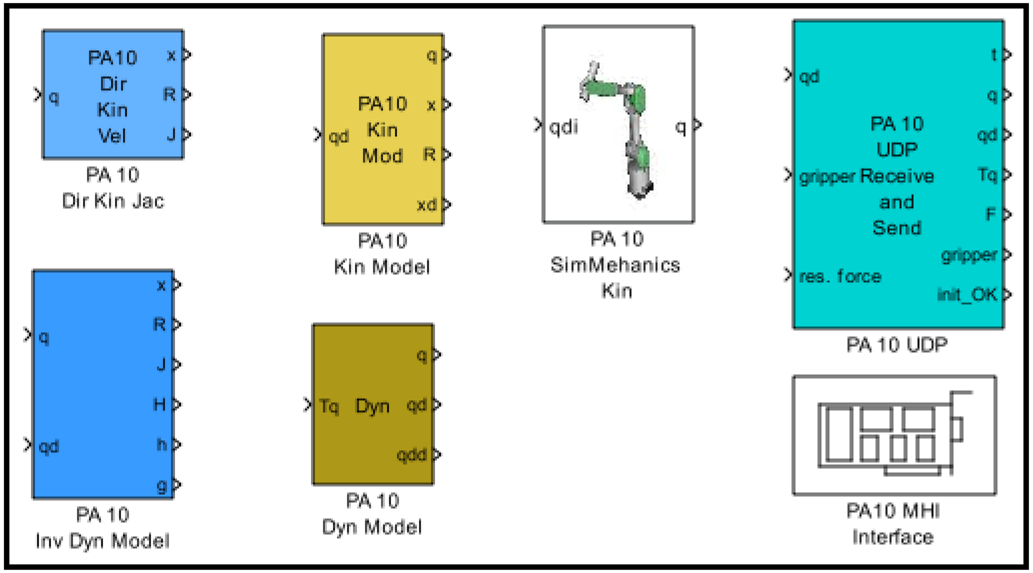


Robot models



Robot model

- For each robot we need:
- Kinematic models
- Dynamic models
- Interfaces
- Special control blocks



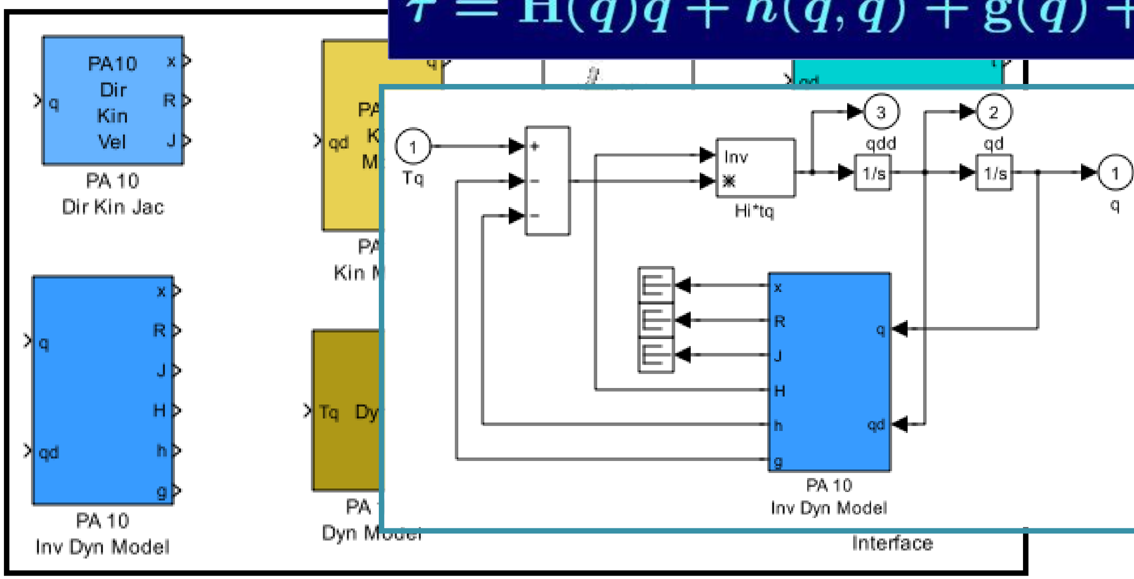
Robot model

- For each robot we need:
- Kinematic models
- Dynamic models
- Interfaces
- Special control blocks

$$x = f(q)$$
$$\dot{q} = \mathrm{J}^{\#}\dot{x} + \mathrm{N}\dot{q}$$
$$\ddot{q} = \mathrm{J}^{\#}(\ddot{x} - \dot{\mathrm{J}}\dot{q}) + \mathrm{N}\ddot{q}$$

