



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

FLÁVIO FABRÍCIO VENTURA DE MELO FERREIRA

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Campina Grande, Paraíba
Julho de 2014

FLÁVIO FABRÍCIO VENTURA DE MELO FERREIRA

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

*Relatório referente ao **Estágio Supervisionado**
submetido à Unidade Acadêmica de Engenharia
Elétrica da Universidade Federal de Campina
Grande como parte dos requisitos necessários
para a obtenção do grau de Bacharel em
Ciências no Domínio da Engenharia Elétrica.*

Área de Concentração: Automação e Controle

Orientador:

Professor George Acioli Júnior, Dr.

Campina Grande, Paraíba
Julho de 2014

FLÁVIO FABRÍCIO VENTURA DE MELO FERREIRA

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Relatório referente ao **Estágio Supervisionado** submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Automação e Controle

Aprovado em / /

Professora Maria de Fátima Queiroz Vieira, PhD
Universidade Federal de Campina Grande
Avaliador

Professor George Acioli Júnior, Dr.
Universidade Federal de Campina Grande
Orientador, UFCG

*“Uma mente,
uma vez
expandida
por uma
nova idéia,
nunca
retornará
a sua dimensão
original.”*

Oliver Wendell Holmes.

RESUMO

Este relatório descreve o trabalho efetuado no decorrer do meu estágio no laboratório *Regelungstechnik und Systemtheorie (RUS)* que faz parte do Departamento de Engenharia Mecânica da universidade *Ruhr-Universität* na cidade de Bochum - Alemanha. O estágio foi realizado entre setembro e novembro de 2013 (320 horas).

O tema do estágio consiste no desenvolvimento de software para controle inteligente de bombas centrífugas a partir de dispositivos móveis.

O algoritmo, antes desenvolvido na linguagem de programação MATLAB para sistemas operacionais desktop, foi então desenvolvido na linguagem JAVA para dispositivos móveis com sistema operacional Android.

Uma apresentação sucinta do laboratório RUS e da universidade *Ruhr-Universität de Bochum* bem como os objetivos globais do estágio serão propostos. O desenvolvimento do software que será apresentado é dividido em duas partes que são a identificação do sistema e a síntese do controle. A linguagem de programação e ambiente de desenvolvimento MATLAB bem como a linguagem e ambiente Android serão introduzidas também.

Por fim, este documento apresenta uma parte do protótipo de um software que faz a identificação e exibe a região de estabilidade dos parâmetros de controle de uma bomba centrífuga.

Palavras-chave: Região de Estabilidade, Controle Inteligente, Dispositivo Móvel, MATLAB, Android, Bomba Centrífuga, PID.

ABSTRACT

This report summarizes the work done during my internship at the laboratory of Automatic Control and Systems Theory (RUS) of the Mechanical Engineering Department at Ruhr-Universität of Bochum – Germany . The internship was between September and November 2013 (320 hours).

The subject of the internship involves the software development to smart control of centrifugal pumps on mobile devices.

The algorithm, earlier developed in MATLAB code for desktop computers was developed in JAVA code for Android mobile devices.

A summary of the laboratory (RUS) and of the Ruhr-Universität of Bochum in addition to the specification of the overall project will initially be presented.

The software development that is presented here is divided in two parts that are the system identification and the controller synthesis. The programming language and the development environment MATLAB in addition to Android language and environment will be presented too.

Finally, this paper presents a part of a software prototype that is capable of to identify and to present the control parameters stable region of a centrifugal pump.

Keywords: Stable Region, Smart Control, Mobile Device, MATLAB, Android, Centrifugal Pump, PID.

SUMÁRIO

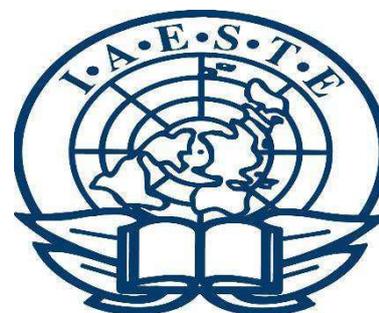
Resumo	v
Abstract.....	vi
Sumário	vii
1.1 IAESTE – International Association for the Exchange of Students for Technical Experience...1	
1.2 <i>Ruhr-Universität</i> de Bochum - RUB	3
2 Desenvolvimento	6
2.1 O Sistema	6
.....	Erro! Indicador não definido.
2.2 Objetivos	8
2.3 Tarefas	9
2.4 Ferramentas.....	10
2.4.1 matlab	10
2.4.2 Android SDK	10
2.5 Android X MATLAB	11
2.5.1 Algumas Telas de Comparação de Trechos de Código.....	12
.....	12
2.5.2 Códigos Nativos MATLAB	13
2.6 Documentação	14
2.7 Interface	15
.....	15
3 Conclusão	16
Bibliografia	Erro! Indicador não definido.

1.1 INTRODUÇÃO

Nos dias de hoje, mesmo com as grandes cidades abarrotadas de pessoas, sabemos que existem milhares e milhares de regiões no mundo inóspitas. Considerando essas regiões e um objeto que em algumas situações fazem parte do mesmo baseamos este trabalho. O objeto seriam as bombas centrífugas que se faz necessária para o manuseio de determinados equipamentos ou transferência de líquidos. A idéia principal do trabalho efetuado neste estágio foi fazer o controle de bombas em locais inóspitos a partir de uma dispositivo móvel.

1.2 IAESTE

A IAESTE, Associação Internacional de Intercâmbios de Estudantes para Experiência Técnica se trata de uma entidade não governamental, apolítica e sem fins lucrativos, membro consultivo da UNESCO, fundada em 1948 pelo Imperial College, em Londres e presente nos 5 continentes.



A IAESTE promove intercâmbio entre estudantes com vínculo universitário em mais de 80 países. Através da IAESTE, os estudantes podem ter uma experiência profissional no exterior, de 1 a 12 meses de duração, dentro da sua área de estudo.

Cada estudante recebe uma bolsa auxílio suficiente para cobrir as suas despesas básicas com acomodação, transporte e alimentação. Já as passagens aéreas e a inscrição no programa são desembolsadas pelos estudantes.

Anualmente, sempre no mês de Janeiro, é realizada a Conferência Internacional da IAESTE. Neste evento são intercambiadas as vagas de estágio captadas por cada um dos países membros da Associação.



