ANALISIS
& IMPLEMENTASI
MENGGUNAKAN
MATLAB

DASAR TEKNIK KENDALI

Media Edukasi INDONESIA

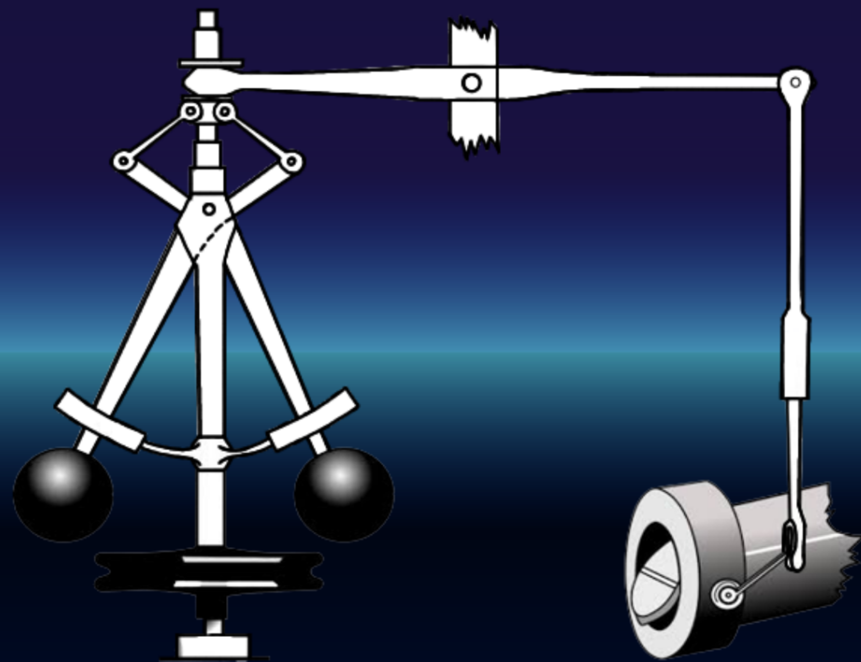
Indonesiamediaedukasi@gmail.com
087871944890
Jalan Lingkar Caringin Cisoka Tangerang
Banten Kode Pos 15730

ISBN 978-623-453-018-6 (PDF)
9 786234 530186

SUHENDAR
ANALISIS & IMPLEMENTASI MENGGUNAKAN MATLAB
DASAR TEKNIK KENDALI

ANALISIS
& IMPLEMENTASI
MENGGUNAKAN
MATLAB

DASAR TEKNIK KENDALI



SUHENDAR

ANALISIS
& IMPLEMENTASI
MENGGUNAKAN
MATLAB

DASAR TEKNIK KENDALI

SUHENDAR

Media Edukasi
INDONESIA

ANALISIS
& IMPLEMENTASI
MENGGUNAKAN
MATLAB

DASAR TEKNIK KENDALI

Penulis	:	SUHENDAR
ISBN	:	978-623-453-018-6
Editor	:	Dema Tesniyadi
Desain Sampul	:	Tim Desain Media Edukasi
Layout	:	Pitriyani

Cetakan Pertama, Oktober 2020
vi + 143 hlm. ; 15 x 23 cm

Penerbit:
Media Edukasi Indonesia (Anggota IKAPI)
Jalan Lingkar Caringin Cisoka Tangerang
Banten Kode Pos 15730
Email: indonesiamediaedukasi@gmail.com
WhatsApp Only: 087871944890

KATA PENGANTAR

Dasar Teknik Kendali merupakan salah satu mata kuliah wajib di Jurusan Teknik Elektro yang ditawarkan kepada mahasiswa semester V. Mata kuliah ini tercantum dalam kurikulum baru hasil revisi dan wajib diambil seluruh mahasiswa Teknik Elektro yang akan memilih konsentrasi Power maupun Non-Power.

Mahasiswa atau siapapun yang baru belajar sistem kendali akan mengalami kesulitan dalam memahaminya. Apa dan bagaimana sistem kendali itu? Dengan demikian, buku ajar ini ditulis dan disajikan dengan harapan dapat memberikan bantuan dalam mempermudah pemahaman mahasiswa, khususnya mahasiswa Jurusan Teknik Elektro Universitas Sultan Ageng Tirtayasa terhadap mata kuliah Dasar Teknik Kendali. Metode lain untuk membantu mempermudah yaitu dengan cara memanfaatkan pemrograman MATLAB UNTUK Analisis dan implementasi mata kuliah ini. Selain itu, pemanfaatan MATLAB untuk sistem multi masukan dan multi keluaran *(MIMO)* pada sistem kendali modern pun dapat ditangani dengan cepat dan akurat. Hal itu terbukti ketika penulis mempraktekan penggunaan MATLAB saat mengajar mata kuliah sistem kendali di kelas dengan bantuan LCD projector. Penggambaran Tempat Kedudukan Akar, diagram BODE, diagram NYQUIST dan diagram lainya diproses oleh MATLAB dapat mempercepat serta mempermudah pemahaman siswa tentang sistem kendali.

Buku ini dapat dimanfaatkan oleh pengajar mata kuliah Dasar Teknik Kendali dan juga para mahasiswa yang membutuhkan referensi guna memahami dan menyelesaikan masalah-masalah yang dihadapi pada sistem kendali. Dalam buku ini diberikan latihan perhitungan dan analisa sistem kendali menggunakan MATLAB termasuk acuan sebagai kunci jawaban terhadap hasil hitungan yang diujicobakan.

Akhirnya dengan segala kerendahan hati penulis menanti saran dan pemberitahuan dari para pembaca demi menyempurnakan dan meningkatkan buku ini menjadi buku referensi atau buku teks di kemudian hari.

Cilegon, Oktober 2020

Penulis

DAFTAR ISI

BAB 1

PENDAHULUAN

A. PERKEMBANGAN MATLAB

MATLAB (kependekan dari *MATrix LABoratory*) pertama kali dibuat di University of Mexico dan Stanford University pada akhir 1970-an yang digunakan pada kuliah teori matriks, aljabar linear dan analisis numerik sebagai pengganti bahasa FORTRAN yang pada waktu itu masih sering digunakan.

Saat ini kemampuan MATLAB jauh melampaui kemampuan *"Matrix Laboratory"* yang semula. Salah satu kemudahan program yang mulai serius dikembangkan sejak tahun 1994 itu adalah pada pemecahan masalah yang dinyatakan dengan notasi matematika biasa.

Secara umum, bahasa ini digunakan untuk matematika dan komputasi, pengembangan algoritma, pemodelan, simulasi, pembuatan prototype, analisis data, visualisasi dan pembuatan aplikasi ber-antarmuka grafis. Dengan MATLAB, permasalahan teknik yang melibatkan matriks dan vektor dapat diselesaikan lebih mudah dan cepat dibanding dengan bahasa C, FORTRAN dan bahasa tingkat tinggi lainnya karena selain formatnya yang interaktif, kita tidak perlu mendeklarasikan elemen dasar basis data array serta ukuran dimensinya.

Setelah versi 4 diperkenalkan, MATLAB mulai menarik perhatian dengan versi 5-nya karena disamping perlengkapan tambahan yang baru, visualisasi dan grafis semakin baik dan cepat. Dan yang lebih penting adalah perbaikan besar pada *toolbox* MATLAB dan SIMULINK. Versi terakhir yang muncul saat ditulisnya buku ini adalah release 14 yang didalamnya termasuk MATLAB 7 dan SIMULINK 6 dengan 26 *upgrade* besar-besaran dan 14 produk baru. Pada release terbaru ini MATLAB makin melebarkan sayapnya ke bidang kedokteran dan farmasi dengan adanya bio informatic toolbox serta tambahan baru :

1. *Instrument Control Toolbox, Data Acquisition Toolbox*
2. *Image Acquisition Toolbox dan Link for Model SIM*, dan lainnya.
3.

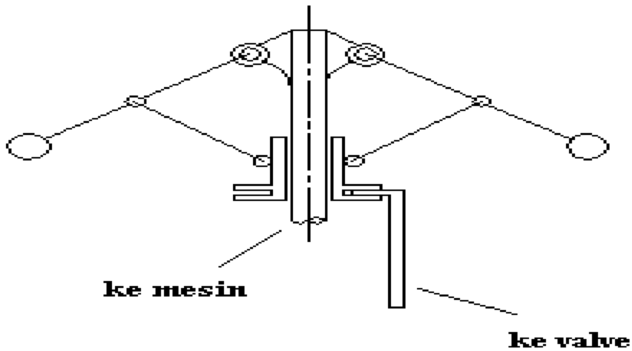
Perangkat keras yang dibutuhkan Release 14 minimal pentium III dengan sistem operasi windows 2000, windows XP, windows NT 4.0 dan LINUX. Tetapi bagi pengguna MATLAB versi di bawahnya jangan berkecil hati karena versi 5 pun sudah cukup memadai untuk menyelesaikan permasalahan pada sistem Kendali dan yang penting adalah program yang telah dibuat bersifat *upward compatible.*

B. PERKEMBANGAN SISTEM KENDALI

Alam semesta sesungguhnya adalah sistem Kendali yang pertama. Gerak planit yang teratur, sirkulasi hujan, sistem biologis dan sistem yang ada di alam merupakan sistem Kendali yang luar biasa hebatnya. Gangguan yang terjadi secara baik direspon dan dikembalikan ke jalurnya yang benar.

Revolusi industri abad XVII menciptakan mesin-mesin yang mempermudah kerja manusia. Tetapi tetap saja dibutuhkan operator yang mengawasi jalannya mesin sehingga memunculkan pemikiran untuk menciptakan mesin tanpa operator yakni mesin yang mampu mengatasi gangguan pada sistem dengan bantuan alat tambahan yang disebut Kendalier.

Kendalier pertama adalah pengontrol kecepatan putar mesin pintal uap dengan bandul berputar yang dibuat oleh James Watt pada abad ke delapan belas. Bandul pada alat yang dinamai governor sentrifugal itu secara alamiah akan naik atau turun mengikuti kecepatan putar mesin. Gerak naik-turun bandul langsung berhubungan dengan poros yang akan membuka dan menutup katup gas. Bila mesin terlalu cepat, bandul naik ke atas dan menutup katup gas sehingga kecepatan mesin turun. Sebaliknya bila mesin terlalu lambat, bandul turun dan membuka katup gas sehingga kecepatan mesin bertambah.



Gambar 1.1 Governor Sentrifugal

Kemudian muncul ilmuwan-ilmuwan yang terjun ke bidang Kendali seperti Nyquist, Minorsky, Rout, Hurwitz, Hazen, Nichols dan sebagainya yang banyak menyumbangkan fikirannya dalam bentuk teorema-teorema yang mempermudah analisa dan hitungan karena pada saat itu komputer belum berkembang. Di tahun 1932 Nyquist mengembangkan suatu prosedur yang relatif sederhana untuk menentukan kestabilan sistem loop tertutup pada basis respon loop terbuka terhadap masukan tunak sinusoida. Metode mereka sering disebut metode klasik yang ciri khasnya adalah *Single Input Single Output (SISO)*.

Kian kompleks mesin/alat membutuhkan pengontrol yang mampu mengontrol sistem *Multi Input Multi Output (MIMO)* dengan persyaratan yang keras terhadap akurasi, kecepatan, berat dan biaya sehingga memunculkan teorema-teorema rumit yang melibatkan matriks berdimensi besar dengan metode yang dinamakan metode ruang keadaan. Diperlukan alat bantu komputer untuk memproses hitungan tersebut dan oleh sebab itu dinamakan juga sistem Kendali modern. Begitu pesatnya perkembangan sistem Kendali modern hingga mampu menciptakan teknik Kendali keseimbangan seperti yang ada pada robot sejenis ASIMO buatan HONDA yang mampu berjalan menuruni tangga.

Gambar 1.2 Robot Asimo (PT. Honda)

C. MANFAAT SISTEM KENDALI

Dalam suatu proses produksi di industri sering dibutuhkan adanya besaran-besaran yang memerlukan kondisi atau persyaratan khusus yang dapat memperlancar tercapainya target proses produksi terebut. Persyaratan khusus ini meliputi ketelitian yang tinggi, nilai yang konstan untuk selang waktu tertentu, nilai yang bervariasi dalam suatu rangkuman tertentu, perbandiangan yang tetap antara dua variable/besaran atau adanya suatu besaran sebagai fungsi dari besaran lainnya.

Semua permasalahan di atas tidak cukup dilakukan hanya dengan melakukan pengukuran saja, tetapi memerlukan suatu cara pengendalian sehingga syarat-syarat tersebut dapat terpenuhi. Dengan alasan seperti ini maka diperkenalkan suatu bentuk konsep pengendalian yang disebut dengan system pengendalian, system kendali, teknik pengontrolan, teknik pengaturan atau system Kendali. Sistem kendali ini ada yang bersifat manual dan otomatis atau lebih dikenal dengan istilah system Kendali otomatik.

Untuk mencapai tujuan tersebut, pada umumnya kita membutuhkan peralatan atau instrumnetasi, sehingga instrumentasi dan kendali merupakan bidang ilmu yang saling menunjang, terutama dalam syarat-syarat khusus seperti dijelakan di atas. Dapat kita perhatkan misalnya alat pemutus dan penghubung arus yang dipasang pada instalasi listrik di rumah-rumah, yang dikenal dengan nama sekering (fuse) atau sekering sejenis dari logam (bimetal) yang disebut dengan circuit breaker (CB).

Jika tehadap sekering atau CB tersebut dialirkan beban arus yang melebihi kapasistas atau kemampuan sekering tersebut maka sakelar penghubungnya dengan otomatis akan turun ke bawah, yang berarti akan memutuskan hubungan arus dari PLN ke rumah dan sebagai akibatnya semua peralatan listrik, lampu dan sebagainya dalam keadaan padam atau off. Circuit beraker ini bekerja berdasarkan banyaknya panas yang dialirkan ke dalamnya yang ditimbulkan oleh listrik yang dialirkannya. Jika arus yang dialirkan terlalu besar (melebihi kapasitasnya) maka sakelar akan terbuka dan arus akan terputus. Kapasitasini dinyatakan dalam amper.

Berdasarkan peristiwa ini dapat kita amati bahwa sebenarnya yang terjadi adalah pengukuran terhadap aliran,membandingkan terhadap kapasitas maksimum, setelah itu melakukan koreksi yaitu dengan cara pemutusan arus. Kejdian inimerupakan suatu contoh dimana proses pengendalian yang terjadi dapat berlangsung secara otomatis.

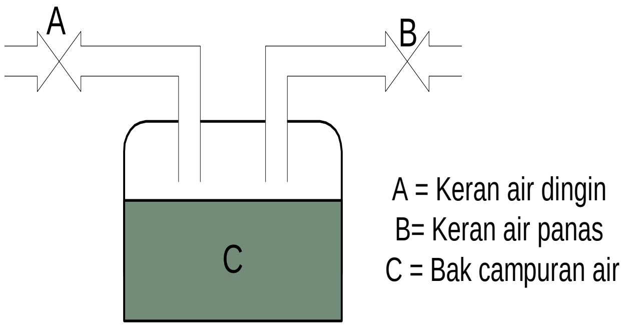
Hal lain yang terjadi dalam kehidupan kita sehari-hari yang dapat dijadikan contoh adalah sewaktu kita mengendarai mobil atau motor. Kecepatan yang diperbolehkan bagi suatu laju kendaraan pada suatu jalan misalnya 60 km/jam, maka kita harus selalu menjaga agar kecepatan kendaraan yang kita kemudikan dijalan tersebut tidak melebihi batas kecepatan yang telah ditentukan. Sebagai alat pencatat/pengukur kecepatan digunakan speedometer yang terdapat pada kendaraan sedangkan sebagai refernsi adalah batas kecepatan 60 km/jam yang terpampang dibahu jalan. Sebagai pem,bandingadalah sipengendara sendiri. Dengan mengamati kecepatan yang dicatat oleh speedometer dan membandingkan serta mengubahnya di bawah kecepatan 60 km/jam jika terjadi penyimpangan, berarti kita telah melakukan pengendalian terhadap kecepatan lajunya kendaraan (mobil/motor) yang kita tumpangi tersebut. Karena pengendalian ini langsung dilakukan oleh manusia maka disebut pengendalian secara manual.

Masih banyak contoh lain yang dapat kita temukan dalam kehidupan sehari-hari, seperti pengendalian level (tinggi) cairan dalam tanki, pengendalian suara radio/TV, pengendalian kecepatan kipas angin sampai pengendalian alat-alat berat (tangga listrik, escalator, alat pengangkat barang) bahkan pengendalian pesawat ruang angkasa dan lain-lain.

Untuk memberikan gambaran yang lebih jelas tetang fungsi dan pentingnya suatu system pengendalian, di bawah ini dapat kita analisis suatu ilustrasi atau gambaran contoh lain, misalnya :

Campuran air dingin dan air panas dalam suatu bak mandi dikehendaki campuaran temperatur tertentu, maka kita dapat mengatur aliran air panas dan aliran air dingin dengan mangalirkannya melalui pipa yang masing-masing dilengkapi dengan keran sebagaimana ditunjukkan pada gambar 1.1 di bawah.

Jika campuran dalam bak tersebut diperlukan untuk mandi, kita cukup mencelupkan jari tangan ke dalam bak dengan merasakan sendiri panas campuran tersebut. Kekurangan atau kelebihan panas dapat kita sesuaikan dengan mengatur keran A dan keran B. Sedangkan jika kita ingin mengetahui dengan lebih pasti berapa suhu campuran air tersebut, perlu dicelupkan termometer ke dalam bak tersebut. Dalam hal ini kita telah melakukan pengukuran.



Gambar 1.3 Pengendalian temperatur campuran air melalui keran

Dari system pengendalian di atas timbul permasalahan lain sebagai berikut :

1. Jika campuran air tersebut tidak hanya diperlukan untuk mandi, tetapi untuk keperluan khusus dengan nilai temperatur yang lebih tepat
2. Jika bak dan alat-alat kendali aliran berada di tempat yang berjauhan
3. Jika aliran ai cukup besar yang tidak apat dikendalikan oleh keran
4. Jika yang dikenadalikan bukan hanya temperatur
5. Jika diinginkan campuran yang konstan atau bervariasi
6. Jika variable-variabel yang dikendalikan dalam jumlah yang banyak

Dengan mengamati kondisi-kondisi seperti itu,kita perlu memberikan analisa sebagai berikut :

1. Untuk pemakaian yang lebih khusus diperlukan pemakaian instrumen-instrumen pengukur temperatur seperti termokopel, termowell, resistance bulb,dan lain-lain dengan dilengkapi alat penunjuk (indicator).
2. Tenaga m,anusia sebagai operatur untuk mengatur aliran keran masih dapat digunakan walaupun untuk tempat yang

saling berjauhan walaupun masih terdapat beberapa esulitan seperti :

a. Akan selalu terjadi keterlambatan waktu antara pemberian perintah terhadap pelaksanaan pengaturan keran
b. Adanya kelemahan ketelitia dan kemampuan operator untuk melakukan pekerjaan dalamjumlah yang banyak
c. Penempatan tenaga operator yang masih cukup sulit terutama ditempat-tempat yang sukar dicapai dan tidak dapat dilihat

Permasalahan yang pokok dari timbulnya masalah-masalah di atas adalah ketepatan (presisi) untuk keadaan-keadaan tertentu seperti diinginkannya nilai variabel yang tetap, pengoperasian yang simultan atau nilai variable dengan toleransi yang cukup kecil.

Dengan demikian melihat kondisi seperti inilah kita membutuhkan konsep pengendalian yang dapat dilakukan secara otomatis sehingga pengendalian-pengendalian harga variable dapat dilakukan dengan lebih teliti dan tepat. Hal ini sangat penting terutama pada industri-industri yang konsumtif dengan mutu produksi serta produktivitas yang tinggi.

D. PENGELOMPOKAN SISTEM KENDALI

Secara sederhana dapat dikatakan bahwa pengertian system kendali adalah suatu proses pengaturan/pengendalian terhadap satu atau beberapa besaran (variable, parameter) sehingga berada pada suatu harga atau dalam suatu rangkuman harga (range) tertentu. Dalam istilah lain disebut juga teknik pengetauran, system pengendalian atau system pengontrolan. Ditinjau dari segi peralatan dan instrumen yang digunakan, system kemdali terdiri dari berbagai susunan komponen fisi yang digunakan untuk mengarahkan aliran energi ke suatu mesin atau proses agar dapat menghasilkan prestasi yang diinginkan.

Tujuan utama dari suati system kendali adalahuntuk mendapatkan optimasi, dalam hal ini dapat diperoleh berdasarkan fungsi dari system kendali itu sendiri, yaitu pengukuran (measurment),membandingkan (comparison), encatatan dan perhitungan (computation) dan perbaikan (correction).

Secara umu system kendali dapat diklompokkan sebagai berikut :

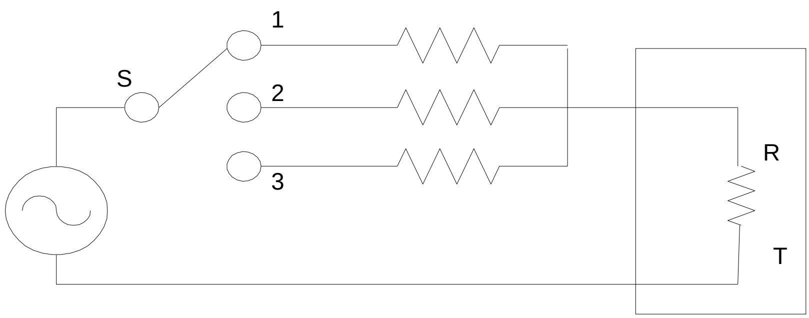
a. Sistem jaringan tertutup (closed loop) dan jaringan terbuka (Open loop)
b. Kontinu (analog) dan diskontinu (diskrit atau digital)
c. Servo dan regulator
d. Menurut sumber penggerak : elektris, pneumatis (udara, angin) hidraulis (cairan) dan mekanis

Di antara keempat jenis pengelompokan tersebut, kelompk (e) pengontrlan secara elektris dan penumatis atau kombinasinya lebih banyak ditemukan dalam industri maupun aplikasi lainnya. Hal ini disebabkan beberapa kelebihan yang diberkannya yaitu pemakaian daya yang lebih kecil, kemampuan untuk melakukan pengendalian jarak jauh, lebih mudah diperoleh dan responnya lebih cepat. Selain dimensinya peralatannya yang dapat dibuat lebih kecil dan lebih sederhana.