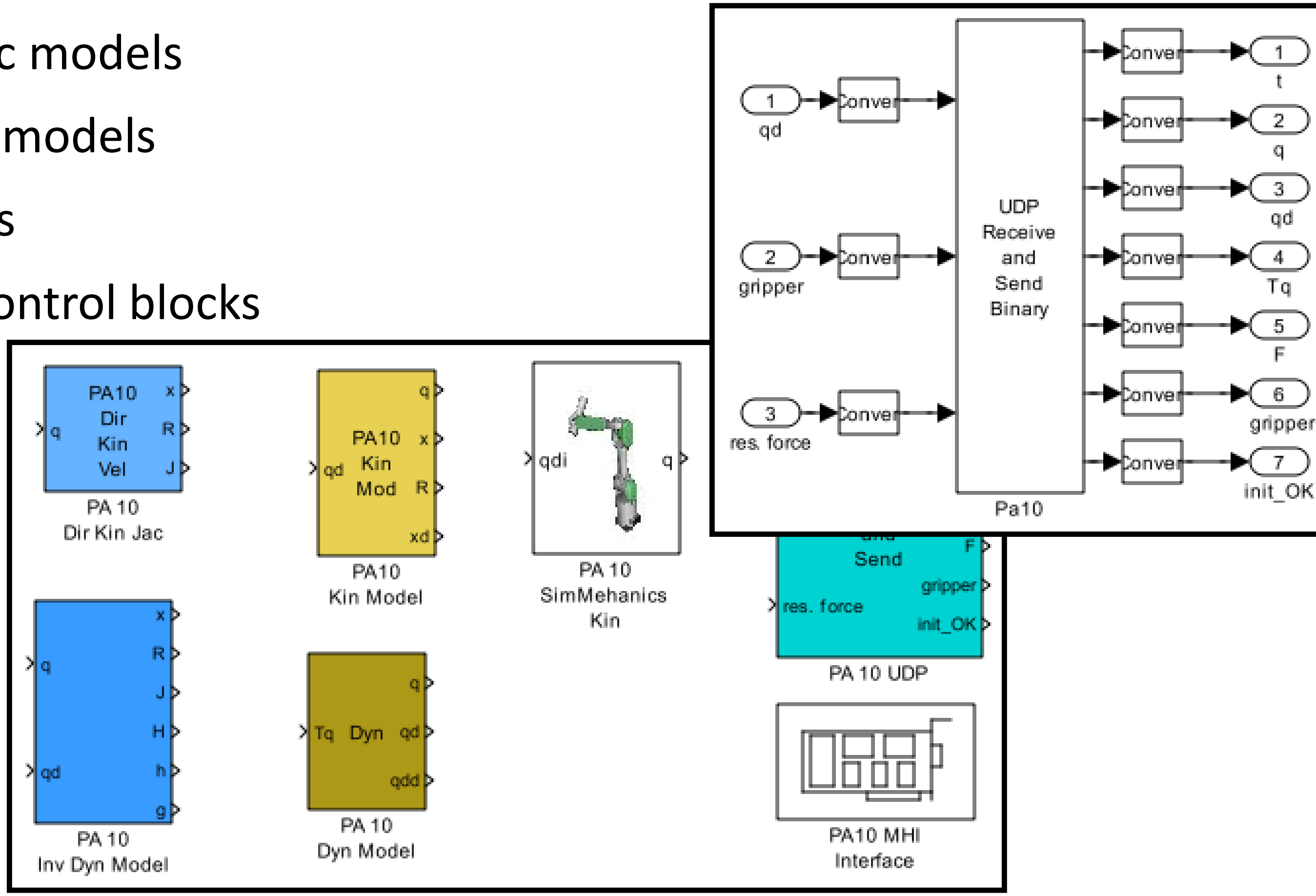
$$\tau = \mathrm{J}^{T}F + \mathrm{N}^{T}\tau$$

$$\tau = \mathrm{H}(q)\ddot{q} + h(q,\dot{q}) + \mathrm{g}(q) + \tau_{\mathrm{fric}} - \tau_{\mathrm{F}}$$



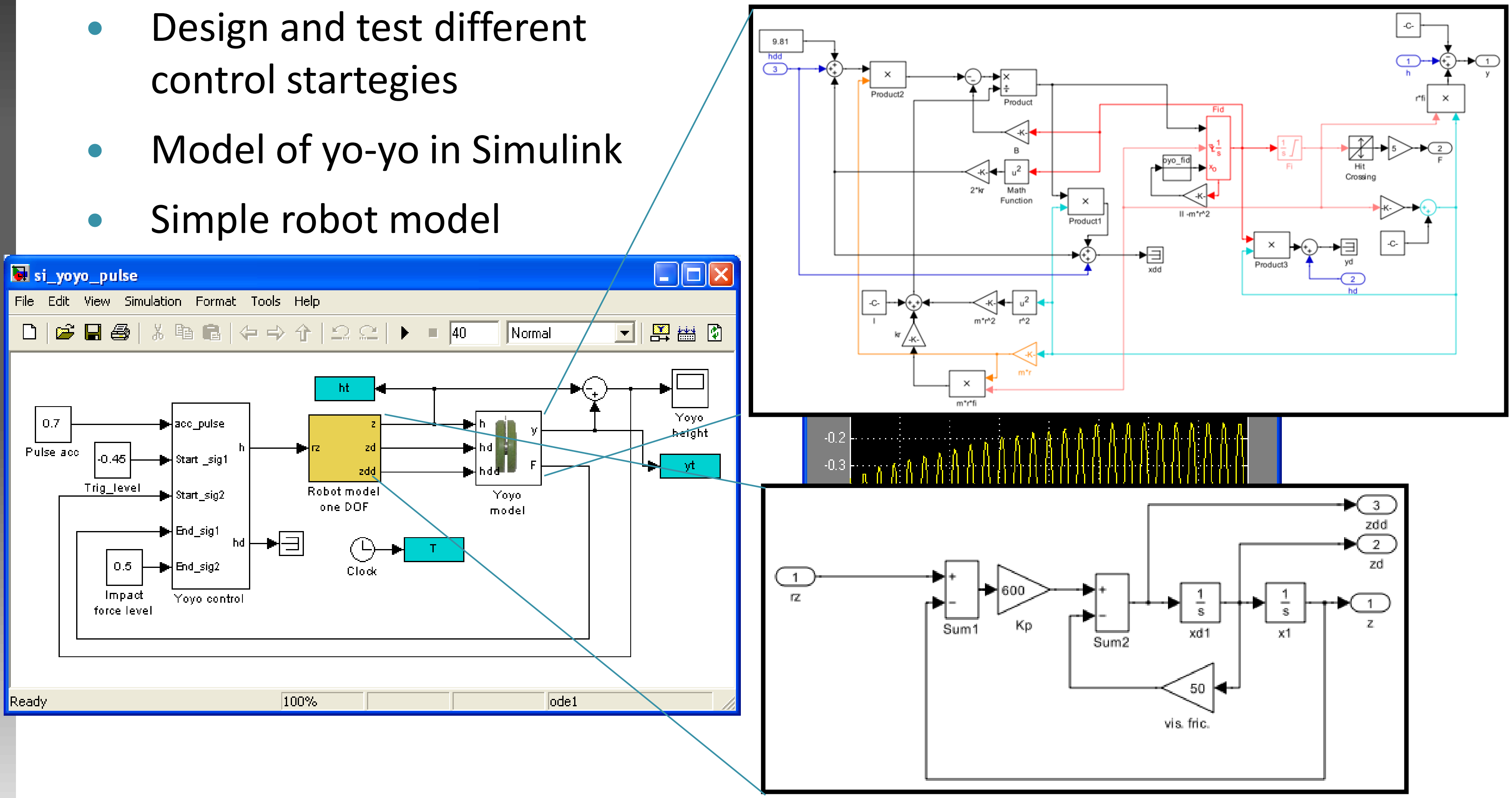
Robot model

- For each robot we need:
- Kinematic models
- Dynamic models
- Interfaces
- Special control blocks



