

# SIMULASI MATLAB UNTUK PERANCANGAN PID CONTROLER

**Pandapotan Siagian, ST, M.Eng**

Dosen Tetap STIKOM-Dinamika Bangsa - Jambi

## ***Abstrak:***

*Perancangan PID Controller dengan simulasi MatLab dapat diterapkan secara tepat. Software Matlab Sangat mudah diaplikasikan, dalam hal simulasi perancangan dan analisa response sinyal karena matlab mempunyai fasilitas Control Toolbox, yang sangat memudahkan dalam perancangan untuk menganalisa response hasil perancangan yang disetting dari beberapa variable input yang bervariasi. Hal ini software MatLab Sangat dibutuhkan dalam perancangan untuk memperoleh output yang diinginkan, karena dengan mengatur variable input juga dapat menggabungkan beberapa sistem dari control PID. Perancangan sistem dapat menggabungkan PI,PD dan mengkombinasikan PID. Selain hal perancangan yang paling perlu di setting dalam kestabilan response yaitu waktu response untuk mendapatkan output sistem yang stabil sesuai dengan keinginan perancang.*

## **I. Pendahuluan**

Pengendalian secara otomatis sudah berkembang dan Sangat berperan penting dalam industri untuk mendapatkan hasil produksi yang diinginkan tercapai dan harus di dukung teknologi yang tepat. Teknologi pengendalian otomatis ini juga sudah banyak di jumpai pada masyarakat luas dan peranan *Automatic System* memberikan peran yang sangat besar dalam kehidupan manusia. Perkembangan teknologi ini harus di aplikasikan dalam bidang ilmu pengetahuan agar dapat membuka wacana pengetahuan dalam analisa dalam dunia kerja baik di industri atau di masyarakat.. Aplikasi yang banyak dijumpai yaitu remote control TV, alarm automatic, puma injection automation dan saat ini yang mengarah teknologi robot pramusaji di restoran. Sistem kontrol dalam penerapannya pada masyarakat berupa alat atau produksi secara otomatis. Simulasi matlab dalam hal ini berperan sebagai langkah utama secara teori untuk mendapatkan response yang stabil. Dengan perolehan response secara teori dapat diterapkan dalam perancangan alat sesungguhnya.

Dalam simulasi matlab saat ini, perancangan sistem kontrol yang yang diinginkan, lebih dahulu menganalisa dan menentukan *response output* yang diinginkan, agar lebih mudah penerapannya dan agar dapat menentukan variable-variabel input. Adapun konsep yang harus diperhatikan yaitu :

1. Respon sistem terhadap variabel masukan seperti : *step function, ramp function, dan impulse function dan juga noise.*
2. Kestabilan sistem dengan menggunakan model : *root locus, frekuensi respon, state space.*

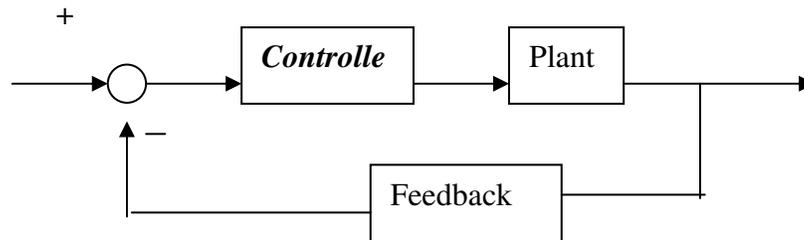
### 3. Respon sistem dengan kombinasi –kombinasi dari PID.

Dalam menganalisa sistem yang sudah dirancang, suatu kendala yang banyak di jumpai yaitu bagaimana cara untuk memperoleh transfer function dari sistem tersebut kemudian menganalisanya secara cepat dan mudah.

Hal ini, sangat sering ditemukan oleh perancang sistem kontrol masih mendapatkan kesulitan untuk mendapatkan transfer function suatu sistem. Masalah tersebut dapat terjawab apabila koonsep teori atau penurunan persamaan matematisnya sudah dapat, sangatlah mudah mengaplikasikan persamaan dan untuk melakukan proses analisisnya tidak terlalu rumit. Karena kendala tersebut sudah dapat di atasi dalam mengembangkan software-software aplikasi yang akan membantu perancangan kendali proses. Salah satu software MatLab.