Pengendalian secara manual adalah system pengendalian yang dapat dilakukan oleh manusia yang bertindak sebagai operator, sedangkan pengendalian secara otomatis adalah system pengendalian yang dilakukan oleh mesin-mesin/peralatan yang bekerja secara otomatis dengan operasi di bawah pengawasan manusia.

Pengendalian secara manual seperti penyetelan suara radio, televisi, pengaturan cahaya, pengaturan aliran air melalui keran, pengetauran kecepetan kendaraan dan lain-lain. Sedangkan pengendalian secara otomatis banyak ditemui dalam proses industri, pengendalian pesawat, pembangkit tenaga listrik listrik dan lain-lain. Sebagai contoh pengendalian aliran, temperatur dan tekanan dengan menggunakan katup pengatur, pengendalian suhu ruangan oleh thermostat, engendalian daya listrik oleh relay, circuit breaker dan lain-lain.

Sistem kendali jaringan terbuka adalah system kendali yang dilakuakn dimana suatu output tidak memberikan efek terhadap besaran input sehingga variable yang dikendalikan tidak dapat dibandingkan terhadap harga yang diinginkan. Secara jelas dapat dilihat pada contoh gambar 1.2 untuk mengendalikan panas pada pen secara elektris melalui elemen pemanas (hater).



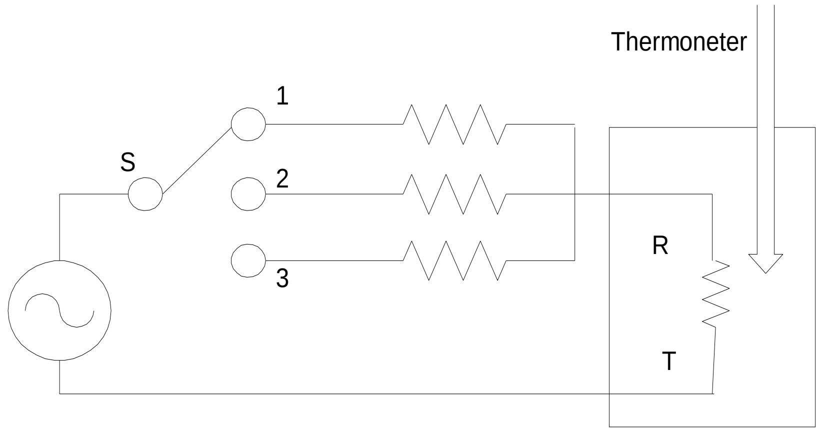
Gambar 1.4 Sistem kendali jaringan (loop) terbuka

T = temperatur oven, R = elemen pemanas

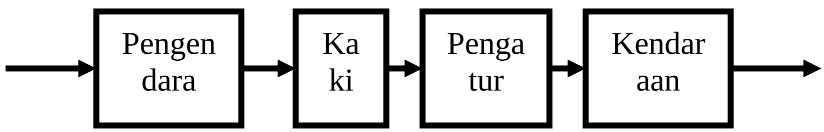
S = selector switch

Dengan system seperti ini berarti banyaknya panas yang dihasilakn dari sumber listrik ke dalam tanur diatur dengan memindahkan sakelar ke posisi 1, 2 dan 3 sehingga tenperatur tanur akan tergantung pada posisi sakelar (s) dan banyaknya panas yang hilang pada dinding-dinding oven tanpa adanya indicator yang memberikan informasi berapa suhu yang terjadi di dalam tanur tersebut.

Sistem kendali jaringan tertutup adalah system pengendalian dimana besaran output dapat memberikan efek terhadap besaran input sehingga besaran yang diKendali dapat dibandingkan terhadap harga yang diinginkan melalui suatu alat pencatat (indicator atau recorder). Secara blok diagram,perbedaan antara system jariangan tertutup dan terbuka dapat dilihat pada gambar blok diagram pengendalian kecepatan sepedar motor oleh seorang pengemudi di bawah ini.



Gambar 1.5 Blok diagram system kendali lup tertutup



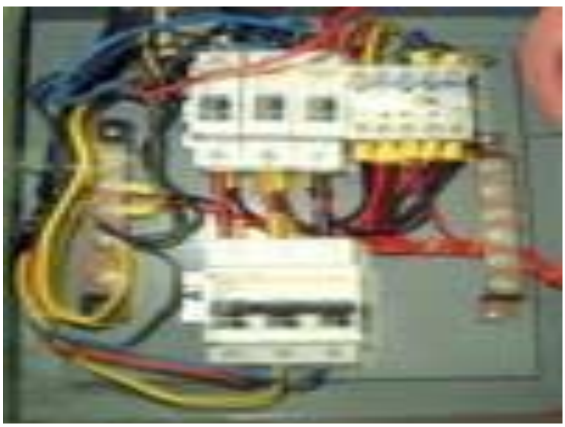
Gambar 1.6 Blok diagram system kendali lup terbuka

Sedangkan system kendali kontinu, biasanya menerapkan system PID, sementara system kendali diskrit dilakukan oleh komonen-komponen diskrit seperti : relay, termostat, level, saklar On-Off, selector switch, floating dan sebagainya.

E. PEMAKAIAN SISTEM KENDALI

Penggunaan system kendali dapat kita temui dalamkehidupan sehari-hari baik dalam pemakaian langsung ataupun tidak langsung. Pemanfaatan system kendali ini dapat dikelompokkan kedalamjenis pemakaian untuk :

a. Pengendalian proses yang meliputi pengendalian temperatur, tekanan, tinggi permukaan cairan viskositas dan lain-lain. Misalnya pada industri kimia, makanan, tekatil, pengilangan minyak dan lain-lain.
b. Pembangkitan tenaga listrik (pengendalian distribusi tenaga listrik)
c. Pengendalian numeric (numerical control, N/C), seperti pengendalian suatu proses operasi yang membutuhkan ketelitian tinggi dalam proses yang berulang-ulang. Misalnya proses pengeboran, pembuatan lobang, tekstil, pengelasan
d. Pengendalian transfortasi seperti elevator, escalator, pesawat terbang, kereta api, conveyor, pengendalian kapal laut, pengendalian melalui servomekanis, dan
e. Pengendalian bidang non teknis seperti ekonomi, sosiologi dan biologi



Gambar 1.7 Contoh pengendalian daya listrik melalui panel daya

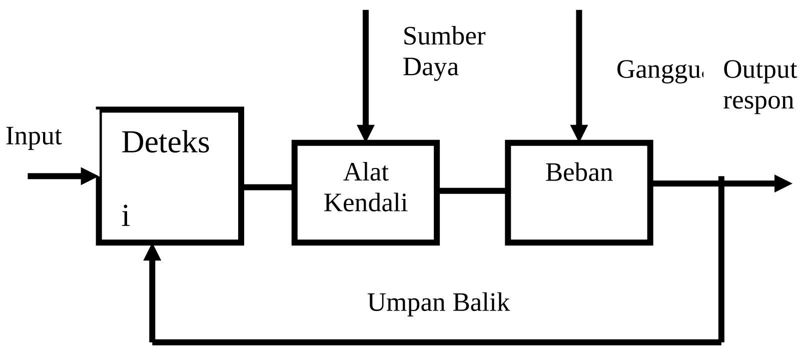
F. ELEMEN SISTEM KENDALI DALAM PRAKTEK

Suatu system kendali dibentuk oleh beberapa unit yang disebut dengan elemen system berupa beberapa komponen.

Secara umum, elemen system dari sebuah system kendali terdiri dari :

a. Elemen masukan (reference input element), berfungsi untuk mengubah besaran yang diKendali menjadi sinyal masukan acuan bagi system kendali tersebut
b. Elemen pengendali (controller), berfungsi untuk memproses kesalahan (error) yang terjadi dan setelah kesalahan tersebut dilewatkan melalui elemen pengendali maka dihasilkan sinyal yang berfungsi sebagai pengontrol proses
c. Elemen system (Proses),berupa proses mekanis, elektris, hidraulis, pneumatis ataupun kombinasi dari beberapa elemen tersebut
d. Elemen umpan balik (feedback element), yaitu bagian system yang mengukur output yang diKendali dan kemudian mengubahnya menjadi sinyal umpan balik
e. Elemen/jalur maju (Forward gain), bagian dai system kendali tanpa umpan balik

Pada umumnya, konfigurasi dari elemen-elemen system kendali tersebut dalam pemakaian sehari-hari dapat ditunjukkan pada blok diagram di bawah ini.



Gambar 1.8 Elmen-elemen system kendali dalam praktek

Dari blok diagram di atas :

1. Beban berupa system fisi yang akan dikendalikan (mekanis, elektris, termis,hidraulis atau pneumatis)

2. Controller merupakan peralatan/rangkaian untuk mengendalikan beban (system), biasanya dapat digabung dengan pengut
3. Respon adalah output yang diperoleh dari alat pencatat
4. Elemen umpan balik menunjukkan atau mengembalikan hasil pencatatan ke detector sehingga bisa dibandingkan terhadap harga yang diinginkan (distel)
5. Error detector adalah suatu aat pendeteksi kesalahan yang menunjukkan selisih antara input dan respons melalui umpan balik

G. SISTEM KENDALI KONVENSIONAL

Proses pengendalian dalan industri selalu berkembang seiring dengan semakin meningkatnya jumlah produksi barang yang harus hasilkan. Mesin-mesin yang digunakan untuk melakukan proses produksi atau produktivitas di industri, pada umumnya digerakkan oleh motor-motor listrik

Pada awalnya proses pengontrolan mesin-mesin industri yang digerakkan oleh motor-motor listrik kebanyakkan menggunakan saklar-saklar biasa yang dioperasikan secara langsung oleh tangan manusia degan proses yang masih manual. Namun proses pengendalian secara manual ini kurang handal dan tidak fleksibel. Sehingga secara bertahap para ahli dan praktisi industri secara terus menerus melakukan percobaan dan penelitian dalam rangka menciptakan suatu system yang dapat melakuklan proses produksi yang lebih efisien, praktis dan otomatis.

Tahap pertama pengendalian proses secara manual (penggunaan saklar-saklar biasa) sudah mulai ditinggalkan dan menggantikannya dengan kontaktor atau saklar elektromagnetik atau relay. Alat ini dapat dioperasikan hanya dengan daya listrik yang relatif rendah untuk mengoperasikan kumparan kerja dari kontaktor atau relay tersebut.

Beberapa keuntungan menggunakan kontaktor sebagai alat pengendali, diantaranya :

a. Dapat dioperasikan untuk pengendalian dari jarak jauh

b. Dapat dioperasikan untuk pengendalian secara otomatis
c. Dapat mengamankan peralatan atau operator jika tegangan sumber hilang atau dating secara tiba-tiba (no voltage release)
d. Dapat dioperasikan dengan mudah (touch oeration)

Pemakaian kontaktor dilengkapi komponen lainnya, seperti push button, time delay relay, thermal overload relay dan alat-alat ukur dapat dirakit dan ditempatkan pada suatu panel atau lemari bagi yang terdistribusi.

Dalam perkembangannya, kendali motor listrik selalu dihadapkan pada hal yang mutakhir dengan alat-alat Kendali yang lebih modern dengan system mekanis, elektris, hidraulis atau kombinasi dari itu. Bahkan pada saat ini, system pengendalian proses di industri sudah banyak menggunakan piranti-piranti elektronik yang dapat diprogram melalui komputer atau sejenisnya seperti penggunaan mikrokontroller dan Programmable Logic Controller (PLC).

SOAL LATIHAN

1. Jelaskan fungsi dan manfaat pemrograman MATLAB!
2. Jelaskan manfaat sistem kendali dalam berbagai bidang aplikasi!
3. Tuliskan kembali kategori dari sistem kendali!
4. Jelaskan kembali fungsi dari masing-masing elemen dalam sistem kendali!
5. Jelaskan perbedaan antara sistem kendali konvensional dengan sistem kendali modern!

DAFTAR PUSTAKA

[1] Anoname. 2004. Matlab Tutorial. Diambil dari: http://www.engin.umich.edu/ group/ctm.

[9] Pakpahan, Sahat. (1994). ***Kontrol Otomatik (Teori Dan Penerapan).*** Erlangga, Jakarta.

[10] Siswosudarmo, Muhammadi., R. Gatot Prio Utomo. 1995. Dasar Sistem Kendali (terj.). Penerbit Universitas Indonesia. Jakarta.

[11] Suhendar, 2005, Programmable Logic Controller dalam Dasar-Dasar Sistem Kendali Motor Listrik Induksi, Graha Ilmu, Yogyakarta

BAB 2

MENGENAL MATLAB

A. PROSES INSTALASI

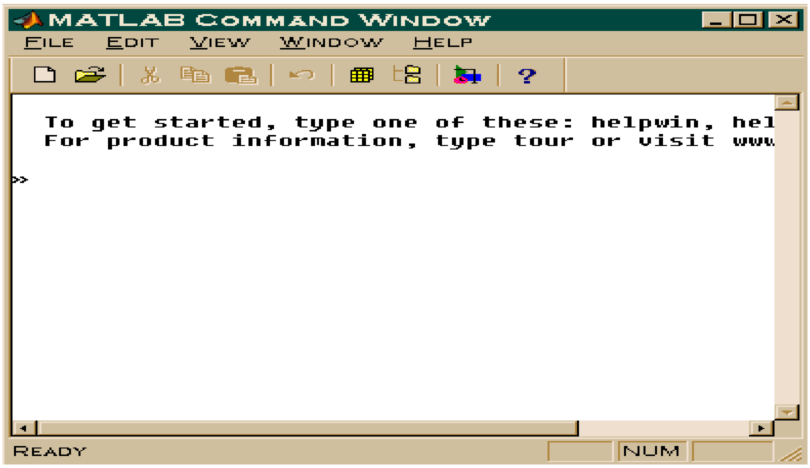
Orang yang biasa menginstall tidak akan mengalami kesulitan dengan menginstal MATLAB kecuali bila sistem operasi yang digunakan LINUX. Untuk pengguna LINUX, tata cara instalasinya dapat dilihat di website http://www.mathworks.com yang merupakan website resmi MATLAB.

Pemakai prosesor pentium III ke atas tidak akan menemui kesulitan dalam menginstall karena mampu mensuport MATLAB R-14 versi 7. Sedangkan untuk pemakai prosesor di bawahnya misalnya pentium I, harus mencari versi MATLAB yang sesuai misalnya MATLAB R-11 versi 5.3 yang sudah memadai karena didalamnya terdapat paket SIMULINK. MATLAB dapat dijalankan pada semua Power Macintosh (dengan mikroprosesor 68020/68030/68040 atau 68881/68882) tetapi tidak bisa jalan pada Macintosh dengan mikroprosesor 68LC040.

Penulis menyarankan menginstall paket MATLAB secara keseluruhan. Tetapi bila kapasitas hardisk-nya kecil, saat memilih peralatan yang akan diinstall, pilihlah yang penting-penting saja dengan mengklik pada kotak pilihan. File help yang berbentuk pdf dan html serta MATLAB untuk server bagi Anda yang sudah mahir mungkin tidak dibutuhkan.

B. COMMAND WINDOW

Saat program MATLAB dijalankan dan setelah muncul secara singkat simbol MATLAB (grafik membran-L) yang Anda lihat sekarang adalah lembar kerja utama kita yang disebut command window. Untuk pengguna Macintosh, tampilannya tidak jauh berbeda.

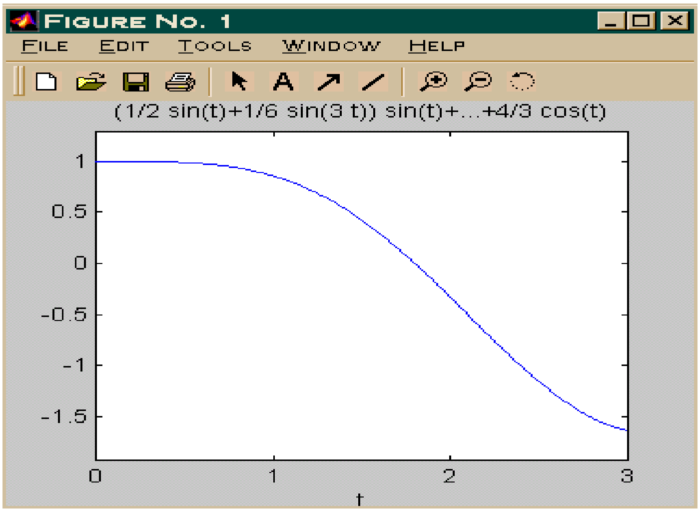


Gambar 2.1 Command Window

Pada lembar kerja *(Workspace)* ini terdapat *toolbox* dan layar putih dengan tanda >> di sebelah kirinya yang siap diberi perintah. Pada lembar kerja ini, MATLAB siap menerima instruksi, baik instruksi langsung seperti pada alat hitung/kalkulator dimana jawabannya langsung diperoleh atau instruksi yang berupa program seperti layaknya bahasa pemrograman tingkat tinggi.

C. GRAFIK/FIGURE

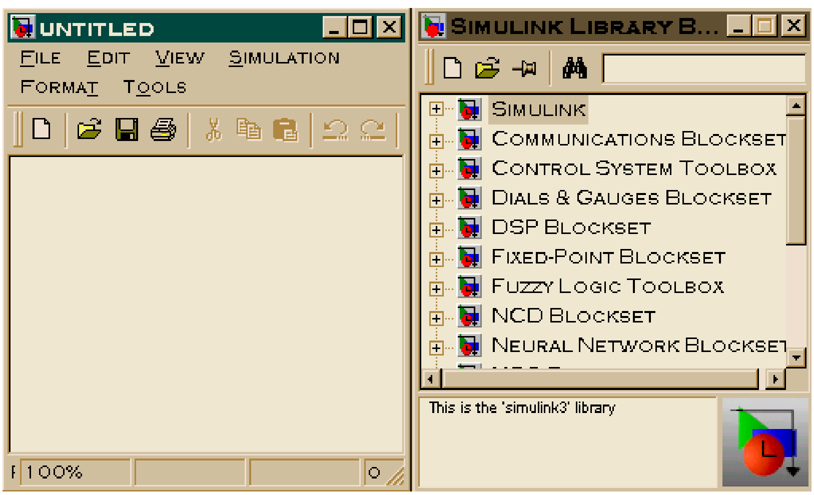
Saat anda memberi instruksi membuat grafik/plot akan muncul jendela grafik/figure pada layar komputer anda. Atau Anda dapat membuka grafik lama yang tersimpan dengan mengklik File – New – Figure. File grafik yang tersimpan berekstensi FIG.



Gambar 2.2 Jendela Grafik

D. JENDELA MODEL/SIMULINK

Disamping command window yang formatnya seperti sistem operasi DOS, pada MATLAB disertakan paket SIMULINK yang berbentuk seperti pada gambar di bawah ini, dapat anda buka dengan mengklik **file** – **new** – **model**.

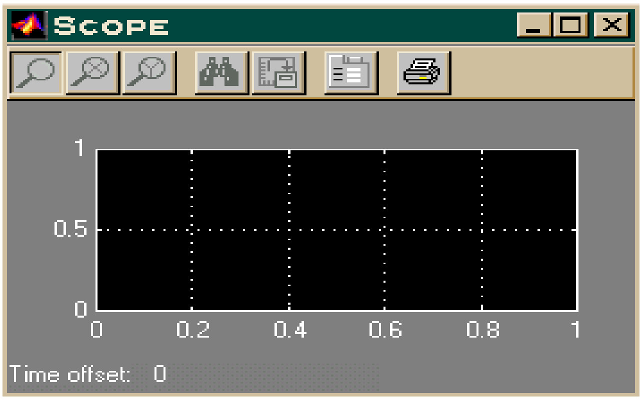


Gambar 2.3 Jendela Model / SIMULINK

Kita dapat menggambar model dengan cara klik dan drug model yang ada di Simulink Library Browser ke jendela Model.

E. JENDELA SIMULASI

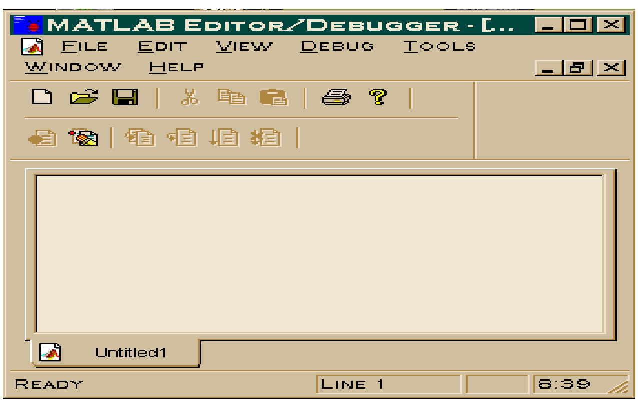
Jendela yang masih merupakan bagian dari SIMULINK ini menggambarkan respon sistem sesuai gambar model.



Gambar 2.4 Jendela SCOPE pada SIMULINK

F. JENDELA M-FILE

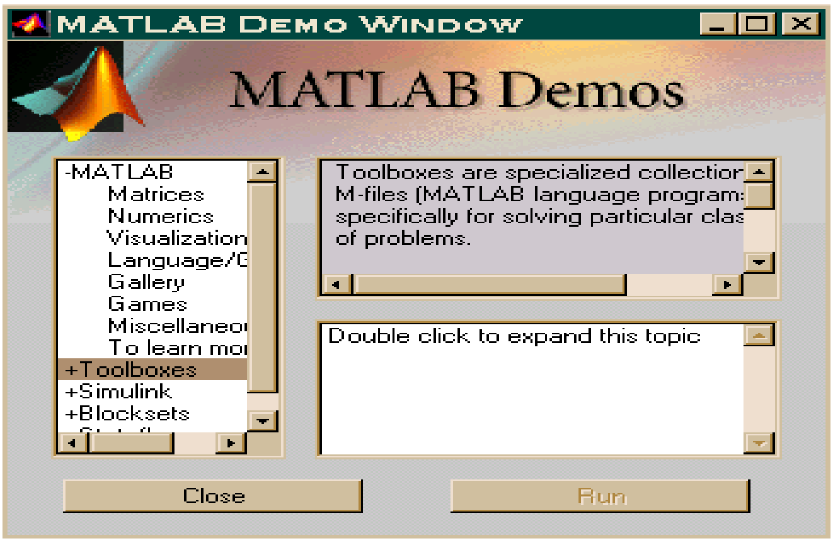
Jendela M-File adalah jendela yang dipergunakan untuk pemrograman. Cara memasukinya adalah dengan klik **file** – **New** – **M-File**. Program disimpan dalam ekstensi m dan cdr.