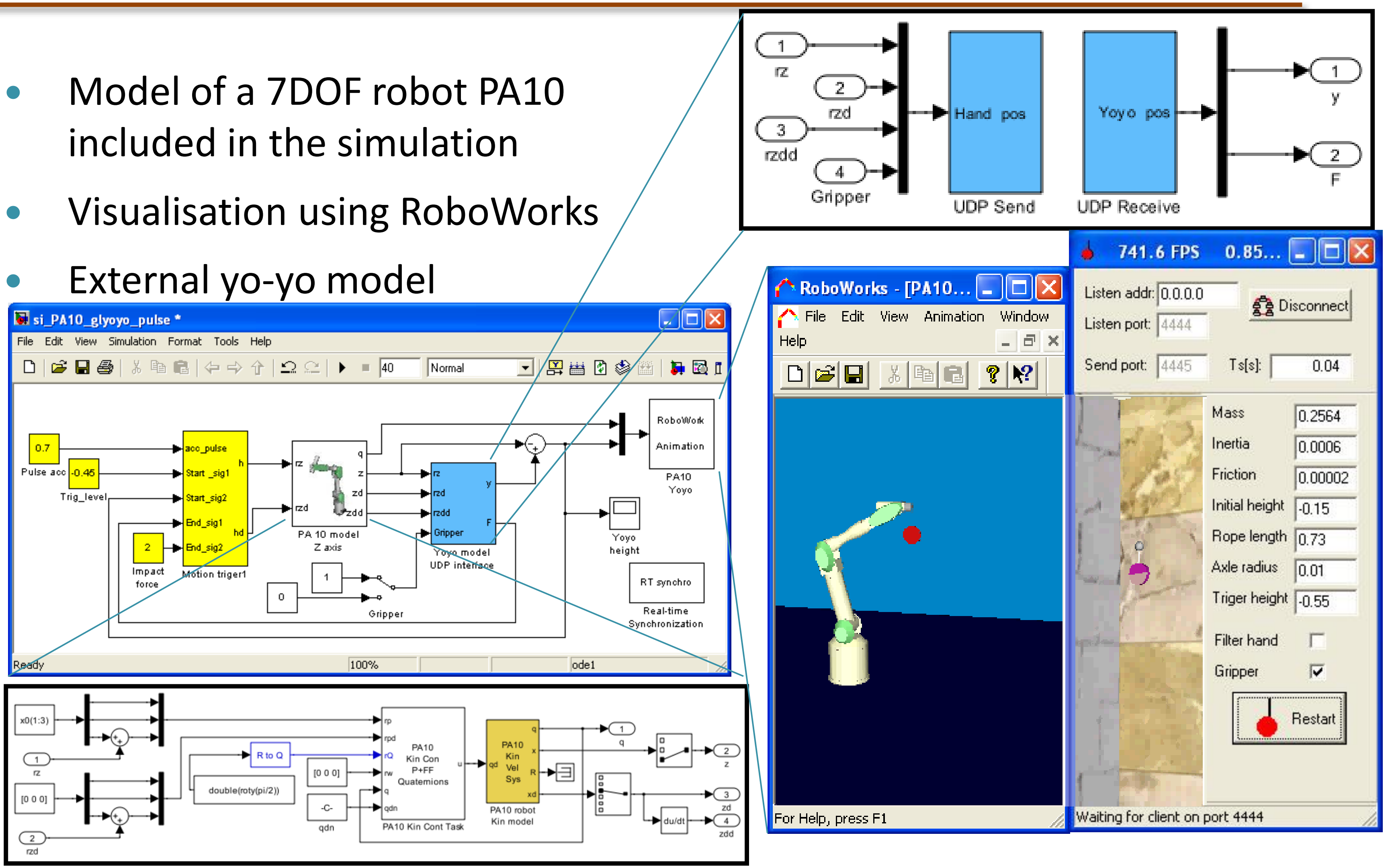
Robot playing Yo-yo: Control strategy

- Design and test different control startegies
- Model of yo-yo in Simulink
- Simple robot model



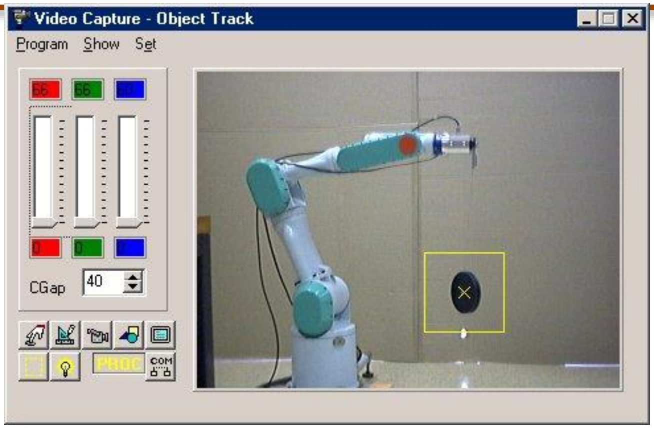
Robot playing Yo-yo: Simulation

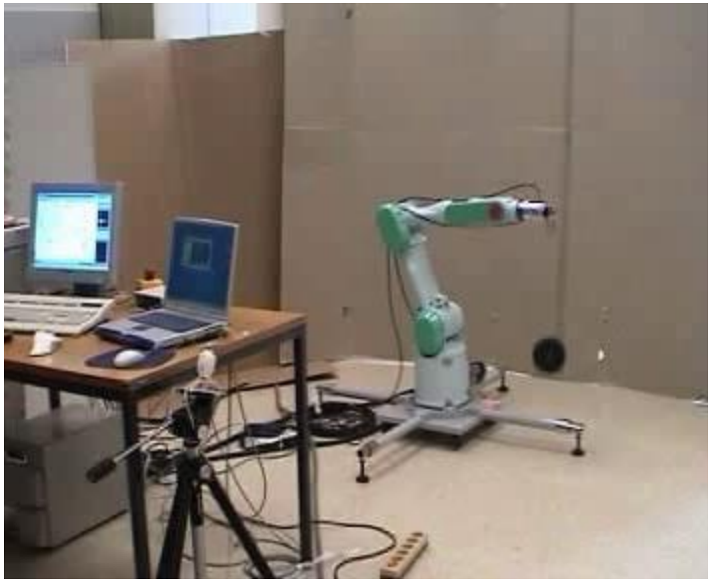
- Model of a 7DOF robot PA10 included in the simulation
- Visualisation using RoboWorks
- External yo-yo model

