RUB

# Smart Pump Control Design on Mobile Devices

Flávio Fabrício Ventura de Melo Ferreira

Automatic Control and Systems Theory
Ruhr-Universität Bochum
Germany

21/11/2013, Bochum - Germany

# IAESTE

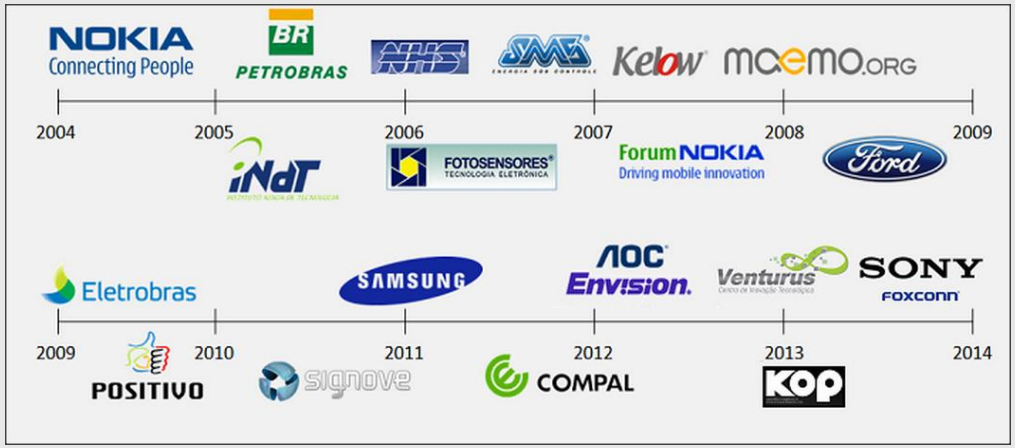
- International Association for the Exchange of Students for Technical Experience
- **Programme Aims:**
  - To provide undergraduates in Science, Engineering and Architecture with course related training overseas
  - To provide employers with highly skilled, highly motivated trainees from around the world
  - To be a source of cultural enrichment for trainees and their host communities


# Maps

# UFCG

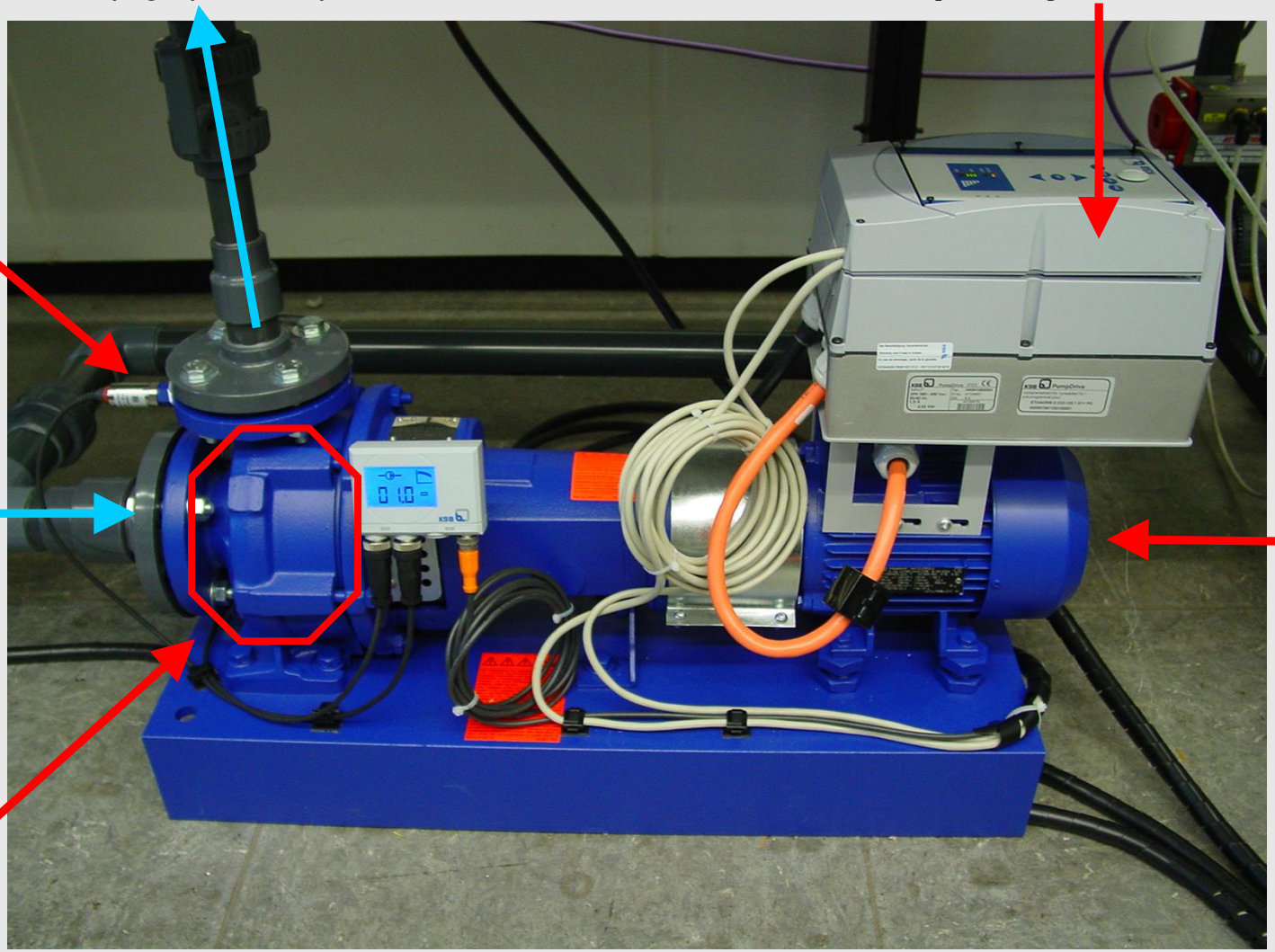
- UFCG
  - Federal University of Campina Grande
  - Public University
  - Main campus is located in the city of Campina Grande , Paraiba , Brazil
  - 10.000 Students
- Campina Grande
  - Greater proportional concentration of doctors in the country
  - "9 New Tech Cities" (*Newsweek*, 2001)
  - "Brazilian Silicon Valley" (*Newsweek*, 2003)