Gambar 2.5 Jendela M-File

G. MATLAB DEMOS

Dalam menggambar grafik maupun simulasi kadang kala kita lupa atau belum mengenal aturan penulisannya. Sedangkan buku yang secara lengkap membahas seluruh fasilitas yang ada pada MATLAB sangat sulit. Maka Anda tidak perlu khawatir karena di MATLAB tersedia suatu fasilitas pembantu yang dinamakan MATLAB DEMOS. Silahkan Anda ketik demo pada command window, maka akan tampak jendela seperti di bawah ini.



Gambar 2.6 Jendela MATLAB DEMO

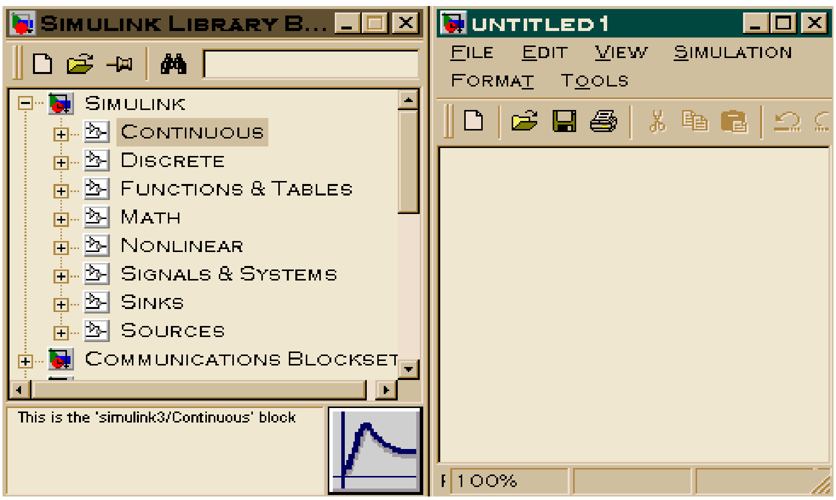
H. MEMBUKA JENDELA SIMULINK

SIMULINK yang masih satu paket dengan MATLAB berguna untuk mempermudah dan memperbaiki visualisasi pemrograman. Sesuai dengan makna katanya, SIMULINK bermanfaat dalam mensimulasikan sistem kontrol yang kita analisa. Versi terakhir SIMULINK saat tulisan ini dibuat adalah versi 6 yang satu paket dengan MATLAB 7 R-14.

SIMULINK dibuka dengan mengklik **File** – **New** – **Model**.

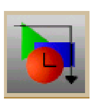
Gambar 2.7 Membuka Jendela SIMULINK

Kemudian muncul dua jendela yaitu Simulink Library Browser dan Jendela Model



Gambar 2.8 Jendela SIMULINK

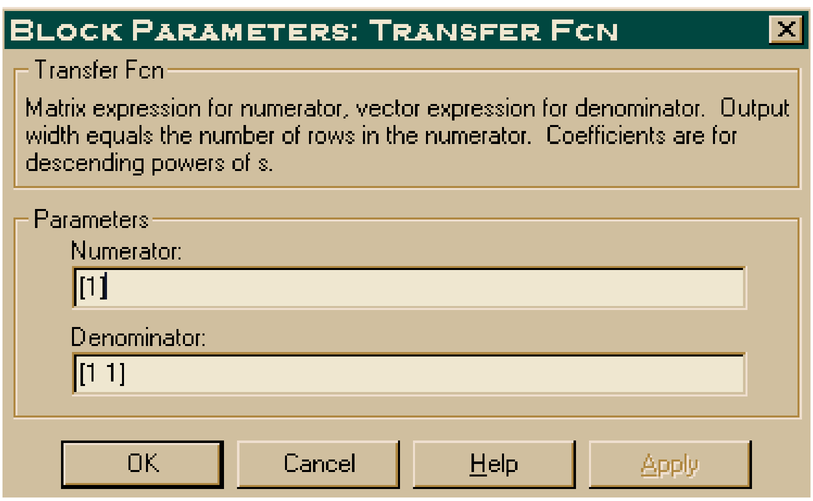
Untuk membuka SIMULINK browser kita dapat juga dengan cara mengklik icon bergambar :



Pada jendela Simulink Library Browser terdapat fasilitas-fasilitas yang berguna sesuai dengan bidang ilmu kita. Kita dapat melihat apa yang tersedia pada tiap-tiap bagian Simulink Browser dengan cara mengklik tanda ‘+’ yang ada di sebelah kiri nama fungsi. Misalnya setelah kita mengklik tanda ‘+’ pada bagian SIMULINK akan muncul continuous, discrete, function and tables dan sebagainya.

I. MENGGAMBAR MODEL

Model kita gambar dalam jendela Model dengan cara klik – drag simbol komponen dari jendela Simulink Library Browser. Misal kita ingin meletakkan diagram blok pada jendela model. Setelah tanda ‘+’ di sebelah kiri kata SIMULINK kita klik lagi tanda ‘+’ di bagian CONTINUOUS yang akan memunculkan DERIVATIVE, INTEGRATOR dan lain-lain termasuk di dalamnya TRANSFER FUNCTION. Pindahkan dengan cara klik – drag ke jendela Model. Untuk mengisi harga fungsi alihnya, double klik pada kotak fungsi alih yang akan memunculkan menu input fungsi alih. Selain dalam format fungsi alih, tersedia pula format lainnya misalnya ruang keadaan dan pole-zero.



Gambar 2.9 Menu Input Fungsi Alih

Silahkan Anda membuka-buka seluruh menu yang tersedia pada Simulink Library Browser agar mengetahui apa saja yang tersedia di SIMULINK.

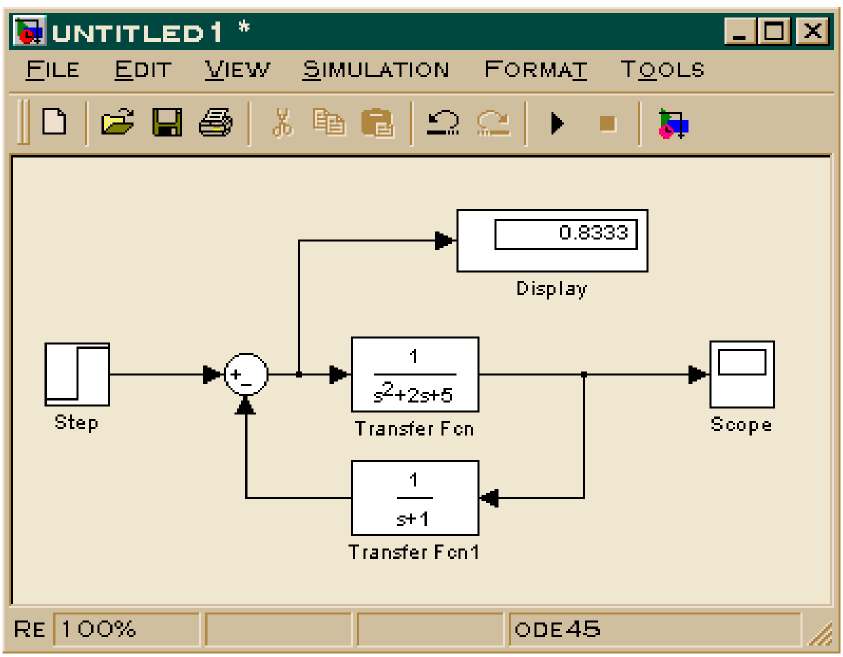
J. PENGGUNAAN SIMULINK DALAM ANALISA RESPON SISTEM KONTROL

Misalnya kita diminta menganalisa respon transien dan kesalahan tunak masukan step sistem lingkar tertutup yang berfungsi alih:

$$G(s) = \frac{1}{s^2 + 2s + 5}$$

dengan umpan balik menggunakan sensor yang berfungsi alih:

$$H(s) = \frac{1}{s+1}.$$

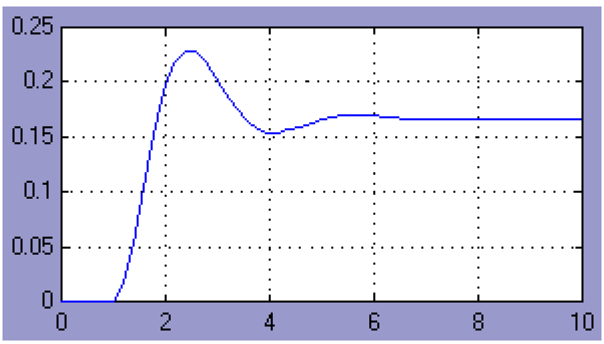


Gambar 2.10 Penggambaran Sistem dengan Jendela Model

Mula-mula buat blok fungsi alih dan umpan balik, double klik untuk memasukkan harganya. Display dihubungkan dengan keluaran titik jumlah guna mengetahui besar kesalahan tunaknya. Scope dipasang di keluaran guna mengetahui respon transien sistem. Baik display maupun scope di klik – drag dari **SIMULINK** – **SINK**. Fungsi alih sensor dibalik dengan cara klik kanan pada **blok** – **FORMAT** – **FLIP BLOCK**. Masukan STEP di ambil dari **SIMULINK** – **SOURCES** – **STEP**.

Dan yang terakhir titik jumlah diambil dari **SIMULINK** – **MATH** – **SUM**. Jangan lupa karena default SUM adalah menjumlah (++) maka agar keluaran mengurangi masukan kita double klik pada titik jumlah lalu diedit agar dihasilkan (+-).

Setelah tombol ►/RUN ditekan diperoleh harga kesalahan tunak pada display sebesar 0,8333 dan untuk mengetahui respon sistem, double klik pada SCOPE setelah itu. Dan klik gambar teropong untuk melihat lebih dekat.



Gambar 2.11 Respon Sistem

Pada grafik di atas terbukti bahwa kesalahan tunak yang terjadi adalah 1 – 0,17 = 0,83 yang sama dengan tampilan pada kotak display di jendela model.

Dari materi pada bab VI Anda tentu dapat mencari harga-harga karakteristik sistem tersebut yaitu td = 1,5 detik, tr = 1,9 detik, tp = 2,5 detik, Mp = 0,045 dan ts = 4 detik. Serta sistem tersebut stabil karena keluaran tidak terus membesar.

SOAL LATIHAN

1. Jelaskan proses dan fitur-fitur dalam MATLAB yang dapat digunakan untuk mengimplementasikan teknik kendali!
2. Gambarkan grafik respon sistem dari persamaan berikut:

$$G(s) = \frac{1}{s^3 + 2s^2 + 3s + 8}$$

DAFTAR PUSTAKA

[1] Anoname. 2004. Matlab Tutorial. Diambil dari: http://www.engin.umich.edu/ group/ctm.

[5] Eko Mursito Budi, Manase Sitorus, Estiyanti Ekawati. 2006. Simulator Untuk Pengajaran Sistem Kontrol. Kelompok Keahlian Instrumentasi & Kontrol ITB, Bandung

[7] Hanselman, Duane., Bruce Littlefield. 1997. Matlab, Bahasa Komputasi Teknis (terj.). Penerbit Andi. Yogyakarta.

BAB 3

MODEL MATEMATIKA MATLAB

Fungsi matematika adalah fungsi umum dan terpenting dari MATLAB. Bahkan seorang profesor yang sekaligus pengguna MATLAB mengatakan bahwa alasan mengapa MATLAB sangat berguna untuk pemrosesan sinyal adalah bukan karena MATLAB dirancang secara khusus untuk memproses sinyal, melainkan karena MATLAB dirancang untuk matematika.

A. DASAR MATEMATIKA

Seperti kalkulator scientific kita tinggal mengetik masalah matematik di command window dengan format seperti matematika biasa. MATLAB mempunyai berbagai fungsi umum yang penting untuk matematika, teknik dan ilmu pengetahuan lain. Sebagai tambahan atas fungsi-fungsi umum tersebut, MATLAB menyediakan ratusan fungsi khusus dan algoritma yang berguna untuk menyelesaikan permasalahan tertentu.. Berikut ini satu contoh perhitungan logaritma.

```
» x=sqrt(3)/2
        x =
            0.8660

» y=acos(x)
        y =
            0.5236
```

```
» y_deg=y*180/pi

    y_deg =

      30.0000
```

Hitungan terakhir di atas adalah jika kita ingin melihat sudut y dalam derajat dikarenakan MATLAB hanya bekerja dalam radian. Pada command window MATLAB saat ini terdapat tiga variabel, yang dapat kita lihat dengan mengetik instruksi who.

```
» who

    Your variables are:

    x        y        y_deg
```

Bila kita ingin menghapus satu atau lebih variabel yang ada di MATLAB maka instruksi yang digunakan adalah clear. MATLAB juga menyediakan panduan help. Ketik help lalu diketik instruksi yang akan dicari informasinya.

```
» help clear

    CLEAR  Clear variables and functions from memory.
    CLEAR removes all variables from the workspace.
    CLEAR VARIABLES does the same thing.
    CLEAR GLOBAL removes all global variables.
    CLEAR FUNCTIONS removes all compiled M-functions.
    CLEAR MEX removes all links to MEX-files and all M-
    functions.
    CLEAR ALL removes all variables, globals, functions and
    MEX links.
    CLEAR CLASSES is the same as CLEAR ALL except
    that class definitions
    are also cleared.  If any objects exist outside the
    workspace (say in userdata or persistent in a locked m-
    file) a warning will be issued and the class definition will
    not be cleared.  CLEAR CLASSES must be used if the
    number or names of fields in a class are changed
```

Tabel 3.1 Fungsi-fungsi Umum

abs(x)	harga mutlak atau besarnya bilangan kompleks
acos(x)	invers kosinus
acosh(x)	invers kosinus hiperbolik
angle(x)	sudut suatu
asin(x)	invers sinus
asinh(x)	invers sinus hiperbolik
atan(x)	invers tangent
atan2(x)	invers tangen untuk empat kuadran
atanh(x)	invers tangen hiperbolik
ceil(x)	pembulatan ke arah plus tak berhingga
conj(x)	konjugat bilangan kompleks
cos(x)	Kosinus
cosh(x)	kosinus hiperbolik
exp(x)	eksponensial: e^x
fix(x)	pembulatan ke arah nol
floor(x)	pembulatan ke arah minus tak berhingga
gcd(x,y)	faktor persekutuan terbesar bilangan bulat x dan y
imag(x)	bagian imajiner suatu bilangan kompleks
lcm(x,y)	kelipatan persekutuan terkecil bilangan bulat x dan y
log(x)	logaritma natural
log10(x)	logaritma biasa
real(x)	bagian real suatu bilangan kompleks
rem(x,y)	sisa pembagian: rem(x,y) menghasilkan sisa pembagian x/y
round(x)	pembulatan ke arah bilangan bulat terdekat
sign(x)	menghasilkan tanda dari argumen, misalnya: sign(1.2)=1, sign(-23.4)=-1, sign(0)=0
sin(x)	Sinus
sinh(x)	sinus hiperbolik

sqrt(x)	akar kuadrat
tan(x)	Tangent

B. PERSAMAAN MATEMATIKA

Sebelumnya Anda diperkenalkan bagaimana MATLAB dapat dipergunakan seperti kalkulator canggih yang dapat diprogram. Akan tetapi kalkulator biasa tidak dapat bekerja untuk memanipulasi ekspresi matematika tanpa menggunakan bilangan. Maka di sini akan dibahas aplikasi MATLAB untuk persamaan-persamaan matematika yang sering dijumpai dalam dunia teknik, khususnya pada sistem kontrol.

1. Membuat Variabel Simbolik

Berikut adalah contoh keunggulan MATLAB dibanding kalkulator scientific biasa karena dapat menghitung simbol-simbol yang bukan angka. Caranya adalah dengan merubah terlebih dahulu variabel numerik menjadi variabel simbolik.

```
» x=sym('x');
» diff(sin(x))
        ans =
        cos(x)
```

Perhatikan format penulisan statemen pertama yang mendefinisikan x sebagai simbol sehingga MATLAB dapat mendiferensialkan sinus. Statemen pertama bisa juga ditulis dalam bentuk: syms x. Untuk mengetahui suatu variabel angka atau simbol dengan cara sebagai berikut:

```
» class(x)
        ans =
        sym
```

2. Persamaan Aljabar

Berikut ini contoh mengoperasikan persamaan aljabar biasa f = $5x^3 + x^2 + 4$ dan g = $x^2 + x - 2$

```
» syms x
» f=5*x^3+x^2+4       % mendefinisikan ekspresi simbolik f dan g
        f =
        5*x^3+x^2+4

» g=x^2+x-2
        g =
        x^2+x-2
» f^(3*x)
        ans =
        (5*x^3+x^2+4)^(3*x)
» h=subs(ans,x,1)
        h =
        1000
```

Instruksi terakhir mengganti/mensubstitusi variabel x pada persamaan ans (kependekan dari *answer*/jawaban) dengan 1 dan diperoleh hasil 1000.

Instruksi double adalah kebalikan dari sym, yaitu merubah simbol menjadi numerik. Untuk mengintegralkan atau mendiferensialkan fungsi f dan g dengan cara sebagai berikut:

```
» diff(f)
        ans =
        15*x^2+2*x
» int(g)
        ans =
        1/3*x^3+1/2*x^2-2*x
» pretty(ans)
```

$1/3.\ x^3 + 1/2.\ x^2 - 2x$

Perintah terakhir adalah untuk memperindah bentuk persamaan jawaban sehinggi memiliki bentuk seperti penulisan matematika biasa.

3. Persamaan Diferensial

Saat kita memodelkan suatu sistem fisis, yang pertama kita peroleh adalah persamaan diferensial. MATLAB membantu menyelesaikan persamaan diferensial yang merupakan kendala bagi orang yang baru belajar sistem kontrol.

Cara penulisan diferensial pada MATLAB adalah dengan huruf kapital D untuk orde 1, D2 untuk orde 2, D3 untuk orde 3 dan seterusnya. Untuk lebih jelasnya misalnya kita diminta untuk menyelesaikan persamaan diferensial dengan harga awal:

$$\frac{d^2y}{dt^2} = \cos(2t) - y \quad \frac{dy}{dt}(0) = 0 \quad y(0) = 1$$

» y=dsolve('D2y=cos(2*t)-y, Dy(0)=0, y(0)=1')

y =

(1/2*sin(t)+1/6*sin(3*t))*sin(t)+(1/6*cos(3*t)-1/2*cos(t))*cos(t)+4/3*cos(t)

» pretty(y)

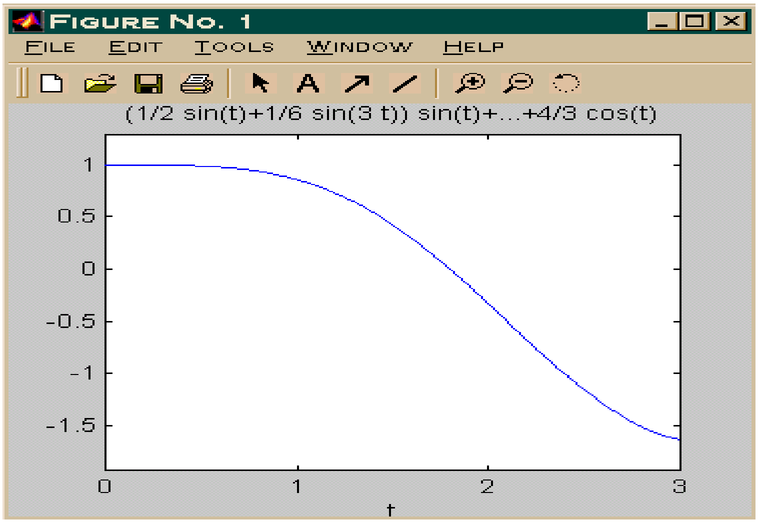
(1/2 sin(t) + 1/6 sin(3 t)) sin(t) + (1/6 cos(3 t) - 1/2 cos(t)) cos(t) + 4/3 cos(t)

C. MEMBUAT GRAFIK

Bila ingin melihat bentuk grafik keluaran persamaan diferensial di atas yang berdimensi dua dapat diketik :

» ezplot(y,[0 3])

Dan kemudian muncul lembar grafik MATLAB yang berbentuk sebagai berikut:



Gambar 3.1 Grafik Persamaan

$$\frac{d^2y}{dt^2} = \cos(2t) - y \quad \frac{dy}{dt}(0) = 0 \quad y(0) = 1$$

Menggambar grafik 3D secara manual amat sulit, tetapi MATLAB mampu menggambarkannya dengan cepat dan akurat. Untuk menjadi ahli dalam menggambar grafik, Anda dapat melihat menu help dengan cara mengetik pada command window.

» help graph3d

Three dimensional graphs.

Elementary 3-D plots.

plot3 - Plot lines and points in 3-D space.
mesh - 3-D mesh surface.
surf - 3-D colored surface.
fill3 - Filled 3-D polygons.

Color control.

colormap	- Color look-up table.
caxis	- Pseudocolor axis scaling.
shading	- Color shading mode.
hidden	- Mesh hidden line removal mode.
brighten	- Brighten or darken color map.
colordef	- Set color defaults.
graymon	- Set graphics defaults for gray-scale monitors.

Lighting.