Principais objetivos do programa IAESTE:

- Proporcionar aos estudantes com vínculo universitário uma experiência prática relevante, dentro da sua área de formação.
- Oferecer às empresas a contribuição temporária de estudantes qualificados e altamente motivados
- Ser uma fonte de enriquecimento cultural tanto para os estudantes como para os ambientes que se beneficiam de sua atuação



Ao contrário do que acontece na maioria dos países onde a IAESTE está presente, no Brasil o programa não conta com subsídios do Governo ou da iniciativa privada e tão pouco ele é gerido por voluntários. Em nosso país, o IAESTE é gerido pela ABIPE (Associação Brasileira de Intercâmbio Profissional e Estudantil) que se trata de uma entidade apolítica e sem fins lucrativos formada por profissionais remunerados. A ABIPE se auto-sustenta com a arrecadação das taxas recebidas. O apoio das universidades e empresas parceiras é fundamental, mas limita-se à concessão de oportunidades de estágio para estudantes estrangeiros.

1.3 RUHR-UNIVERSITÄT DE BOCHUM – RUB



A Ruhr-Universität Bochum mais conhecida como RUB está localizada na cidade de Bochum, cidade de aproximadamente 360 mil habitantes que fica no estado de North Rhine-Westphalia na região noroeste da Alemanha. A Universidade foi fundada em 1962 sendo a primeira nova universidade pública após a segunda guerra mundial.

A universidade de Bochum foi uma das primeiras universidades da Alemanha a introduzir a titulação internacional de bacharel e mestre substituindo assim os tradicionais títulos alemães de diploma e magistério. A RUB é atualmente financiada pelo estado de North Rhine-Westphalia e possui aproximadamente 39 mil estudantes e 5 mil funcionários tornando-a uma das 10 maiores universidades de toda a Alemanha.



Diferentemente das demais universidades alemãs, os prédios da universidade de ruhr são todos centralizados em um campus com arquitetura da década de 60, consistindo de 14 prédios idênticos localizados do lado de um cinturão verde.



Boc

1.4 REGELUNGSTECHNIK UND SYSTEMTHEORIE (RUS)

O Regelungstechnik und Systemtheorie (RUS), palavra em alemão que significa Grupo de Controle Automático e Teoria de Sistemas, se trata de um laboratório vinculado ao departamento de Engenharia Mecânica localizado no conhecido como IC da Ruhr-Universität Bochum mais conhecida como RUB. Neste mesmo prédio estão presentes ainda a Faculdade de Engenharia Mecânica, Faculdade de Engenharia Civil e Ambiental além do centro interdisciplinar para simulação de materiais avançados.

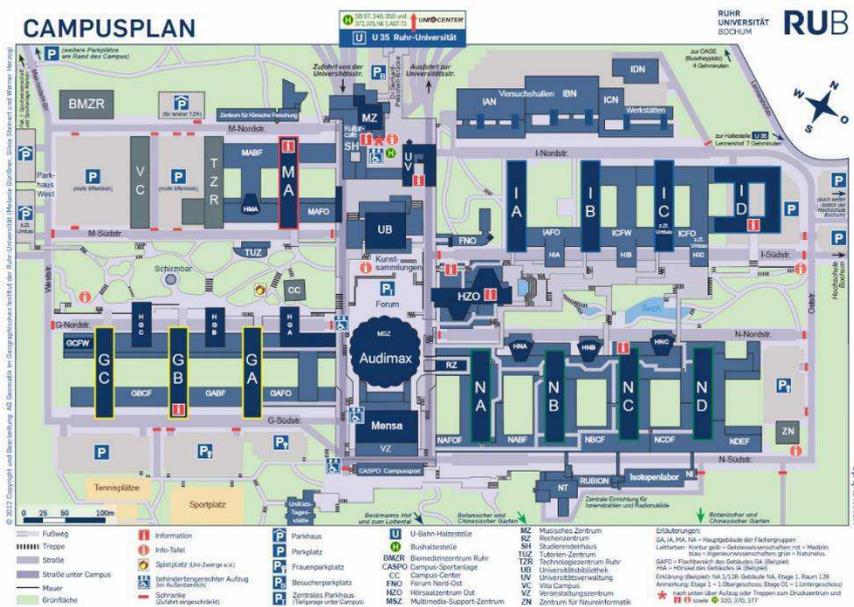


FIGURA 1- MAPA DA UNIVERSIDADE DE BOCHUM

No ano de 2013 o prédio foi reformado devido a um plano de reformas da universidade dos prédios que completaram 50 anos. Foram gastos aproximadamente 70 milhões de euros nessa reforma.



FIGURA 2 - REFORMA DO PRÉDIO IC

Neste laboratório, as pessoas com as quais trabalhei foram:

	PROF. DR.-ING. MARTIN MÖNNIGMANN Posição : Head
	DIPL.-INFORM. M. KASTSIAN Posição : Scientific Staff
	DR.-ING. DARYA KASTSIAN Posição : Scientific Staff

	DIPL.-ING. SEBASTIAN LEONOW Posição : Scientific Staff
	DIPL.-ING. SEBASTIAN LEONOW Posição : Technical Staff