# Electrical Engineering at UFCG

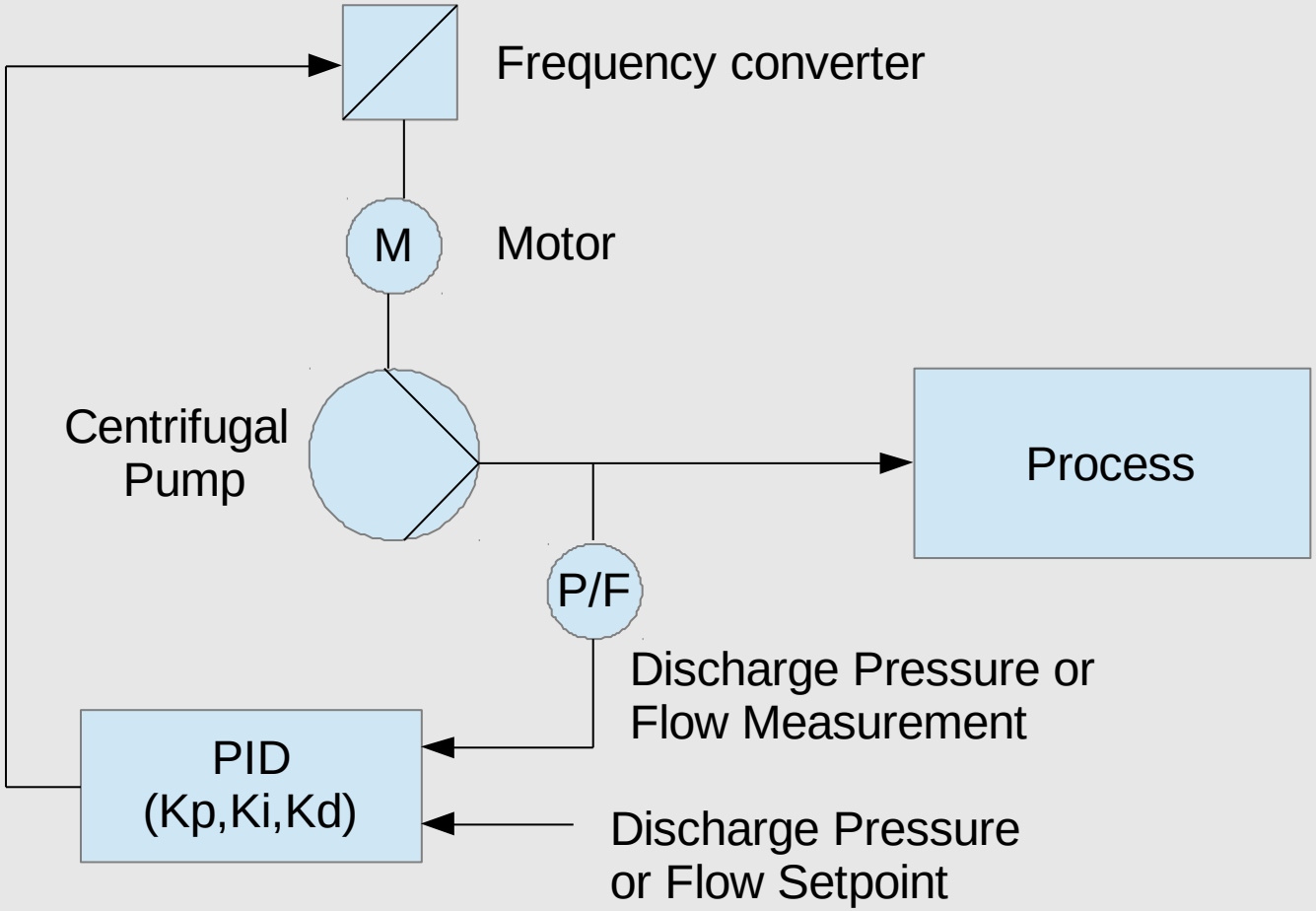




- Among the top 8 in the country
- Embedded Systems and Pervasive Computing Lab
- Part of the Center of Electrical Engineering and Informatics (CEEI)