## II. Kontrol PID Controler

Simulasi matlab untuk sistem kontrol *Propositional Integral Derivatif (PID)*. Kontrol PID merupakan salah satu jenis pengatur yang banyak digunakan adapun software yang lainnya View Lab dll. Sistem dengan software matlab ini mudah dikombinasikan dengan software atau sistem atau metoda pengaturan yang lain seperti Fuzzy, wavelet dan Robust. Hal ini akan mendapatkan sistem pengatur yang semakin baik Perancangan PID yang di bahas dalam tulisan ini yaitu gambaran untuk *Feedback System*, sesuai dengan gambar 1 sebagai berikut.



Gambar 1 : Blok diagram *Feedback Systems*

Kontrol PID untuk blok diagram yang akan dirancang dan akan di analisa responnya yaitu menggunakan persamaan transfer function sebagai sebagai berikut :

$$H(s) = \frac{K_D s^2 + K_P s + K_I}{s^3 + K_D s^2 + K_P s + K_I} \quad (1)$$

Kombinasi dari PID untuk persamaan transfer fuction dapat dirancang dengan beberapa kombinasi dan mempunyai variable konstanta untuk masing masing kombinasi. Untuk kontrol PID dapat menggabungkan system dan pengaturan antara lain : *P (Proportional) Controller, D (Derivative) Controller, dan I (Integral) Controller*. Masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang

disebut sebagai *konstanta*. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel 1 di bawah ini :

Tabel 1 : Respon Sistem Kontrol PID Terhadap Perubahan Variabel Konstanta

<b>Closed-Loop Response</b>	<b>Rise Time</b>	<b>Overshoot</b>	<b>Settling Time</b>	<b>SS Error</b>
K <sub>p</sub>	Decrease	Increase	Small change	Decrease
K <sub>i</sub>	Decrease	Increase	Increase	Eliminate
K <sub>d</sub>	Small change	Decrease	Decrease	Small change

Parameter-parameter tersebut, tidak bersifat independent, sehingga pada saat salah satu nilai konstantanya diubah, maka mungkin sistem tidak akan bekerja atau memberi respon seperti yang kita inginkan. Dimana table 1 tersebut hanya dipergunakan sebagai acuan apabila melakukan perubahan konstanta pada kontrol PID. Untuk merancang suatu kontrol PID, biasanya dipergunakan metoda *trial & error*. Trial and error terhadap rancangan dilakukan III. Model Konvensional

Perancangan untuk sistem kontrol, lebih awal melakukan flowchart atau diagram blok sistem rancangan. Blok diagram yang mempunyai komponen transfer function, transfer function yang terdapat pada sistem perlu di analisa dengan menggunakan model pengontrolan yang berbeda-beda. Respon yang diperoleh merupakan hasil, yang dipengaruhi perubahan sinyal input sehingga perancang dapat melihat respon sistem jika mendapat input sinyal tertentu. Kombinasi antara sinyal input dan jenis aksi pengontrolan ini akan menghasilkan respon yang berbeda-beda.

dengan mengubah variabel perancangan dan juga kombinasi rancangan. Hal ini dilakukan untuk mendapatkan kombinasi yang baik atau kombinasi yang mendapatkan error yang minimal sesuai yang dirancang tersebut. Perancang harus mencoba kombinasi pengatur beserta konstantanya untuk memilih perancangan mana yang akan diterapkan, diantara perancangan mana hasil terbaik yang paling sederhana dan memperoleh respon kestabilan yang lebih baik.

Sebelum di kenal matlab, untuk melihat hasil sinyal output atau untuk melihat respon suatu sistem dengan gabungan berbagai model kombinasi sinyal input dan set pengontrolan merupakan hal yang sulit dan untuk mendapatkan respon yang cukup lama. Proses yang harus dipun prosedur yang harus lakukan sebagai berikut :

1. Mendapatkan transfer function sistem (dalam s-domain) dengan Laplace Transform.
2. Menentukan jenis aksi pengontrolan beserta dengan konstantanya.
3. Menggabungkan transfer function yang sudah didapatkan dengan jenis aksi pengontrolan.
4. Menentukan sinyal input yang akan dimasukkan (biasanya fungsi step, fungsi ramp dan pulse) dan menggabungkannya ke dalam transfer function yang baru.
5. Melakukan perhitungan invers Laplace Transform untuk mendapatkan fungsi dalam t-domain.
6. Menggambar respon berdasarkan fungsi dalam t-domain.