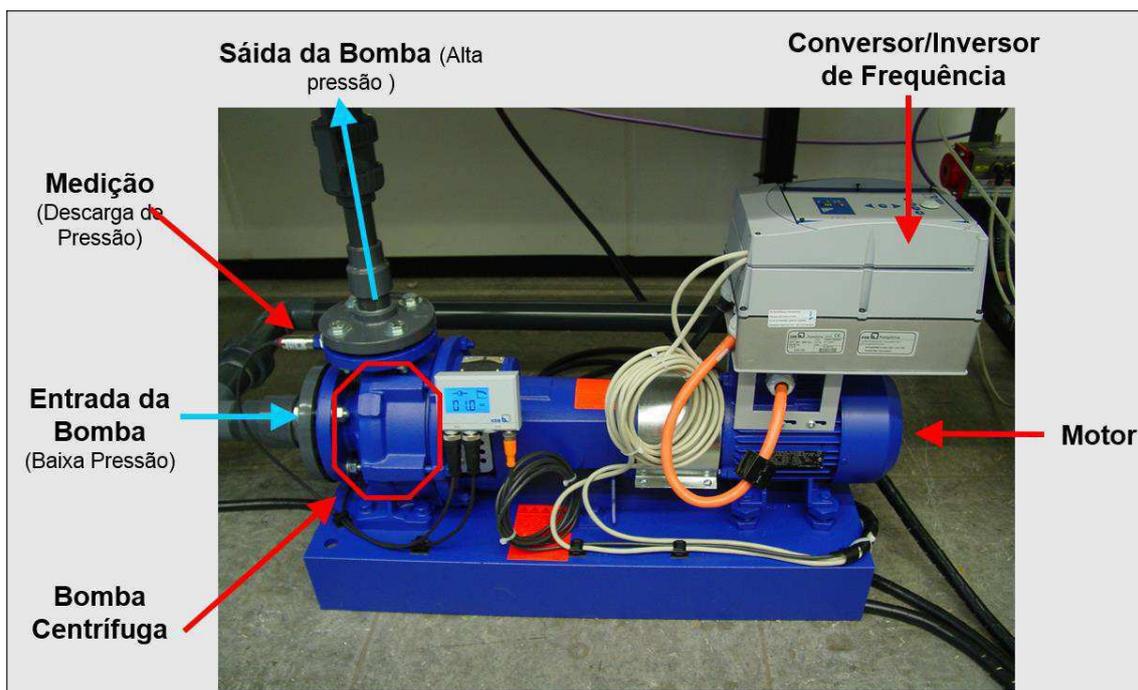
2 DESENVOLVIMENTO

2.1 O SISTEMA

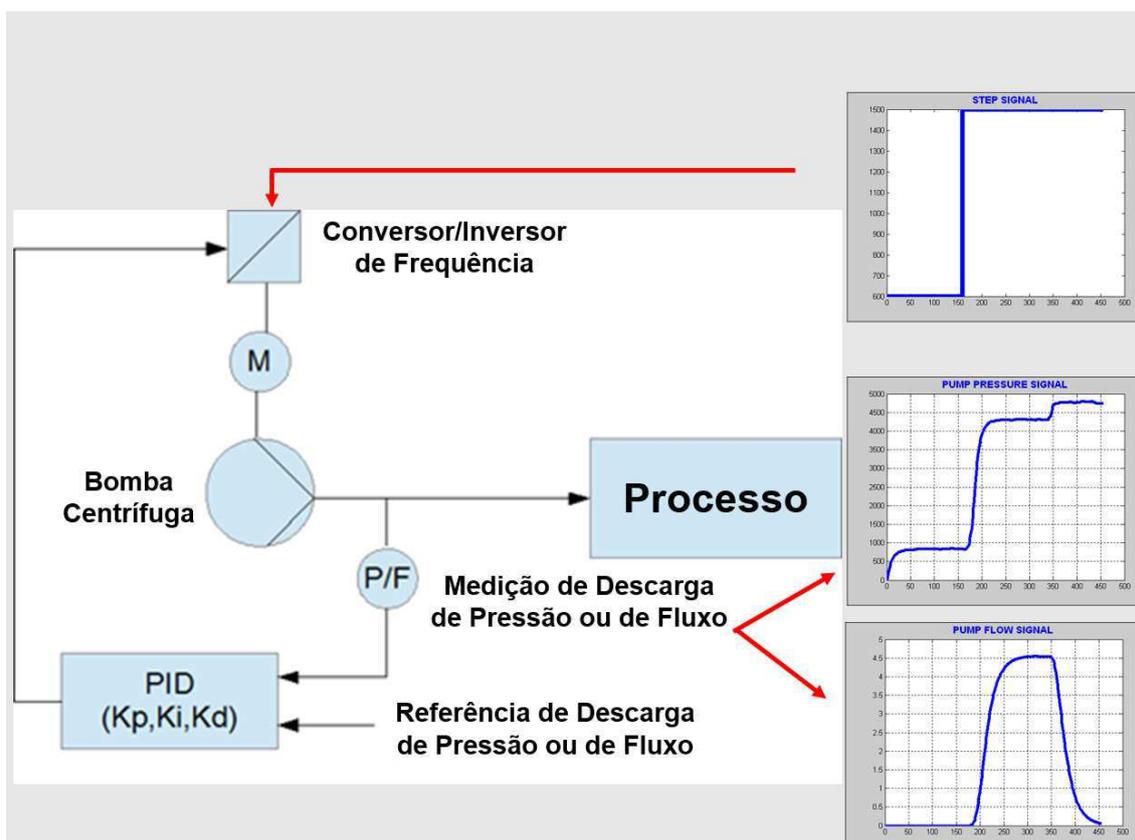
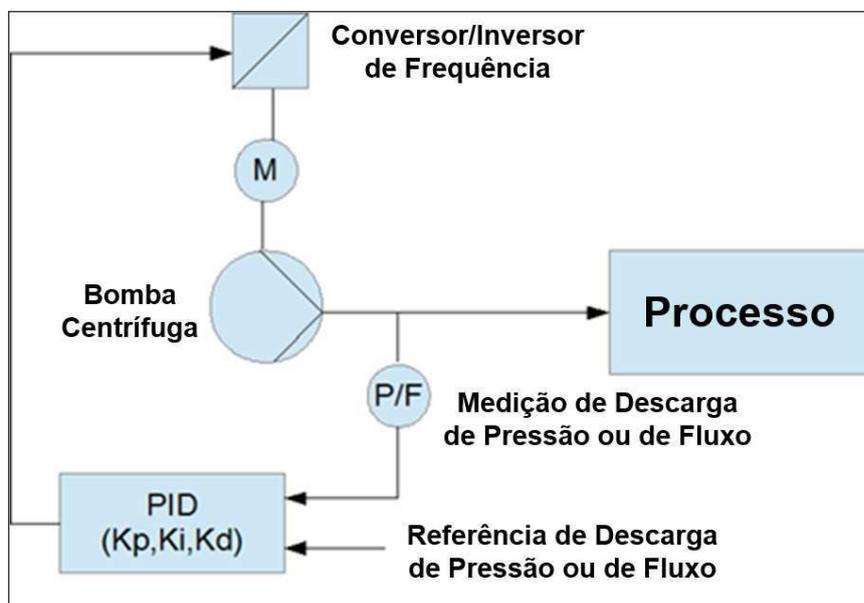
Nos dias atuais o equipamento mais utilizado para bombear líquidos são as bombas centrífugas. Ela está presente elevando, pressurizando ou transferindo líquidos de um local para o outro em vários setores como:

- Saneamento básico.
- Irrigação de lavouras.
- Edifícios residenciais.
- Indústria em Geral

A figura X apresenta os componentes básicos de uma bomba centrífuga. A quantidade de fluxo ou pressão de saída da bomba é proporcional à velocidade do motor que por sua vez é controlado pelo conversor de frequência.