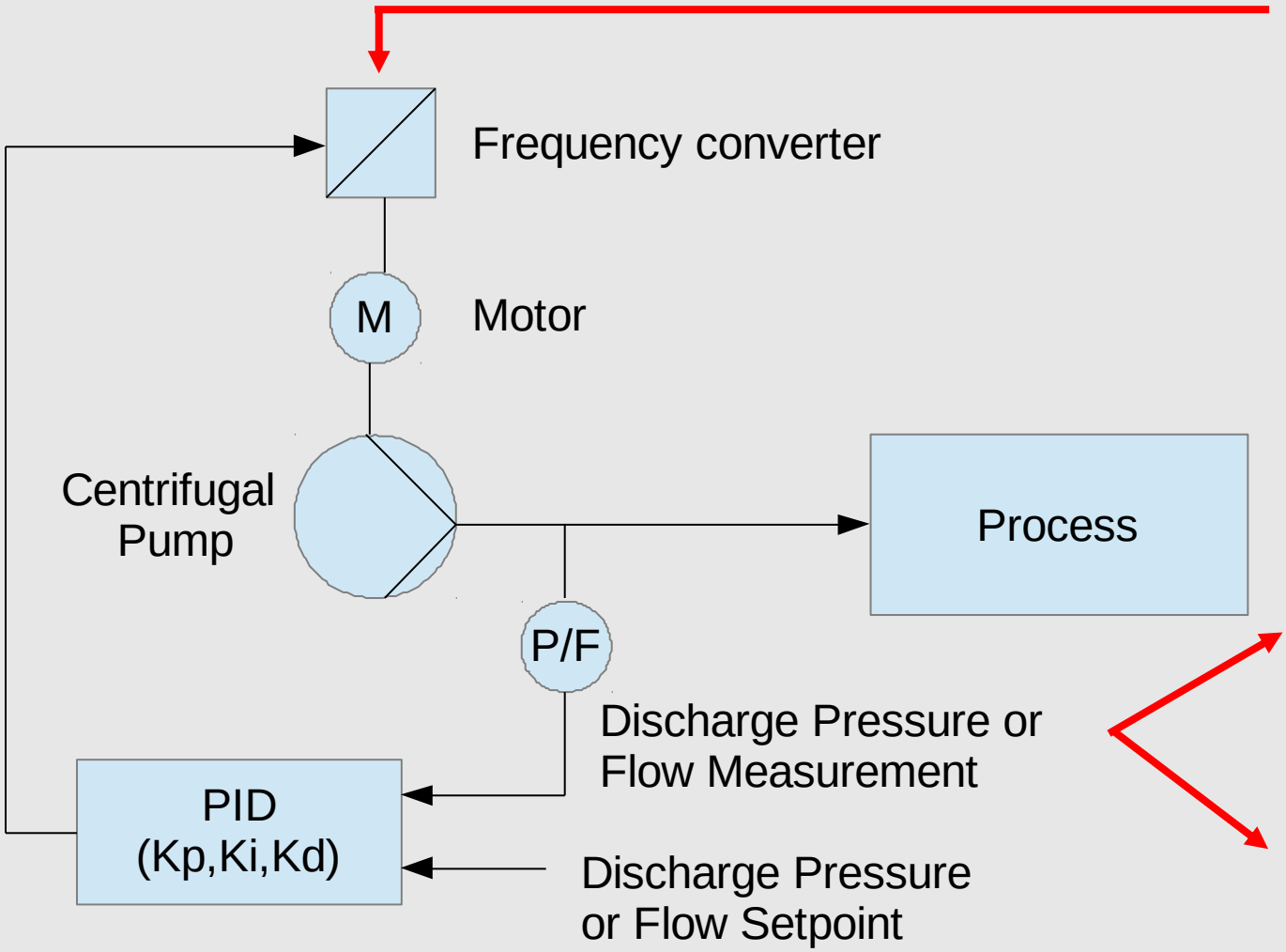
# The System

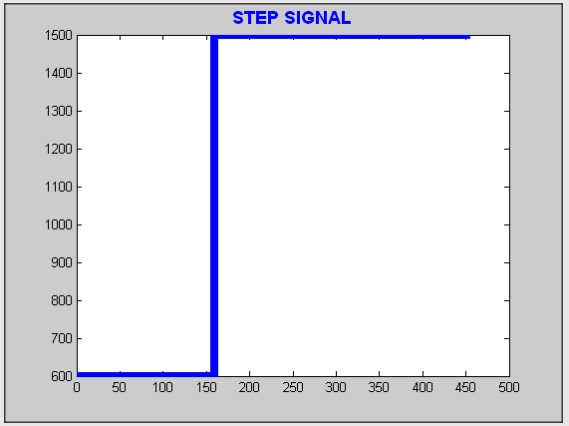

Pump Outlet
(high pressure)
Frequency Converter
Measurement
Discharge
Pressure
Pump Inlet
(low pressure )
Motor
Centrifugal
Pump