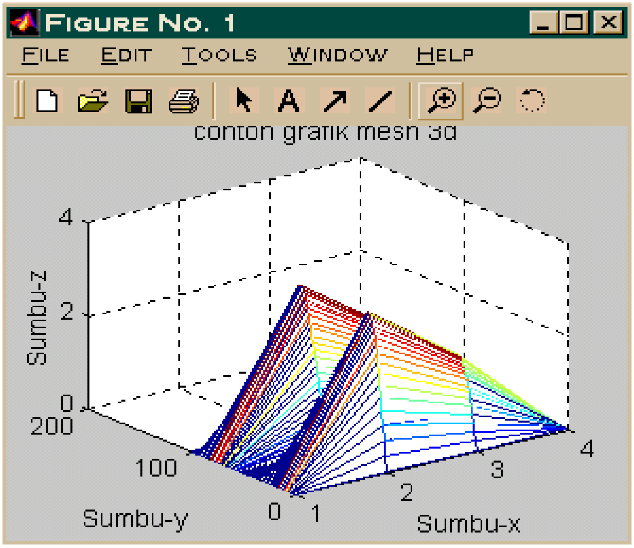
surfl	- 3-D shaded surface with lighting.
lighting	- Lighting mode.
material	- Material reflectance mode.
specular	- Specular reflectance.
diffuse	- Diffuse reflectance.
surfnorm	- Surface normals.

Sedangkan untuk tatacara penulisannya ada baiknya Anda belajar sendiri dengan cara mengetik demo pada command window hingga muncul jendela MATLAB DEMOS.

Berikut ini adalah contoh penggunaan salah satu fasilitas yang ada di grapik tiga dimensi MATLAB yaitu mesh/jala.

```
» [X,Y]=meshgrid(-pi:pi/50,-pi:pi/50:pi);
» Z=X.*sin(X).*Y.*sin(Y);
» mesh(Z)
» title(‘contoh grafik mesh 3d’)
» xlabel('Sumbu-x'),ylabel('Sumbu-y'),zlabel('Sumbu-z');
```

Saat mesh(Z) di-enter akan muncul grafik sesuai fungsi yang ditulis di command window. Untuk memberi penjelasan nama judul dengan instruksi title sedangkan xlabel, ylabel dan zlabel memberi keterangan pada tiap sumbu.



Gambar 2.2 Grafik Mesh/Jala 3D

D. ARRAY DAN MATRIKS

Operasi dengan satu besaran skalar saja memang merupakan dasar matematika. Namun jika dalam sesaat Anda ingin melakukan operasi yang sama pada beberapa bilangan, perulangan operasi skalar akan menghabiskan waktu dan tentu saja tidak praktis.

1. Penulisan Array

Array sesungguhnya adalah matris vektor yang mempunyai satu baris atau kolom saja. Misalnya kita ingin mencari harga cosinus sudut dari 0 sampai π.

```
» x=[0 0.1*pi 0.2*pi 0.3*pi 0.4*pi 0.5*pi]
    x =
    0    0.3142    0.6283    0.9425    1.2566    1.5708
» y=cos(x)
    y =
    1.0000    0.9511    0.8090    0.5878    0.3090    0.0000
```

Perhatikan tata cara penulisannya! Setelah nama variabel dan tanda sama dengan array ditulis di antara kurung [ ]. Antar sel dipisah dengan spasi (boleh lebih dari satu spasi). Di antara tanda kurung [ ] dapat juga kita sisipi data dari Microsoft Excel® dengan cara copy-paste biasa.

2. Alamat Array

Array contoh sebelumnya adalah matriks 1x6 atau dengan kata lain matriks dengan satu baris dan enam kolom yang lebih dikenal dengan nama vektor. Dalam MATLAB elemen-elemen array diakses dengan *subscript*. Misalnya x(1) berarti elemen pertama, x(2) elemen kedua dan seterusnya.

```
» x(1)
        ans =
        0
» x(2)
        ans =
        0.3142
```

Untuk mengakses suatu blok elemen, MATLAB menyediakan notasi kolom.

```
» x(1:4)
        ans =
        0    0.3142    0.6283    0.9425
```

3. Membentuk Array

Untuk jumlah elemen array yang banyak, tidak praktis dengan cara mengetikan satu persatu array-nya. Ada dua cara memasukan nilai x dengan notasi kolom. Cara pertama:

```
» x=(0:0.1:0.5)*pi
        x =
        0    0.3142    0.6283    0.9425    1.2566    1.5708
```

Dengan cara pertama, notasi kolom (0:0.1:0.5) menghasilkan array yang dimulai dengan 0, meningkat setiap 0,1 dan berhenti pada 0,5 (Matlab menggunakan notasi matematika eropa dimana titik berarti koma untuk notasi kita). Sedangkan cara kedua:

```
» x=linspace(0,0.5*pi,6)
        x =
        0    0.3142    0.6283    0.9425    1.2566    1.5708
```

Dengan cara kedua, notasi kolom (0,0.5*pi,6) menghasilkan array yang dimulai dengan 0, diakhiri 0,5π dengan jumlah elemen enam. Cara di atas sering digunakan di MATLAB dimana jarak antar elemen linear.

Untuk yang tidak linear misalnya logaritmik, yang biasa dipakai pada diagram BODE, adalah sebegai berikut:

```
» logspace(0,2,6)
        ans =
        1.0000    2.5119    6.3096   15.8489   39.8107  100.0000
```

Maksud notasi kolom (0,2,6) adalah suatu array yang dimulai 10^0, diakhiri 10^2 dengan jumlah elemen enam.

Tabel 3.2 Instruksi Pada Array

Pengalamatan Array	
x=[2 2*pi sqrt(2) 2-3j]	menciptakan vektor baris x yang memuat elemen-elemen yang diberikan
x=awal: akhir	membuat vektor baris x dimulai dengan awal, kenaikan satu, diakhiri pada atau sebelum akhir
x=awal:kenaikan:akhir	membuat vektor baris x diawali dengan awal, kenaikan sebesar kenaikan, diakhiri pada atau sebelum akhir
x=linspace(awal,akhir,n)	menciptakan vektor baris x diawali dengan awal, berakhir dengan akhir, mempunyai n elemen

x=logspace(awal,akhir,n)	menciptakan vektor kolom dengan elemen berjarak logaritmis dimulai dengan 10^{awal}, diakhiri dengan 10^{akhir}, mempunyai n elemen
Pengalamatan Array	
A(r,c)	mengalamti subarray dalam A dengan indeks baris yang diingini dalam r dan kolom yang diingini dalam c
A(r,:)	mengalamati subarray dalam A dengan indeks baris yang dialamati dalam r dan semua kolom diambil
A(:,r)	mengalamati subarray dalam A dengan semua baris diambil dan indeks kolom yang diingini dalam r
A(:)	mengalamati semua elemen dalam A sebagai vektor kolom diambil kolom per kolom
A(i)	mengalamati subarray dalam A dengan indeks tunggal i, dianggap A merupakan vektor kolom a(:)
A(x)	mengalamati subarray dalam A dengan array logika x; x harus mempunyai ukuran yang sama dengan A
Pencarian Array	
i=find(x)	menghasilkan indeks dari array x dimana elemen-elemennya tidak nol
[r,c]=find(X)	menghasilkan indeks baris dan kolom dari array X dimana elemen-elemennya tidak nol

4. Membentuk Matriks

Matriks adalah perluasan dari array/vektor. Cara penulisannya hampir sama dengan array hanya saja untuk memisahkan antara baris satu dengan lainnya digunakan titik koma (;) atau menekan enter.

Berikut ini adalah operasi dasar pada matriks, yaitu matriks transpose, determinan dan invers.

```
» A=[1 2 3; 4 5 6; 8 9 0]
  A =
      1    2    3
      4    5    6
      8    9    0
» B=A'                  % transpose matriks A
  B =
      1    4    8
      2    5    9
      3    6    0
» C=det(A)              % determinan matriks A
  C =
     30
» D=inv(A)              % invers matriks A
  D =
    -1.8000    0.9000   -0.1000
     1.6000   -0.8000    0.2000
    -0.1333    0.2333   -0.1000
```

5. Memanipulasi Matriks

Sekali suatu matriks dibentuk, MATLAB menyediakan cara mudah untuk menyisipkan, mengambil dan mengatur kembali sebagian dari matriks tersebut dengan mengidentifikasi *subscript* yang berkaitan. Penguasaan akan hal ini merupakan kunci untuk menggunakan MATLAB secara efisien. Berikut ini contoh memanipulasi array dan matriks.

```
» A=[1 2 3; 4 5 6; 7 8 9]
  A =
      1    2    3
      4    5    6
      7    8    9
» A(3,3)=0              % mengubah elemen pada baris3 kolom 3
  menjadi nol
  A =
      1    2    3
      4    5    6
```

```
    7   8   0
» A(:,3)=4          % mengubah seluruh elemen kolom 3
  menjadi 4
  A =
    1   2   4
    4   5   4
    7   8   4
» B=A(3:-1:1,1:3)   % matriks B dibuat dengan urutan baris A
  yang terbalik
  B =
    7   8   4
    4   5   4
    1   2   4
» C=[A B(:,[1 3])]  % membuat C dgn A yg disisipi kolom 1
  & 3 matriks B
  C =
    1   2   4   7   4
    4   5   4   4   4
    7   8   4   1   4
```

6. Karakteristik Matriks

Karakteristik matriks menggambarkan perihal suatu matriks yaitu ukuran matriks, rank dan nilai eigen.

```
» C
  C =
    1   2   4   7   4
    4   5   4   4   4
    7   8   4   1   4
» size(C)                  % mengetahui ukuran matriks C
  ans =
    3   5
```

Untuk mengetahui seluruh ukuran matriks yang telah dibuat pada command window, dengan cara sebagai berikut:

» whos

Name	Size	Bytes	Class

```
A        3x3     72  double array
B        3x3     72  double array
C        3x5    120  double array
D        3x4     96  double array
E        3x3     72  double array
F        3x4     96  double array
ans      1x2     16  double array
```

Grand total is 68 elements using 544 bytes

Nilai eigen λ adalah salah satu karakteristik matriks yang sering dijumpai pada sistem kontrol. Misal kita ingin mencari nilai λ matriks A.

```
» A
  A =
        1    2    3
        4    5    6
        7    8    9
» eig(A)
  ans =
       16.1168
       -1.1168
       -0.0000
```

Rank (peringkat) matriks secara teori dicari dengan mengubah matriks menjadi bentuk eselon baris tereduksi. Misal kita ingin mencari rank matriks G:

```
» G
  G =
    1    2    4
    2    4    8
    1    2    3

» [H,pivot]=rref(G)
  H =
    1    2    0
    0    0    1
    0    0    0
  pivot =
```

```
        1     3
»  length(pivot)
    ans =
        2
```

Instruksi terakhir menghasilkan rank matriks G =2. Atau kita dapat dengan cepat menghitung rank matriks tanpa melakukan pivot (fungsi rref) dengan bantuan fungsi-fungsi khusus yang telah disediakan MATLAB.

```
»  rank(G)
    ans =
        2
```

Berikut ini tabel yang berisi fungsi-fungsi yang berkaitan dengan matriks.

Tabel 3.3 Instruksi MATLAB pada Matriks

Fungsi-fungsi Matriks	
balance(A)	penskalaan untuk meningkatkan ketepatan nilai eigen
cdf2rdf(A)	bentuk kompleks diagonal ke bentuk real diagonal
chol(A)	faktorisasi Cholesky
cholinc(A,droptol)	faktorisasi Cholesky tidak lengkap
cond(A)	bilangan kondisi matriks
condest(A)	perkiraan bilangan kondisi matriks 1-norm
condeig(A)	kondisi w.r.t nilai eigen
det(A)	Determinan
d=eig(A), [V,D]=eig(A)	nilai eigen dan vektor eigen
eigs(A)	sedikit nilai eigen dan vektor eigen
expm(A)	pemangkatan matriks
expm1(A)	implementasi M-file dari expm
expm2(A)	pemangkatan matriks menggunakan deret Taylor
expm3(A)	pemangkatan matriks menggunakan nilai dan vektor eigen
funm(A,’fun)	menghitung fungsi matriks umum

hess(A)	bentuk Hessenberg
inv(A)	invers matriks
logm(A)	logaritma matriks
lscov(A,b,V)	kuadrat terkecil dengan kovarian diketahui
lu(A)	faktor dari eliminasi Gauss
luinc(A,droptol)	faktorisasi LU tidak lengkap
nnls(A,b)	kuadrat terkecil non-negatif
norm(A)	normal matriks dan vector
norm(A,1)	normal-1
norm(A,2)	normal-2 Euclidan
norm(A,inf)	ketidak-terbatasan
norm(A,p)	normal-P (vektor saja)
norm(A,’fro’)	normal-F
normest(A)	memperkirakan normal-2 untuk matriks jarang besar
null(A)	bidang nol
orth(A)	Ortogonalisasi
pinv(A)	Pseudoinverse
planerot(A)	rotasi bidang Given
poly(A)	karakteristik polynomial
polyeig(A1,A2,...)	menyelesaikan masalah nilai eigen polynomial
polyvalm(A)	mengevaluasi matriks polynomial
qr(A)	dekomposisi ortogonal-triangular
qrdelete(Q,R,j)	menghapus kolom dari faktorisasi QR
qrinsert(Q,R,j,x)	menyisipkan kolom dalam faktorisasi QR
qz(A,B)	nilai eigen umum
rank(A)	jumlah baris atau kolom bebas linear
rcond(A)	perkiraan komdisi resiprok
rref(A)	bentuk eselon baris tereduksi
rsf2csf(U,T)	bentuk real Schur ke bentuk kompleks Schur
sqrtm(A)	akar kuadrat matriks
subspace(A,B)	sudut antara dua bidang
svd(A)	dekomposisi nilai singular
svds(A,K)	sedikit nilai-nilai singular
trace(A)	jumlah elemen-elemen diagonal

7. Operasi Pada Matriks

MATLAB mampu melaksanakan operasi penjumlahan, pengurangan, perkalian, pembagian dan pemangkatan hanya bila kondisi terpenuhi. Tidak ada salahnya pembaca membuka kembali teori aljabar linear matriks. Penjumlahan dan pengurangan matriks membutuhkan orde yang sama sedangkan perkalian boleh dilaksanakan jika kolom matriks pertama sama dengan jumlah baris matriks kedua.

```
» A=[1 2 3 4; 5 6 7 8; 9 10 11 12]
  A =
       1     2     3     4
       5     6     7     8
       9    10    11    12

» B=[1 1 1 1; 2 2 2 2; 3 3 3 3]
  B =
       1     1     1     1
       2     2     2     2
       3     3     3     3

» C=[1 2 3; 4 5 6; 7 8 9; 10 11 12]
  C =
       1     2     3
       4     5     6
       7     8     9
      10    11    12

» D=A+B
  D =
       2     3     4     5
       7     8     9    10
      12    13    14    15

» E=A*C
  E =
      70    80    90
     158   184   210
     246   288   330
```

```
» F=A.^B
  F =
        1      2      3      4
        25     36     49     64
        729    1000   1331   1728
```

Matriks D pada instruksi di atas adalah hasil operasi penjumlahan, matriks E hasil perkaliah sedangkan matriks F hasil pemangkatan dengan matriks B.

8. Persamaan Linear

Salah satu masalah pertama pada MATLAB adalah menyelesaikan persamaan aljabar linear dari sekumpulan persamaan linear, misalnya:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 8 & 9 & 0 \end{bmatrix} \cdot \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$$

A . X = B

Jika terdapat penyelesaian, maka terdapat beberapa metode untuk menemukan penyelesaian itu, seperti eliminasi Gauss – Jordan, faktorisasi LU, atau penggunaan langsung dari A^{-1}. Secara analitis, penyelesaian ditulis sebagai X= A^{-1}.B.

MATLAB menyelesaikan persamaan di atas dengan cepat dan akurat:

```
» A=[1 2 3; 4 5 6; 8 9 0]
  A =
     1   2   3
     4   5   6
     8   9   0
```

```
» B=[100; 200; 300]
  B =
```

```
      100
      200
      300
» X=inv(A)*B
  X =
    -30.0000
     60.0000
      3.3333
```

Mungkin kita masih dapat menghitung penyelesaian persamaan di atas secara manual, tetapi jika matriks yang akan dihitung berukuran besar, misalnya 1000x1000, tentu saja akan mengalami kesulitan.

E. PEMROGRAMAN MATLAB

Untuk membuat program dalam bahasa MATLAB pertama-tama bukalah jendela M-File dengan cara seperti yang telah dijelaskan di bab II. Ada dua jenis m-file yaitu *script file* dan *function file*. *Script file* tidak menggunakan argumen input atau mengembalikan argumen output. *Function file* dapat menggunakan argumen input atau menghasilkan argumen output.

1. Membuat *Script File*

Langkah pertama membuat program m-file adalah dengan mengetik instruksi pada jendela MATLAB Editor/ Debugger. Misalnya kita ingin membuat program *script file* yang menggambarkan perbandingan respon dua buah sistem :

$$y1 = 1 - e^{-0,5t}\sin t$$
$$y2 = 1 - e^{-1t}\sin t$$

Ketikan pada jendela MATLAB Editor/Debugger program sebagai berikut:

%program membandingkan respon

```
t=0:.1:15.;
y=1.-(exp(-.5.*t).*sin(t));
plot(t,y),grid
title('perbandingan respon sistem')
gtext('1.-(exp(-.5.*t).*sin(t))')
hold;
y=1.-(exp(-1.*t).*sin(t));
plot(t,y),grid
gtext('1.-(exp(-1.*t).*sin(t))')
```

Gambar 3.3 Tampilan Grafik1.m pada MATLAB Editor/Debugger

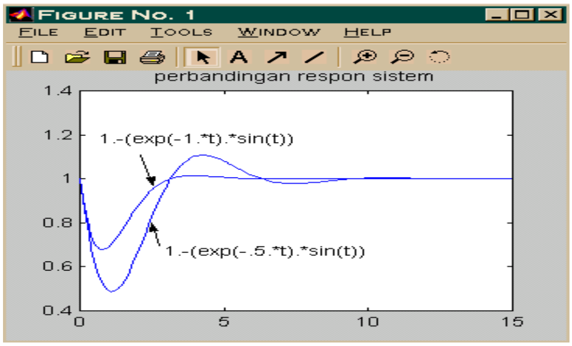
Kemudian simpanlah program Anda dengan cara mengklik **File** dalam layar **MATLAB Editor/Debugger** dan pilih **Save As...**. Tuliskan nama file yang baru Anda buat misalnya grafik1 dan tekan tombol **Save**. Pastikan file Anda tersimpan dalam direktori yang ada dalam jalur pencarian *(path)* MATLAB, yang default-nya pada subdirektori work.

Untuk menjalankan program yang baru saja Anda buat, aktifkan kembali command window. Lalu tekan enter setelah mengetik grafik1:

» grafik1

Current plot held

Di bawah grafik muncul kalimat current plot held, yang sesuai dengan instruksi pada program yaitu hold;. Maka akan muncul hasil program yaitu grafik respon:



Gambar 3.4 Grafik Hasil Pemrograman Script File

Anda dapat menambahkan tanda panah pada persamaan sebagai penunjuk dengan cara mengklik gambar panah pada toolbox.

2. Membuat *Function File*

Seperti telah dijelaskan di atas bahwa *function file* memproses input agar dihasilkan output sesuai dengan program m-file yang dibuat. Misal kita ingin membuat program yang mensortir suatu array dimana nanti diharapkan memperoleh array yang elemennya urut dari kecil ke besar.

```
function [b, j]=sortir(a)
%fungsi sortir, dengan urutan menaik
%keluarannya, parameter j dengan permutasi
%menemukan vektor b dari vektor a
[b, j]=sort(a);
```

Gunakan prosedur yang sama seperti membuat *script file*. Misalnya kita beri nama sortir. Untuk menjalankannya kita kembali ke command window. Tentu saja kita tidak dapat langsung menjalankannya tanpa adanya input. Oleh karena itu kita harus memberi input vektor yang akan kita urutkan.

```
» a=[13 15 19 22 100 44 4 98];
» [b, j]=sortir(a)          % mengambil function file dari m-file
    b =
        4    13    15    19    22    44    98   100
    j =
        7     1     2     3     4     6     8     5
» sortir(a)
    ans =
        4    13    15    19    22    44    98   100
```

Vektor b adalah hasil sortir dari vektor a yang kita masukkan sedangkan vektor j jumlah permutasi yang dilakukan. Perintah yang terakhir kita ketik apabila hanya ingin mensortir saja.

3. Kontrol Program

MATLAB menyediakan 4 alat yang dapat digunakan programer saat menulis program. yaitu:

- For
- While
- If – else – end , dan
- Switch – Case

Berikut ini contoh penggunaan kontrol program for untuk menghitung harga kosinus dari n=0 hingga π. Di sini akan dihitung sinus dari 0 hingga π dengan 11 point.

```
»  for t=0:10
    y(t+1)=sin(pi*t/10);
    end
»  y
```

```
y =
  Columns 1 through 7
  0    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511
  Columns 8 through 11
  0.8090    0.5878    0.3090    0.0000
```

Program di atas dapat juga ditulis dengan kontrol program while sebagai berikut:

```
»  while t>10
   t=0
   y(t+1)=sin(pi*t/10);
   end
»  y
   y =
   Columns 1 through 7
   0    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511
   Columns 8 through 12
   0.8090    0.5878    0.3090    0.0000   -0.3090
```

Berikut adalah contoh penggunaan kontrol program if – else – end. Bila barang yang dibeli lebih dari 10, maka dapat diskon 20%. Jika pembelian kurang dari 10, tidak dapat diskon. Pertama-tama buat *function file* dari MATLAB Debugger.

```
      if barang>10
         harga=(1-0.2)*hargabrg*barang
      else
         harga=hargabrg*barang
   end
```

Lalu simpan dengan nama harga, sehingga diperoleh m-file dengan file harga.m. Coba program dijalankan lewat command window.

```
»  barang=5
   barang =
        5
»  hargabrg=100
   hargabrg =
      100
```

```
» harga               % memanggil m-file harga.m
  harga =
    500
» clear        % membersihkan variabel
» barang=100
  barang =
    100
  » hargabrg=100
  hargabrg =
          100
» harga               % memanggil m-file harga.m
  harga =
    8000
```

Pada program di atas saat jumlah barang yang dibeli lima dengan harga satuan 100 didapat harga = 500. Tetapi jika pembelian 100 (lebih dari 10 buah) harganya bukan 10.000 melainkan diskon 20%=8.000.

