

# Development of a control strategy to move a robot arm into the flight trajectory of a ping pong ball on a 2D curve

Michael Garstka

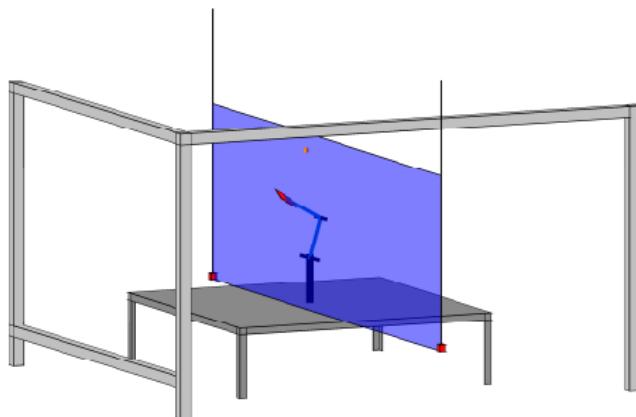
Institute of Control Systems  
Hamburg University of Technology  
Germany

## Project Work

23.03.2016

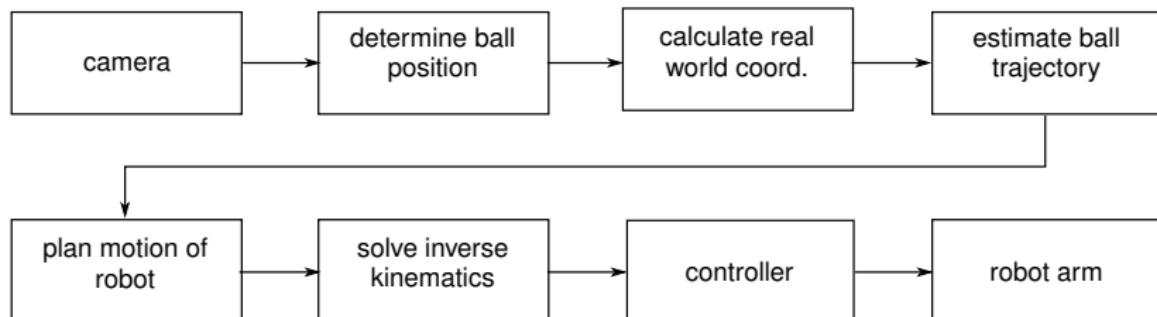
# Motivation

- **Project goal:** Move the robot arm to hit a ping pong ball that is thrown in a 2D plane.
- Subtasks: Position measurement, position estimation, motion planning, control strategy
- Simulate the behavior of the whole system



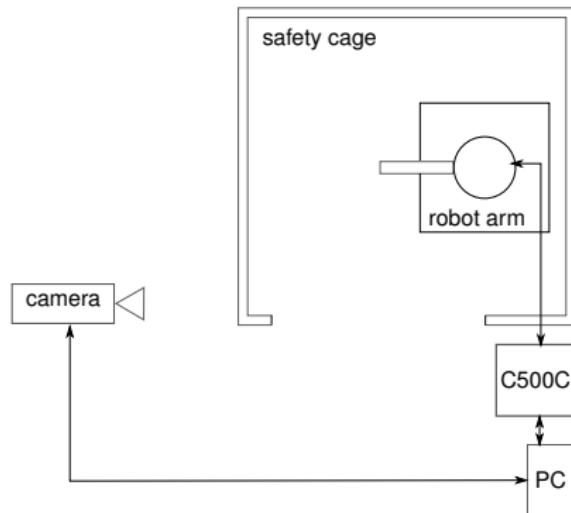
# Contents

- ① Overall Setup of the System
- ② Image Processing and Position Tracking
- ③ Trajectory Prediction
- ④ Robot Model
- ⑤ Motion Planning & Inverse Kinematics
- ⑥ Controller Design
- ⑦ Evaluation of Simulation Results



# Overall Setup of the System

- Basler camera  
( $1920 \times 1200$  px, 165 FPS)
- Personal Computer (Intel i7, 64 bit, 3.4 GHz, 16 GB RAM)
- C500C controller for low-level robot control and security functions



# Camera Calibration and Positioning

- **Goal of Camera Calibration:**

Measurements in the image in pixel domain and real world coordinates

- **2 Steps:**

- ① Find intrinsic camera parameters to undistort image

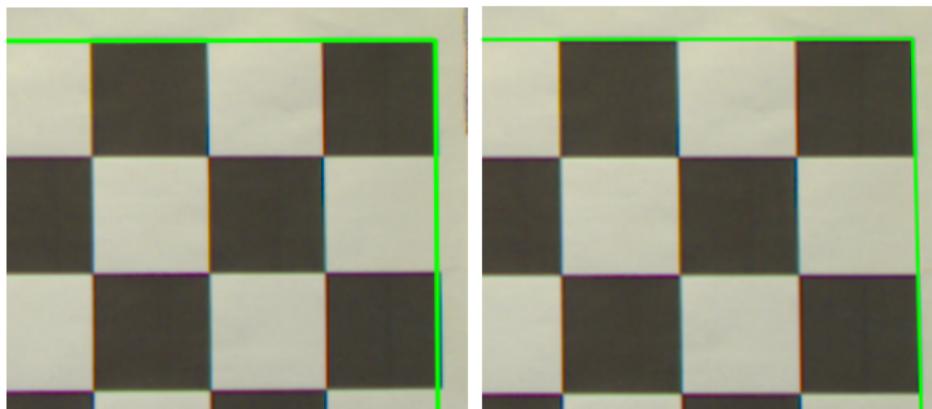
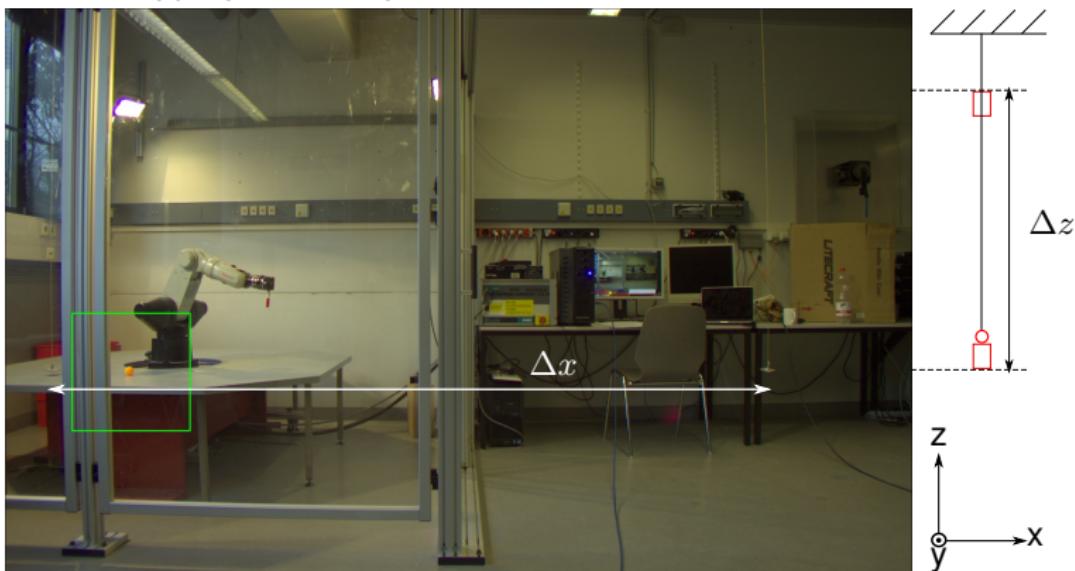


Figure: Distorted (left) and undistorted image (right).