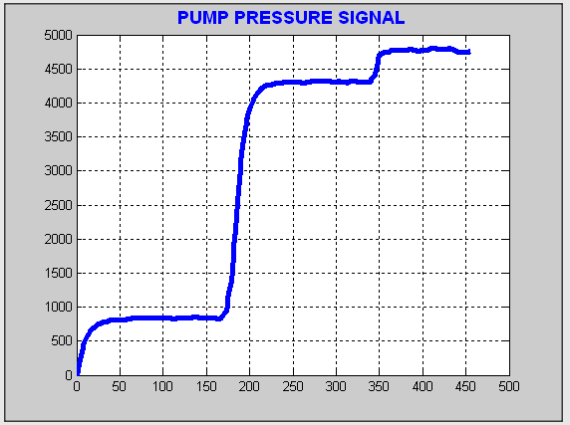
# The System

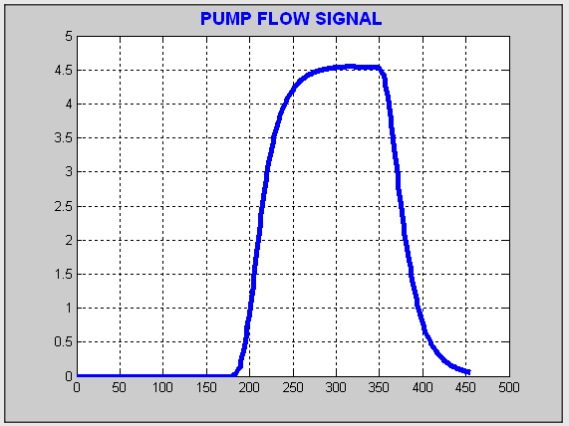

Frequency converter
M
Motor
Centrifugal
Pump
Process
P/F
Discharge Pressure or
Flow Measurement
PID
(Kp,Ki,Kd)
Discharge Pressure
or Flow Setpoint

# The System


Frequency converter
M
Motor
Centrifugal
Pump
Process
P/F
Discharge Pressure or
Flow Measurement
PID
(Kp,Ki,Kd)
Discharge Pressure
or Flow Setpoint