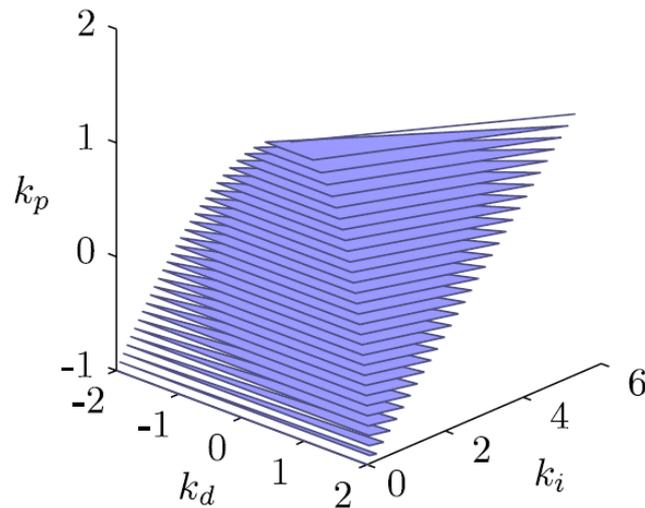


A partir da comparação da medida do fluxo ou pressão de saída com um valor de referência pré-determinado, pode-se então ser feito o controle PID desse sistema aumentando ou diminuindo a velocidade do motor a partir do conversor de frequência, alterando assim consequentemente a quantidade de fluxo ou pressão de saída. O diagrama desse sistema pode ser visto na figura X.

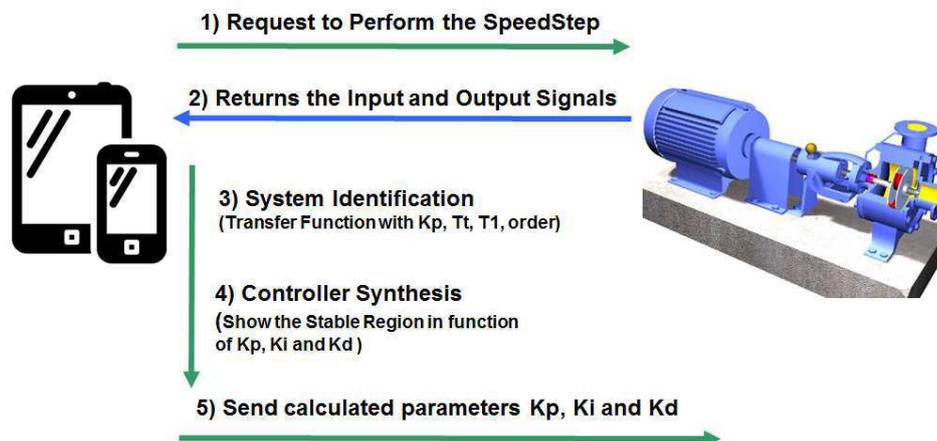


2.2 OBJETIVOS

O objetivo deste trabalho se resume a identificação e visualização em um dispositivo móvel da região tridimensional de estabilidade na qual o sistema em questão se mantenha estável. Esta região tridimensional possui como eixos os parâmetros de um controlador PID, ou seja, K_p , K_i e K_d . Na figura X pode ser visualizado um exemplo de região tridimensional de estabilidade como descrito acima.



No contexto maior, o projeto também envolve a parte de comunicação entre o dispositivo móvel e a bomba para a recepção dos sinais de entrada e saída da bomba e envio dos parâmetros calculados, mas essa parte do projeto está fora do escopo das atividades a mim atribuídas durante o estágio. A partir da figura X podemos ter uma visão geral do escopo do projeto tendo sido a mim atribuídas as atividades 3 e 4.



2.3 TAREFAS

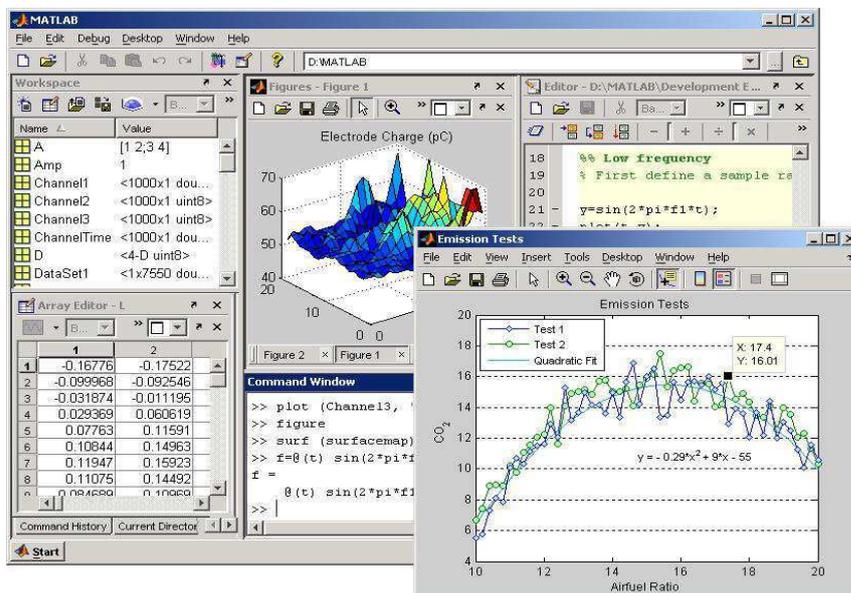
As atividades foram implementadas em JAVA para Android trechos de códigos já implementados em MATLAB. Como visto anteriormente na figura X, as atividades desenvolvidas foram a atividade 3 e 60% da atividade 4 como pode ser visualizado abaixo junto com as referentes subatividades.