# Camera Calibration and Positioning

- ② Find a mapping between pixel domain and real world coordinates

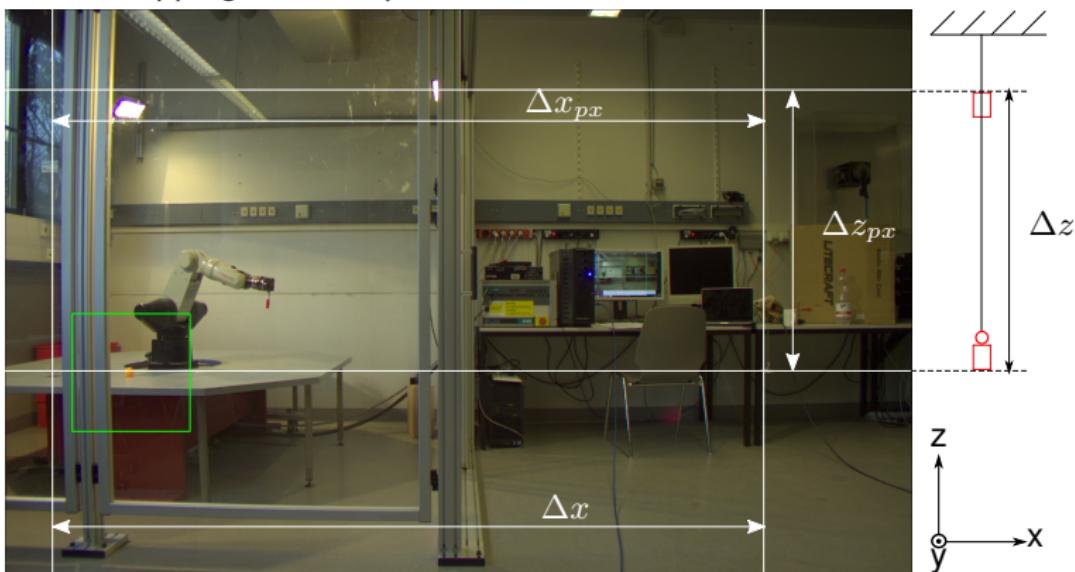


$$x = x_{px} \cdot \frac{\Delta x}{\Delta x_{px}}$$

$$z = (z_{max} - z_{px}) \cdot \frac{\Delta z}{\Delta z_{px}}.$$

# Camera Calibration and Positioning

- ② Find a mapping between pixel domain and real world coordinates

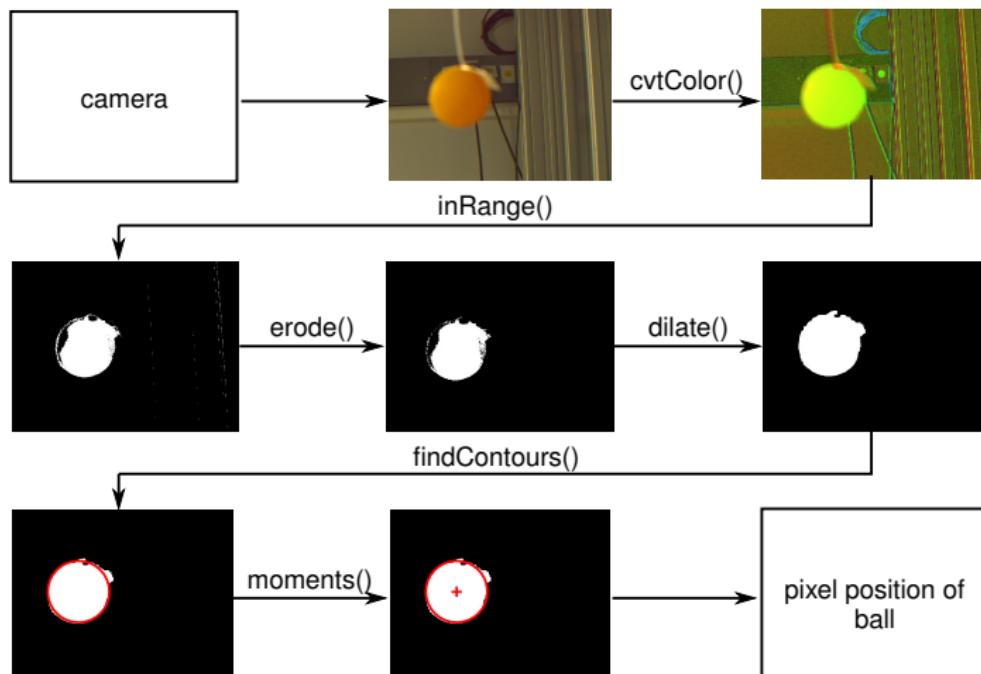


$$x = x_{px} \cdot \frac{\Delta x}{\Delta x_{px}}$$

$$z = (z_{max} - z_{px}) \cdot \frac{\Delta z}{\Delta z_{px}}.$$

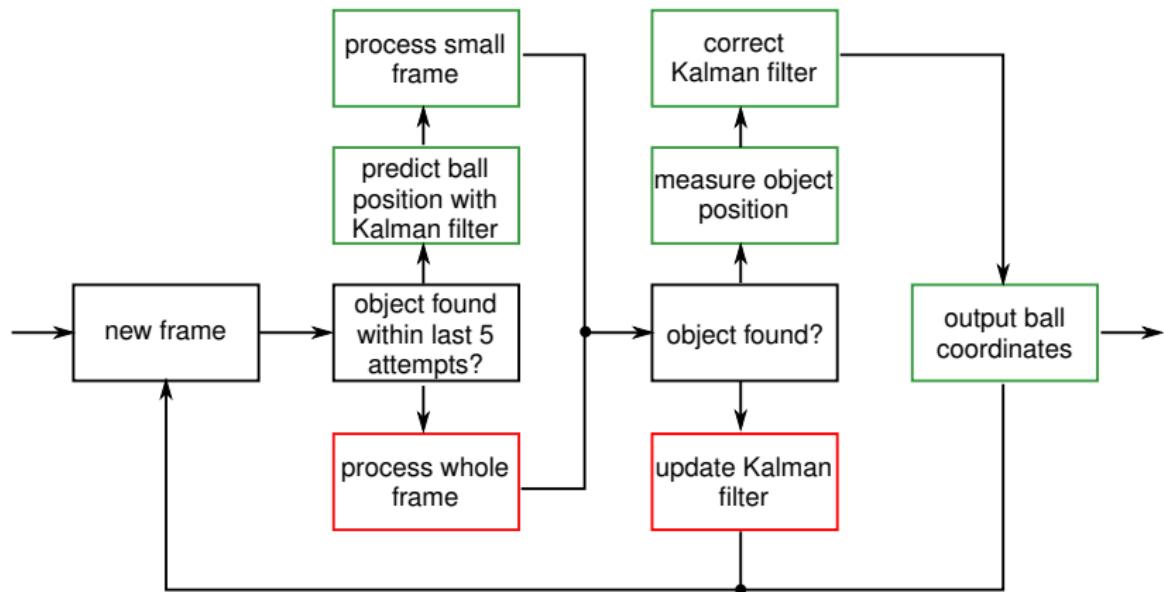
# Image Processing & Object Detection

- Approach: Detect ball by its orange color
- OpenCV C/C++ library provides basic image processing functions



# Position Tracking Algorithm

- **Task:** Detect and track the ball in incoming camera images
- Speed of algorithm: 30 FPS (whole frame); 70 FPS (dynamic windowing)



# Video of Position Tracking

Video: `wurf1geschnitten.mp4`

# Evaluation of Tracking Algorithm

- 34 measurements were taken with the camera and the described algorithm

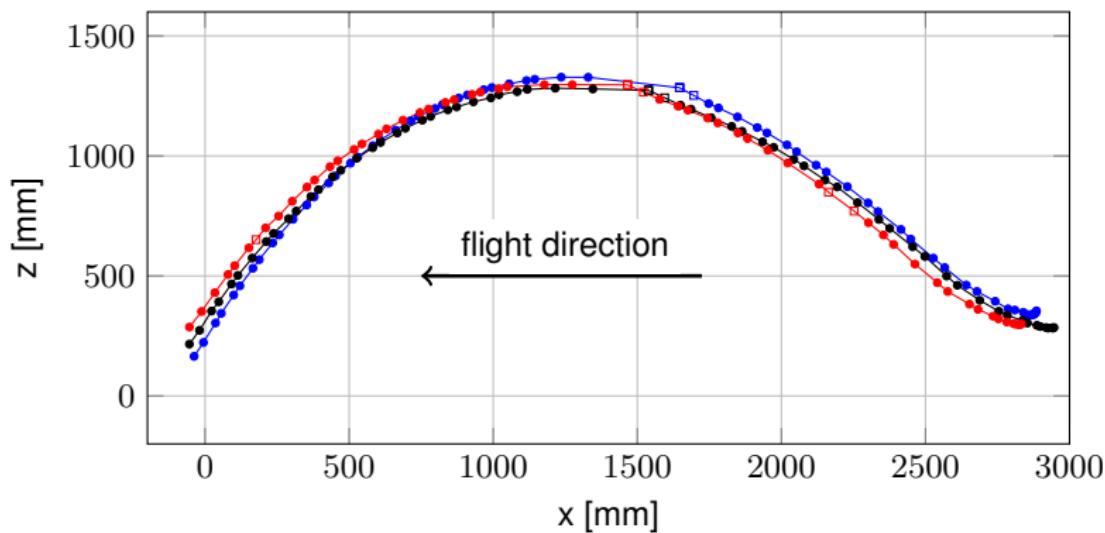


Figure: Three measurements of the ball trajectory. Dots: detected position; Rectangle: predicted position.