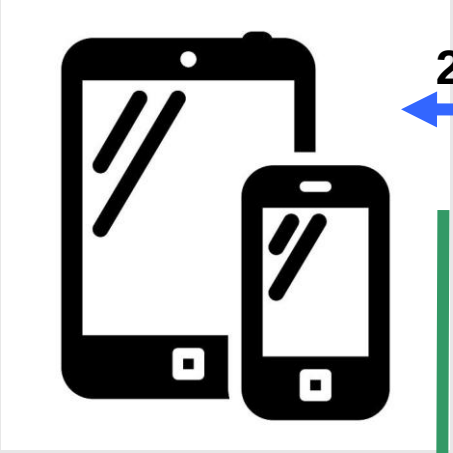

STEP SIGNAL


PUMP PRESSURE SIGNAL

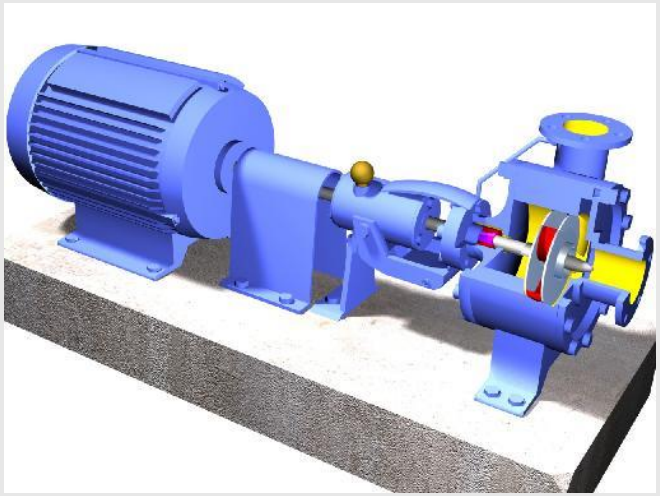

PUMP FLOW SIGNAL

# The Aim

**1) Request to Perform the SpeedStep**



**2) Returns the Input and Output Signals**



**3) System Identification**
**(Transfer Function with Kp, Tt, T1, order)**

**4) Controller Synthesis**
**(Show the Stable Region in function of Kp, Ki and Kd )**

**5) Send calculated parameters Kp, Ki and Kd**

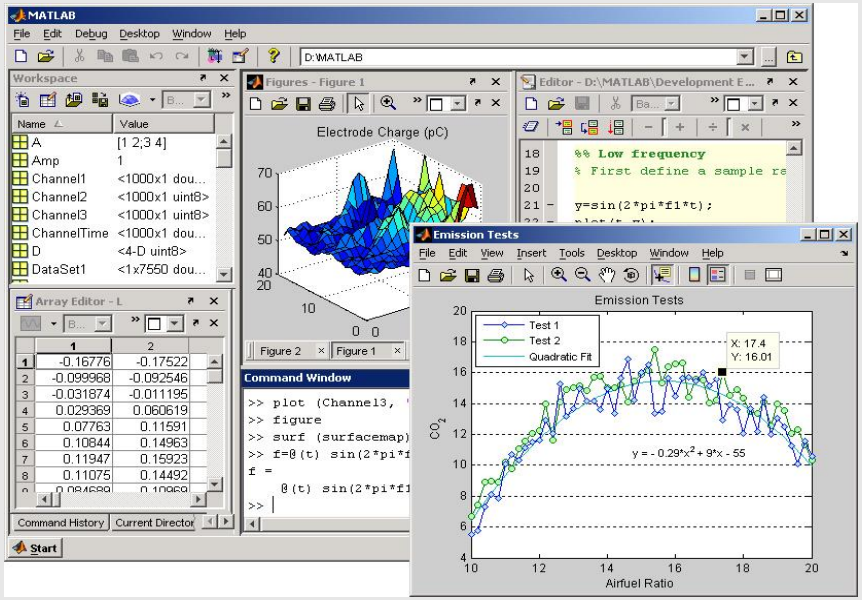
# Tasks

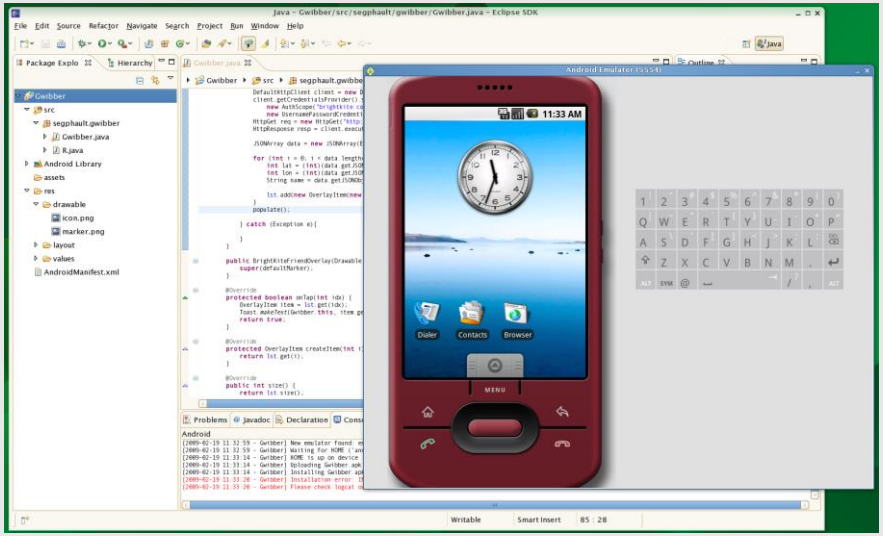
- TASK1 - Mobile device request to pump to perform the speedStep
- TASK2 - Pump send the input and output signals to the mobile device
- **TASK3 - System identification**
- **TASK4 - Controller synthesis**
- TASK5 - Mobile device sends to the pump the calculated parameters

# Tasks

- **TASK3 - System Identification**
  - Import ScopeData  
  - Calculate Gain K  
  - Calculate T  
  - Calculate Dead Time  
  - Calculate Order  
- **TASK4 - Controller Synthesis**
  - Nyquist Decoposition  
  - D Composition  
  - Calculate Singular Frequencies  
  - Calculate Stabilizable KP Interval 
  - Calculate Stable Region 

# Requirements



- Matlab
- Android SDK
  - Eclipse + ADT (Android Developer Tools) plugin
  - Android SDK Tools
  - Android Platform-tools
  - The latest Android platform
  - The latest Android system image for the emulator

# Android x Matlab



MATLAB R2012a

File Edit Text Go Cell Tools Debug Desktop Window Help

Current Folder: C:\Dokumente und Einstellungen\Flavio\Eigene Dateien\MATLAB

Shortcuts How to Add What's New

Editor - C:\Dokumente und Einstellungen\Flavio\Eigene Dateien\MATLAB\automatic_controller_NOPDT.m*

```
13
14      % perform nyquist-decomposition
15 -    [De,Do,Ne,No,X,Y,Z,n,m]=nyquist_decomposition(D,N);
16
17      % perform d-composition
18 -    [Ra,Rb,Ia,Ib,f1,f2,fn,n,m,l]=d_composition(D,N);
19
20      % calc singular frequencies for KP=0
21 -    [omega0 omegaplus omegaminus]=calc_singular_frequencies_delay(f1,f2,fn,0,L,D,N,l,0.1,1);
22
```

Java - MoPumpCo/src/screen/MainActivity.java - ADT

File Edit Refactor Source Navigate Search Project Run Window Help

CalcSingula... InputSelect... MoPumpCo Man... InputSelect... CalcSingula... *MainActivit... SplashScreen... OtherFunctio...

```
        List<Double> numeratorN = Gs.getTfData().get(0);

        List<Double> denominatorD = Gs.getTfData().get(1);

//      perform nyquist-decomposition
        DecompositionNyquist DN = new DecompositionNyquist(denominatorD, numeratorN);

//      perform d-composition
        CompositionD D = new CompositionD(denominatorD, numeratorN);

        int L= deadTime;

//      calc singular frequencies
        CalcSingularFrequenciesDelay CSF = new CalcSingularFrequenciesDelay(D.getF1(),D.getF2(),
                                                                            D.getFn(),0, L,
                                                                            denominatorD, numeratorN,
                                                                            D.getL(), 0.1, true);
```

# Android x Matlab



MATLAB R2012a

File Edit Text Go Cell Tools Debug Desktop Window Help

Current Folder: C:\Dokumente und Einstellungen\Flavio\Eigene Dateien\MATLAB

Shortcuts How to Add What's New

Editor - C:\Dokumente und Einstellungen\Flavio\Eigene Dateien\MATLAB\original_code\d_composition.m

Stack: Base

```
39
40      %% KP generator function
41 -    f1=([0 -conv(Ra,Rb)-conv(Ia,Ib)]);
42 -    f2=([0 conv(Ia,Rb)-conv(Ra,Ib)]);
43 -    fn=[conv(Ra,Ra)+conv(Ia,Ia) 0];
44
45
```