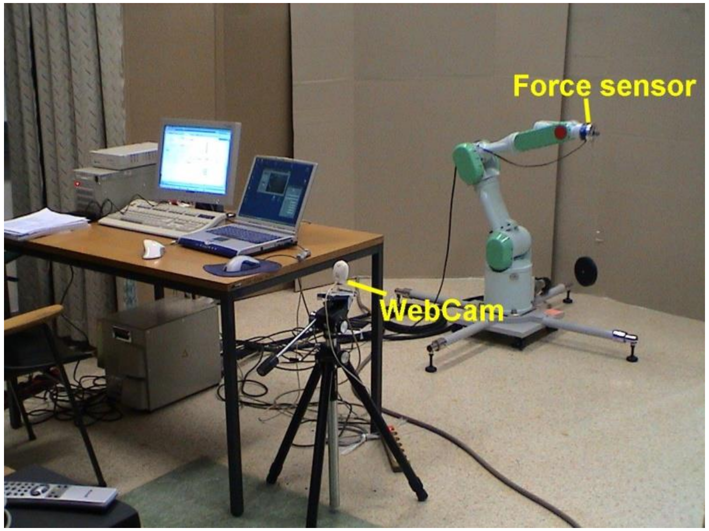


Robot playing Yo-yo: Experiment on a real robot

- Testing on a real robot:
- the position of the yo-yo is now obtained from the vision system
- force sensor is used to measure string forces