# Evaluation of Tracking Algorithm

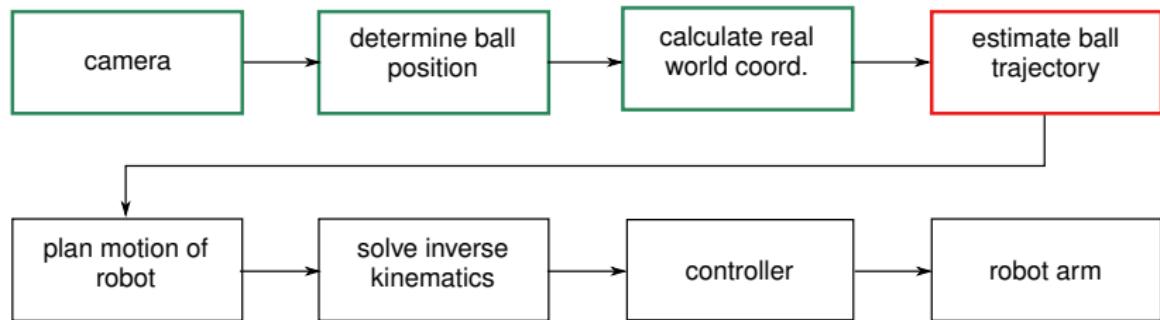
- 34 measurements were taken with the camera and the described algorithm

Table: Key numbers for 34 measurements

Name	Mean Value $\mu(X)$	Std. Dev. $\sigma(X)$
Flight time	0.981 s	0.081 s
Distance travelled	2872.1 mm	74.1 mm
Num. of data points	58.353	6.4
Num. of detections	50.5	7.3
mm/detection	57.054 mm	7.49 mm
Detection rate	86.54 %	
Avg. processing speed	60 FPS	

- Avg. processing speed and detection rate comparable to other approaches.
- High sensitivity of algorithm to changes in lighting and reflections

# Overview - Flow Chart



# Trajectory Prediction

- Goal: Predict flight trajectory of the ball
- ① Model dynamics of the ball
  - ② Determine initial conditions from set of measured positions
  - ③ Numerically integrate equations of motion

$$m\dot{v} = F_G + F_M + F_D$$

$$F_G = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T$$

$$F_D = -\frac{1}{2}\rho S C_D ||v|| v$$

$$F_M = \rho \omega r_b S C_L v$$

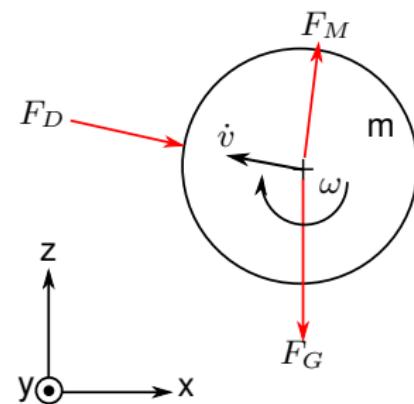
$\rho$  air density

$C_L$  lift coefficient

$S$  cross-sectional area

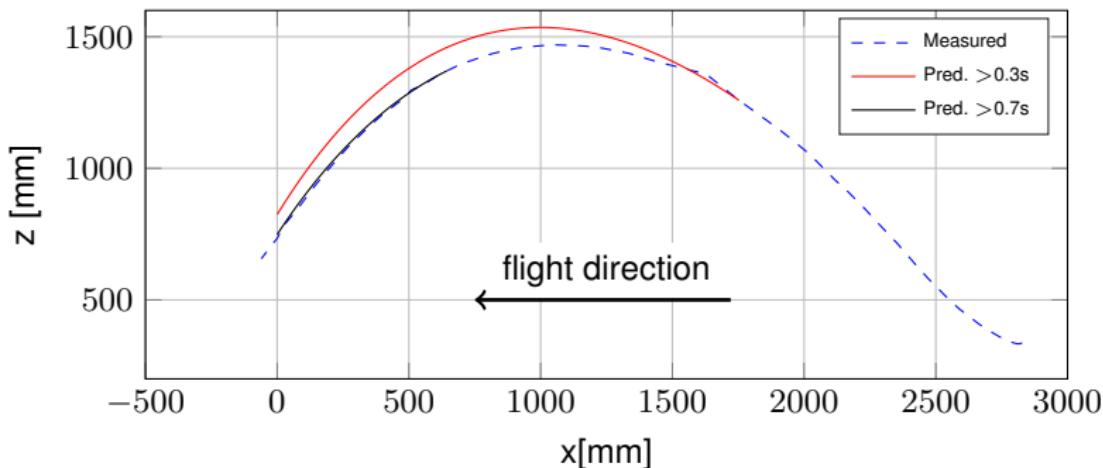
$r_b$  radius of ball

$C_D$  drag coefficient



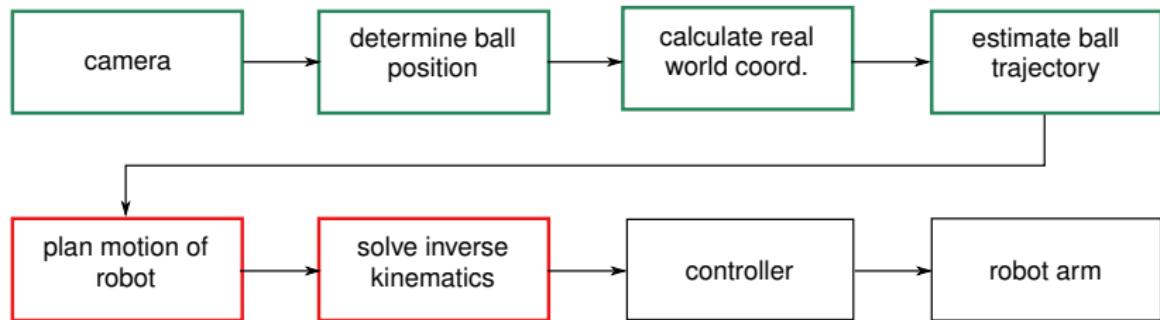
## Trajectory Prediction - Example

- Two predictions after 0.3 s and 0.7 s
- Ten previous data points were used to determine  $v_{x,0}$ ,  $v_{y,0}$



- Measured position:  $z_{\text{meas}} = 701.4 \text{ mm}$
- Predicted position error ( $t_F = 0.3 \text{ s}$ ):  $e_{p,0.3} = 90 \text{ mm}$
- Predicted position error ( $t_F = 0.7 \text{ s}$ ):  $e_{p,0.7} = 14 \text{ mm}$

# Overview - Flow Chart



# Robot Model

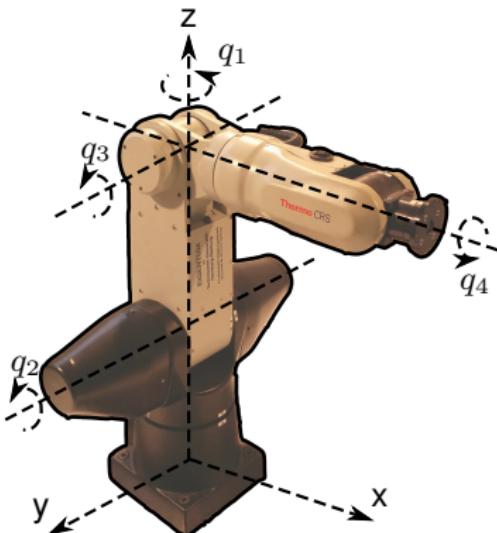
- Six rotational joints controlled by incremental encoders
- First four DOFs are used during this project

- Equations of motion:

$$M(q)\ddot{q} + c(q, \dot{q}) + g(q) + \tau_f(\dot{q}) = \tau$$

$$\ddot{q} = M(q)^{-1} (\tau - c(q, \dot{q}) - g(q) - \tau_f(\dot{q}))$$

- The system was identified in previous projects



# Motion Planning & Inverse Kinematics

- **Task:** Turn information about predicted position of ball into reference signal for robot arm in joint space
- ① Plan motion of end effector, i.e center point of ping pong paddle
- ② Solve inverse kinematics problem

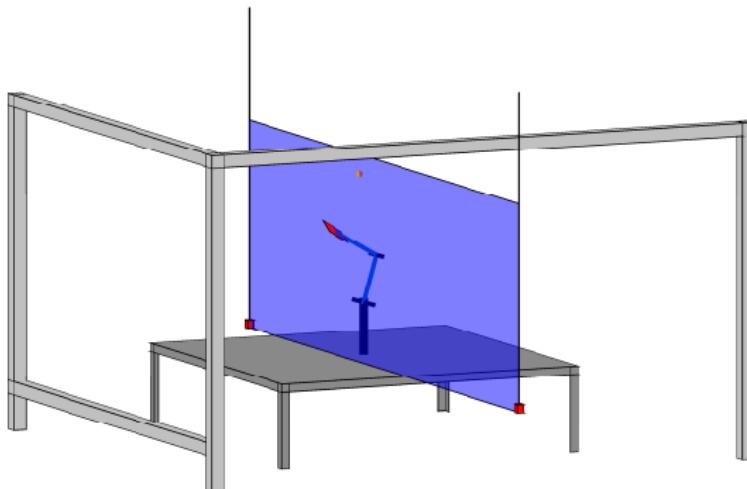


Figure: Plane of motion of the ball (blue) and robot arm in desired position.