Java - MoPumpCo/src/controlFunctions/CompositionD.java - ADT

File Edit Refactor Source Navigate Search Project Run Window Help

CompositionD.java

```
100 //      KP generator function
101         List<Double> convRaRb = OtherFunctions.convolution(Ra,Rb);
102         List<Double> convIaIb = OtherFunctions.convolution(Ia,Ib);
103         List<Double> convIaRb = OtherFunctions.convolution(Ia,Rb);
104         List<Double> convRaIb = OtherFunctions.convolution(Ra,Ib);
105         List<Double> convRaRa = OtherFunctions.convolution(Ra,Ra);
106         List<Double> convIaIa = OtherFunctions.convolution(Ia,Ia);
107
108         List<Double> SumConvRaRbConvIaIb = OtherFunctions.SumOfVectors(convRaRb,convIaIb);
109         SumConvRaRbConvIaIb = OtherFunctions.MultOfVectorByOneNumber(SumConvRaRbConvIaIb,-1);
110         setF1(OtherFunctions.shiftRightVector(SumConvRaRbConvIaIb, 1));
111
112         List<Double> SubConvIaRbConvRaIb = OtherFunctions.SubOfVectors(convIaRb,convRaIb);
113         setF2(OtherFunctions.shiftRightVector(SubConvIaRbConvRaIb, 1));
114
115         List<Double> SumConvRaRaConvIaIa = OtherFunctions.SumOfVectors(convRaRa,convIaIa);
116         setFn(OtherFunctions.shiftLeftVector(SumConvRaRaConvIaIa, 1));
```

# Matlab Native Methods



Java - MoPumpCo/src/controlFunctions/OtherFunctions.java - ADT

File Edit Refactor Source Navigate Search Project Run Window Help

CompositionD.java | OtherFunctions.java

```
271    public static List<Double> convolution(List<Double> inputX,List<Double> inputH) {
272
273        //OBJECTS ARE PASSED BY REFERENCE AND PRIMITIVE VALUES BY VALUE
274        // Then in this case when i to flip the inputH or the InputX ,
275        // i am flipping the input values and not a copy.
276        // The solution is to transform  the two inputs in a copy of the input
277
278        List<Double> inputXConv = zeros(inputX.size());
279        for(int iX=0;iX<inputXConv.size();iX++){
280            inputXConv.set(iX, inputX.get(iX));
281        }
282
283        List<Double> inputHConv = zeros(inputH.s
284        for(int iH=0;iH<inputHConv.size();iH++){
285            inputHConv.set(iH, inputH.get(iH));
286        }
287
288        int lenghtL1X = inputX.size();
289        int lenghtL2H = inputH.size();
290        int lenghtL3SUM = lenghtL1X + lenghtL2H
291        int lenghtL4SUB = lenghtL2H -1;
292        int lenghtL5 = 0,p1=0;
293        double c2 = 0,sum = 0;
294
295        List<Double> inputHFliped = flipLeftRigh
296        List<Double> convolutionOutput = zeros(l
297        List<Double> c1;
```

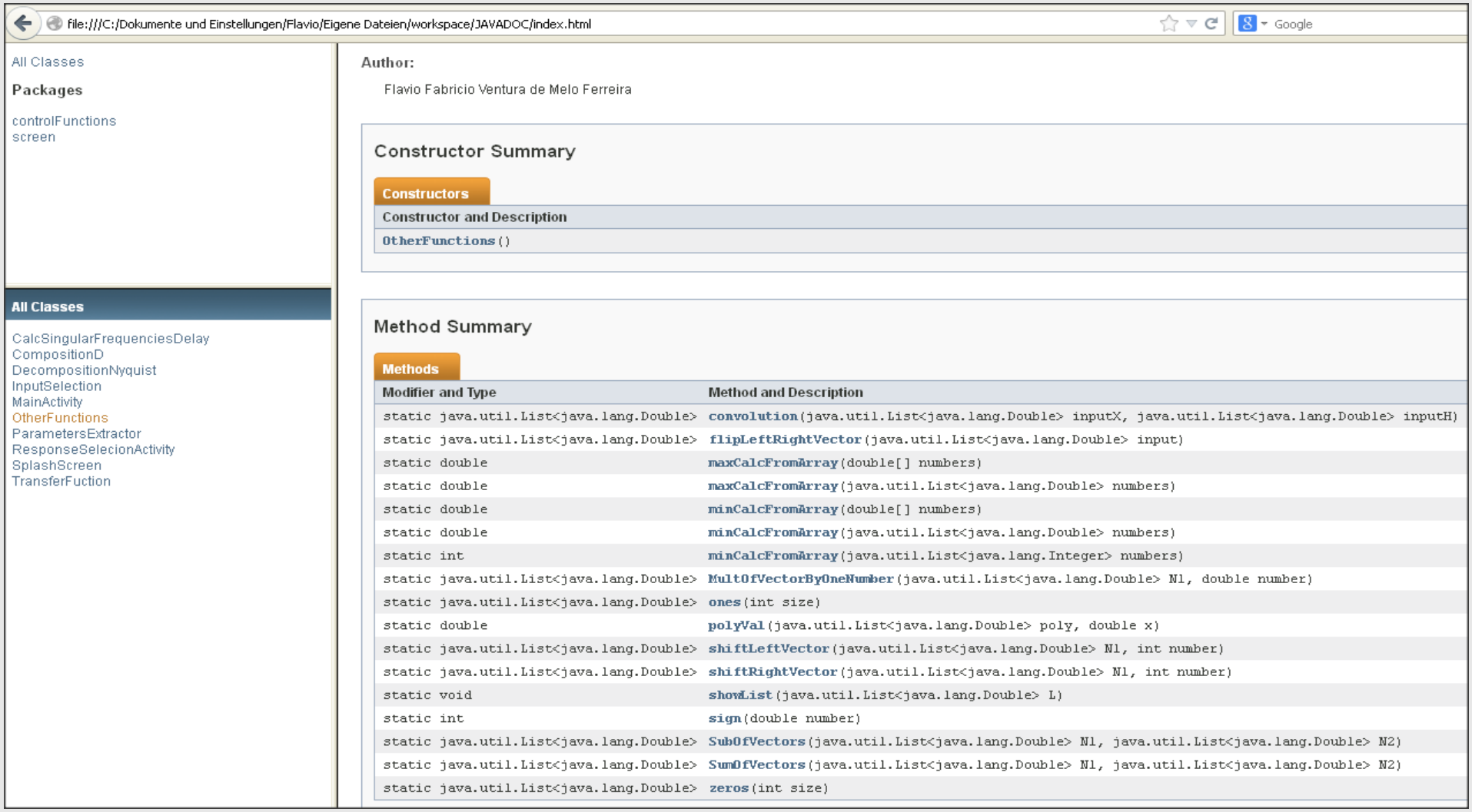
Java - MoPumpCo/src/controlFunctions/OtherFunctions.java - ADT