Proses dalam melakukan langkah-langkah di atas diperlukan ketelitian yang tinggi dan hasil gambaran sistem sering kali kurang akurat. Untuk perancang ingin mengamati respon sistem terhadap sinyal input yang lain, maka proses-proses tersebut sebagian besar akan diulang kembali. Hal ini bertambah kompleks jika perubahan yang dilakukan tidak terbatas pada sinyal input, tetapi juga pada jenis aksi pengontrolannya.

Sehingga untuk mendapatkan respon dari berbagai macam kombinasi, membutuhkan waktu yang relatif lama. Selain itu, perancang juga melakukan proses perhitungan yang rumit.

#### IV. Model Simulasi Matlab

Perkembangan teori kontrol juga diikuti oleh software pendukungnya. Mulai dari software untuk pemrograman sistem, sampai dengan software untuk proses simulasinya. Salah satu software yang dapat dipergunakan untuk simulasi tersebut adalah *MatLab* dari *Mathworks, Inc.*

Software ini dilengkapi dengan berbagai toolbox yang memudahkan pemakai untuk melakukan perhitungan-perhitungan tertentu. Bahkan saat ini sudah dikembangkan toolbox khusus untuk simulasi yang diberi nama *Simulink*.

Aplikasi MatLab dalam bidang pengaturan dilengkapi Control Toolbox. Toolbox ini sudah dilengkapi dengan berbagai macam fungsi pendukung yang dipergunakan dalam analisa sistem kontrol. Beberapa fungsi pendukung yang sering dipergunakan untuk menganalisa suatu sistem adalah : feedback, step, rlocus, series, dll. Untuk menganalisa suatu sistem, software hanya memerlukan masukan berupa transfer function yang ditulis dalam Laplace Transform dalam s- domain atau matriks. Untuk selanjutnya, pemakai tinggal memilih analisa yang akan dipergunakan. Tulisan ini akan membahas penggunaannya secara khusus untuk merancang PID Controller pada suatu sistem.

Sebagai contoh, suatu sistem kontrol memiliki transfer function sebagai berikut :

$$H(s) = \frac{1}{s^2 + 15s + 30} \quad (2)$$

kriteria perancangan sebagai berikut :

1. memiliki rise time yang cepat
2. overshoot sekecil mungkin
3. tidak memiliki steady state error.

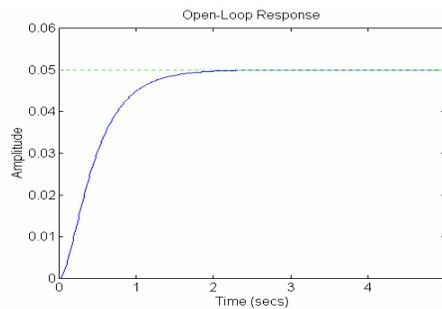
Untuk fungsi di atas, maka parameter-parameter yang dimasukkan berupa koefisien pembilang dan penyebutnya. Biasanya dipergunakan variabel *num* untuk pembilang dan *den* untuk penyebut. Kedua nama variabel tersebut tidak mutlak, jadi penggunaan nama variabel yang lain juga diperbolehkan. Setelah itu komputer sudah siap untuk menganalisa sistem kontrol.

Langkah kedua yang perlu dilakukan adalah memilih jenis input yang akan dimasukkan ke dalam sistem. Input ini bisa berupa *step*, *pulse*, *ramp*, *sinus*, dan sebagainya. Sebagai dasar analisa akan diperlunakan fungsi *step*.

Fungsi dasar yang akan sering dipergunakan adalah *step*, dengan syntax : `step(num,den,t)` untuk s-domain atau `step(A,B,C,D)` untuk state space. Fungsi ini menghasilkan gambar respon sistem bila diberi input unit step dalam t-domain.

```
num = [1];  
den = [1 15 30];  
step(num,den)  
title('OpenLoop Response')
```

Respon sistem terbuka (*open loop response*) dapat dilihat pada gambar 2.