# Motion Planning & Inverse Kinematics

**Task:** Predicted position of ball:  $(x_p, y_p, z_p) \rightarrow$  Robot joint angles  $q_1, q_2, q_3, q_4$

① Set  $q_1 = -90^\circ$

② Compute  $q_2$  and  $q_3$ :

$$d_y = y_p - l_{23} \sin(-q_2) - l_{34} \sin(-\tilde{q}_3)$$

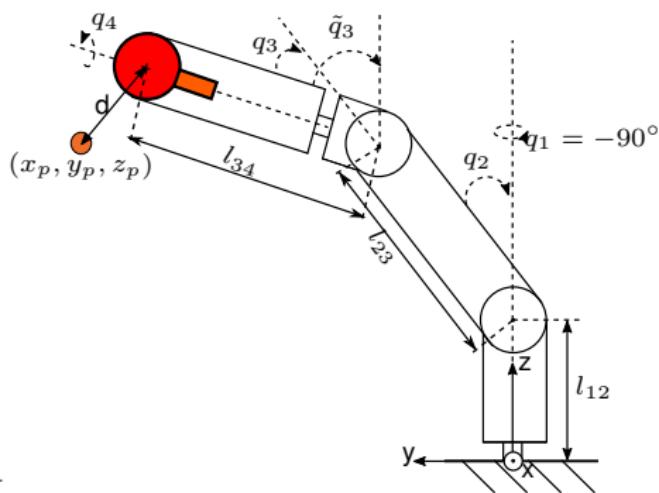
$$d_z = z_p - l_{12} - l_{23} \cos(-q_2) - l_{34} \cos(-\tilde{q}_3)$$

$$d = \| \begin{bmatrix} d_y & d_z \end{bmatrix}^T \|_2$$

$$\min_{q_2, \tilde{q}_3} d(q_2, \tilde{q}_3) \text{ such that: } \begin{cases} q_{2,\min} \leq q_2 \leq q_{2,\max} \\ \tilde{q}_{3,\min} \leq \tilde{q}_3 \leq \tilde{q}_{3,\max} \end{cases}$$

③ Compute  $q_4$ :

$$q_4 = \theta + \beta = \theta + (90^\circ - q_4) \Leftrightarrow q_4 = \frac{\theta + 90^\circ}{2}$$



# Motion Planning & Inverse Kinematics

**Task:** Predicted position of ball:  $(x_p, y_p, z_p) \rightarrow$  Robot joint angles  $q_1, q_2, q_3, q_4$

① Set  $q_1 = -90^\circ$

② Compute  $q_2$  and  $q_3$ :

$$d_y = y_p - l_{23} \sin(-q_2) - l_{34} \sin(-\tilde{q}_3)$$

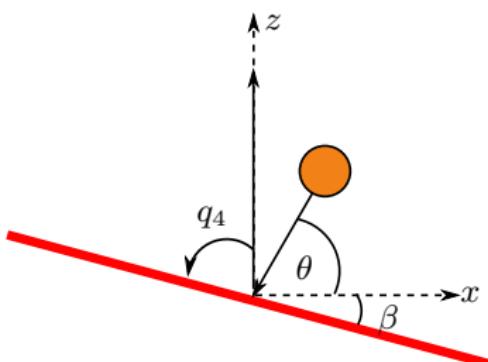
$$d_z = z_p - l_{12} - l_{23} \cos(-q_2) - l_{34} \cos(-\tilde{q}_3)$$

$$d = \left\| \begin{bmatrix} d_y & d_z \end{bmatrix}^T \right\|_2$$

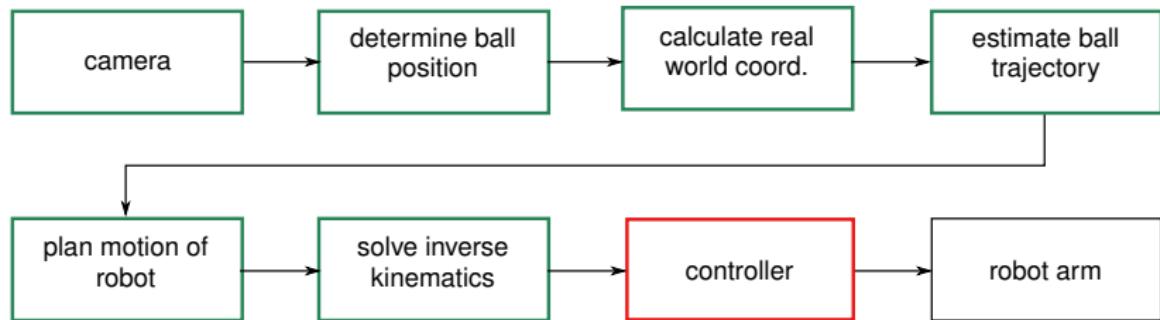
$$\min_{q_2, \tilde{q}_3} d(q_2, \tilde{q}_3) \text{ such that: } \begin{cases} q_{2,\min} \leq q_2 \leq q_{2,\max} \\ \tilde{q}_{3,\min} \leq \tilde{q}_3 \leq \tilde{q}_{3,\max} \end{cases}$$

③ Compute  $q_4$ :

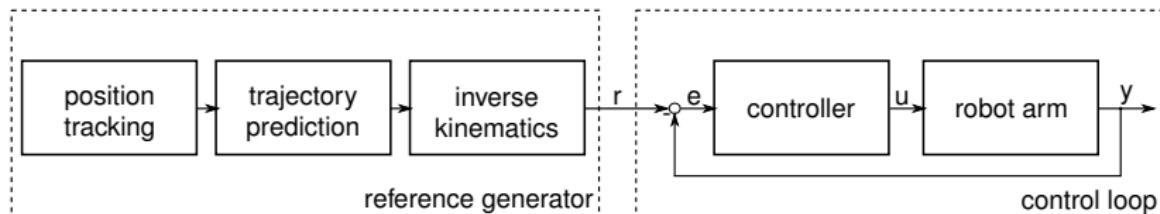
$$q_4 = \theta + \beta = \theta + (90^\circ - q_4) \Leftrightarrow q_4 = \frac{\theta + 90^\circ}{2}$$



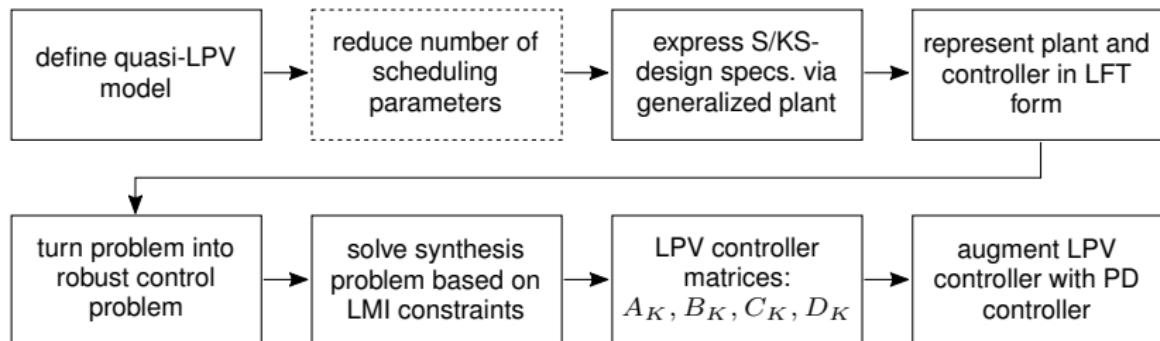
# Overview - Flow Chart



# Controller Design - Control Loop



- Two reference vector signals  $r$  are calculated based on two predictions
- DOFs  $q_4, q_5, q_6$  are well-decoupled from DOFs  $q_1, q_2, q_3$   
→ LPV controller for  $q_1, q_2, q_3$  + PD controller for  $q_4$



# Controller Design - Quasi-LPV model

- LPV model  $P_\theta$  for 3DOF-robot:

$$\dot{x} = A(\theta(t))x + B(\theta(t))u$$

$$y = Cx + Du$$

with:  $A(\theta(t)) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \theta_1 & 0 & 1 \\ 0 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \\ 0 & \theta_7 & \theta_8 & \theta_9 & \theta_{10} & \theta_{11} \end{bmatrix}$ ,  $B(\theta(t)) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \theta_{12} & 0 & 0 \\ 0 & \theta_{13} & \theta_{14} \\ 0 & \theta_{15} & \theta_{16} \end{bmatrix}$ ,

$$C = \begin{bmatrix} I_3 & 0_{3 \times 3} \end{bmatrix}, \quad D = 0_{3 \times 3}.$$

- 5 Scheduling signals:

$$\rho(t) = [\rho_1 \quad \rho_2 \quad \rho_3 \quad \rho_4 \quad \rho_5]^T = [q_2 \quad \tilde{q}_3 \quad \dot{q}_1 \quad \dot{q}_2 \quad \dot{\tilde{q}}_3]^T.$$

- 9 rational LFT parameters  $\delta(\rho(t))$ :

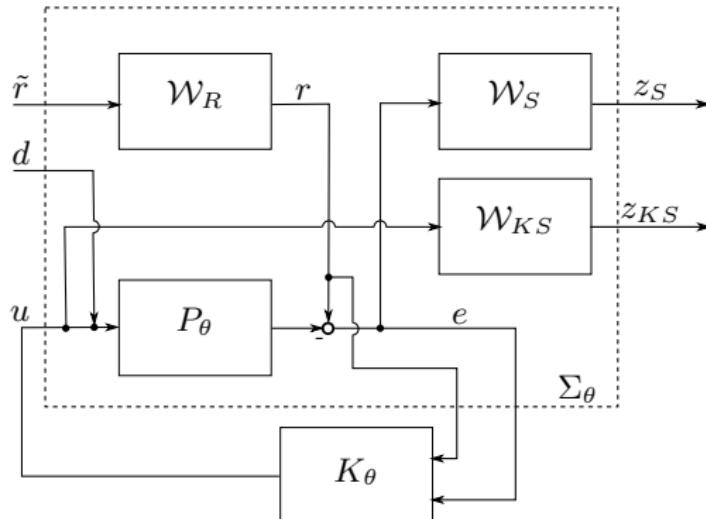
$$\begin{array}{llllll} \delta_1 & \cos(\rho_1) & \delta_4 & \sin(\rho_2) & \delta_7 & \rho_5 \\ \delta_2 & \cos(\rho_2) & \delta_5 & \rho_3 & \delta_8 & \text{sinc}(\rho_1) \\ \delta_3 & \sin(\rho_1) & \delta_6 & \rho_4 & \delta_9 & \text{sinc}(\rho_2) \end{array}$$