File Edit Refactor Source Navigate Search Project Run Window Help

CompositionD.java | *OtherFunctions.java

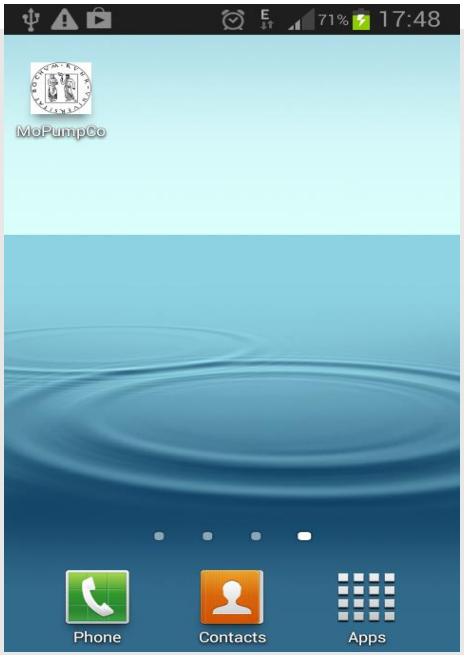
```
86     /** Fill with Zeros the array with length size
87      * @param size is the lenght of the array
88      * @return list is the array of zeros with length size
89      */
90     public static List<Double> zeros(int size) {
91         List<Double> list = new ArrayList<Double>();
92         for(int i=0;i<size;i++){
93             list.add(0.0);
94         }
95         return list;
96     }
97
98     /** Calculate the maximum value of the array
99      * @param numbers is the array thats we will search the maximum value
100     * @return max is the maximum value found at array
101     */
102    public static double maxCalcFromArray(double[] numbers) {
103
104        double max = numbers[0];
105        for(int i=0; i < numbers.length ; i++){
106            if(numbers[i]>max)
107            {
108                max=numbers[i];
109            }
110        }
111        return max;
112    }
```