Gambar 2 Respon Sistem Loop Terbuka Menggunakan P Controller

Sistem di atas memiliki steady state error yang tinggi, yaitu 0,95. Sebab respon tertinggi hanya didapatkan pada amplitudo 0,05. sistem steady state error dan amplitudo seperti diatas diperoleh dengan rise time yang cukup besar (sekitar 1,7 detik). Rise time yang memperoleh sebesar 1,7 kurang menguntungkan. Hal ini lah yang perlu diperhatikan dalam perancangan kombinnasi mana yang akurat dan sesuai dengan hasil respon yang di inginkan perancang. Sistem open loop diatas kurang tepat di aplikasikan karena error dan rise time yang masih relative besar. Dalam menghasilkan sistem kontrol yang baik, diperlukan sistem yang tertutup (close loop system). Sesuai dengan blok diagram sistem yang terdapat pada gambar 1 sistem kontrol yang efektif harus menggunakan feedback, dimana dalam feedback dapat distabilkan keseimbangan sistem. Sistem yang memiliki feedback, yang akan membandingkan kondisi sesungguhnya dengan seting poin yang diberikan.

### Kontrol Proportional

Kontrol proposional dapat mengurangi rise time, menambah overshoot, dan mengurangi steady state error. Hal ini dapat dikombinasikan dengan melihat tabel1. Closed-loop transfer function sistem di atas dengan menggunakan kontrol proposional (P Controller) sebagai berikut :

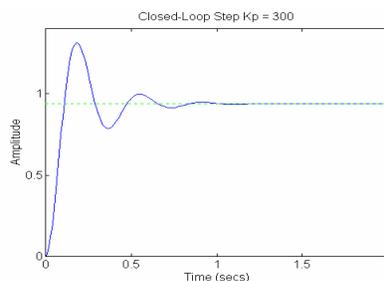
$$H(s) = \frac{K_p}{s^2 + 15s + (30 + K_p)} \quad (3)$$

Contoh, dengan variabel konstanta  $K_p = 300$ , maka :

$K_p = 300$ ; num = [Kp]; den = [1 15 30+Kp]; t = 0 : 0.01 : 2; step(num,den) title('Closed-Loop Step Kp = 300')

Penambahan variabel t=0:0.01:2 dimaksudkan untuk melihat respon sistem dari t=0 s/d t=2, dengan step 0,01 detik.

Gambar 3 Respon Sistem Loop Tertutup Menggunakan P Controller



Dari gambar 2 di atas, dapat dilihat bahwa penambahan kontrol proposional ( P Controller) mengurangi rise time dan steady state error, tetapi menambah overshoot. Overshoot yang terjadi masih terlalu besar. Jika konstanta  $K_p$  diperbesar, maka overshoot yang terjadi juga semakin besar, settling time juga semakin besar, tetapi rise time nya menjadi kecil. Kebalikan dari keadaan itu terjadi jika konstanta  $K_p$  diperkecil.

## VI. Proportional-Derivative Controller

Kontrol dengan loop tertutup (*Closed-Loop*) transfer function sistem di atas dengan PD Controller adalah :

$$H(s) = \frac{K_D s + K_p}{s^2 + (15 + K_D)s + (30 + K_p)} \quad (4)$$

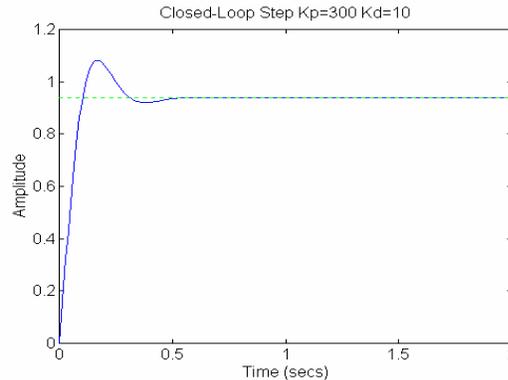
Misal,  $K_p = 300$  dan  $K_d = 10$ , maka :

```

Kp = 300;
Kd = 15;
num = [Kd Kp];
den = [1 15+Kd 30+Kp];
t = 0 : 0.01 : 2;
step(num,den)
title('Closed-Loop Step Kp=300 Kd=10')

```

Respon sistem tergambar terdapat pada gambar 4.



Gambar 4 Respon Sistem Tertutup Menggunakan PD Controller

Berdasarkan gambar di atas, penggunaan kontrol PD dapat mengurangi overshoot dan settling time, tetapi tidak memberikan dampak apa pun terhadap steady state error.