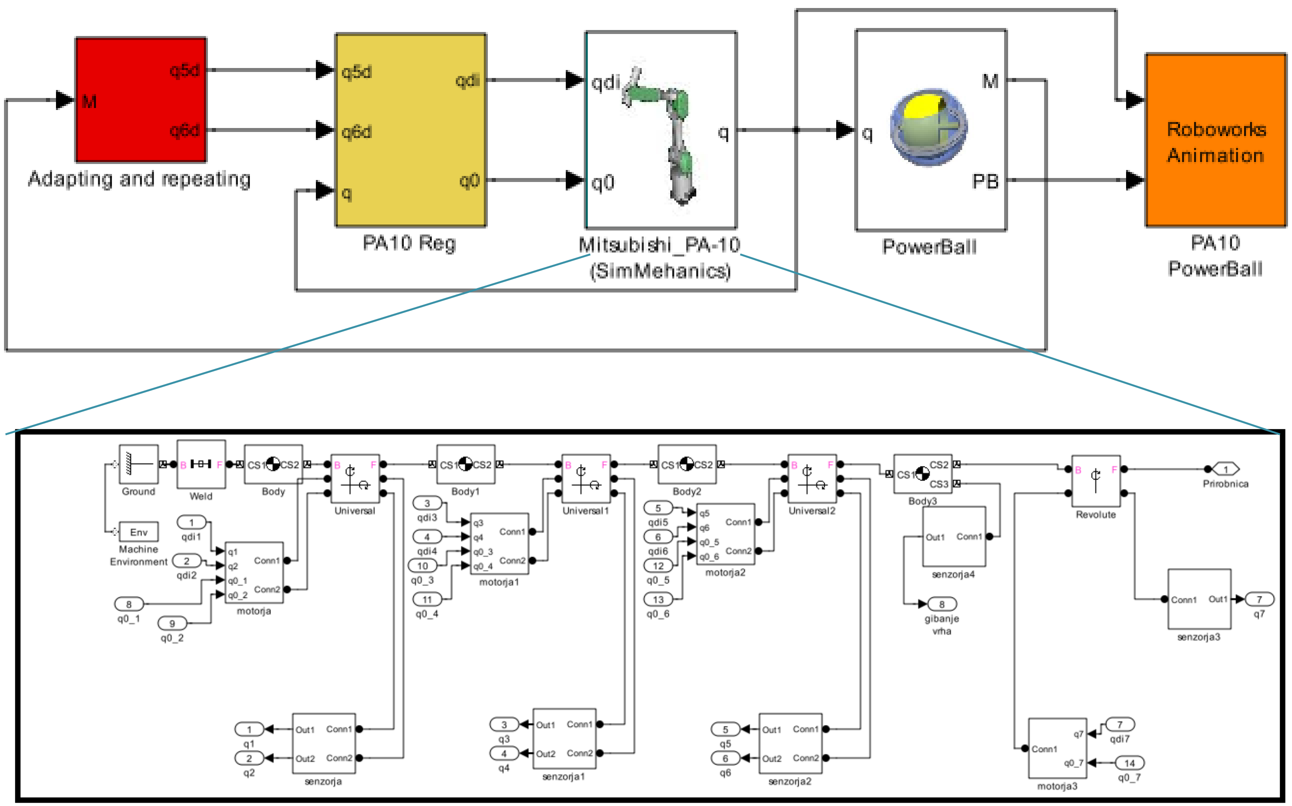


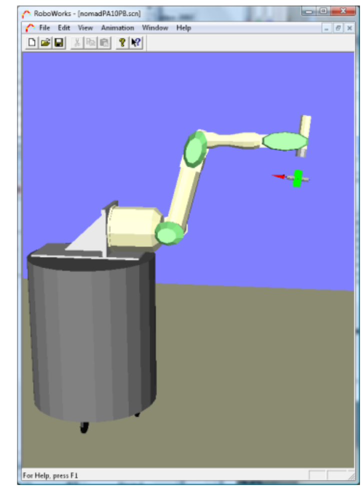


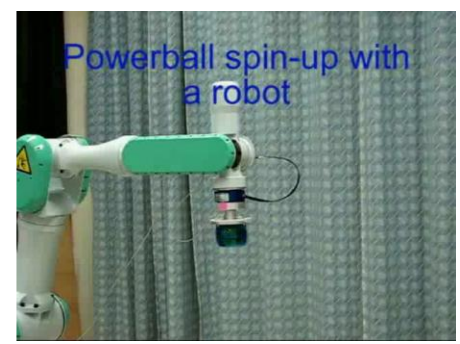


Playing PowerBall

- The robot should perform the spinning of a hand held gyroscopic exerciser



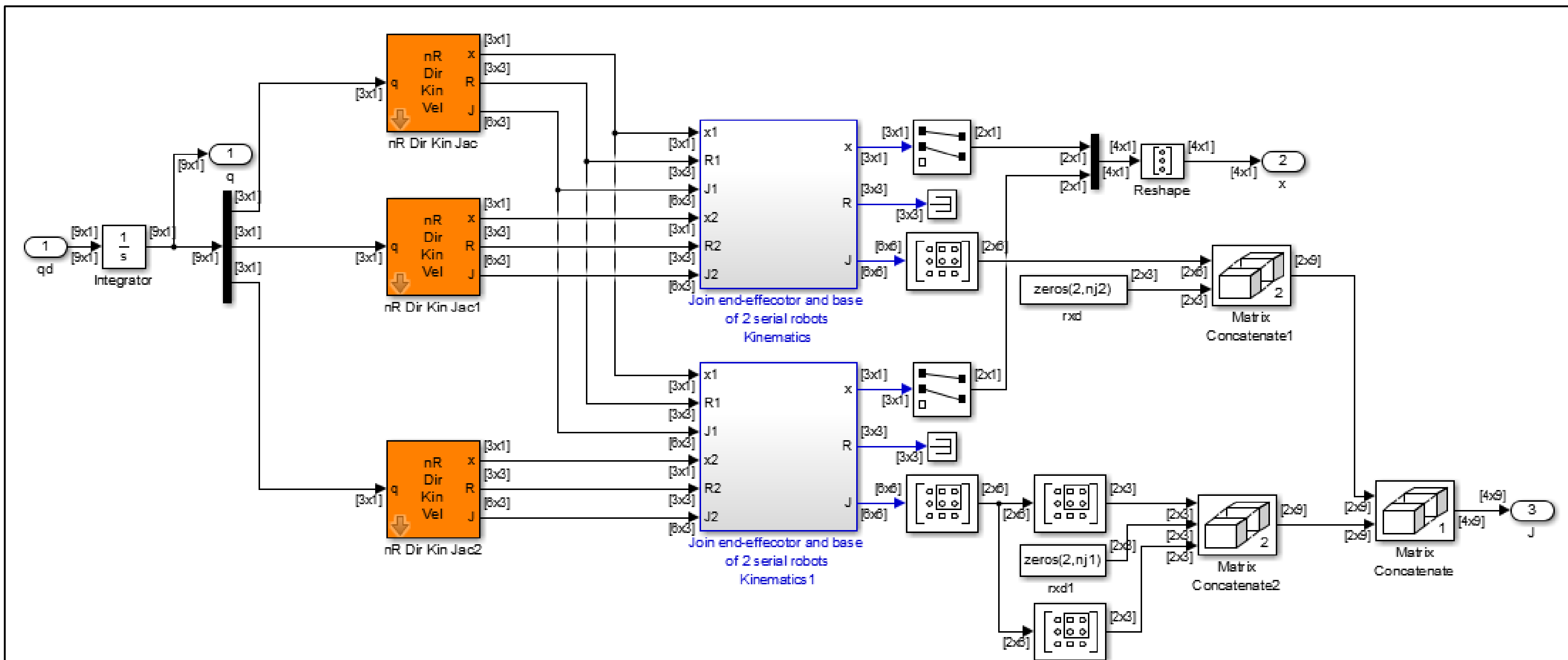




Joining robot arms

There are many tasks that can be performed in cooperation, i.e. using multi-arm robots.

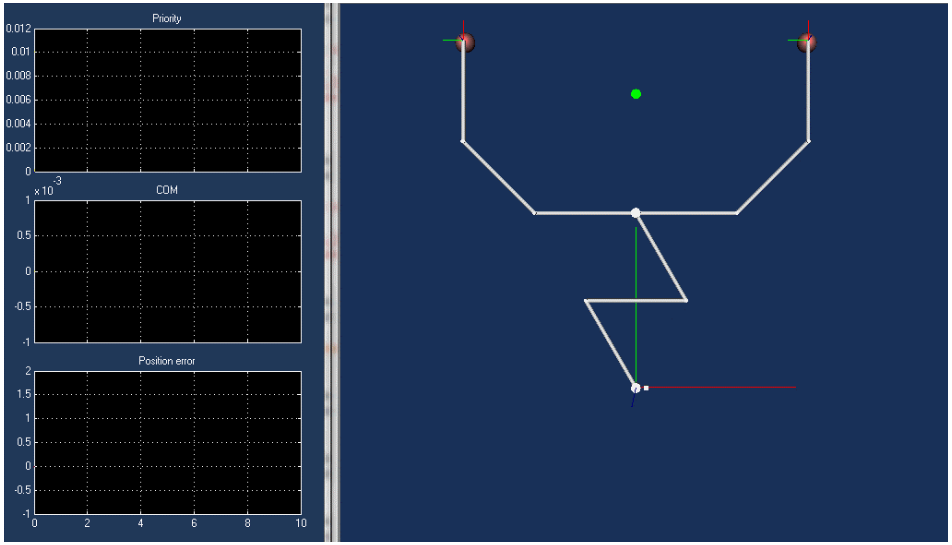
We provide blocks for joining robot arms, i.e. a model of a complete robot system consists of a models of particular robot arm models which are connected using special Simulink blocks.