- 16 affine LFT parameters  $\theta(t) = [\theta_1, \dots, \theta_{16}]^T \rightarrow$  Reducible to 3 or 4 parameters  
(via PCA)

[Hashemi et al. (2013)]

# Controller Design - Mixed S/KS-Design

- **Design goals:** Fast and accurate tracking, good noise and disturbance rejection



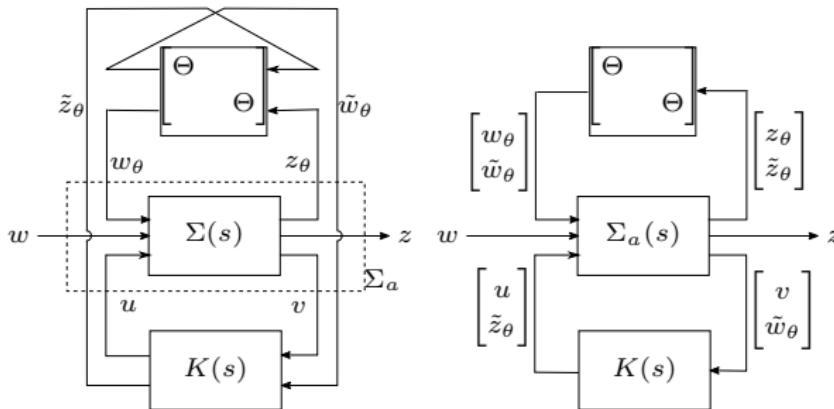
- Two-degrees-of-freedom-control: Separate feed-through of  $r$ .
- Problem: Find gain-scheduled controller  $K_\theta$ , that stabilizes closed-loop and achieves

$$\|T_{wz}\|_{\mathcal{L}_2} = \left\| \begin{bmatrix} \mathcal{W}_S S \\ \mathcal{W}_{KS} K_S \end{bmatrix} \right\|_{\mathcal{L}_2} < \gamma.$$

[Hoffmann et al. (2013)]

# Controller Design - Synthesis based on LFT-framework

- LFT-framework to transform LPV problem into a robust control problem

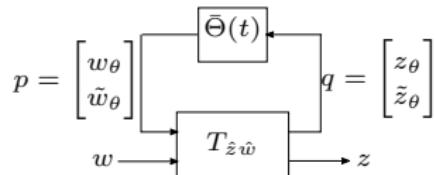


Find a LTI controller  $K(s)$  and a scaling matrix  $L \in \mathcal{L}$  such that the nominal closed-loop system  $\mathcal{F}_l(\Sigma_a, K)$  is stable and satisfies:

$$\left\| \begin{pmatrix} L^{-1} & 0 \\ 0 & \gamma^{-1} \end{pmatrix} \mathcal{F}_l(\Sigma_a, K(s)) \begin{pmatrix} L & 0 \\ 0 & I \end{pmatrix} \right\|_{\infty} < 1.$$

- The gain-scheduled controller  $K_\theta$  is then given by  $\mathcal{F}_l(K(s), \Theta)$ .

## Controller Design - Full Block S-Procedure



$$\begin{bmatrix} \dot{\zeta} \\ q \\ z \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B}_p & \mathcal{B}_w \\ \mathcal{C}_q & \mathcal{D}_{qp} & \mathcal{D}_{qw} \\ \mathcal{C}_z & \mathcal{D}_{zp} & \mathcal{D}_{zw} \end{bmatrix} \begin{bmatrix} \zeta \\ p \\ w \end{bmatrix}.$$

The closed loop LPV system  $T_\theta = \mathcal{F}_u(T_{\hat{z}\hat{w}}, \bar{\Theta})$  is well-posed, stable and  $\|T_\theta\| < \gamma$  if and only if there exists a  $X > 0$  and a symmetric multiplier  $W$  with  $Q = Q^T$ ,  $R = R^T$  and  $S$  such that

$$\begin{bmatrix} \bar{\Theta} \\ I \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}}_W \begin{bmatrix} \bar{\Theta} \\ I \end{bmatrix} > 0 \quad \forall \bar{\Theta}(t) \in \Delta. \quad (\text{multiplier condition})$$

and

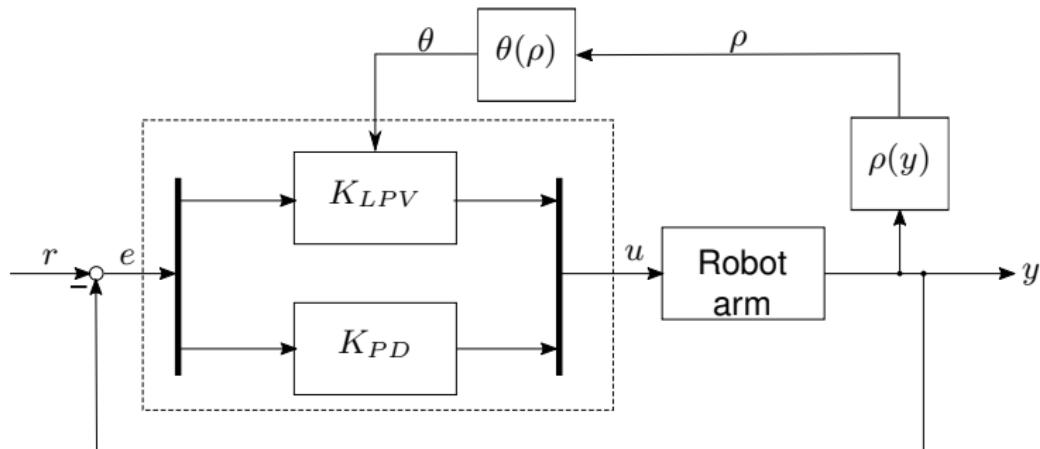
$$[*] \quad \begin{bmatrix} 0 & X & 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q & S & 0 & 0 \\ 0 & 0 & S^T & R & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma I & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma^{-1} I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ \mathcal{A} & \mathcal{B}_p & \mathcal{B}_w \\ 0 & I & 0 \\ \mathcal{C}_q & \mathcal{D}_{qp} & \mathcal{D}_{qw} \\ 0 & 0 & I \\ \mathcal{C}_z & \mathcal{D}_{zp} & \mathcal{D}_{zw} \end{bmatrix} < 0. \quad (\text{nominal condition})$$

- Inequalities can be turned into LMIs to find  $A_K, B_K, C_K, D_K$ .

[Scherer (2001)]

# Controller Design - Augmented LPV-PD-controller

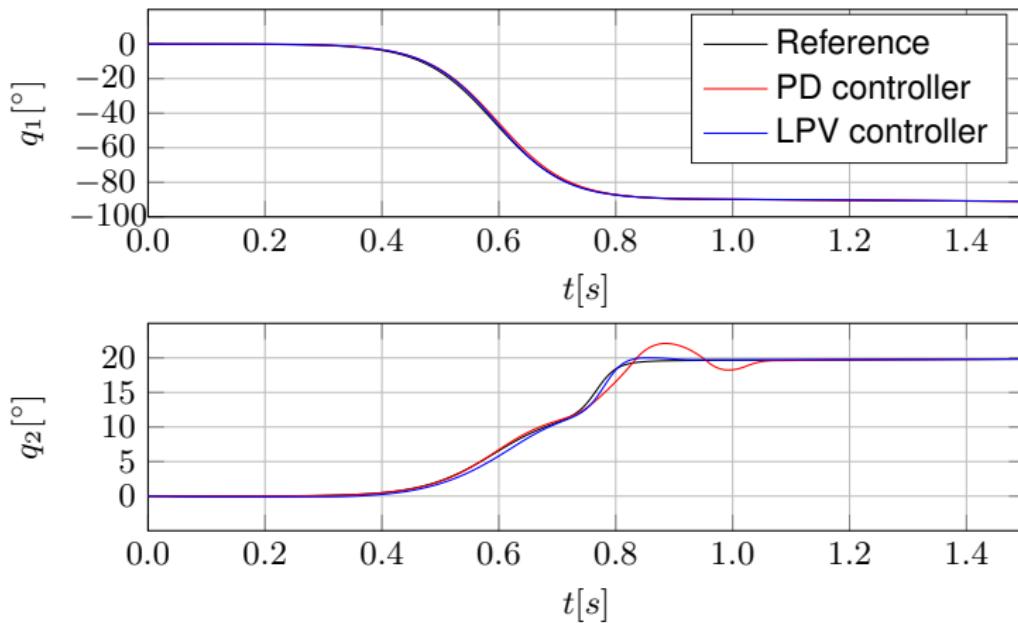
- LPV controller for  $q_1, q_2, q_3$ ; PD controller for  $q_4$



- Reduction of complexity needed for implementation on real system
  - PCA to reduce number of scheduling parameters
  - Polytopic approach