- Atividade 1 - Mobile Device Request to Pump to Perform the SpeedStep
- Atividade 2 - Pump Send the Input and Output Signals to the Mobile Device
- Atividade 3 - System Identification
 - Import ScopeData  
 - Calculate Gain K  
 - Calculate T  
 - Calculate Dead Time  
 - Calculate Order  
- Atividade 4 - Controller Synthesis
 - Nyquist Decoposition  
 - D Composition  
 - Calculate Singular Frenquecies  
 - Calculate Stabilizable KP Interval  
 - Calculate Stable Region 
- Atividade 5 - Mobile Device Send to the Pump the Calculated parameters

Algumas atividades não foram completadas durante o período do estágio devido ao fato de que um dos orientadores do estágio estava ainda finalizando alguns códigos em MATLAB, no passo, em que, o desenvolvimento em Android levava uma duração de tempo menor pois se tratava de, a grosso modo, uma tradução.

2.4 FERRAMENTAS

2.4.1 MATLAB

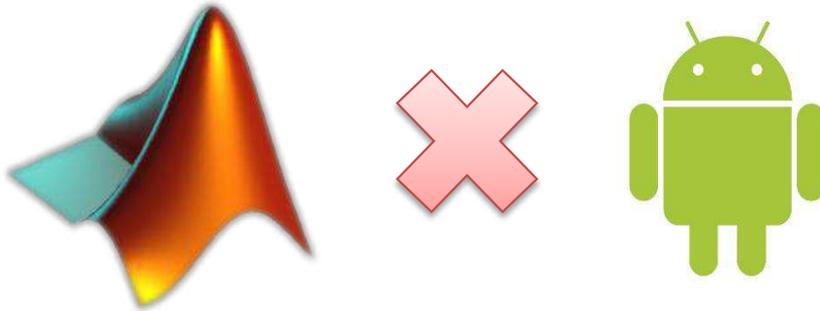


2.4.2 ANDROID SDK



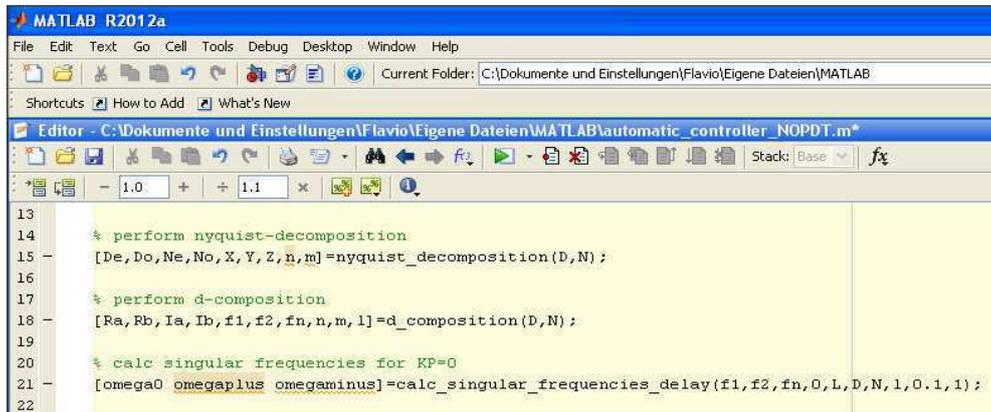
2.5 ANDROID X MATLAB

No momento do desenvolvimento propriamente dito houveram várias diferenças que em alguns momentos ajudaram e em outros nem tanto. A simplicidade da linguagem MATLAB catalisou o desenvolvimento em Android mas já um pequeno detalhe fez com que o desenvolvimento de vários métodos atrasasse bastante.



Esse detalhe foi a diferença da indexação de arrays em MATLAB e Android. Enquanto que em Android o índice inicial de cada array é 0, em MATLAB isso não ocorre, sendo o primeiro índice o número 1. A primeira vista esse detalhe é mínimo e faz com que se venha a pensar que basta uma adição unitária para resolver tudo. Infelizmente não, dado que nos códigos em MATLAB vários métodos fazem cálculos relativos, somando e subtraindo índices e valores complicando assim a implementação.

2.5.1 ALGUMAS TELAS DE COMPARAÇÃO DE TRECHOS DE CÓDIGO



MATLAB R2012a

File Edit Text Go Cell Tools Debug Desktop Window Help

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

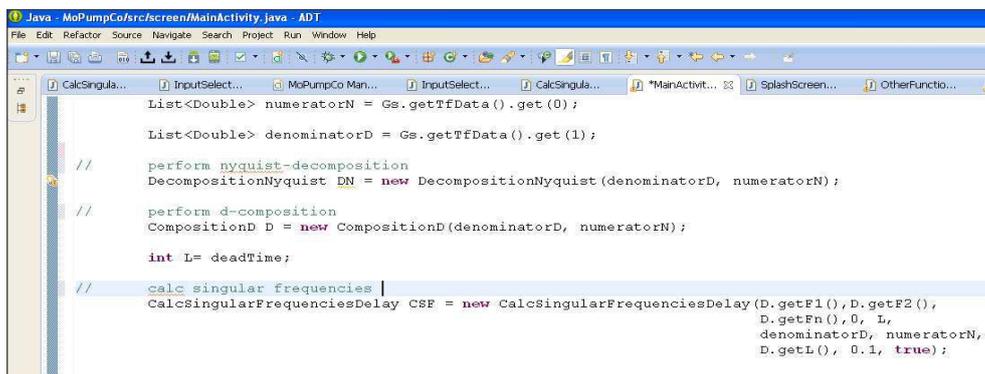
Shortcuts How to Add What's New

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,1,0.1,1);
22

```



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

File Edit Refactor Source Navigate Search Project Run Window Help

CalcSingular... InputSelect... MoPumpCo Man... InputSelect... CalcSingular... *MainActivit... SplashScreen... OtherFuncio...