Bila sederetan perintah harus dikerjakan dengan didasarkan pada penggunaan berulang-ulang suatu tes dengan argumen yang sama, konstruksi switch-case akan lebih tepat digunakan. Konstruksi control ini memiliki bentuk:

switch ekspresi
case test_ekspresi1
deret_perintah1
case {test_ekspresi2, test_ekspresi3, test_ekspresi4}
deret_perintah2
otherwise
deret_perintah 3
end

Berikut ini adalah contoh penggunaan switch case dimana merubah beberapa satuan menjadi centimeter, cm. Buatlah m-file sebagai berikut lalu simpan dengan nama ubahunit.

```
switch units
case{'inch','in'}
  y=x*2.54;
```

```
case{'meter','m'}
  y=x/100;
case{'centimeter','cm'}
  y=x;
otherwise
  disp{['Unit tidak diketahui: 'units]}
  y=nan;
end;
```

Setelah itu masuk ke command window. Misal kita masukan harga x=10 in, dan akan diubah menjadi centimeter, cm.

```
» x=10.;
» units='in';
» ubahunit              % panggil m-file ubahunit.m
» y
   y =
     25.4000
```

4. Fungsi Inline dan Feval

Fungsi inline dipakai selama satu sesi MATLAB tanpa membuka MATLAB Debugger. Berikut ini contoh membuat *inline function* guna mencari harga diskriminan persamaan kuadrat.

```
» disk=inline('sqrt(b.^2-4*a*c)','a','b','c')
   disk =
       Inline function:
       disk(a,b,c) = sqrt(b.^2-4*a*c)


» disk(1,3,2)
   ans =
       1
```

Feval memanfaatkan m-file yang telah kita buat. Misalnya kita lihat kembali *function file* harga.m yang menyatakan bila jumlah barang yang dibeli lebih dari 10 akan mendapatkan diskon 20% (lihat subbab 3.6.3).

```
» barang=100.
  barang =
     100

» hargabrg=100
  hargabrg =
     100

» feval('harga')          % memanggil m-file harga.m
  harga =
          8000
```

5. Operator Logika dan Relasi

Perbandingan-perbandingan dalam MATLAB dilakukan dengan bantuan operator-operator berikut:

Tabel 3.4 Operator Logika

Operasi	Simbol
<	Kurang dari
<=	Kurang dari sama dengan
>	Lebih dari
>=	Lebih dari sama dengan
==	Sama dengan
~=	Tidak sama dengan

Operator == membandingkan dua variabel dan mengembalikan yang satu jika mereka sama dan jika sebaliknya diisi angka nol. Fungsi relasi dan logika serta fungsi-fungsi penguji dapat dilihat pada tabel berikut ini:

Tabel 3.5 Tabel Fungsi Logika pada MATLAB

Fungsi Relasi dan Logika	
xor(x,y)	Operasi eksklusif OR. Menghasilkan satu jika salah satu dari x atau y tidak nol (benar). Menghasilkan nol jika baik x maupun y adalah nol (salah)/keduanya tidak nol (benar).
any(x)	Menghasilkan satu jika ada elemen dalam suatu vektor yang tidak nol. Menghasilkan satu untuk setiap kolom suatu matriks yang mempunyai elemen tidak nol.
all(x)	Menghasilkan satu jika semua elemen dalam suatu vektor x tidak nol. Menghasilkan satu untuk setiap kolom dalam suatu matriks x yang semua elemennya tidak nol.
Fungsi-fungsi Penguji	
isa(X,’nama’)	Benar jika X mempunyai objek kelas ‘nama’
iscell(X)	Benar jika argumennya adalah suatu sel array.
iscellstr(S)	Benar jika argumennya adalah sela suatu array string.
ischar(S)	Benar jika argumennya adalah karakter.
isempty(X)	Benar jika argumennya kosong
isequal(A,B)	Benar jika A dan B identik
isfield(S,’nama’)	Benar jika ‘nama’ adalah field dari struktur S.
isfinite(X)	Benar jika elemen terhingga.
isglobal(X)	Benar jika argumennya adalah variabel global.
ishandle(h)	Benar jika argumennya adalah objek handle valid.
ishold	Benar jika status hold plot saat ON.
isieee	Benar jika komputer melakukan aritmatika IEEE
isinf(X)	Benar jika elemen tak terhingga.
isletter(S)	Benar jika elemen adalah huruf-huruf alfabet.
islogical(X)	Benar jika argumennya adalah array logika.

ismember(A,B)	Benar jika elemen di A juga terdapat di B.
isnan(X)	Benar jika elemen adalah nan.
isnumeric(X)	Benar jika argumen adalah array numerik
Isppc	Benar untuk Macontosh dengan prosesor powerpc
isprime(X)	Benar jika elemen adalah bilangan prima.
isreal(X)	Benar jika argumen tidak mempunyai bagian imajiner.
isspace(S)	Benar jika argumen adalah karakter putih.
issparse(A)	Benar jika argumen adalah matriks jarang.
isstruct(S)	Benar jika argumen merupakan suatu struktur.
isstudent	Benar jika digunakan The Student Edition of MATLAB.
isunix	Benar jika sistem operasinya adalah UNIX.
isvms	Benar jika komputernya adalah VMS.

Berikut adalah contoh penggunaan fungsi logika untuk mencari elemen berangka satu.

```
» a=[ 1 1 2 3 4 7 1 12]
   a =
       1     1     2     3     4     7     1    12
» indx=(a==1)                       % mencari angka 1 di vektor a
   indx =
       1     1     0     0     0     0     1     0
» b=a(indx)               % vektor b berisi angka 1 saja dari a
   b =
       1     1     1
» indx=find(a==1)                   % letak angka 1 pada vektor a
   indx =
       1     2     7
```

6. String

String adalah sederetan karakter yang bukan angka/numerik. Karakter yang tampak berformat ASCII. Berikut ini adalah contoh penggunaan string:

```
» str='SUHENDAR'
  str =
  SUHENDAR
» str1=double(str)              % merubah string ke numerik
  str1 =
  Columns 1 through 12
  82   97  104  109   97  100  121   97   32   84  114  105
  Columns 13 through 24
  97  115   32   72   97  110  100   97  121   97  110  116
  Column 25
  111
```

Tabel berikut ini merangkum fungsi-fungsi pada MATLAB yang berkaitan dengan operasi string.

Tabel 3.6 Konversi dan Fungsi-fungsi String

Konversi String	
base2dec	string berbasis x ke bilangan bulat desimal
bin2dec	string biner ke bilangan bulat desimal
Char	string ke ASCII
dec2base	bilangan bulat desimal ke string berbasis x
dec2bin	bilangan desimal ke string biner
dec2hex	bilangan desimal ke string heksadesimal
double	ASCII ke string
fprintf	menulis teks ke file atau monitor
hex2dec	string heksadesimal ke bilangan bulat desimal
hex2num	string heksadesimal ke bilangan floating point IEEE
int2str	integer ke string

mat2str	matriks numerik ke string yang dapat dioperasikan oleh fungse eval
num2str	bilangan ke string
sprintf	bilangan ke string dengan kontrol format
sscanf	konversi string ke bilangan dengan kontrol format
str2num	string ke bilangan
Fungsi-fungsi String	
blank(n)	menghasilkan string dengan n spasi
deblank(s)	menghasilkan karakter blang akhir dari string
eval(string)	mengevaluasi string sebagai perintah MATLAB
feval(f,x,y,...)	mengevaluasi fungsi yang diberikan dengan string
findstr(s1,s2)	menemukan suatu string dalam string yang lain
ischar(s)	benar jika inputnya adalah string
isletter(s)	benar jika terdapat karakter alfabet
isspace(s)	benar jika terdapat karakter putih
lasterr	string pesan kesalahan MATLAB terakhir
lower(s)	mengkonversi string ke huruf kecil
strcat(s1,s2,...)	melakukan penggabungan string horisontal
strcmp(s1,s2)	benar jika kedua string identik
strjust(s)	membuat string rata kanan
strmatch(s1,s2)	menemukan kemungkinan kesesuaian string
strncmp(s1,s2,n)	benar jika n karakter pertama identik
strrep(s1,s2)	mengganti suatu string dengan string yang lain
strtok(s)	menemukan token pertama dalam suatu string
strvcat(s1,s2,...)	menggabungkan string secara vertikal
upper(s)	konversi string ke huruf kapital

SOAL LATIHAN

1. Jelaskan fungsi matematika dalam MATLAB!
2. Tuliskan contoh persamaan matematika dan gambarkan dalam bentuk grafik pada pemrograman MATLAB!
3. Hitunglah invers dari variabel X dalam matriks berikut menggunakan pemrograman MATLAB:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 8 & 9 & 0 \end{bmatrix} \cdot \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$$

DAFTAR PUSTAKA

[1] Anoname. 2004. Matlab Tutorial. Diambil dari: [http://www.engin.umich.edu/ group/ctm](http://www.engin.umich.edu/group/ctm).

[5] Eko Mursito Budi, Manase Sitorus, Estiyanti Ekawati. 2006. Simulator Untuk Pengajaran Sistem Kontrol. Kelompok Keahlian Instrumentasi & Kontrol ITB, Bandung

[7] Hanselman, Duane., Bruce Littlefield. 1997. Matlab, Bahasa Komputasi Teknis (terj.). Penerbit Andi. Yogyakarta.

BAB 4

TRANSFORMASI LAPLACE

Sistem dinamik melibatkan kecepatan dan percepatan sehingga persamaan matematiknya akan melibatkan diferensiasi. Persamaan matematik itu disebut persamaan diferensial. Ada banyak cara menyelesaikan persamaan diferensial, misalnya metode operator D, bernoulli dan sebagainya. Salah satu cara yang sering dipakai adalah transformasi laplace. Transformasi laplace mampu merubah persamaan diferensial menjadi persamaan aljabar biasa dalam variabel laplace. Kemudian setelah persamaan itu diselesaikan, dilakukan transformasi balik menjadi persamaan dalam variabel semula.

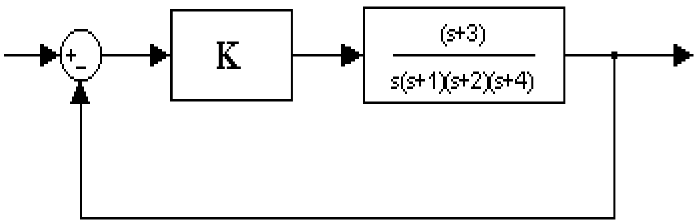
A. FUNGSI ALIH

Fungsi alih atau fungsi transfer adalah perbandingan keluaran terhadap masukan dalam variabel laplace. Variabel laplace diberi simbol s yang didefinisikan: $s = \sigma + j\omega$ dimana σ bagian real dan $j\omega$ bagian imajiner. Disebut fungsi alih karena mengalihkan/mentransfer masukan menjadi keluaran tertentu.

B. DIAGRAM BLOK

Diagram blok menggambarkan aliran proses sistem kontrol yang berupa kotak dan anak panah. Anak panah menggambarkan arah proses yang menggambarkan besaran fisika misalnya listrik, panas dan lain-lain tergantung sistem fisiknya. Kotak merupakan pemroses sinyal yang masuk menjadi sinyal keluar yang merupakan hasil pemodelan matematik dalam variabel laplace.

Terdapat dua titik pada diagram blok yaitu titik jumlah dan titik pisah. Titik jumlah menjumlah/mengurang dua sinyal sedangkan titik pisah tidak melakukan operasi hanya membagi dua sinyal masing-masing identik (tidak bertambah/berkurang).



Gambar 4.1 Diagram Blok

C. TRANSFORMASI LAPLACE

Pada sistem kontrol klasik, transformasi laplace sangat terkenal karena mampu merubah persamaan diferensial yang rumit menjadi persamaan aljabar biasa. Awalnya kita mentransformasikan persamaan dalam variable t dengan rumus transformasi:

$$\wp\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}\,dt$$

Tetapi kita tidak perlu mengintegralkan semua fungsi yang terlibah dalam persamaan diferensial. Kita tinggal melihat pada tabel bentuk yang sesuai dicari.

Tabel 4.1 Transformasi Laplace

No	f(t)	F(s)
1.	Impuls satuan δ(t)	1
2.	Langkah satuan 1(t)	1/s

3.	T	$1/s^2$
4.	$\frac{t^{n-1}}{(n-1)!}\ (n=1,2,3,...)$	$1/s^n$
5.	$t^n\ (\mathrm{n}=1,2,3,...)$	$\frac{n!}{s^{n+1}}$
6.	e^{-at}	$\frac{1}{s+a}$
7.	te^{-at}	$\frac{1}{(s+a)^2}$
8.	$\frac{1}{(n+1)!}t^{n-1}e^{-st}\ (n=1,2,3,...)$	$\frac{1}{(s+a)^n}$
9.	$t^n e^{-st}$ (n=1, 2, 3, ..)	$\frac{n!}{(s+a)^{n+1}}$
10.	sin ωt	$\frac{\omega}{s^2+\omega^2}$
11.	cos ωt	$\frac{s}{s^2+\omega^2}$
12.	sinh ωt	$\frac{\omega}{s^2-\omega^2}$
13.	cosh ωt	$\frac{s}{s^2-\omega^2}$
14.	$\frac{1}{a}\left(1-e^{-at}\right)$	$\frac{1}{s(s+a)}$
15.	$\frac{1}{(b-a)}\left(e^{-at}-e^{-bt}\right)$	$\frac{1}{(s+a)(s+b)}$
16.	$\frac{1}{(b-a)}\left(be^{-bt}-ae^{-at}\right)$	$\frac{s}{(s+a)(s+b)}$
17.	$\frac{1}{ab}\left[1+\frac{1}{(b-a)}\left(be^{-at}-ae^{-bt}\right)\right]$	$\frac{1}{s(s+a)(s+b)}$
18.	e^{-st} sin ωt	$\frac{\omega}{(s+a)^2+\omega^2}$
19.	e^{-st} cos ωt	$\frac{s+a}{(s+a)^2+\omega^2}$
20.	$\frac{\omega_n}{\sqrt{1-\xi^2}}e^{-\xi\omega_n t}\sin\omega_n\sqrt{1-\xi^2 t}$	$\frac{\omega_n{}^2}{s^2+2\xi\omega_n s+\omega_n{}^2}$

21.	$-\frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin\left(\omega_n \sqrt{1-\xi^2}\, t - \phi\right)$ $\phi = \tan^{-1} \frac{\sqrt{1-\xi^2}}{\xi}$	$\frac{s}{s^2 + 2\xi\omega_n s + \omega_n^{\ 2}}$
22.	$1-\frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin\left(\omega_n \sqrt{1-\xi^2}\, t + \phi\right)$ $\phi = \tan^{-1} \frac{\sqrt{1-\xi^2}}{\xi}$	$\frac{\omega_n^{\ 2}}{s^2 + 2\xi\omega_n s + \omega_n^{\ 2}}$
23.	1-cos ωt	$\frac{\omega^2}{s\left(s^2+\omega^2\right)}$
24.	ωt -sin ωt	$\frac{\omega^3}{s^2\left(s^2+\omega^2\right)}$
25.	$\frac{d}{dt} f(t)$	sF(s) – f(0+)
26.	$\int_{0+}^{t} f(t) dt$	F(s)/s

Dengan bantuan MATLAB kita dapat langsung menghitung transformasi laplace berbagai macam fungsi, dengan cara sebagai berikut:

Sintaks: L=Laplace(F)
L=Laplace(F,t)
L=Laplace(F,w,z)

```
» laplace(exp(-a*t).*sin(4*t))
    ans =
    4/((s+a)^2+16)
» pretty(ans)
```

$$\frac{4}{(s + a)^2 + 16}$$

Kebalikan transformasi laplace adalah invers transformasi laplace yang mengembalikan hasil operasi matematika dalam variabel s menjadi variabel t.

Sintaks:

F=iLaplace(L)
F=iLaplace(L,t)
F=iLaplace(L,y,x)

Berikut ini contoh seperti pada tabel laplace no. 6:

```
» syms s a
» y=ilaplace(1/(s+a))
  y =
  exp(-a*t)
```

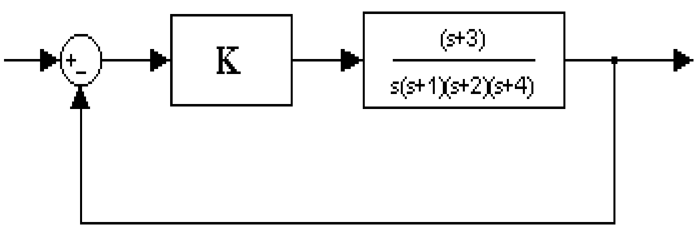
Berikut ini dirangkum transformasi-transformasi integral lainnya yang tersedia di MATLAB.

Tabel 4.2 Transformasi Integral

Transformasi Integral	
Fourier	Transformasi Fourier
Laplace	Transformasi Laplace
Ztrans	Transformasi Z
Ifourier	Invers Transformasi Fourier
Ilaplace	Invers Transformasi Laplace
Iztrans	Invers Transformasi Z

SOAL LATIHAN

Tuliskan persamaan karakteristik dari blok diagram sistem kendali di bawah ini jika nilai K = $S^2 + 4$



DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

BAB 5

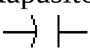
PEMODELAN MATEMATIKA

Sampai sejauh ini Anda telah diperkenalkan dengan bahasa komputasi dan teknik MATLAB yang mampu menghitung dengan kecepatan dan akurasi yang sangat tinggi. Namun tetap saja hanya kita yang mampu membuat model matematik dari sistem fisik. Komputer hanya bisa membantu (simulasi dan menghitung) setelah kita selesai membuat model matematik. Bagi Anda yang sudah menguasai, bab ini boleh dilewati.

A. SISTEM LISTRIK

Tiga elemen dasar pembentuk rangkaian listrik yaitu resistor, kapasitor dan induktor. Masing-masing elemen mempunyai hubungan antar variabel sinyal yang berbeda-beda. Secara rinci hubungannya dapat dilihat pada tabel berikut ini:

Tabel 5.1 Hubungan Tegangan – Arus Sistem Elektrik

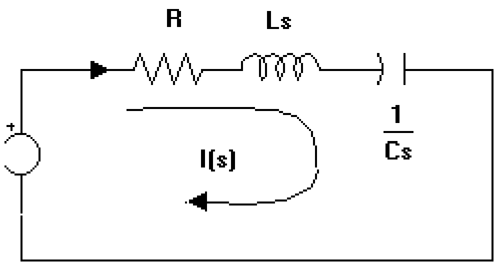
Komponen	Tegangan – Arus	Tegangan–Muatan	Impedansi Z(s)=V(s)/I(s)
Kapasitor 	$v(t) = \frac{1}{C}\int_0^t i(\tau)d\tau$	v(t)=q(t)/C	$\frac{1}{Cs}$
Resistor 	v(t)=Ri(t)	$v(t) = R\frac{dq(t)}{dt}$	R

Induktor	$v(t) = L\dfrac{di(t)}{dt}$	$v(t) = \dfrac{Ld^2q(t)}{dt^2}$	Ls

Impedansi sistem listrik dinyatakan sebagai Z(s)=V(s)/I(s). Secara umum prosedur penyelesaian sistem listrik adalah:

1. Ganti harga elemen pasif dengan nilai impedansinya
2. Asumsikan arah arus di masing-masing mesh
3. Tuliskan hukum tegangan Kirchoff di masing-masing mesh
4. Selesaikan persamaan-persamaannya
5. Bentuk fungsi alih dari persamaan-persamaan tersebut

Berikut ini contoh mencari fungsi alih sistem listrik sederhana:



Gambar 5.1 Contoh Sistem Listrik Sederhana

Pada kasus ini, tegangan melintasi kapasitor adalah impedansi kapasitor dibagi impedansi total, dikalikan tegangan inputnya. Sehingga perumusannya:

$$V_c(s) = \frac{\frac{1}{Cs}}{R + Ls + \frac{1}{Cs}} \quad sehingga$$

$$\frac{V_c(s)}{V(s)} = \frac{\frac{1}{Cs}}{R + Ls + \frac{1}{Cs}}$$

Selain mempergunakan pembagian tegangan, persoalan ini dapat juga diselesaikan dengan analisis lingkar ataupun metoda persamaan simpul yang telah dipelajari pada kuliah rangkaian listrik.

B. SISTEM MEKANIS

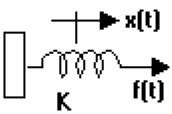
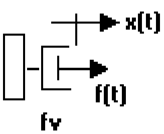
Sistem mekanis banyak dijumpai pada sistem kontroler robotik dalam industri. Di sini melibatkan perpindahan kecepatan dan percepatan.

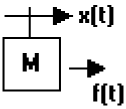
1. Translasi

Sistem mekanis hampir menyerupai sistem elektris. Sistem mekanis, seperti halnya rangkaian listrik, mempunyai tiga komponen pasif linear. Dua di antaranya, yaitu massa dan pegas, adalah elemen penyimpan energi. Sementara peredam membuang energi.

Tabel berikut dapat dijadikan patokan untuk menganalisis sistem mekanik. K, fv dan M berturut-turut disebut konstanta pegas, koefisien gesek dan massa.