## VII. Proportional-Integral Controller

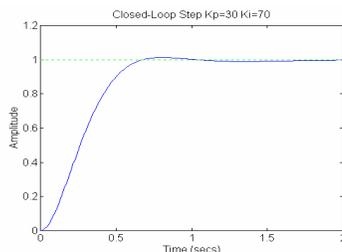
Kontrol dengan Proporsional dengan Integral dalam sistem Closed-Loop transfer function sistem di atas dengan PD Controller untuk persamaan :

$$H(s) = \frac{K_p s + K_i}{s^3 + 15s^2 + (30 + K_p)s + K_i} \quad (5)$$

Misal,  $K_p = 30$  dan  $K_i = 70$ , maka :

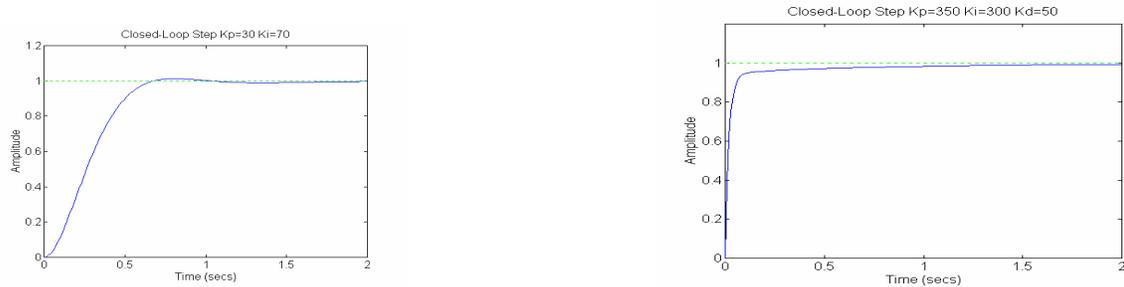
$K_p = 300$ ;  $K_i = 70$ ;  $\text{num} = [K_p \ K_i]$ ;  $\text{den} = [1 \ 15 \ 30+K_p \ K_i]$ ;  $t = 0 : 0.01 : 2$ ;  
 $\text{step}(\text{num}, \text{den})$   $\text{title}('Closed-Loop Step Kp=30 Ki=70')$

P dan I Controller memiliki karakteristik yang sama dalam hal rise time dan overshoot. Maka perlu melakukan minimum nilai variabel  $K_p$ , atau nilai konstanta  $K_p$  harus turunkan nilainya agar dapat mengatasi overshoot yang besar nilainya. Nilai  $K_i$  diambil



lebih besar dari  $K_p$ , karena diinginkan untuk meniadakan steady state error. Jika nilai  $K_p$  lebih besar dari  $K_i / K_p > K_i$ , maka steady state errornya tidak dapat dihilangkan atau diminimalkan.

Gambar 5 di bawah ini memperlihatkan respon sistem dengan PI Controller



Gambar 5 Respon Sistem Tertutup Menggunakan PI Controller

Dari gambar 5, dapat dilihat dari respon sistem bahwa rise time sistem menurun, dengan overshoot yang kecil, serta steady state errornya dapat dihilangkan.

### VIII. Proportional-Integral-Derivative Controller

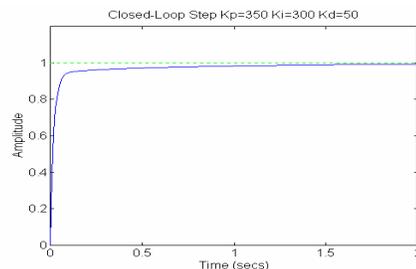
Bagian akhir dari simulasi ini adalah PID Controller, yang memiliki transfer function untuk sistem di atas adalah :

$$\frac{K_D s^2 + K_p s + K_I}{s^3 + (15 + K_D)s^2 + (30 + K_p)s + K_I} \quad (6)$$

$$K_p = 350;$$

Ki = 300; Kd = 50; num = [Kp Ki Kd]; den = [1 15+kd 30+Kp Ki]; t = 0 : 0.01 : 2; step(num,den) title('Closed-Loop Step Kp=350 Ki=300 Kd=50')