```

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

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

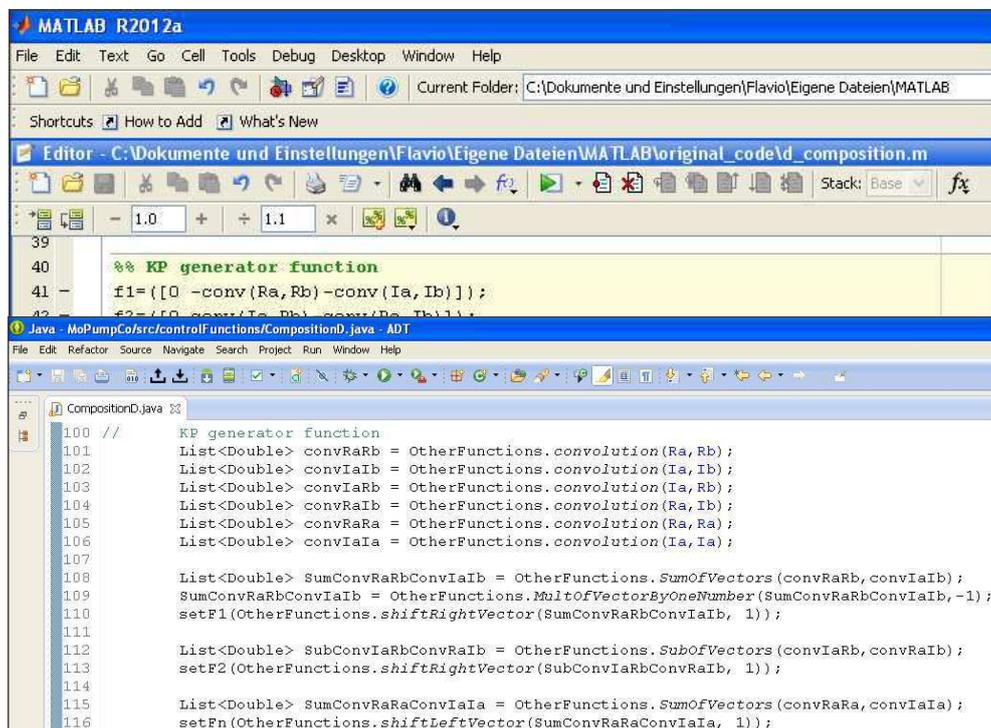
// perform nyquist-decomposition
DecompositionNyquist DN = new DecompositionNyquist(denominatorD, numeratorN);

// 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);

```



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\l_d_composition.m

```

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

```

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

File Edit Refactor Source Navigate Search Project Run Window Help

CompositionD.java

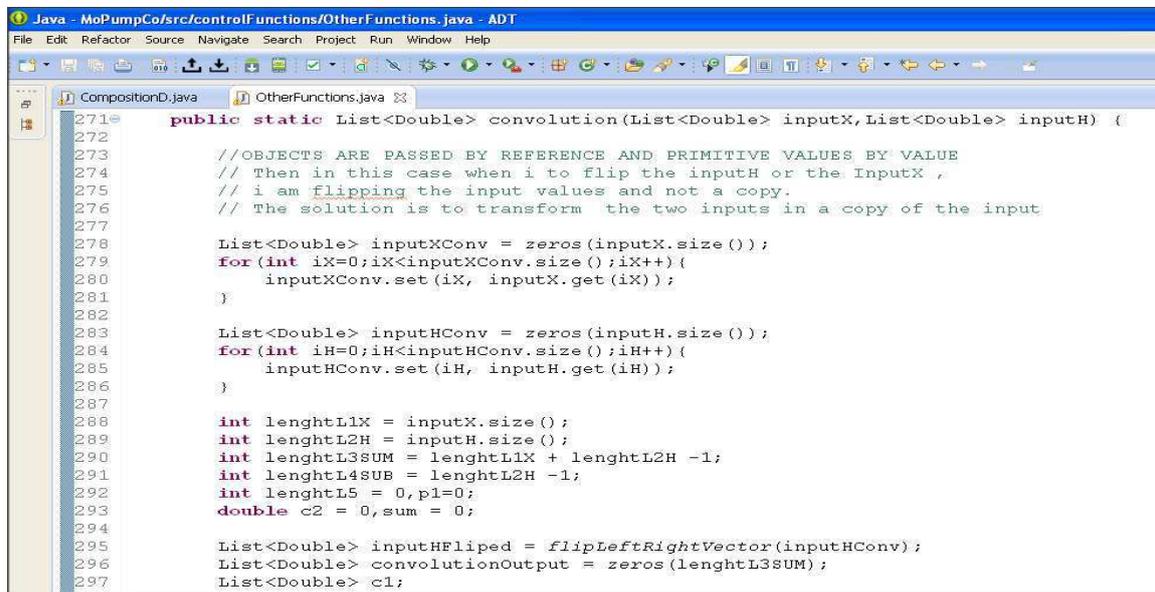
```

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.MultiplyVectorByOneNumber(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));

```

2.5.2 CÓDIGOS NATIVOS MATLAB

Dado que a linguagem de programação MATLAB possui uma vasta variedade de métodos matemáticos para tratamentos de dados, o mesmo não ocorre em JAVA, tendo sido assim necessário implementar alguns métodos como por exemplo o método de Convolução.

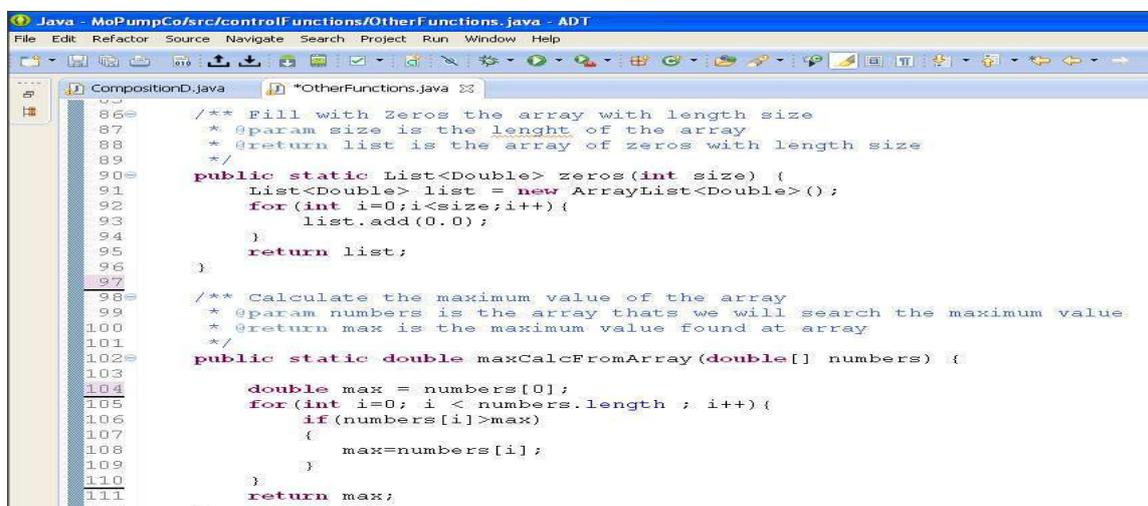