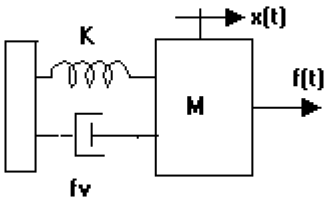
Tabel 5.2 Hubungan Gaya – Perpindahan

Komponen	Gaya – Kecepatan	Gaya–pergeseran	Impedansi $Z_m(s)=F(s)/X(s)$
Pegas 	$f(t) = K\int_0^t v(\tau)d\tau$	f(t)=Kv(t)	K
Peredam 	$f(t)=f_v v(t)$	$f(t) = f_v \frac{ds(t)}{dt}$	$f_v s$

Massa 	$f(t) = M\dfrac{dv(t)}{dt}$	$f(t) = \dfrac{Md^2 s(t)}{dt^2}$	Ms^2

Jika sebelumnya impedansi pada rangkaian listrik dinyatakan sebagai perbandingan antara tegangan dan arus dalam bentuk transformasi laplace, maka impedansi pada sistem mekanis dinyatakan sebagai: $Z_m(s)=F(s)/X(s)$

Berikut ini contoh cara mencari fungsi alih X(s)/F(s) sistem mekanis:



Gambar 5.2 Contoh Sistem Mekanis

Gaya f yang bekerja akan dilawan oleh gaya pegas, peredam dan kelembaman massa, sehingga menurut hukum Newton:

$$Ms^2X(s) + f_v sX(s) + KX(s) = F(s)$$

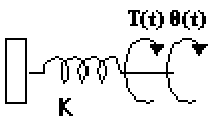
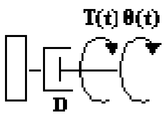
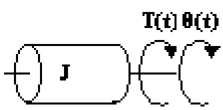
Sehingga fungsi alihnya adalah :

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{Ms^2 + f_v s + K}$$

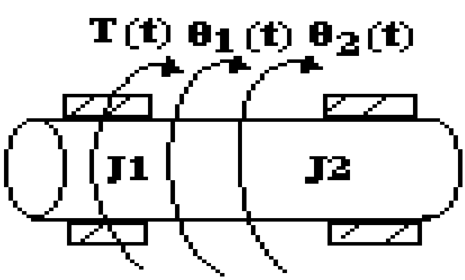
2. Rotasi Mekanis

Penanganan sistem rotasi mekanis, menyerupai sistem translasi, kecuali bahwa torsi menggantikan gaya dan pergerakan sudut menggantikan perpindahan translasi. Komponen mekanis antara kedua sistem tidak berbeda, hanya pada rotasi, gerak berputar diperhitungkan lebih dominan dibanding gerak translasinya.

Tabel 5.3 Hubungan Torsi – Sudut

Komponen	Torsi – Kecepatan sudut	Torsi– sudut	Impedansi $Z_m(s)=T(s)/\theta(s)$
Pegas 	$T(t) = K\int_0^t \omega(\tau)d\tau$	$T(t)=K\omega(t)$	K
Peredam 	$T(t)=D\ \omega(t)$	$T(t) = D\dfrac{d\theta(t)}{dt}$	Ds
Massa 	$f(t) = J\dfrac{d\omega(t)}{dt}$	$f(t) = \dfrac{Jd^2\theta(t)}{dt^2}$	Js^2

Berikut ini adalah contoh perhitungan mencari fungsi alih sistem rotasi mekanis:



Gambar 5.3 Sistem Rotasi Mekanis

Asumsikan torsi bekerja seperti sebuah pegas yang dikonsentrasikan pada satu titik khusus pada silinder, dengan inersia J1 dan J2 di sebelah kanannya. Analogi dengan sistem translasi menghasilkan persamaan:

$$(J1s^2 + D1s + K)\theta_1(s) - K\,\theta_2(s) = T(s)$$

$$-K\theta_1(s) + (J2s^2 + D2s + K)\,\theta_2(s) = 0$$

Dengan aturan crammer ataupun analisis matematis lainnya, didapat fungsi alih:

$$\frac{\theta_2(s)}{T(s)} = \frac{K}{\Delta}$$

$$\text{dimana: } \Delta = \begin{vmatrix} (J_1s^2 + D_1s + K) & -K \\ -K & (J_2s^2 + D_2s + K) \end{vmatrix}$$

3. Sistem dengan Roda Gigi (Gir)

Gir adalah penemuan manusia yang cukup spektakuler. Dengan ditemukannya gir, kerja-kerja mekanis menjadi sederhana. Untuk memudahkan analisis diasumsikan gir sebagai gir ideal dimana tidak terjadi *backlash*.

Hubungan antara dua gir adalah :

$$r_1\theta_1 = r_2\theta_2$$

$$\frac{\theta_2}{\theta_1} = \frac{r_1}{r_2} = \frac{N_1}{N_2}$$

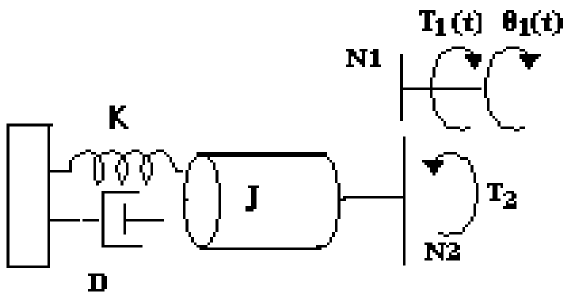
Dimana:

- r_1 = Jari-jari gir-1
- r_2 = Jari-jari gir-2

- θ_1 = Sudut putar gir-1
- θ_2 = Sudut putar gir-2
- N_1 & N_2 = Jumlah gigi pada gir-2 dan gir-2

Sedangkan hubungan antara torsi masukan dan torsi keluaran adalah:

$$T_1\theta_1 = T_2\theta_2$$
$$\frac{T_2}{T_1} = \frac{\theta_1}{\theta_2} = \frac{N_2}{N_1}$$



Gambar 5.4 Sistem Rotasi dengan Gir

Persamaan geraknya menjadi:

$$\left(Js^2 + Ds + K\right)\theta_2\left(s\right) = T_2 s)$$

Berdasarkan hubungan sudut, torsi dan jumlah gir:

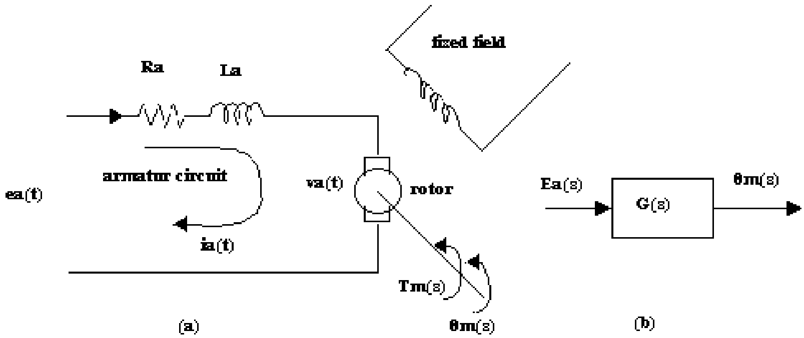
$$\left(Js^2 + Ds + K\right)\frac{N_1}{N_2}\theta_2\left(s\right) = T_1(s)\,\frac{N_2}{N_1}$$

Secara umum, impedansi rotasi mekanis dapat dicerminkan melalui rangkaian gir dengan mengalikan impedansi mekanisnya dengan perbandingan:

$$\left(\frac{\sum N\ tujuan}{\sum N\ sumber}\right)^2$$

4. Sistem Rotasi Motor

Motor banyak digunakan pada robotika. Oleh karena itu pemahaman tentang model matematika rotasi pada motor mutlak diperlukan.



Gambar 5.5 (a). Motor DC, (b). Fungsi Alih

Medan magnet dibangkitkan oleh sebuah elektromagnet tetap yang disebut fixed field. Sementara rangkaian berputar dinamakan *armature*, dimana mengalir arus $i_a(t)$ dan bekerja gaya sebesar $F=Bxlxi_a(t)$. B adalah kuat medan magnet dan l merupakan panjang konduktor.

Fenomena lain yang terjadi adalah pembangkitan tegangan akibat pergerakan konduktor terhadap medan magnet e=Blv, dengan e adalah tegangan dan v kecepatan gerak konduktor. Jika armatur pembawa arus berotasi sepanjang medan magnet, tegangan sebanding dengan kecepatan:

$$v_a(t) = Kb\frac{d\theta_m(t)}{dt} \qquad (1)$$

$v_a(t)$ disebut gaya gerak listrik balik, Kb adalah konstanta ggl balik, dan $d\theta_m(t)/dt$ kecepatan sudut motor. Dengan transformasi laplace, diperoleh

$$\mathbf{V_a(s)=Kbs\theta_m(s)} \qquad (2)$$

Persamaan lingkar sepanjang rangkaian armatur,

$$R_aI_a(s) + L_asI_a(s) + V_a(s) = E_a(s) \qquad (3)$$

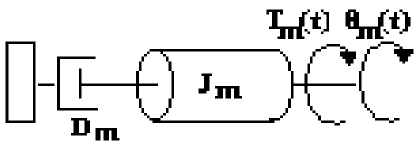
Torsi motor sebanding dengan arus armatur, sehingga kita dapat menulis:

$$T_m(s) = K_tI_a(s) \qquad (4)$$

T_m adalah torsi motor, sedangkan K_t adalah konstanta torsi motor yang bergantung kepada motor dan karakteristik medan magnet. Subsitusi persamaan (2) dan (4) ke persamaan (3) menghasilkan:

$$\frac{(R_a + L_a s)T_m(s)}{K_t} + K_b s\theta_m(s) = E_a s \qquad (5)$$

Sekarang harus dicari nilai $T_m(s)$ dalam fungsi $\theta_m(s)$ sehingga dihasilkan fungsi alih $\theta_m(s)/E_a(s)$



Gambar 5.6 Ekivalensi Pembebanan Mekanis Pada Motor

Dari gambar di atas dapat dibentuk persamaan

$$T_m(s) = (J_m s2 + D_m s)\ \theta_m(s) \qquad (6)$$

Substitusi ke persamaan (5) menghasilkan:

$$\frac{(R_a + L_a s)(Jms^2 + D_m s)\theta_m(s)}{K_t} + Kbs\theta_m(s) = E_a(s) \qquad (7)$$

Jika diasumsikan induktansi *armature* sangat kecil, persamaan di atas disederhanakan menjadi:

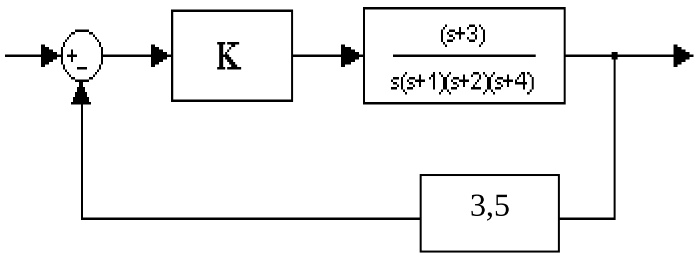
$$\left[\frac{R_a}{K_t}(J_m s + D_m) + Kb\right]s\theta_m(s) = E_a(s) \qquad (8)$$

dan didapat fungsi alih:

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t / R_a J_m}{s\left[s + \frac{1}{J_m}\left(D_m + \frac{K_t K_b}{R_a}\right)\right]} \tag{9}$$

SOAL LATIHAN

Tuliskan model persamaan matematika dari blok diagram sistem kendali di bawah ini!



DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

BAB 6

KINERJA SISTEM KENDALI

Dengan bermodalkan pengetahuan dari bab terdahulu yaitu pemodelan matematik dan penyelesaian persamaan diferensial dengan transformasi laplace sekarang kita mulai menganalisa unjuk kerja dasar sistem kendali yang berkaitan dengan kecepatan dan akurasi.

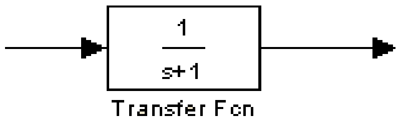
A. RESPON TRANSIEN

Respon transien adalah reaksi awal sistem kontrol terhadap masukan tertentu. Transien berarti transisi dari keadaan sebelum menjadi keadaan setelah diberi masukan. Jenis masukan yang sering digunakan untuk menguji respon transien sistem antara lain:

1. Masukan Impulse, $r(t)=\delta(t)$
2. Masukan Step, $r(t)=1$
3. Masukan Ramp, $r(t)=t$
4. Masukan Sinusoida, $r(t)=\sin \omega t$

Sebagian besar alat bekerja dengan respon step, misalnya AC, penggerak antena dan mekanisme servo. Pada sistem kontrol elektronik biasanya melibatkan frekuensi sehingga masukan yang tepat untuk analisa adalah sinus.

Contoh sistem orde-1.

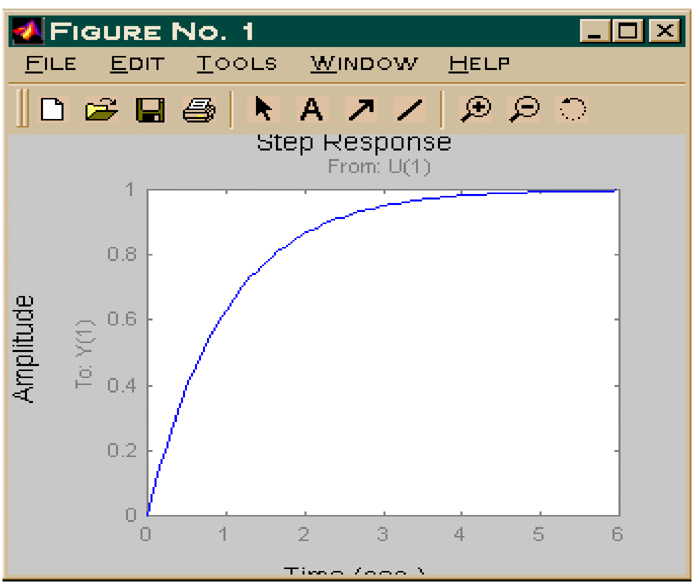


Gambar 6.1 Sistem Orde-1

Berikut ini contoh penulisan fungsi alih dalam MATLAB.

```
» num=[0 1];           % cara pertama menulis fungsi alih
» den=[1 1];
» g=tf([1],[1 1])      % cara kedua menulis fungsi alih
  Transfer function:
         1
       -----
       s + 1

» step(num,den)        % masukan step
» step(g)              % masukan step untuk cara kedua
```

Pada jendela grafik akan muncul grafik sebagai berikut:



Gambar 6.2 Respon Sistem Contoh Terhadap Masukan Step

Cara praktis menganalisa tanggapan step suatu sistem adalah dengan menganggap kondisi awal sistem nol. Lima karakteristik pada tanggapan transien adalah:

1. Waktu tunda, td, yaitu waktu yang dibutuhkan sistem naik setengah dari set point.
2. Waktu naik, tr, yaitu waktu yang dibutuhkan sistem naik menyentuh 90%-100% set point.
3. Waktu puncak, tp, yaitu waktu yang dibutuhkan saat sistem menyentuh nilai tertinggi.
4. Overshoot Maksimum, Mp, Jarak lebih respon sistem terhadap set point atau biasanya dinyatakan dalam prosentase.
5. Waktu turun, ts, adalah waktu yang ditempuh sistem untuk mencapai harga 2%-5% kesalahan terhadap setpoint.

Pembaca tentu saja dapat menjawab lima karakteristik sistem dengan melihat grafik respon pada gambar VI.2. td=1s, tr=2s, ts=3,5s dan tidak ada overshoot.

Bagaimana dengan masukan ramp, sinus dan masukan lain? Selain step dan impulse, kita harus memodifikasi fungsi alih. Misalnya kita ingin mengetahui respon ramp. Berdasarkan teori, fungsi alih adalah perbandingan keluaran dengan masukan. Jadi keluaran adalah masukan dikali fungsi alih. Karena masukan ramp=$1/s^2$, kita dapat melihat respon dengan:

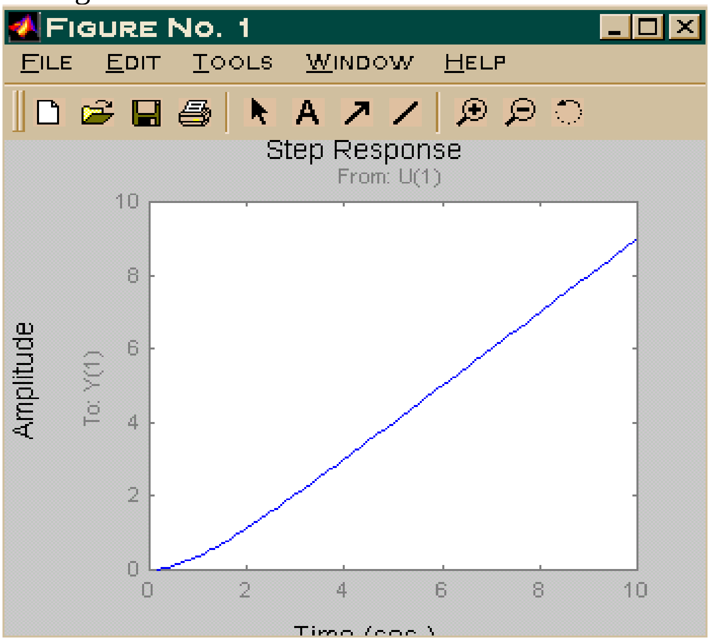
1. Masukan impulse terhadap fungsi alih yang dikali dengan $1/s^2$
2. Dengan masukan step terhadap fungsi alih yang dikali dengan 1/s.

Berikut ini cara melihat respon ramp dengan cara kedua.

```
» g=tf([1],[1 1 0])      % fungsi dikali 1/s
        Transfer function:
           1
        -------
        s^2 + s
```

» step(g) % walau step, sebenarnya ini adalah masukan ramp

Diperoleh grafik:



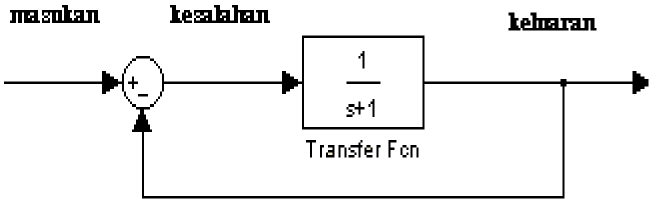
Gambar 6.3 Grafik Sistem Respon Ramp yang memanfaatkan fungsi Step

Di bab VII nanti Anda akan menemukan fasilitas dalam MATLAB yang lebih mudah dalam menganalisa respon transien sistem.

B. KESALAHAN TUNAK

Kesalahan adalah penyimpangan sistem kontrol dari set point sedangkan tunak *(steady state)* adalah kondisi mantap yang secara matematis dinyatakan limit mendekati waktu tak hingga. Secara praktek, kesalahan tunak berarti kesalahan yang muncul dengan cara mengukur kesalahan sistem setelah menunggu sistem itu diam/mantap.

Kita menghitung kesalahan tunak secara matematik dengan teorema limit dari selisih masukan dengan keluaran. Kita ambil contoh menghitung kesalahan tunak sistem di bawan ini.



Gambar 6.4 Kesalahan Tunak 1/(s+1)

Fungsi alih lingkar tertutup gambar di atas adalah:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} \quad C(s) = E(s)\cdot G(s)$$

$$\frac{E(s)}{R(s)} = \frac{1}{1+G(s)} \quad E(s) = \frac{R(s)}{1+G(s)}$$

Dengan teorema final value problem dalam laplace diperoleh:

$$e(\infty) = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sF(s) = \lim_{s\to 0} \frac{sR(s)}{1+G(s)}$$

Bila pada soal kita ingin menganalisis kesalahan tunak sistem untuk masukan step R(s)=1/s didapat:

$$e(\infty) = \lim_{s\to 0} \frac{sR(s)}{1+G(s)} = \lim_{s\to 0} \frac{s(1/s)}{1+\dfrac{1}{s+1}} = 0{,}5$$

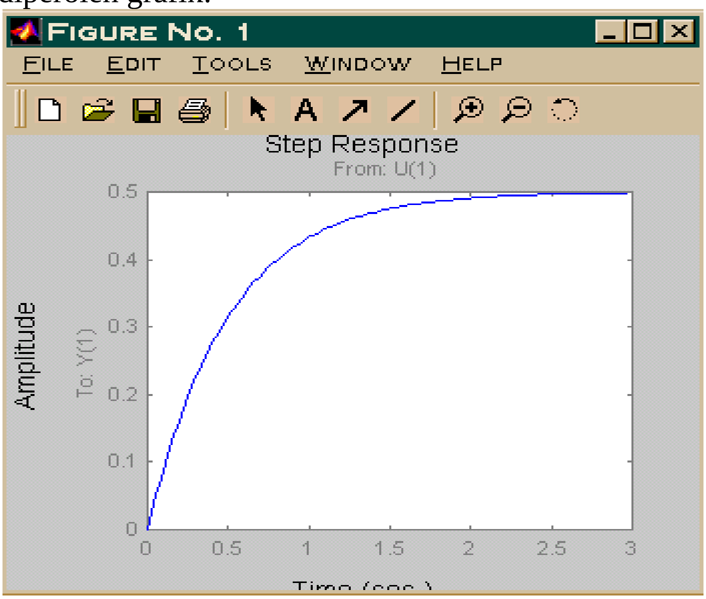
Sekarang mari kita coba dengan MATLAB. Fungsi alih lingkar tertutup sistem di atas adalah :

$$\frac{C(s)}{R(s)}=\frac{G(s)}{1+G(s)}=\frac{\frac{1}{s+1}}{1+\frac{1}{s+1}}=\frac{1}{s+2}$$

Coba analisa respon sistem untuk masukan step R(s)=1/s

```
» gclose=tf([1],[1 2])  % fungsi alih lingkar tertutup sistem
          Transfer function:
            1
          -----
          s + 2

» step(gclose)
```

Dan diperoleh grafik:



Gambar 6.5 Grafik Respon Sistem 1/(S+1) Lingkar tertutup

Kalau Anda cermati harga keluaran tunaknya yang sebesar 0,5 dapat diambil kesimpulan bahwa kesalahan tunak sistem tersebut adalah : masukan – keluaran = 1 – 0,5 = 0,5 yang sesuai dengan hitungan manual sebelumnya.

Kita dapat juga menggambar respon lingkar tertutup dengan hanya melihat fungsi alih lingkar terbukanya:

```
» g=tf([1],[1 1])              % lingkar terbuka
       Transfer function:
         1
       -----
       s + 1
» gclose=minreal(g/(1+g))      % merubah ke lingkar tertutup
       Transfer function:
         1
       -----
       s + 2
 » step(gclose)
```

Yang akan memunculkan grafik yang serupa dengan sebelumnya. Berikut ini tabel yang memuat fungsi-fungsi lain pada MATLAB yang bermanfaat dalam menganalisa respon.