Respon sistem tergambar di bawah ini



Gambar 6 Respon Sistem tertutup Menggunakan PID Controller

Dari gambar 6 di atas terlihat bahwa kriteria sistem yang diinginkan sudah terpenuhi, yaitu tidak memiliki overshoot, rise time yang cepat, dan tidak memiliki steady state

error. Nilai-nilai konstanta yang terdapat pada simulasi ini menggunakan coba- coba dalam memberikan variabel dan konstanta sebagai input atau simulasi ini dilakukan dengan *trial and error*. Perancang yang dilakukan akan mendapatkan hasil respon yang berbeda, karena perancangan yang berbeda akan mendapatkan variabel input yang berbeda dan hasil respon juga akan berbeda atau dengan mendapatkan nilai yang berlainan untuk memenuhi kriteria di sesuai respon yang terdapat pada percobaan atas. Hal itu terjadi karena perubahan pada salah satu konstanta akan berpengaruh pada konstanta yang lain. Artinya tidak akan didapatkan hasil sesuai dengan tabel 1. Tabel tersebut hanya dipergunakan sebagai acuan perancangan.

Berikut ini beberapa konsep yang dapat dilakukan untuk mendapatkan respon yang diinginkan :

1. Dapatkan respon sistem terbuka sistem (open-loop) untuk menentukan bagian mana yang harus tukar nilainya agar mendapatkan *rise time*, *settling time*, *overshoot*, *steady state error* sesuai yang di kehendaki perancang.
2. Perlu menambahkan P Controller untuk memperbaiki rise time.
3. Menambahkan D Controller untuk memperbaiki overshoot.
4. Menambahkan I Controller untuk menghilangkan steady state error.
5. Kombinasikan konstanta dan variabel input yang ada untuk mendapatkan respon yang diinginkan.

Implementasikan sistem kontrol, sebenarnya tidak perlu menggunakan PID Controller. Jika sistem sudah memberikan respon yang cukup baik hanya dengan PI Controller, maka tidak perlu menambahkan D Controller ke dalamnya. Sehingga sistem menjadi lebih sederhana atau kombinasi yang main banyak membuat sistem menjadi makin kompleks.

Percobaan dalam merancang sistem dan menganalisa, sama halnya pada perancangan pada model diatas, yang dilakukan dengan input *unit step*. Jika menginginkan analisa dengan input yang berbeda, maka harus dilakukan modifikasi transfer function yang berbeda pula.

Untuk menganalisa sistem dengan input impulse function, maka transfer function dikalikan dengan faktor  $s$ . Demikian juga untuk input ramp function, perlu dikalikan dengan faktor  $1/s$ .

## **IX. Kesimpulan**

1. Untuk melakukan perancangan suatu PID Controller dengan simulasi matlab tidak mudah, karena perlu menentukan konstanta dan variabel lainnya dan si beri nilai input dengan melakukan secara *trial and error* dan juga perubahan salah satu konstanta akan berpengaruh terhadap parameter yang lain. Respon yang dihasilkan akan dapat melakukan percobaan. Apabila melakukan perancangan selanjutnya kesulitan sudah dapat diatasi dengan bantuan matlab.
2. Aplikasi MatLab tidak hanya pada perancangan PID Controller, dapat juga dilakukan untuk analias-analisa sistem kontrol seperti Bode Diagram, analisa kestabilan Nyquist, Root Locus, State Space, pengolahan citra, wavelet, dll.

**Daftar Pustaka :**

1. D C Reddy,. Control and Signal Processing. Mc GrawHill, Inc. 2001.
2. Ogata, Katsuhiko. Modern Control Engineering. 3<sup>rd</sup> ed. Prentice Hall International. 1997
3. Ogata, Katsuhiko. Solving Control Engineering Problems with MatLab. Englewood Cliffs, New Jersey : Prentice Hall Inc. 1994
4. \_\_\_\_\_. MatLab High Performance Numeric Computation and Visualization Software ver,7.0. The Mathworks, Inc. 1992
5. William and Dawn Tilbury. Control Tutorials for MatLab and Simulink. A Web Based Approach. Addisson Wesley, Inc. 1999.
6. William and Dawn Tilbury. Tool Boox for MatLab and Simulinks. Addisson Wesley, Inc. 2000.
7. <http://www.mathworks.com>