```

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.size());
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 -1;
291 int lenghtL4SUB = lenghtL2H -1;
292 int lenghtL5 = 0,p1=0;
293 double c2 = 0,sum = 0;
294
295 List<Double> inputHFliped = flipLeftRightVector(inputHConv);
296 List<Double> convolutionOutput = zeros(lenghtL3SUM);
297 List<Double> c1;

```

Uma das grandes vantagens da linguagem JAVA é permitir a orientação a objetos facilitando assim a estruturação do código. Um das classes criada para facilitar a codificação foi a classe FunçãoDeTransferência, classe bastante utilizada nos algoritmos matemáticos de controle.



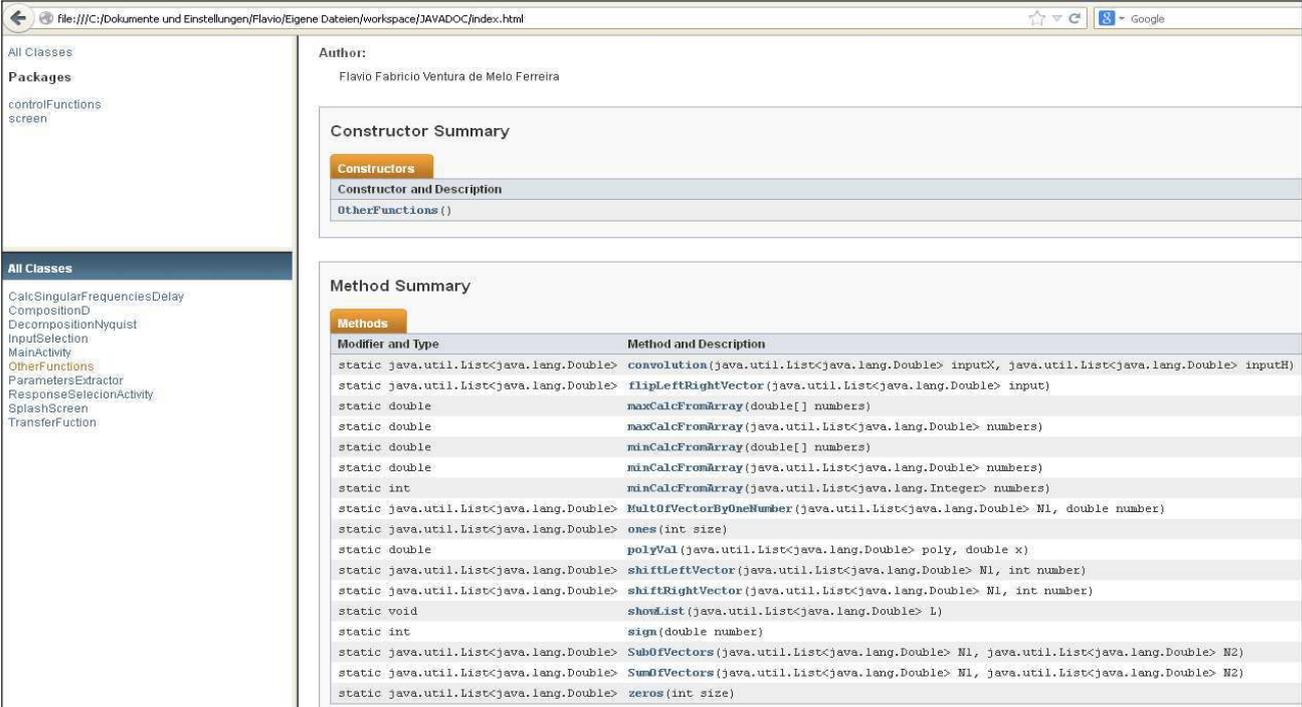
```