Tabel 6.1 Fungsi MATLAB untuk Respon Sistem Kontrol

Step	respon step
impulse	respon impulse
initial	respon sistem status-ruang dengan keadaan awal ditentukan
Lsim	respon terhadap sembarang input
ltiview	analisis respon berbasis GUI
gensig	menghasilkan signal input untuk lsim
stepfun	menghasilkan input step

C. STABILITAS

Sistem dikatakan stabil jika tanggapan sistem terhadap gaya pemaksa mengecil. Sedangkan sistem dikatakan tidak stabil jika tanggapan sistem terhadap gaya pemaksa terus membesar menuju besaran tak hingga menyebabkan sistem rusak seperti yang terjadi pada kasus meledaknya pesawat ulang alik Colombia milik Amerika Serikat. Kasus ketidakstabilan kerap terjadi pada sistem lingkar tertutup karena adanya umpan balik yang bila rancangannya kurang tepat malah dapat menimbulkan kerusakan sistem.

Berikut ini cara perhitungan kestabilan klasik yang dinamakan Routh-Hurwitz. Suatu sistem memiliki persamaan karakteristik: $as^4 + bs^3 + cs^2 + ds + e = 0$

Perhatikan cara memasukan nilai pada tabel berikut ini sesuai dengan ordenya.

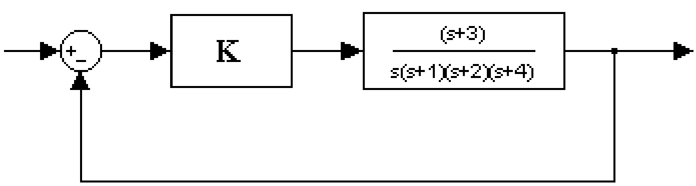
Tabel 6.2 Tabel Rout-Hurwitz

s^4	a	C	e
s^3	b	d	0
s^2	$\frac{(bxc)-(axd)}{b} = f$	$\frac{(bxe)-(ax0)}{b} = g$	0
s^1	$\frac{(fxd)-(bxg)}{f} = h$	$\frac{(fx0)-(bx0)}{f} = 0$	
s^0	$\frac{(hxg)-(fx0)}{h} = i$		

Lihat pada kolom pertama berisi angka a, b, f h dan i. Routh-Hurwitz mengatakan bahwa sistem stabil jika di kolom pertama ini tidak ada perubahan tanda.

Misal sistem yang kita analisa memiliki fungsi alih:

$$G(s) = \frac{s+3}{s(s+1)(s+2)(s+4)}$$



Gambar 6.6 Contoh Perhitungan Kestabilan

Bila harga penguatan K = 5, apakah sistem stabil? Mari kita mulai menghitung dengan kriteria Routh-Hurwitz. Penyederhanaan blok sistem di atas menghasilkan fungsi alih:

$$G_{tertutup}(s) = \frac{5(s+3)}{s(s+1)(s+2)(s+4)+5(s+3)}$$

Bila yang ingin kita analisa adalah respon terhadap masukan impulse R(s)=1 maka keluarannya didapat:

$$C(s) = G_{tertutup}(s) = \frac{5(s+3)}{s(s+1)(s+2)(s+4)+5(s+3)}$$

Diperoleh :
$$C(s) = \frac{5s+15}{s^4+7s^3+14s^2+13s+15}$$

Persamaan karakteristik untuk sistem lingkar tertutup di atas adalah :

$S^4 + 7s^3 + 14s^2 + 13s + 15 = 0$ (penyebut disama dengankan dengan nol)

Kemudian buat tabel Routh-Hurwitz berdasarkan orde pangkatnya yang dapat dilihat teknik peletakan nilainya pada tabel berikut ini:

Tabel 6.3 Tabel Rout-Hurwitz

s^4	1	14	15
s^3	7	13	0
s^2	$\frac{(7x14)-(13x1)}{7} = 12{,}14$	$\frac{(7x15)-(1x0)}{7} = 15$	0
s^1	$\frac{(12{,}14x13)-(15x7)}{12{,}14} = 4{,}35$	$\frac{(12{,}14x0)-(0x7)}{12{,}14} = 0$	
s^0	$\frac{(4{,}35x15)-(0x12{,}14)}{4{,}35} = 15$		

Yang perlu diperhatikan adalah cara perhitungan ordo-2 ke bawah. Lihat pada kolom pertama berisi angka 1,7, 12.14, 4.35 dan 0 yang semuanya positif dan tidak ada perubahan tanda. Routh-Hurwitz mengatakan bahwa sistem stabil jika di kolom pertama ini tidak ada perubahan tanda.

Bila kita ingin memperkuat sistem dengan meningkatkan K, apakah kinerja sistem akan bertambah baik, terutama kestabilannya? Gunakan cara yang sama dengan cara di atas.

Keuntungan cara Routh-Hurwitz adalah kemampuan mengetahui kestabilan sistem tanpa bantuan komputer. Sedangkan kelemahan utamanya adalah kita hanya tahu kestabilan tanpa mengetahui bentuk respon sistem tersebut. Oleh karena itu cara Routh-Hurwitz mulai ditinggalkan seiring berkembangnya komputer.

Berikut ini program singkat melihat respon sistem untuk K=5 dan K yang lebih besar misalnya 10. Apakah kinerjanya baik?

```
» g5=zpk([-3],[0 -1 -2 -4],5)          % fungsi alih untuk K=5 terbuka
  Zero/pole/gain:
      5 (s+3)
  -------------------
  s (s+1) (s+2) (s+4)

» g10=zpk([-3],[0 -1 -2 -4],10)        % fungsi alih untuk K=10 terbuka
  Zero/pole/gain:
      10 (s+3)
  -------------------
  s (s+1) (s+2) (s+4)

» gc5=minreal(g5/(1+g5))               % fungsi alih untuk K=5 tertutup
  Zero/pole/gain:
                5 (s+3)
  -----------------------------------------
  (s+2.473) (s+4.203) (s^2 + 0.3238s + 1.443)
```

» gc10=minreal(g10/(1+g10)) % fungsi alih untuk K=10 tertutup

Zero/pole/gain:

```
              10 (s+3)
----------------------------------------
(s+2.63) (s+4.391) (s^2 - 0.02073s + 2.598)
```

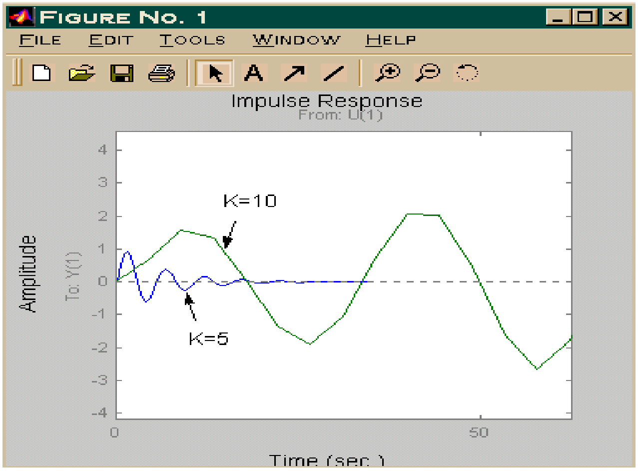
» impulse(gc5) % masukan impuls pada K=5

» hold % grafik sebelumnya tetap dipakai

Current plot held

» impulse(gc10) % input impulse pada K=10

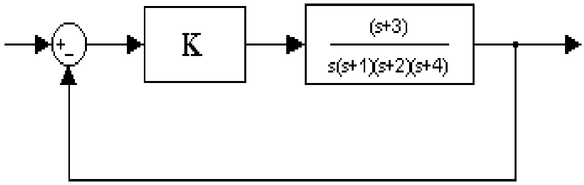
Diperoleh grafik sebagai berikut:



Gambar 6.7 Grafik Respon Sistem dengan K=5 dan K=10

Pada grafik di atas ternyata bila K diperbesar dari 5 menjadi 10 bukannya kinerjanya baik malah sistem menjadi tidak stabil. Untuk menggambar hubungan K dengan letak-letak pole di bidang-s yang mempengaruhi kestabilan dapat Anda lihat di bab yang membahas Tempat Kedudukan Akar.

1. Gambarkan grafik respon transien dari blok diagram sistem kendali di bawah ini!



2. Tentukan dengan menggambarkan grafiknya, apakah sistem dengan persamaan di bawah ini stabil/tidak stabil?

$$G(s) = \frac{s+4}{s\left(s^2+1\right)\left(s+2\right)\left(s+2\right)}$$

DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

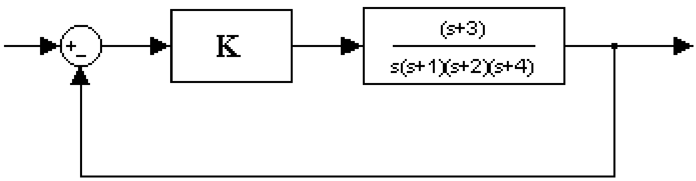
[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

BAB 7

TEMPAT KEDUDUKAN AKAR

A. PENGARUH PENGUATAN TERHADAP STABILITAS

Lihat kembali bab VI mengenai kinerja sistem kendali terutama bab 6.3 tentang stabilitas. Berikut disajikan diagram blok sistem orde-3 seperti gambar 6.6.



Gambar 7.1 Pengaruh K terhadap Stabilitas

Telah dihitung baik dengan kriteria Routh-Hurwitz maupun dengan simulasi MATLAB bahwa K=5 stabil sedangkan K=10 tidak stabil. Dalam perancangan kita tidak hanya berpatokan pada beberapa harga K Sebagai perancang kita ingin tahu letak akar-akar pada bidang-s untuk seluruh harga penguatan K. Letak pole-pole untuk K=1 didapat dengan menyelesaikan persamaan karakteristik:

$$\mathbf{s(s+1)(s+2)(s+4)+(s+3)=0}$$

$$\mathbf{s^4+7s^3+14s^2+9s+3=0}$$

yang merupakan persamaan s pangkat 4. Bila setelah kita hitung diperoleh akar s ada bagian real-nya yang positif maka sudah dipastikan sistem itu tidak stabil. Dengan MATLAB:

```
» pk=[1 7 14 9 3]            % polinomial pers. karakteristik
    pk =
        1    7    14    9    3
» s=roots(pk)                % mencari akar polinomial
    s =
      -4.0415
      -2.1764
      -0.3910 + 0.4338i
      -0.3910 - 0.4338i
```

Sistem untuk K=1 stabil terbukti dari hasil akar s yang seluruh real-nya berharga negatif (-4.0415, -2.1764 dan -0.3910). Namun kita akan kerepotan bila diminta menggambar tempat kedudukan akar dengan cara seperti itu. Saat komputer belum berkembang ada teknik menggambar tempat kedudukan akar dengan metode grafis yang aturannya sebagai berikut:

a. Jumlah Percabangan
   Jumlah cabang dalam TKA sebanding dengan jumlah pole lingkar tertutup

b. Simetri
   Tempat Kedudukan Akar akan simetris di sekitar sumbu real.

c. Segmen sumbu real.
   Untuk K>0, TKA terletak di sebelah kiri jumlah ganjil sumbu real

d. Titi awal dan titik akhir
   TKA berawal pada pole dan berakhir pada zero

e. Aturan Tambahan:

$$\sigma = \frac{\sum Poles - \sum Zeros}{Poles - Zeros}$$

$$M = \tan \frac{(2k+1)\pi}{Poles - Zeros}$$

dengan k=0,± 1, ±2, ...

f. Syarat Sudut:

$$\angle G(s)H(s) = \pm 180^0 (2k+1) \qquad (k = 0,1,2,...)$$

g. Syarat besaran:

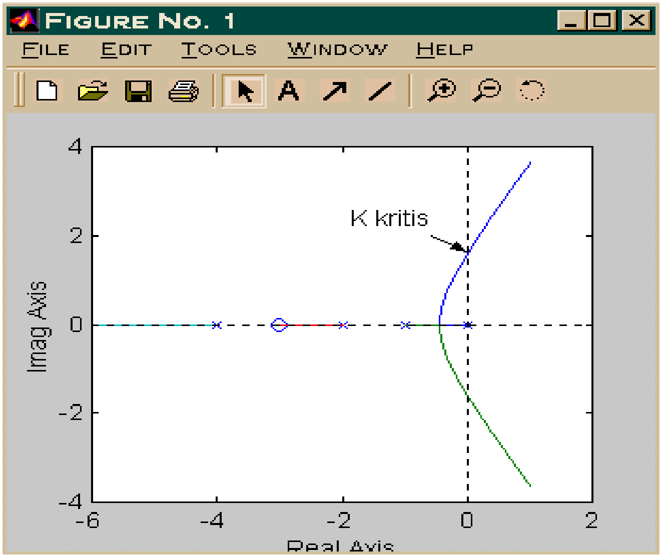
$$|G(s)H(s)| = 1$$

Namun di sini kita akan berkonsentrasi penggambaran tempat kedudukan akar dengan program MATLAB.

B. MENGGAMBAR TEMPAT KEDUDUKAN AKAR DENGAN MATLAB

Dengan MATLAB penggambaran tempat kedudukan akar menjadi lebih mudah, kita tinggal mengetik perintah sebagai berikut pada command window:

```
» g=zpk([-3],[0 -1 -2 -4],1)          % fungsi alih lingkar terbuka
    Zero/pole/gain:
          (s+3)
    ------------------
    s (s+1) (s+2) (s+4)

» rlocus(g)
```



Gambar 7.2 Tempat Kedudukan Akar

Karena pada grafik di atas ada garis yang menyeberangi sumbu imajiner ke kanan, maka tidak seluruh K menghasilkan sistem yang stabil. Ada harga K batas yang disebut K kritis dimana bila harga K tersebut dilampaui, sistem menjadi tidak stabil. Untuk mencari harga K kritis itu kita butuh kriteria Routh-Hurwitz.

Karena persamaan karakteristiknya:

$$S^4 + 7s^3 + 14s^2 + (8+K)s + 3K = 0$$

didapat tabel kriteria routh Hurwitz.

Tabel 7.1 Tabel Routh -Hurwitz

s^4	1	14	3K
s^3	7	8+K	0
s^2	$\frac{90-K}{7}$	$\frac{21K-0}{7}=3K$	

s^1	$\frac{-K^2-65K+720}{90-K}$		
s^0	3K		

Syarat stabil adalah kolom pertama tidak boleh negatif. Sehingga titik kritisnya diperoleh dari:

$$\frac{-K-65K+720}{90-K}=0$$

» p=[-1 -65 720]

p =

-1 -65 720

» roots(p)

ans =

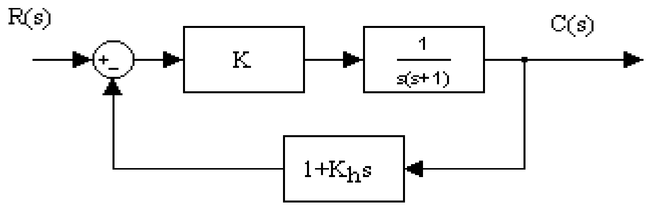
-74.6456

9.6456

Didapat K=9,6456. Ini berarti agar sistem tersebut stabil, penguatan yang diijinkan maksimal 9,6456. Berarti bila K > 9,6456 ada pole yang terletak di sebelah kanan sumbu imajiner pada bidang-s dan sistem akan tidak stabil.

C. PERANCANGAN SISTEM KONTROL DENGAN TEMPAT KEDUDUKAN AKAR

Gambar berikut ini adalah contoh sistem servo dengn umpan balik kecepatan. Kita ingin mendisain mesin ini agar didapat overshoot maksimum sebesar 0,2 dan waktu puncak 1 detik. Berapakah besar penguat K dan K_h yang cocok?



Gambar 7.3 Diagram Blok Sistem Servo Berumpan balik Kecepatan

Overshoot maksimum M_p diberikan oleh persamaan berikut:

$$M_p = e^{-(\zeta / \sqrt{1-\zeta^2})\pi}$$

Karena nilai ini harus sama dengan 0,2 maka:

$$e^{-(\zeta / \sqrt{1-\zeta^2})\pi} = 0{,}2$$

$$atau \quad \frac{\zeta\pi}{\sqrt{1-\zeta^2}} = 1{,}61$$

$$dan \quad \zeta = 0{,}456$$

Waktu puncak diketahui 1 detik. Oleh karena itu diperoleh:

$$tp = \frac{\pi}{\omega_d} = 1$$

$$\omega_d = 3{,}14$$

Karena ζ adalah 0,456, maka ω_n sama dengan

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} = 3{,}53$$

Karena frekuensi alami ω_n sama dengan akar K:

$$K = \omega_n^{\;2} = 12{,}5$$

Maka K_h diperoleh dari persamaan sebagai berikut dengan berdasarkan soal B=1 dan J=1:

$$\zeta = \frac{B + KK_h}{2\sqrt{KJ}}$$

$$K_h = \frac{2\sqrt{K}\zeta - 1}{K} = 0{,}178$$

Waktu naik, t_r: waktu naik (rise time) didefinisikan:

$$t_r = \frac{\pi - \beta}{\omega_d}$$

dengan

$$\beta = \tan^{-1}\frac{\omega_d}{\sigma} = \tan^{-1} 1{,}95 = 1{,}10$$

Jadi, tr diperoleh : t_r = 0,65 detik

Waktu turun, t_s: Misal kita pakai standar 2%,

$$t_s = \frac{4}{\sigma} = 2{,}48\ \text{det}$$

Sekarang kita coba buktikan dengan MATLAB. Tempat kedudukan akar sistem servo di atas adalah:

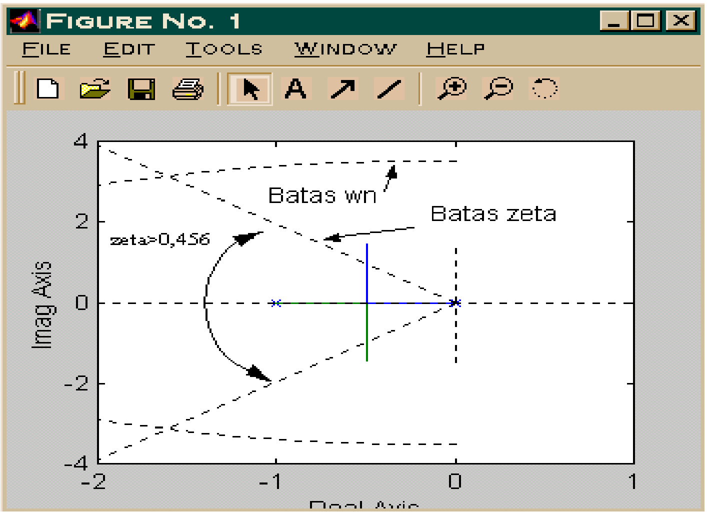
```
» g=zpk([],[0 -1],1)                % lingkat terbuka sistem servo
    Zero/pole/gain:

       1
    -------
    s (s+1)

» rlocus(g)
```

Nilai redaman ζ (zeta) diketahui = 0,456 dan ω_n = 3,53 maka kita dapat mengetahui batasan-batasan harga K dengan instruksi:

```
» g=zpk([],[0 -1],1)
```

Zero/pole/gain: 1
----
s (s+1)

» rlocus(g)
» zeta=0.456;
» wn=3.53;
» sgrid(zeta, wn)   % membuat batas wilayah zeta dan wn

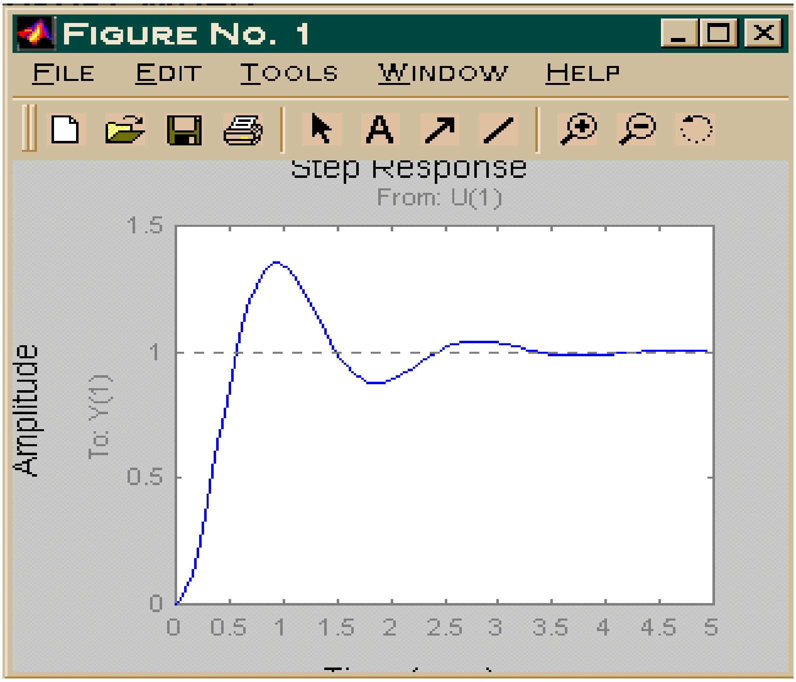


Gambar 7.4 Tempat Kedudukan Akar Sistem Servo

Responnya adalah:
» g=tf([12.5],[1 2.225 12.5])   % lingkar tertutup mesin servo

Transfer function:
12.5
----
s^2 + 2.225 s + 12.5

» step(g)



Gambar 7.5 Grafik Step Respon Sistem Servo

Berikut ini tabel yang memuat instruksi-instruksi penting mengenai tempat kedudukan akar.

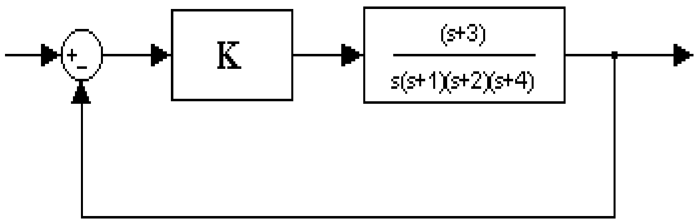
Tabel 7.2 Alat Disain Klasik

Rlocus	akar locus Evan
Rlocfind	penentuan akar perolehan locus secara interaktif
Acker	penempatan kutub SISO
Place	penempatan kutub MIMO
Estim	estimator form dengan estimator perolehan diberikan
Reg	regulator form dengan status umpan balik dan estimator perolehan diberikan

SOAL LATIHAN

Gambarkan grafik tempat keududkan akar dari blok diagram dan persamaan di bawah ini!

1.