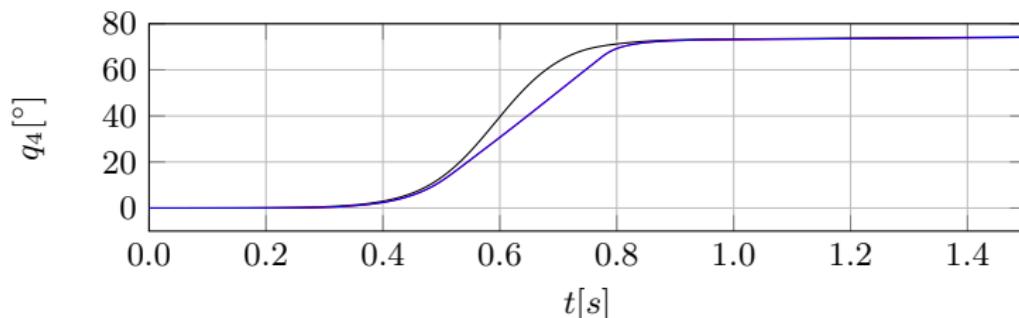
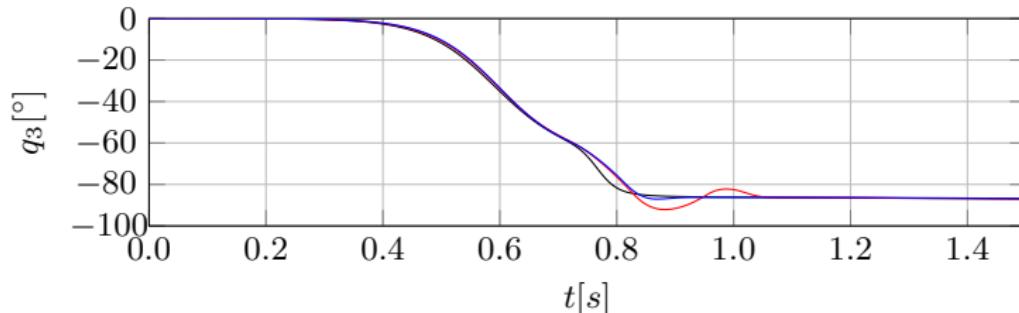
# Simulation results

- Simulation of the system in MATLAB and SIMULINK
- Two predictions after 0.3 s and 0.7 s



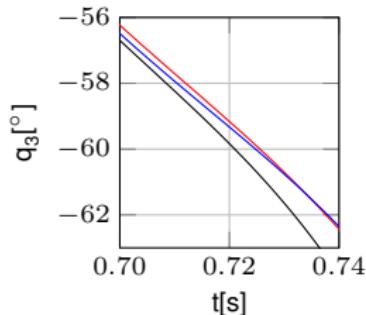
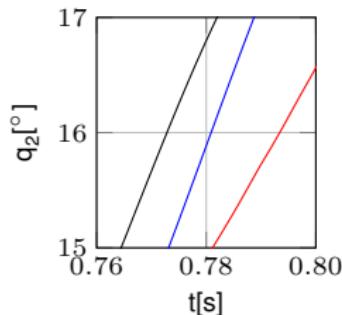
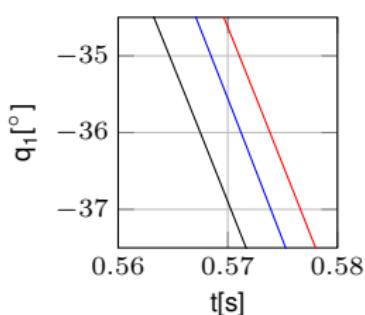
# Simulation results

- Simulation of the system in MATLAB and SIMULINK
- Two predictions after 0.3 s and 0.7 s



# Simulation results

- Simulation of the system in MATLAB and SIMULINK
- Two predictions after 0.3 s and 0.7 s



**Table:** Root mean square errors (RMSE) between reference signals  $q_{r,i}$  and system output  $q_i$  for the first three degrees-of-freedom.

	RMSE( $q_1$ )	RMSE( $q_2$ )	RMSE( $q_3$ )	Combined
PD	0.741	0.682	1.857	2.112
LPV	0.419	0.311	1.316	1.416

## Animation of the system

Video: animatedsystem.mp4

# Summary and Outlook

## Main results:

- Position tracking algorithm achieves fast and reliable position detections
- The trajectory of the ball can be accurately estimated
- A simple motion planning & inverse kinematics strategy was developed
- A LFT-LPV controller for the robot arm was synthesized and successfully tested

## Future work:

- Test of proposed approach in experiments
- Components will be extended for 3D measurements
- Active movement of the robot arm

# The End

**Thank you very much for your attention!**

# Camera Calibration

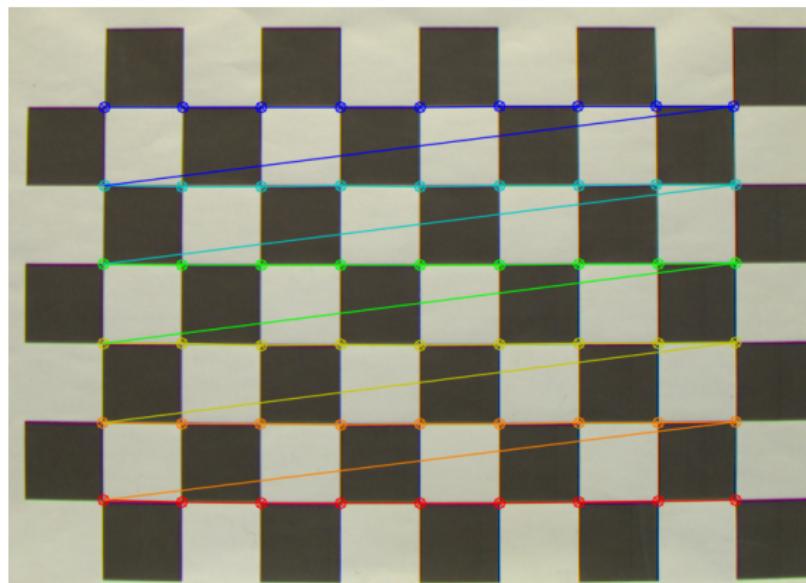


Figure: Corners of chessboard image detected by opencv's  
`findChessboardCorners()`.

# Kinematic limits of robot

Table: Kinematic limits of the joints.

<b>Angle</b>	<b>Range [°]</b>	<b>Maximum Speed</b>	<b>Value [<math>^{\circ}s^{-1}</math>]</b>
$q_1$	$[-175, \dots, 175]$	$\dot{q}_1$	180
$q_2$	$[-90, \dots, 90]$	$\dot{q}_2$	180
$q_3$	$[-110, \dots, 110]$	$\dot{q}_3$	180
$q_4$	$[-180, \dots, 180]$	$\dot{q}_4$	171

Table: Bounds on scheduling signals  $\rho_i(t)$  that define the compact set  $\mathcal{P}$ .

<b>Parameter</b>	<b>Range [°]</b>	<b>Parameter</b>	<b>Speed [<math>^{\circ}s^{-1}</math>]</b>
$\rho_1 = q_2$	$[-90, \dots, 90]$	$\rho_3 = \dot{q}_1$	$\pm 180$
$\rho_2 = \tilde{q}_3$	$[-190, \dots, 135]$	$\rho_4 = \dot{q}_2$	$\pm 180$
		$\rho_5 = \dot{\tilde{q}}_3$	$\pm 360$

# Affine LFT parameters

Shorthand notation for denominators:

$$v_1 = b_6\delta_1^2 + b_7\delta_2^2 + b_5 + b_3\delta_3\delta_4$$

$$v_2 = (b_3^2(\delta_1\delta_2 + \delta_3\delta_4)^2)/2 + 2b_{17}b_3(\delta_1\delta_2 + \delta_3\delta_4) - 2b_{13}b_{16} + 2b_{14}b_{17}$$

Parameters in  $A(\theta(t))$ :

$$\theta_1 = -(b_1 + b_2\delta_6\delta_1\delta_3 + b_3\delta_6\delta_1\delta_4 + b_3\delta_7\delta_2\delta_3 + b_4\delta_7\delta_2\delta_4)/v_1$$

$$\theta_2 = (2b_{16}b_8\delta_8)/v_2$$

$$\theta_3 = -(2b_9(b_{14} - b_{16}) + b_3b_9(\delta_1\delta_2 + \delta_3\delta_4))\delta_9/v_2$$

$$\theta_4 = -(2b_{12}\delta_5\delta_4\delta_2(2b_{14} - 2b_{16}) - (b_3^2\delta_5(\delta_1\delta_2 + \delta_3\delta_4)(\delta_3\delta_2 + \delta_1\delta_4))/4 + \dots$$

$$\dots + (b_{14}b_3\delta_5(\delta_4\delta_1 - \delta_2\delta_3))/2 + (b_3^2\delta_5(\delta_1\delta_2 + \delta_3\delta_4)(\delta_4\delta_1 - \delta_2\delta_3))/4 + \dots$$

$$\dots + b_3\delta_5(\delta_3\delta_2 + \delta_1\delta_4)(b_{14}/2 - b_{17}) - 2b_{11}b_{16}\delta_5(2\delta_3\delta_1) + \dots$$

$$\dots + b_{12}b_3\delta_5(\delta_1\delta_2 + \delta_3\delta_4)(2\delta_4\delta_2)/v_2$$

$$\theta_5 = (2b_{10}b_{16} + 2b_{14}b_{15} + b_{15}b_3(\delta_1\delta_2 + \delta_3\delta_4) - (b_3^2\delta_6(\delta_1\delta_2 + \delta_3\delta_4)(\delta_4\delta_1 - \delta_2\delta_3))/2 - \dots$$