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 }

```

2.6 DOCUMENTAÇÃO

A partir de um gerador de documentação chamado Javadoc foi gerado a documentação dos métodos a partir do código-fonte. O resultado é expresso em HTML. É constituído, basicamente, por algumas marcações muito simples inseridas nos comentários do programa. Este sistema é o padrão de documentação de classes em Java, e muitas dos IDEs (Integrated Development Environment) desta linguagem geram automaticamente um Javadoc em HTML como é o exemplo do Android SDK utilizado no desenvolvimento deste trabalho.



The screenshot shows a web browser window with the address bar displaying the file path: `file:///C:/Dokumente und Einstellungen/Flavio/Eigene Dateien/workspace/JAVADOC/index.html`. The browser's address bar also shows a search engine icon for Google.

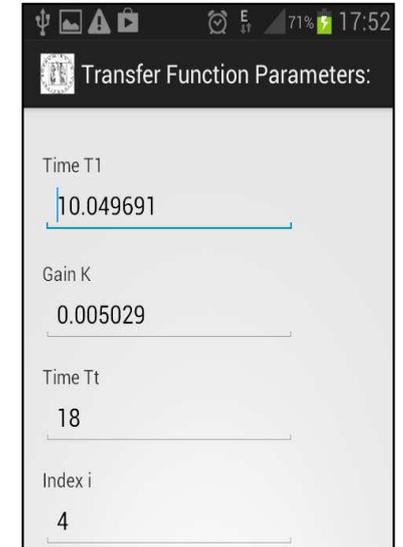
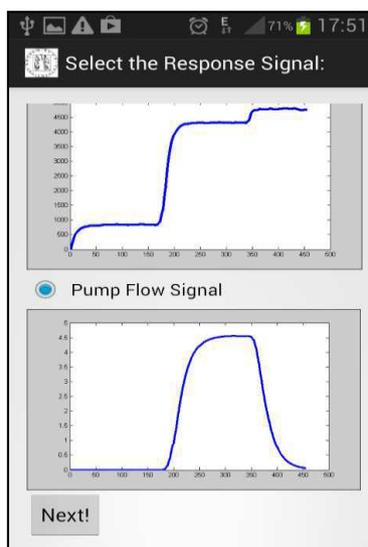
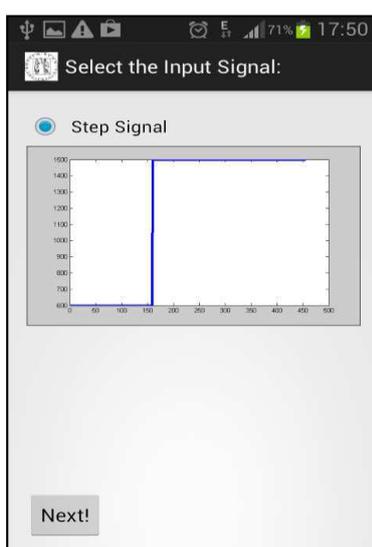
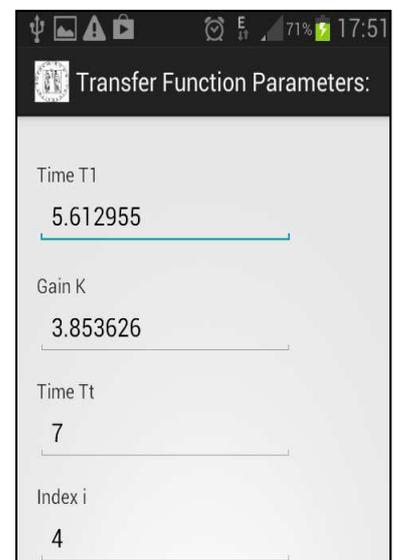
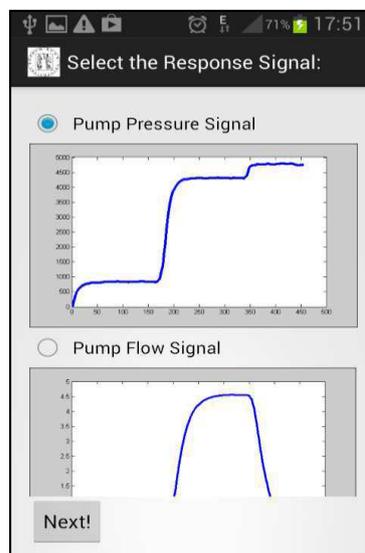
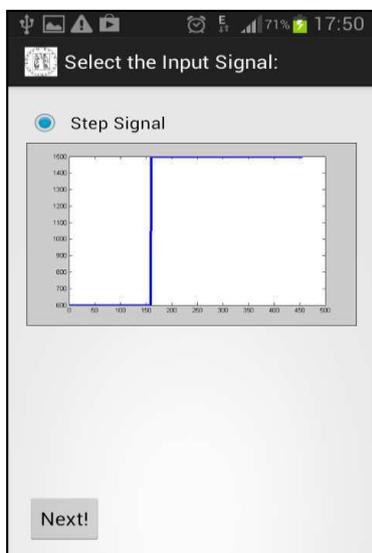
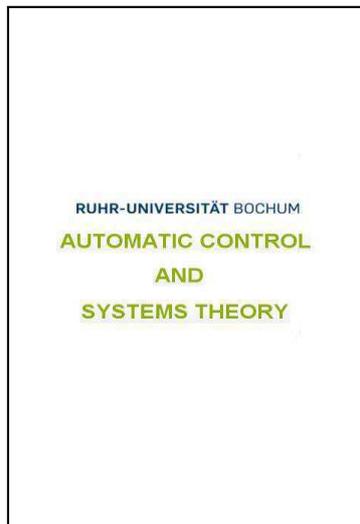
The page content is organized into several sections:

- All Classes:** A sidebar on the left lists various classes and packages, including `controlFunctions` and `screen`.
- Author:** The author's name is listed as `Flavio Fabricio Ventura de Melo Ferreira`.
- Constructor Summary:** This section includes a sub-section for **Constructors** and a table with the following entry:

Constructor and Description
<code>OtherFunctions ()</code>
- Method Summary:** This section includes a sub-section for **Methods** and a table with the following entries:

Modifier and Type	Method and Description
<code>static java.util.List<java.lang.Double></code>	<code>convolution(java.util.List<java.lang.Double> inputX, java.util.List<java.lang.Double> inputH)</code>
<code>static java.util.List<java.lang.Double></code>	<code>flipLeftRightVector (java.util.List<java.lang.Double> input)</code>
<code>static double</code>	<code>maxCalcFromArray (double[] numbers)</code>
<code>static double</code>	<code>maxCalcFromArray (java.util.List<java.lang.Double> numbers)</code>
<code>static double</code>	<code>minCalcFromArray (double[] numbers)</code>
<code>static double</code>	<code>minCalcFromArray (java.util.List<java.lang.Double> numbers)</code>
<code>static int</code>	<code>minCalcFromArray (java.util.List<java.lang.Integer> numbers)</code>
<code>static java.util.List<java.lang.Double></code>	<code>MultOfVectorByOneNumber (java.util.List<java.lang.Double> M1, double number)</code>
<code>static java.util.List<java.lang.Double></code>	<code>ones (int size)</code>
<code>static double</code>	<code>polyVal (java.util.List<java.lang.Double> poly, double x)</code>
<code>static java.util.List<java.lang.Double></code>	<code>shiftLeftVector (java.util.List<java.lang.Double> M1, int number)</code>
<code>static java.util.List<java.lang.Double></code>	<code>shiftRightVector (java.util.List<java.lang.Double> M1, int number)</code>
<code>static void</code>	<code>showList (java.util.List<java.lang.Double> L)</code>
<code>static int</code>	<code>sign (double number)</code>
<code>static java.util.List<java.lang.Double></code>	<code>SubOfVectors (java.util.List<java.lang.Double> M1, java.util.List<java.lang.Double> M2)</code>
<code>static java.util.List<java.lang.Double></code>	<code>SumOfVectors (java.util.List<java.lang.Double> M1, java.util.List<java.lang.Double> M2)</code>
<code>static java.util.List<java.lang.Double></code>	<code>zeros (int size)</code>

2.7 INTERFACE



3 CONSIDERAÇÕES

Esse estágio foi de grande valia por todo o conhecimento e experiências adquiridas durante os 2 meses de aprendizado. O aprendizado em relação a língua foi o mais importante influenciando bastante nas ofertas de emprego recebidas após o estágio.

Devido a base de conhecimento adquiridos nas disciplinas da UFCG acerca de Engenharia de Software, projeto de Interface Gráfica , Controle Analógico dentre outras não existiram dificuldades técnicas no desenvolvimento das minhas atividades na Alemanha.