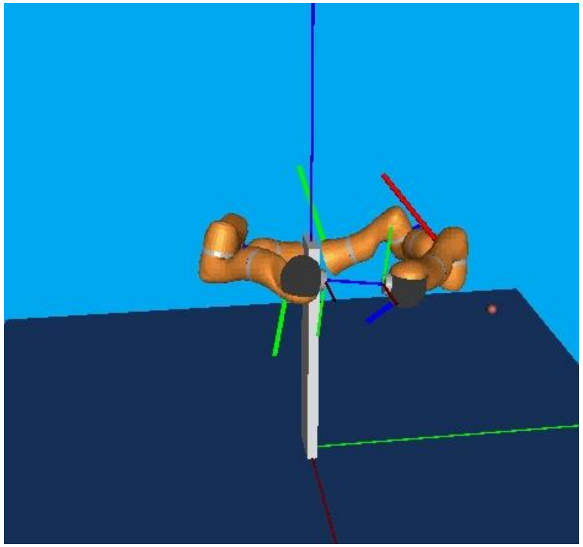


Joining robot arms

There are many tasks that can be performed in cooperation, i.e. using multi-arm robots.

We provide blocks for joining robot arms, i.e. a model of a complete robot system consists of a models of particular robot arm models which are connected using special Simulink blocks.

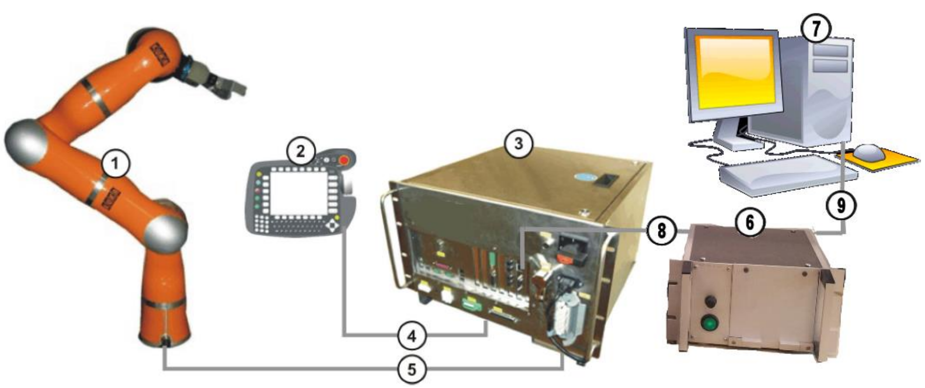




Kuka LWR interface

The KUKA lightweight robot (LWR) is the very versatile robot produced by KUKA with collaboration of the Institute of Robotics andMechatronics at DLR. The Fast Research Interface (FRI) enables to interact with the robot controller. It is also possible to use own control strategies, methods and sensory systems.

For many robotics research topics, like motion planning or task generation at higher levels the update rates are usually relatively low and they are not always in „real-time“.

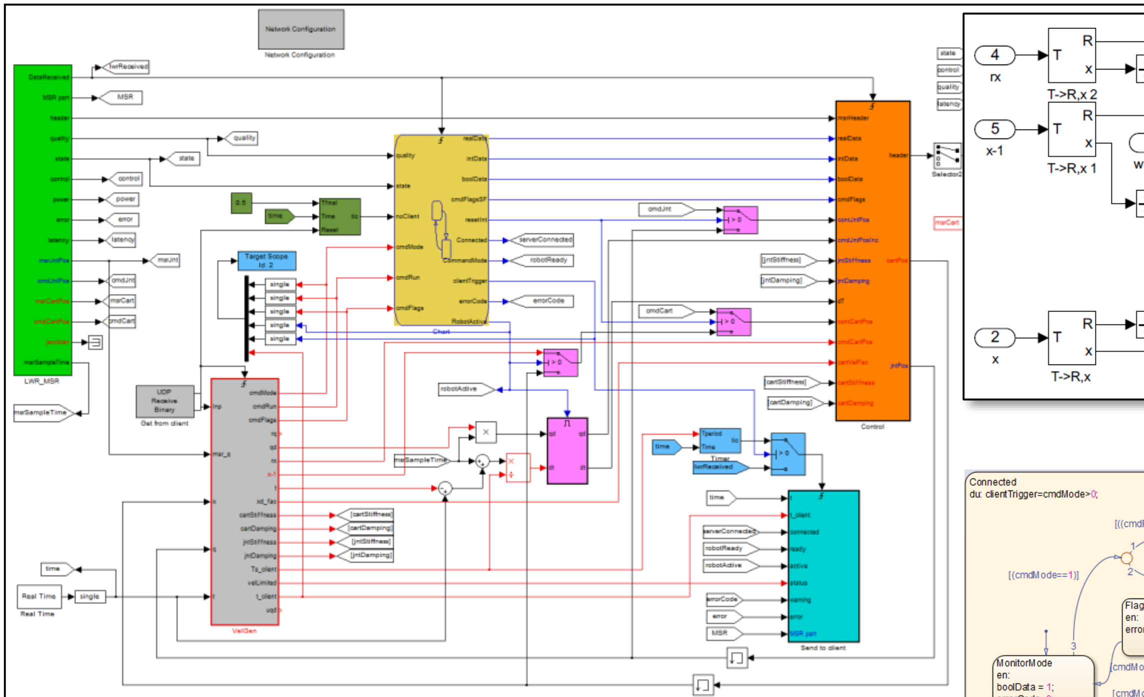


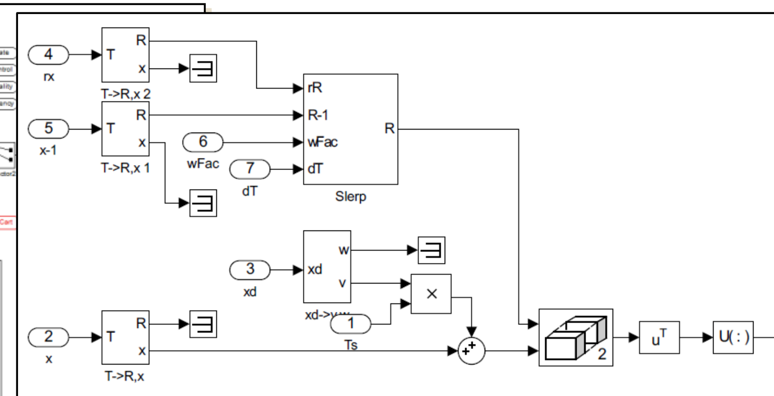
We have developed a server which operates in real-time and has basically the following tasks:

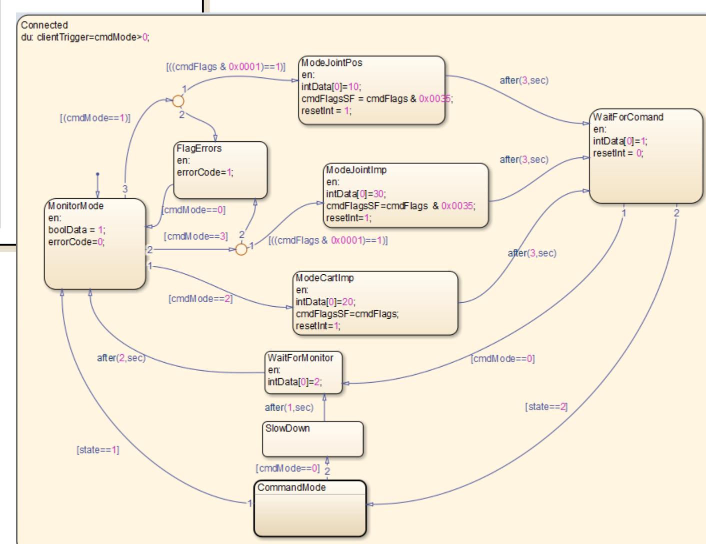
- communication with the FRI fulfilling all the requirements of the FRI UDP socket communication,
- communication with the client computer via UDP socket communication at approximately constant rate, which can be lower than the rate for FRI,
- interpolation of input signals send from the client computer, and
- assures proper change of robot control modes via FRI.

The server is implemented on a computer using MATLAB xPC Target real-time platform.