$$\dots - b_3\delta_6(\delta_4\delta_1 - \delta_2\delta_3)(b_{14} - b_{16}))/v_2$$

$$\theta_6 = -(2b_{14}b_{15} + b_{15}b_3(\delta_1\delta_2 + \delta_3\delta_4) + b_{16}b_3\delta_7(\delta_4\delta_1 - \delta_2\delta_3))/v_2$$

$$\theta_7 = -(2b_{17}b_8 + b_3b_8(\delta_1\delta_2 + \delta_3\delta_4))\delta_8/v_2$$

$$\theta_8 = (b_9(2b_{13} - 2b_{17}))\delta_9/v_2$$

# Affine LFT parameters

$$\begin{aligned}\theta_9 = & -((b_3\delta_5(\delta_3\delta_2 + \delta_1\delta_4)(b_{13} - 2b_{17}))/2 - b_{12}\delta_5(2\delta_4\delta_2)(2b_{13} - 2b_{17}) - \dots \\ & \dots - (b_3^2\delta_5(\delta_1\delta_2 + \delta_3\delta_4)(\delta_3\delta_2 + \delta_1\delta_4))/4 + (b_{13}b_3\delta_5(\delta_3\delta_2 - \delta_1\delta_4))/2 + \dots \\ & \dots + (b_3^2\delta_5(\delta_1\delta_2 + \delta_3\delta_4)(\delta_3\delta_2 - \delta_1\delta_4))/4 + 2b_{11}b_{17}\delta_5(2\delta_3\delta_1) + \dots \\ & \dots + b_{11}b_3\delta_5(\delta_1\delta_2 + \delta_3\delta_4)(2\delta_3\delta_1))/v_2\end{aligned}$$

$$\begin{aligned}\theta_{10} = & -(2b_{10}b_{17} + 2b_{13}b_{15} + b_3(\delta_1\delta_2 + \delta_3\delta_4)(b_{10} + b_{15}) + b_3\delta_6(\delta_3\delta_2 - \delta_1\delta_4)(b_{13} - b_{17}))/v_2 \\ \theta_{11} = & (2b_{13}b_{15} + b_{15}b_3(\delta_1\delta_2 + \delta_3\delta_4) - b_{17}b_3\delta_7(\delta_3\delta_2 - \delta_1\delta_4) - \dots \\ & \dots - (b_3^2\delta_7(\delta_1\delta_2 + \delta_3\delta_4)(\delta_3\delta_2 - \delta_1\delta_4))/2)/v_2\end{aligned}$$

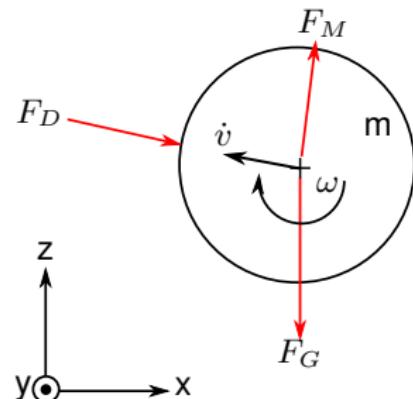
Parameters in  $B$  ( $\theta(t)$ ):

$$\begin{array}{ll}\theta_{12} = 1/v_1 & \theta_{13} = -(2b_{16})/v_2 \\ \theta_{14} = (2b_{14} + b_3(\delta_1\delta_2 + \delta_3\delta_4))/v_2 & \theta_{15} = (2b_{17} + b_3(\delta_1\delta_2 + \delta_3\delta_4))/v_2 \\ \theta_{16} = -(2b_{13} + b_3(\delta_1\delta_2 + \delta_3\delta_4))/v_2 & \end{array}$$

# Trajectory Prediction - State Space model

- Goal: Predict flight trajectory of the ball
- ① Model dynamics of the ball.
  - ② Determine initial conditions from set of measured positions.
  - ③ Numerically integrate equations of motion.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -K_m ||v|| v_x \\ -K_m ||v|| v_y \\ -K_m ||v|| v_z - g \end{bmatrix}$$



# Trajectory Prediction - Forward Euler Method

- Determine initial conditions  $x_0, z_0, v_{x,0}, v_{z,0}$
- Polynomial fitting on the last 10 data points  $(X, Y, Z, T)$

$$x = p_1 t^2 + p_2 t + p_3$$

$$y = q_1 t^2 + q_2 t + q_3$$

$$z = w_1 t^2 + w_2 t + w_3$$

- These relationships can then be used to compute the initial velocities at the corresponding time  $t_p$ :

$$v_x(t_p) = \dot{x}(t_p) = 2p_1 t_p + p_2$$

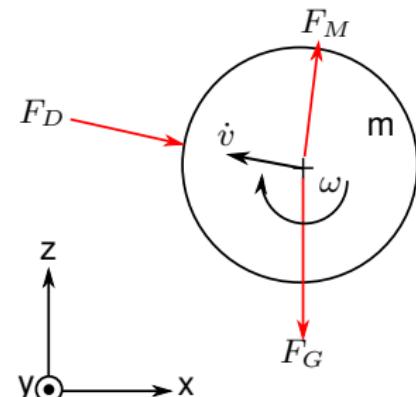
$$v_y(t_p) = \dot{y}(t_p) = 2q_1 t_p + q_2$$

$$v_z(t_p) = \dot{z}(t_p) = 2w_1 t_p + w_2.$$

# Trajectory Prediction - Forward Euler Method

- Goal: Predict flight trajectory of the ball
- ① Model dynamics of the ball.
  - ② Determine initial conditions from set of measured positions.
  - ③ Numerically integrate equations of motion.

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \\ v_x(k+1) \\ v_y(k+1) \\ v_z(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix} + T_s \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_z(k) \\ -K_m ||v(k)|| v_x(k) \\ -K_m ||v(k)|| v_y(k) \\ -K_m ||v(k)|| v_z(k) - g \end{bmatrix}$$



# Ball model - Typical values

Table: Typical parameters for the ball model.

Parameter	Value
$g$	$9.81 \text{ m/s}^2$
$m$	$0.0027 \text{ kg}$
$\rho$	$1.29 \cdot 10^3 \text{ kg/m}^3$
$C_D$	$0.4 \sim 0.5$
$r_b$	$0.02 \text{ m}$

## Reference generation - Sigmoids

- Sigmoids are used to generate reference signal
- Ensures smooth signal and just in time arrival of robot arm

$$r_i(t) = \frac{q_i}{2} + \frac{q_i}{2} \tanh \left( \left( t - t_{st} + \frac{t_{end} - t_{st}}{2} \right) \frac{4}{t_{end} - t_{st}} \right). \quad (1)$$

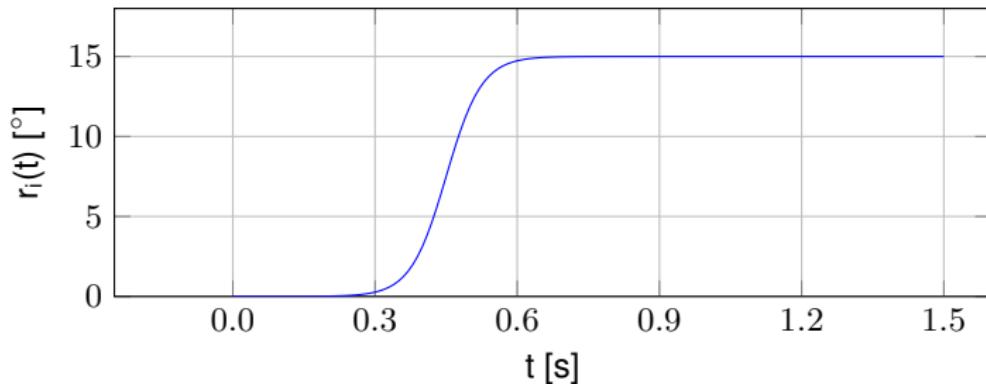


Figure: Reference signal  $r_i(t)$  as sigmoid with  $q_i = 15^\circ$ ,  $t_{st} = 0.3\text{ s}$  and  $t_f = 0.6\text{ s}$ .

# PD Controller

- PD controller proposed by manufacturer

$$K_{P,i} = 2.5, \quad K_{D,i} = 0.05 \quad \text{for: } i = 1, \dots, 4,$$

$$K_P = \underset{i=1}{\overset{4}{\text{diag}}}(K_{P,i}), \quad K_D = \underset{i=1}{\overset{4}{\text{diag}}}(K_{D,i}).$$

These gains are then used to compute the control input  $u(t)$  from the error signal  $e(t)$ :

$$u(t) = K_P e(t) + K_D \frac{de(t)}{dt}.$$

## Parameter Set Mappings based on PCA (1)

- For affine LPV systems a parameter set mapping based on PCA can be used to reduce the number of scheduling signals
- ① Generate a data matrix  $\Xi = [\theta(\rho^1), \dots, \theta(\rho^m)]$  from scheduling signal vectors  $\rho^k$ ,  $k = 1, \dots, m$ .
  - ② Normalize the data matrix row-wise to achieve scaled, zero mean data:

$$\Xi^n = \mathcal{N}(\Xi). \quad (2)$$

- ③ Perform a singular value decomposition on the normalized data matrix:

$$\Xi^n = [U_s U_n] \begin{bmatrix} \Sigma_s & 0 & 0 \\ 0 & \Sigma_n & 0 \end{bmatrix} \begin{bmatrix} V_s^T \\ V_n^T \end{bmatrix}. \quad (3)$$

## Parameter Set Mappings based on PCA (2)

The first 10 singular values are shown in Fig. 5.