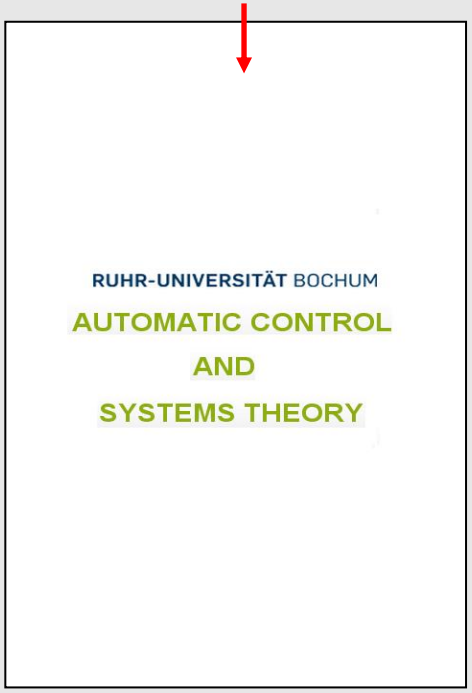
# Documentation

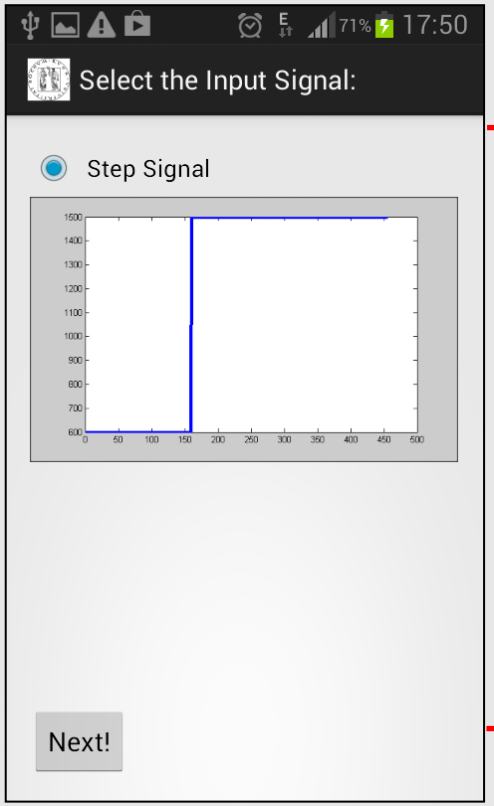


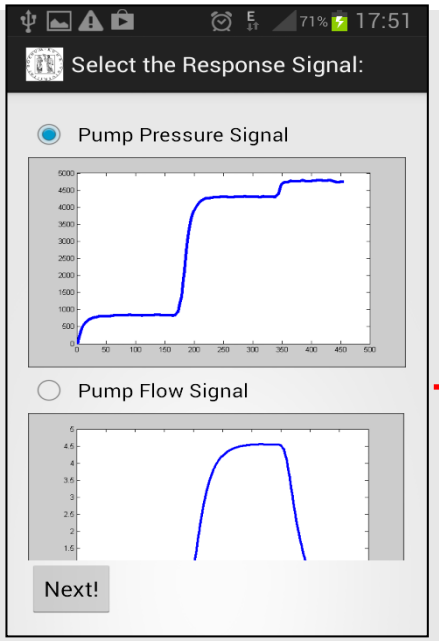
# Interface

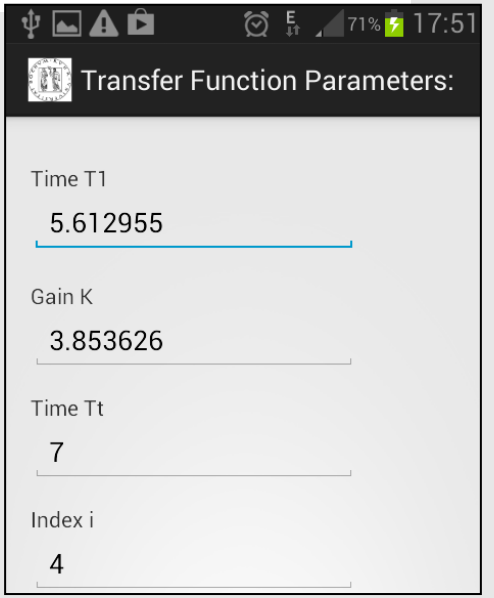


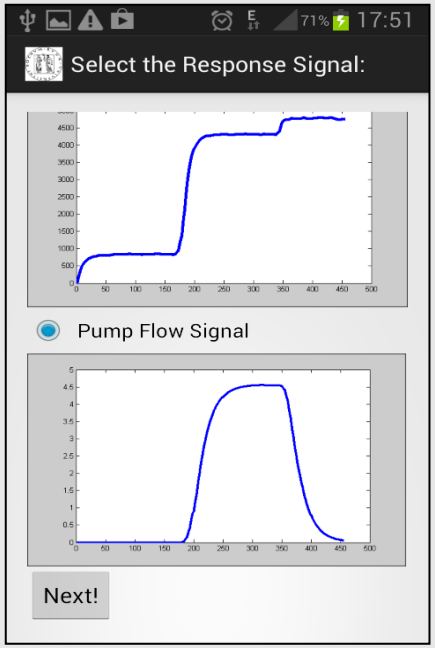

17:48
MoPumpCo
Phone
Contacts
Apps

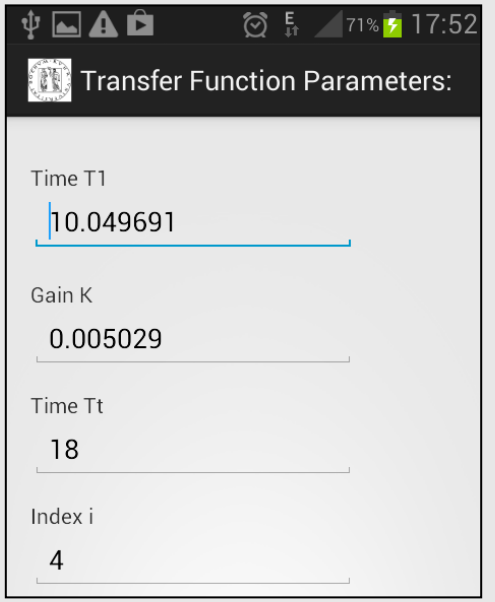

RUHR-UNIVERSITÄT BOCHUM
AUTOMATIC CONTROL
AND
SYSTEMS THEORY


17:50
Select the Input Signal:
Step Signal
Next!


17:51
Select the Response Signal:
Pump Pressure Signal
Pump Flow Signal
Next!

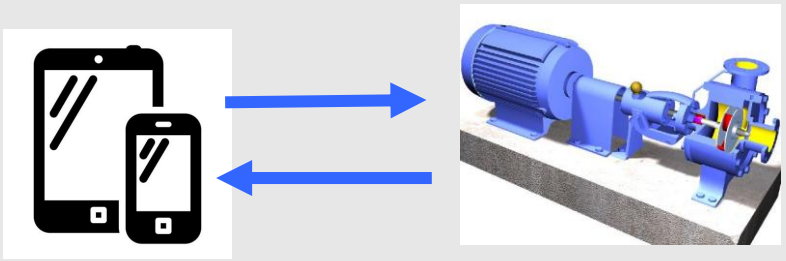

17:51
Transfer Function Parameters:
Time T1
5.612955
Gain K
3.853626
Time Tt
7
Index i
4


17:51
Select the Response Signal:
Pump Flow Signal
Next!


17:52
Transfer Function Parameters:
Time T1
10.049691
Gain K
0.005029
Time Tt
18
Index i
4

# Future Tasks

- Full documentation
- Clear the code
- Unit test
- Complete the task 4
- Connection interface between mobile device and the pump

# To Finish my Engineering Course

- Digital Control
- Digital Control Laboratory
- Electrical Machines
- Electrical Machines Laboratory
- Electrical Energy Generation
- Internship Thesis
- **Bachelor Thesis**

# Vielen Dank!!

# Everyone is Invited