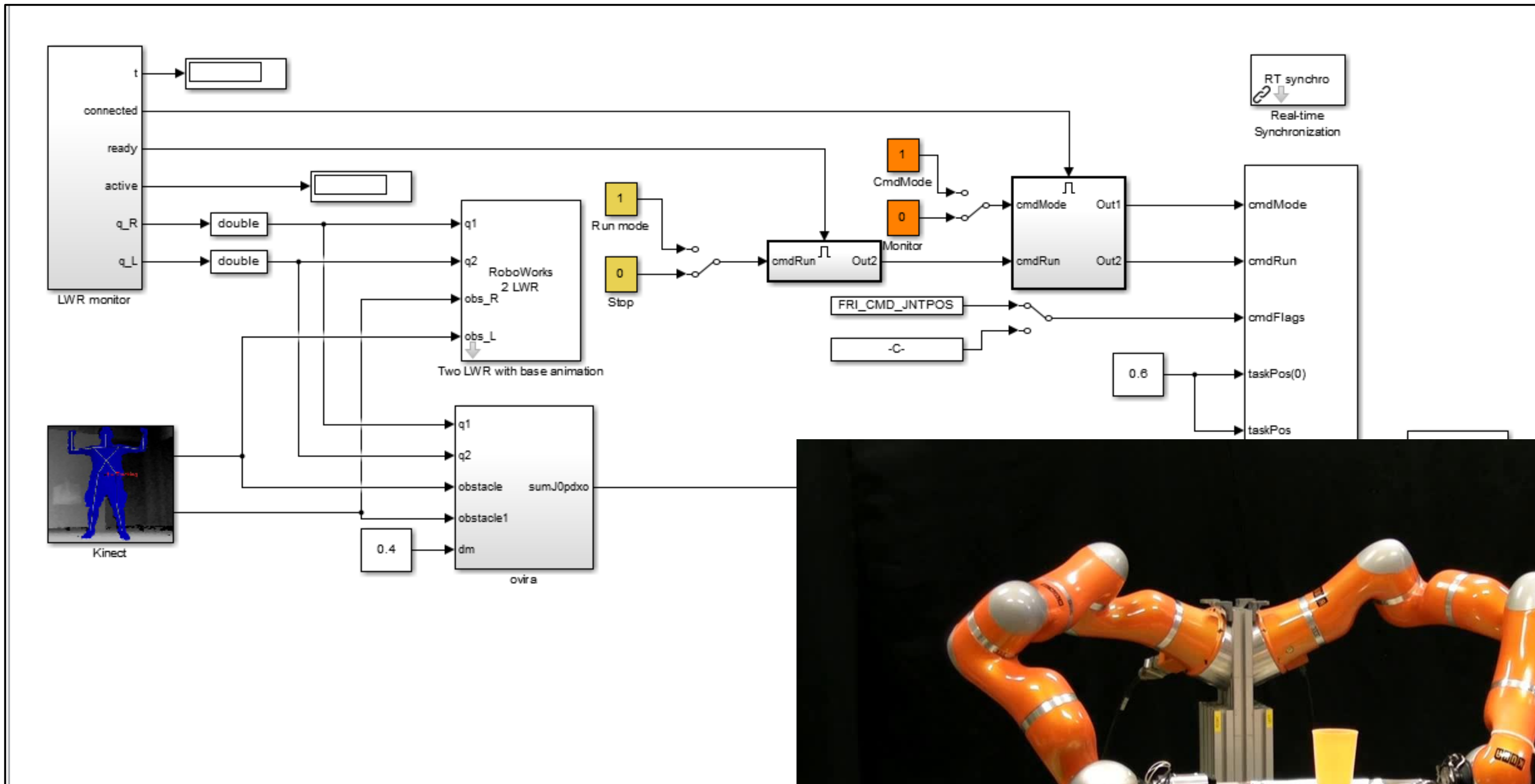
FRI-xPC server for Kuka FRI controller

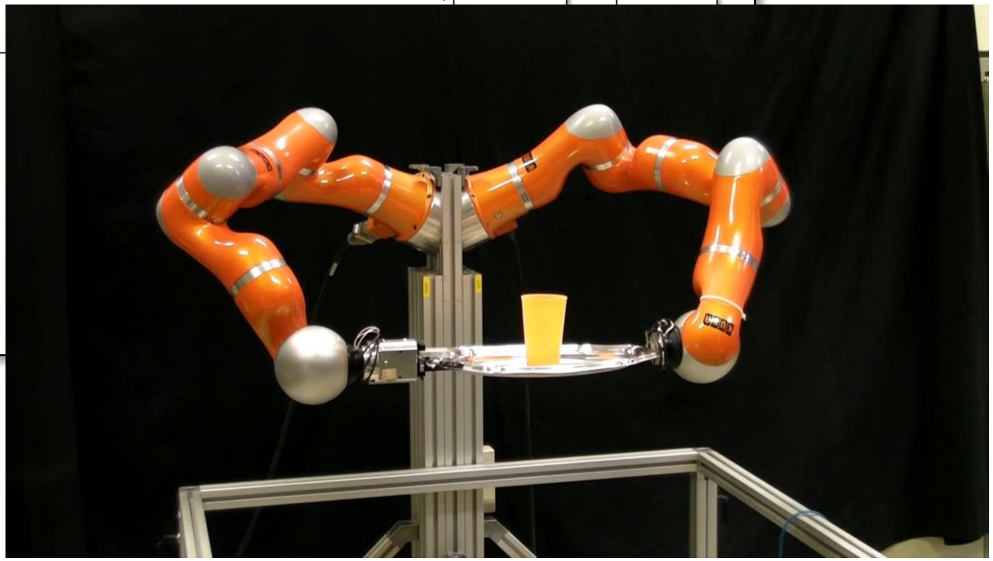






FRI-xPC client for Kuka FRI controller





Simulation of multiple robots in unstructured environment

In last years the research focus in robotics is on:

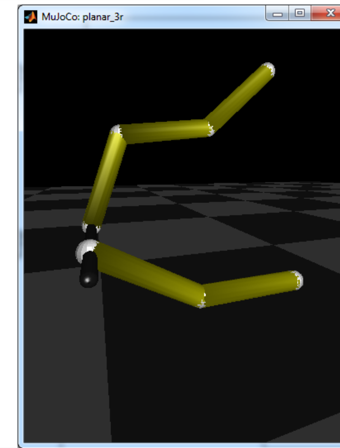
- development of control strategies for multi-arm and cooperating robots (humanoid robots)
- considering physical interaction between the robots and with the environment including contacts

There are different multi-body dynamics simulation engines, which include also contacts.

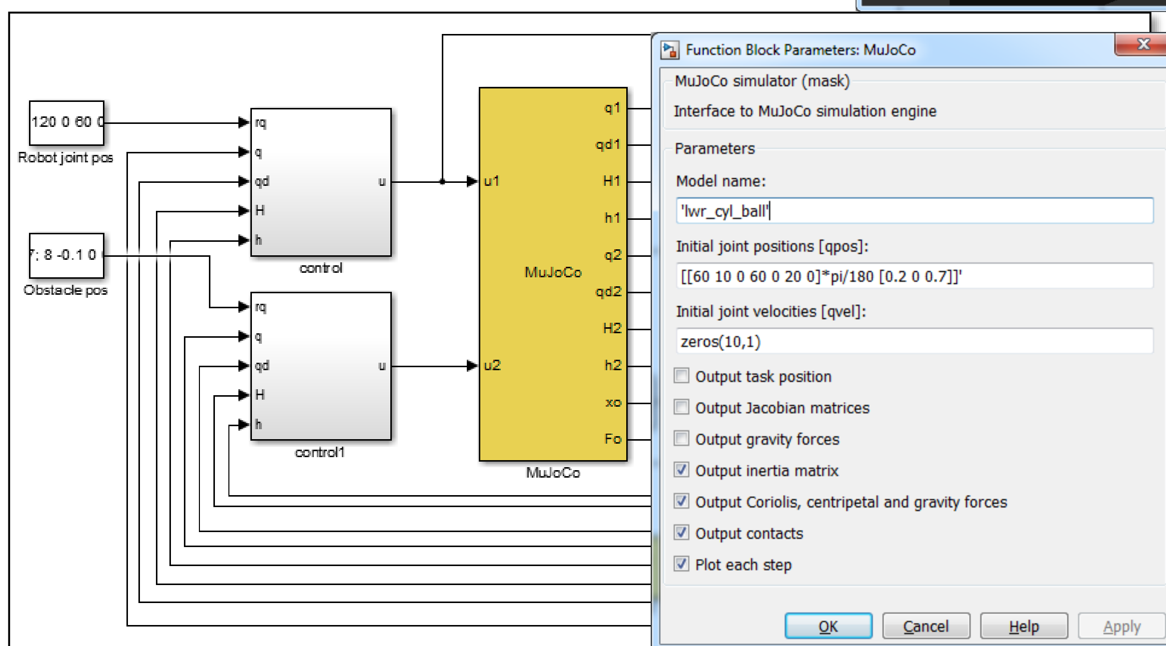
Out latest activities in the simulation in the direction of using these existing physics engines together within our MATLAB/Simulink environment using our already developed simulation blocks and interfaces.

Connecting to MuJoCo engine

- MuJoCo is a physics engine, which offers for a user easy modeling of complex multi-body systems, acurate simulation including dynamic contacts.
- For us it was also important that it is also availabe as a MATLAB toolbox.



- To be used within Simulink environment, we have developed a special block as an interface to MuJoCo MEX file.
- Using the tags in the XML model definition file, we are able to control independantly multiple mechanisms.
- The user can also select different outputs



Conclusions

- The simulation environment can be composed of more than one application (communication via Ethernet)
- The main part is implemented in MATLAB/Simulink (including robot models, sensors, transformations, ...)
- The simulation is augmented with the animation
- External applications which simulate certain robot subsystem and use the same interface as a real system
- Easy testing on real robots (models are replaced with interfaces)

The presented control design environment is a very useful and effective tool for fast and safe development and testing of advanced control schemes and task planning algorithms, including force control and visual feedback in research and education.

Thank you!