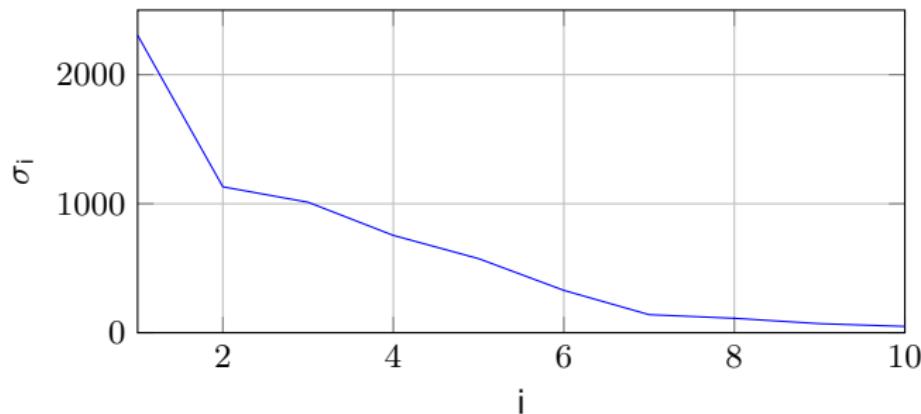


Figure: First ten singular values  $\sigma_i$  of the normalized data matrix  $\Xi^n$  ordered with decreasing significance.

- ④ First three singular values lead to an accuracy of 87.9 % (4: 94.47 %).

$$\phi(t) = U_s^T \mathcal{N}(\theta(t)). \quad (4)$$

## Parameter Set Mappings based on PCA (3)

- ⑤ The system matrices  $\hat{A}(\phi), \hat{B}(\phi), \hat{C}(\phi), \hat{D}(\phi)$  for the reduced system  $\hat{P}(\phi)$  can be computed by:

$$\hat{P}(\phi) = \begin{bmatrix} \hat{A}(\phi) & \hat{B}(\phi) \\ \hat{C}(\phi) & \hat{D}(\phi) \end{bmatrix} = \begin{bmatrix} \hat{A}(\hat{\theta}) & \hat{B}(\hat{\theta}) \\ \hat{C}(\hat{\theta}) & \hat{D}(\hat{\theta}) \end{bmatrix}, \quad (5)$$

where

$$\hat{\theta}(t) = \mathcal{N}^{-1}(U_s \phi(t)) = \mathcal{N}^{-1}(U_s U_s^T \mathcal{N}(\theta(t))). \quad (6)$$

# Sensitivity Shaping Filters

$$W_{S,i} = \frac{\omega_{S,i}/M_{S,i}}{s + \omega_{S,i}}, \quad W_{KS,i} = \frac{c_{KS,i}/M_{KS,i}(s + \omega_{KS,i})}{s + c_{KS,i}\omega_{KS,i}}, \quad W_{R,i} = \frac{M_{r,i}}{s + \omega_{r,i}},$$

$$\mathcal{W}_S = \underset{i=1}{\overset{3}{\text{diag}}}(W_{S,i}), \quad \mathcal{W}_{KS} = \underset{i=1}{\overset{3}{\text{diag}}}(W_{KS,i}), \quad \mathcal{W}_R = \underset{i=1}{\overset{3}{\text{diag}}}(W_{R,i}).$$

The filter parameters gain  $M$ , crossover-frequency  $\omega$  and constant  $c$  are listed int the following Table.

Table: Shaping filter parameters.

<b>Channel i</b>	$\omega_{S,i}$	$M_{S,i}$	$\omega_{KS,i}$	$M_{KS,i}$	$c_{KS,i}$	$\omega_{R,i}$	$M_{R,i}$
1	0.09	0.0001	100	1000	1000	20	20
2	0.09	0.0001	100	500	1000	20	20
3	0.1	0.0001	100	500	1000	20	20

# Linear Fractional Transformation

Consider two complex matrices  $A$  and  $B$ :

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbb{C}^{(l_1+l_2) \times (m_1+m_2)}, \quad B \in \mathbb{C}^{l_1 \times m_1}.$$

Then upper and lower LFT of  $A$  with respect to  $B$  are defined as:

$$\mathcal{F}_u(A, B) = A_{22} + A_{21}B(I - A_{11}B)^{-1} + A_{12} \quad (7)$$

$$\mathcal{F}_l(A, B) = A_{11} + A_{12}B(I - A_{22}B)^{-1} + A_{21}. \quad (8)$$

fig:block, where A has the two inputs  $u$  and  $w$  and the two outputs  $z$  and  $v$ .

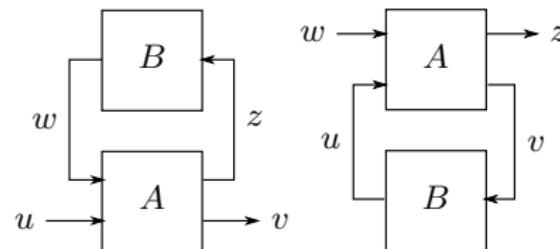


Figure: Upper and lower LFT of a system A with respect to a system B.

# Augmented system with scalings

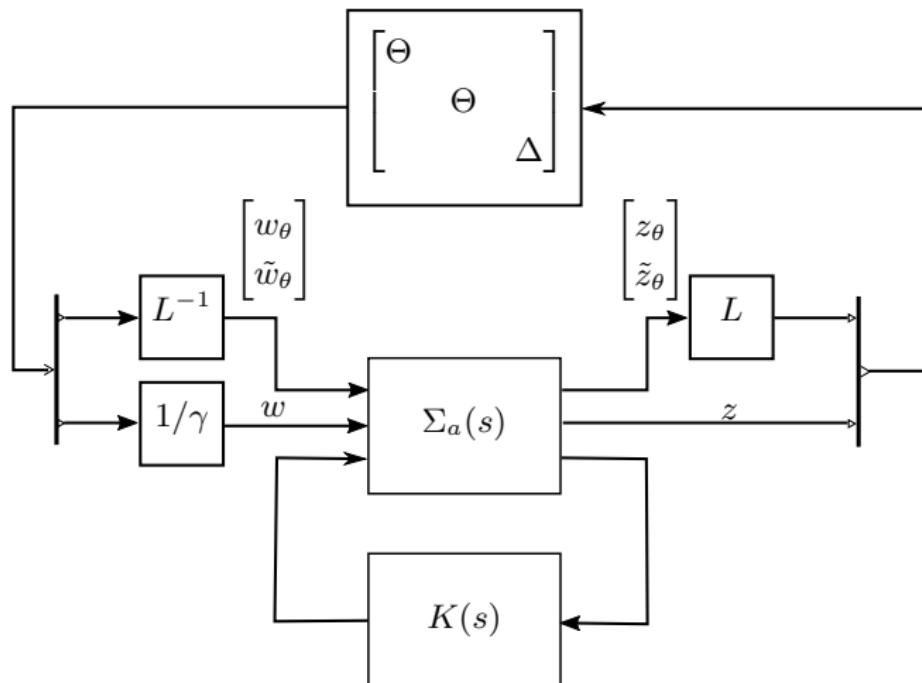


Figure: Augmented system with scaling blocks  $L$ ,  $L^{-1}$ ,  $1/\gamma$  to formulate the robust performance problem.

## Simulations based on these predictions

First prediction:

$$q_{r,0.3} = \begin{bmatrix} q_{r,1} & q_{r,2} & q_{r,3} & q_{r,4} \end{bmatrix}^T = \begin{bmatrix} -90^\circ & 12.34^\circ & -65.81^\circ & 74.28^\circ \end{bmatrix}^T$$

Second prediction:

$$q_{r,0.7} = \begin{bmatrix} -90^\circ & 19.68^\circ & -86.23^\circ & 73.43^\circ \end{bmatrix}^T \quad (\text{second prediction}).$$

# Further Reading

- Pierre Apkarian. A convex characterization of gain-scheduled  $H\infty$  controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, 1995.
- Sayedali Shetab Boushehri. Development of a linearizing state-feedback for a robotic manipulator. Master's thesis, University of Hamburg, 2015.
- Duane C Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.
- Seyed Mahdi Hashemi, Hossam Seddik Abbas, and Herbert Werner. Low-complexity linear parameter-varying modeling and control of a robotic manipulator. *Control Engineering Practice*, 20(3):248–257, 2012.
- Seyed Mahdi Hashemi, Utku Gürcüoğlu, and Herbert Werner. Interaction control of an industrial manipulator using LPV techniques. *Mechatronics*, 23(6):689–699, 2013.
- Christian Hoffmann, Seyed Mahdi Hashemi, Hossam Seddik Abbas, and Herbert Werner. Benchmark problem—nonlinear control of a 3-DOF robotic manipulator. In *IEEE 52nd Annual Conference on Decision and Control (CDC)*, 2013, pages 5534–5539. IEEE, 2013.
- Carsten W Scherer. Lpv control and full block multipliers. *Automatica*, 37(3):361–375, 2001.