2. $$G(s) = \frac{s+4}{s(s^2+1)(s+2)(s+2)}$$

DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis nd Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

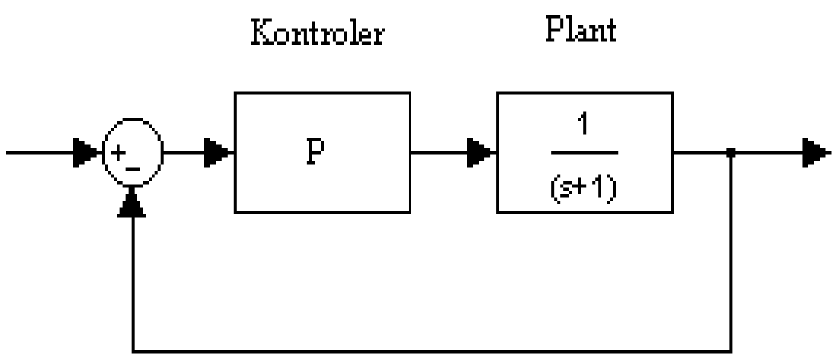
BAB 8

KONTORLER P, PI, DAN PID

Pada bab sebelumnya Anda telah diperkenalkan dengan penguat K dan hubungannya dengan stabilitas sistem pada sistem kontrol lingkar tertutup. Bab ini membahas lebih jauh kinerja sistem kontrol bukan hanya pada kestabilan melainkan juga pada kecepatan dan akurasi.

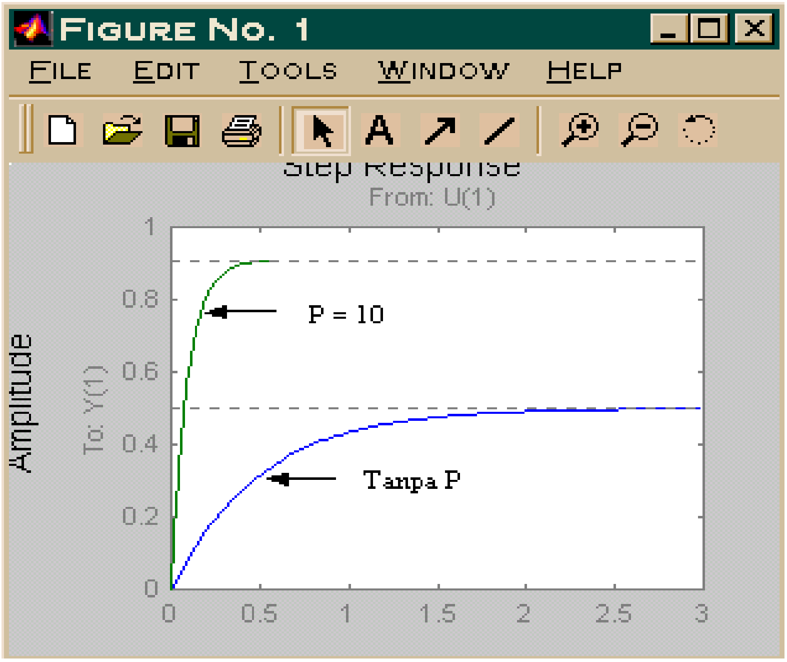
A. KONTROLER PROPORTIONAL (P)

Konroler proportional (P) identik dengan penguat K pada bab yang lalu. Agar lebih jelas berikut ini diberikan contoh ilustrasi.



Gambar 8.1 Kontroler Proportional

```
» g1=zpk([],[-1],1)                % fungsi alih tanpa
  kontroler P (P=1)
      Zero/pole/gain:
       1
     -----
     (s+1)

» g1c=minreal(g1/(1+g1))           % lingkar tertutup g1
      Zero/pole/gain:
       1
     -----
     (s+2)
» step(g1c)                        % input step g1 tertutup

» g2=zpk([],[-1],10)               % fungsi alih dengan
  kontroler  P=10

     Zero/pole/gain:
      10
     -----
     (s+1)

» g2c=minreal(g2/(1+g2))           % lingkar  tertutup g2
      Zero/pole/gain:
       10
     ------
     (s+11)

» hold                       % grafik yang lalu tetap dipakai
     Current plot held

» step(g2c)                        % input step g2 tertutup
```



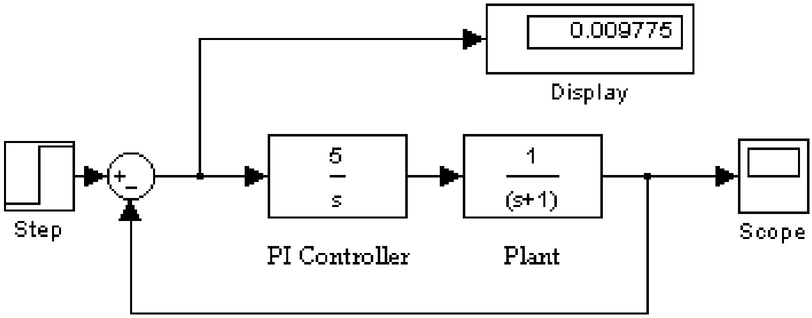
Gambar 8.2 Perbandingan Kontroler P dengan Tanpa Kontroler

Kalau Anda perhatikan dapat disimpulkan bahwa kontroler proportional menambah akurasi (dari kesalahan tunak 0,5 menjadi sekitar 0,1) dan kecepatan/waktu turun (dari 2 detik menjadi 0,5 detik).

B. KONTROLER PROPORTIONAL INTEGRATOR (PI)

Integrator ideal adalah penambahan pole di titik asal (titik nol) pada sistem. Tetapi integrator menambah lambat suatu sistem sehingga dibutuhkan tambahan kontroler proportional.

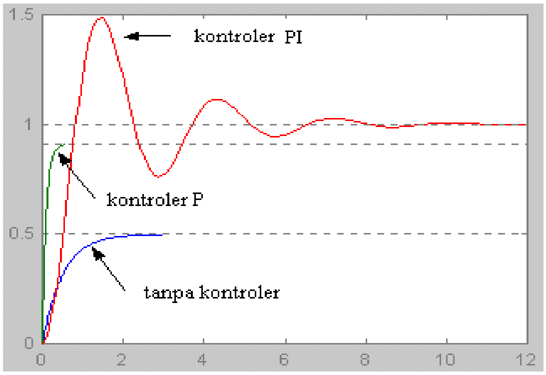
Masih dengan sistem gambar IX.1, misal kita tambah kontroler PI, $PI = \frac{5}{s}$, pada sistem itu. Kita coba dengan cara lain yaitu dengan jendela model.



Gambar 8.3 Efek Kontroler PI (simulasi dengan SIMULINK)

Dan ternyata kesalahan tunaknya berkurang dari 0,5 menjadi mendekati nol (tanpa kesalahan). Untuk melihat kecepatan sistem cobalah double klik gambar SCOPE. Berikut ini cara melihat perbandingan kecepatan lewat command window.

```
» gpi=zpk([],[0 -1],5)          % fungsi alih dengan
  kontroler PI=5/s
      Zero/pole/gain:
                5
             -------
             s (s+1)

» gpic=minreal(gpi/(1+gpi))     % fungsi lingkar tertutup
       Zero/pole/gain:
          5
      -------------
      (s^2 + s + 5)

» step(gpic)
```



Gambar 8.4 Perbandingan Kontroler PI, P dan Tanpa Kontroler

Perhatikan, ternyata PI memiliki kelemahan yaitu berosilasi dan waktu turunnya jadi lama sekitar enam detik. Ingat bahwa integrator murni memakai prinsip menambah pole di titik asal (titik nol) yang pada hakikatnya adalah titik perbatasan antara stabil dan tidak stabil yang sangat berbahaya untuk perancangan. Bergeser saja sedikit ke kanan letak pole integrator, sistem itu sudah tidak stabil.

Ada teknik lain untuk mengatasi hal itu yaitu dengan kontroler lag. Prinsipnya adalah menambah pole di sekitar titik asal (tentu saja di sebelah kirinya) dan untuk mengurangi efek pole tambahan itu, ditambah zero yang letaknya dekat pole tambahan itu. Guna mengurangi efek lag (ketinggalan) dan memperbaiki margin fasa (lihat bab X) ditambahkan kompensator lead, sehingga dikenal kontroler Lag – lead.

C. KONTROLER PROPORTIONAL DIFERENSIATOR (DERIVATIF)

Prinsip dari kontroler ini adalah menambah zero pada sistem kontrol. Penambahan zero ini meningkatkan kecepatan sistem. Kelemahannya adalah pada kontroler ini muncul gangguan noise yang harus diperhitungkan bila sistem bekerja dengan input yang memiliki frekuensi besar. Disamping itu kontroler ini memerlukan catu daya yang berarti menambah biaya.

D. KONTROLER PID

Ini merupakan jenis kontroler termahal gabungan antara proportional, integrator dan diferensiator (disingkat PID) yang rumus umumnya:

$$K_p = \left(1 + \frac{1}{T_i} + T_d s\right) \tag{8.1}$$

dimana $K_p/T_i = K_i$ dan $K_p T_d = K_d$.

Di sini kita menggunakan penulisan:

$$K_p + K_d s + \frac{K_i}{s} \tag{8.2}$$

Atau bila disederhanakan:

$$\frac{K_d s^2 + K_p s + K_i}{s} \tag{8.3}$$

Kita cari harga-harga Kd, Kp dan Ki yang sesuai agar didapat hasil yang terbaik. Dengan bantuan komputer khususnya program MATLAB prinsip *trial and error* jadi lebih cepat. Bayangkan kalau kita menghitung tiap harga K tertentu tentu saja akan banyak menghabiskan waktu.

Trial and Error yang paling nyaman adalah dengan jendela model SIMULINK karena kita tinggal merubah besar K tanpa menulis ulang instruksi, gambarlah pada jendela model sistem tersebut.

E. ATURAN ZIEGLER-NICHOLS

Ziegler dan Nichols mengusulkan aturan-aturan untuk menentukan nilai penguatan proporsional K_p, waktu integral T_i dan waktu turunan T_d yang didasarkan pada karakteristik respon transien suatu sistem yang diketahui. Dasarnya adalah persen overshoot maksimum 25% dan respon terhadap masukan step. Ada dua metode yang mereka lontarkan yang sebagian besar penggunaannya dengan cara eksperimental (trial dan error).

Kita ambil contoh lagi. Misalnya diketahui sistem lingkar tertutup yang fungsi alih lingkar terbukanya,

$$G(s) = \frac{1}{s^2 + 2s + 5}$$

Akan kita rancang dengan spesifikasi:

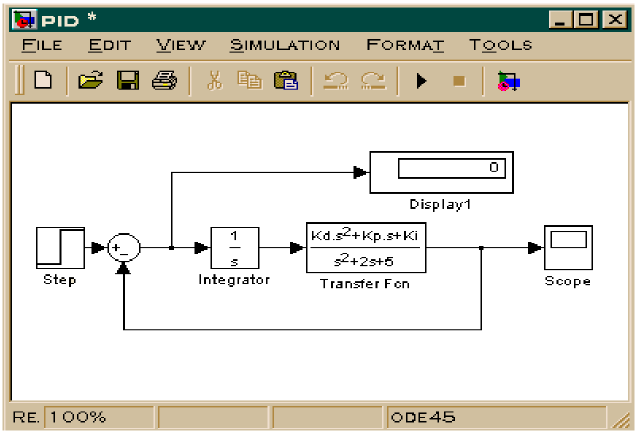
a. Sistem Stabil
b. Mp < 0,3 (30% overshoot)
c. Waktu turun < 3 detik, Kesalahan tunak < 0,1

Langkah yang ditempuh adalah sebagai berikut:

a. Simulasikan sistem tanpa kompensator, apakah sudah sesuai dengan spesifikasi.
b. Rancang kontroler P (Kp), simulasikan!
c. Rancang kontroler PD (Kd), simulasikan! Rancang kontrol PI (Ki), simulasikan!
d. Bila kurang memuaskan, kembali ke langkah 2!

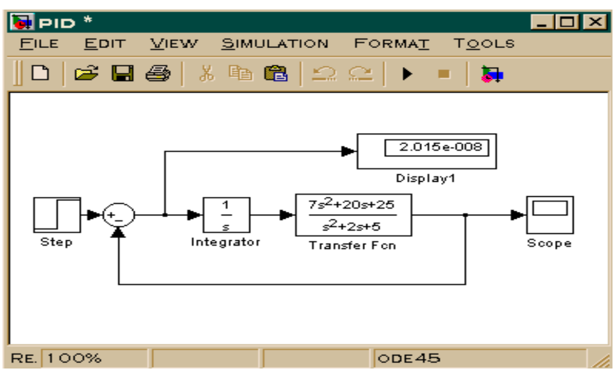
Untuk simulasi yang cepat dapat digunakan jendela model SIMULINK yang tersedia di MATLAB. Tanpa kontroler setelah disimulasi diperoleh kesalahan tunak = 0,8333 dan waktu turun sekitar 4 detik yang berarti masih jauh dari spesifikasi yang diharapkan.

Kita tidak dapat membentuk fungsi alih seperti pada persamaan umum PID, persamaan (8.3), karena fungsi alih tidak mengijinkan orde pembilang lebih besar dari penyebut. Oleh karena itu diperlukan sedikit modifikasi seperti gambar berikut.



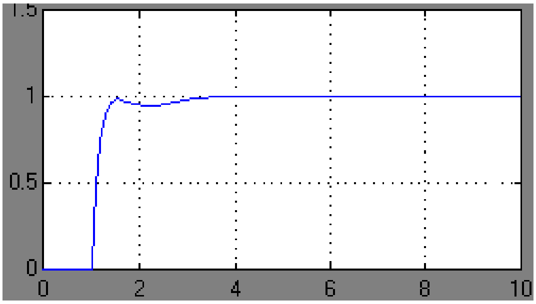
Gambar 8.5 Perancangan Kontroler PID dengan SIMULINK dengan Modifikasi Fungsi Alih

Fungsi alih sistem di atas adalah identik dengan soal hanya saja pembilang PID di pindah ke plant untuk menghindari PID yang orde pemblingnya lebih besar dari penyebut. Double klik pada blok fungsi alih *(transfer function)* dan ganti Kd, Kp dan Ki dengan angka lalu jalankan (RUN). Bila hasilnya belum sesuai dengan spesifikasi yang diharapkan, ganti lagi harga Kp, Kd dan Ki. Berdasarkan tata cara trial and error, sebaiknya cari dahulu Kd (derivatif) yang sesuai baru kemudian Kp dan Ki. Berikut ini hasil trial and error penulis, dan pembaca mungkin saja menemukan hasil yang lebih baik.



Gambar 8.6 Hasil Perancangan kontroler PID

Penulis menemukan kontroler yang sesuai dengan spesifikasi dengan fungsi alih $PID(s) = 20 + 7s + \frac{25}{s}$. Dari gambar dilihat kesalahan tunaknya memenuhui spesifikasi (mendekati nol). Untuk mengetahui waktu turun dan overshoot apakah sudah sesuai, tinggal mendouble klik pada blok scope.



Gambar 8.7 Respon Sistem dengan PID.

Kita lihat pada grafik di atas, overshoot hampir tidak ada dan yang lebih penting kecepatan sistem kontrol (waktu naik kira-kira 1,5 detik dan waktu turun kira-kira 2,2 detik) yang sesuai dengan spesifikasi. Pembaca dapat memahami sendiri saat perancangan kontroler di atas perubahan apa yang terjadi bila harga-harga K di atas dirubah. Mungkin secara tidak sengaja pembaca menemukan kondisi dimana harga-harga K menyebabkan sistem tidak stabil.

Anda juga dapat langsung mendesain blok PID dengan cara klik tanda '+' pada **SIMULINK EXTRAS** dan **ADDITIONAL LINEAR** yang kemudian tampak blok PID CONTROLLER. Tabel berikut merinci efek yang diberikan oleh masing-masing kontroler.

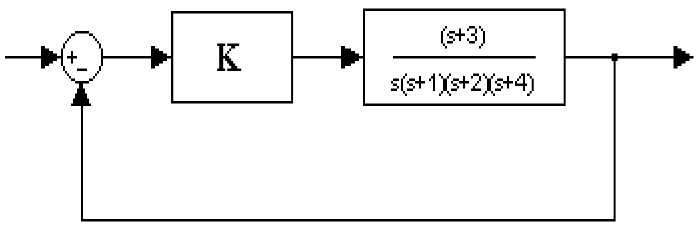
Tabel 8.1 Efek Kontroler Terhadap Kinerja Sistem

Respon L. tertutup	**Waktu Naik**	**Overshoot**	**Waktu Turun**	**Kesalahan Tunak**
K_P	Turun	Naik	Sedikit	Turun

			Berubah	
K_i	Turun	Naik	Naik	Hilang
K_d	Sedikit Berubah	Turun	Turun	Sedikit Berubah

SOAL LATIHAN

Gambarkan grafik sistem Proporsional (P), Integral (I), Derivatif (D), PI, dan PID berdasarkan blok diagram berikut:



DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

BAB 9

RESPON FREKUENSI

Bab yang lalu membahas analisa sistem kontrol dengan masukan step dan impuls tanpa frekuensi. Tetapi banyak alat/mesin yang beroperasi dengan masukan yang melibatkan frekuensi (sinus/kosinus). Jadi bukan penguatan saja yang mempengaruhi kestabilan. Bab ini membahas analisa respon frekuensi dengan diagram bode dan nyquist.

A. DIAGRAM BODE

Diagram bode menggambarkan hubungan penguatan dan fase dengan frekuensi masukan dalam skala logaritmik. Skala logaritmik memiliki keunggulan:

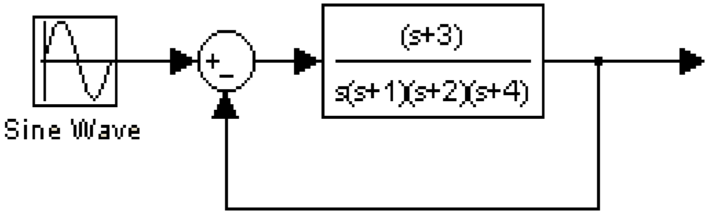
1. Pada titik-titik rendah lebih lebar dan lebih mudah dilihat
2. Perkalian berubah menjadi penjumlahan

Untuk gaya masukan yang sama ternyata penguatan dan pergeseran fasenya akan berbeda jika frekuensi masukannya berbeda. Diagram bode direpresentasikan dalam dua grafik, yaitu grafik magnetudo – frekuensi dan fase – frekuensi.

Analisa diagram bode hanya berlaku untuk sistem yang lingkar terbukanya stabil. Untuk sistem yang lingkar terbukanya tidak stabil, analisa menggunakan diagram nyquist.

1. Diagram Bode dengan MATLAB

Misal kita ingin melihat respon frekuensi sistem :

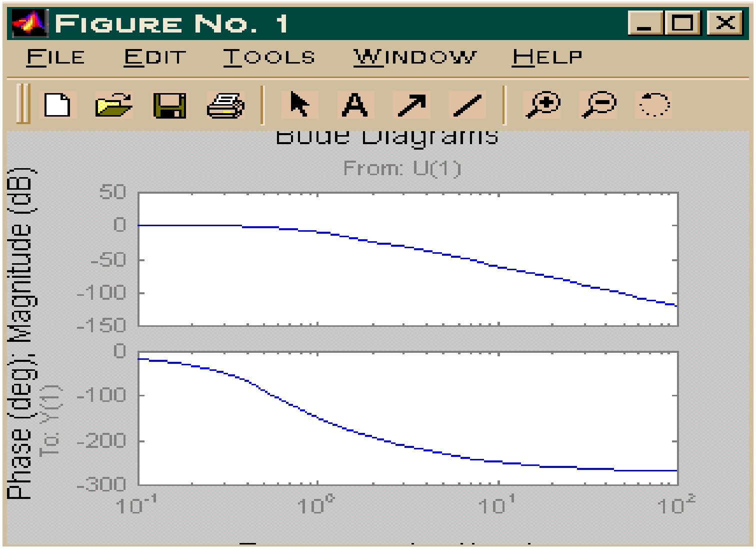


Gambar 9.1 Diagram Blok Sistem $\dfrac{(s+3)}{s(s+1)(s+2)(s+4)}$ dengan penguat K=1.

Buka MATLAB anda dan ketik pada command window sebagai berikut:

```
» g=zpk([-3],[0 -1 -2 -4],1)                    % lingkar terbuka
       Zero/pole/gain:
            (s+3)
       ------------------
       s (s+1) (s+2) (s+4)

» gl=minreal(g/(1+g))                           % lingkar tertutup
        Zero/pole/gain:
                     (s+3)
       ------------------------------------------
       (s+2.176) (s+4.042) (s^2 + 0.7821s + 0.3411)

» bode(gl)                                      % plot diagram
  bode
```

Dan diperoleh grafik dengan karakteristik low pass filter.



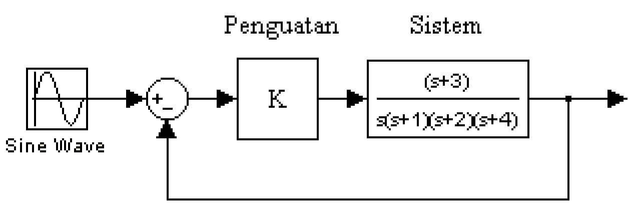
Gambar 9.2 Diagram Bode Sistem Loop Tertutup

Anda dapat mencoba menggambar diagram Bode sistem lain, misalnya untuk sistem high pass filter, coba Anda buat diagram bode sistem dengan fungsi alih loop terbuka

$$G(s) = \frac{s+1}{s+2}\,.$$

2. Margin Penguatan dan Margin Fase

Masih dengan sistem sebelumnya, kita ingin menambah penguat pada sistem itu. Pada bab yang lalu diketahui sistem itu memiliki batas penguatan 9,65 untuk masukan impuls dan step. Tetapi di sini masukannya memiliki frekuensi. Berapakah batasan K?



Gambar 9.3 Diagram Blok Sistem dengan Penguat K

Margin penguatan adalah batasan penguatan lingkar terbuka yang bila dilampaui menyebabkan lingkar tertutupnya tidak stabil. Sistem dengan margin penguatan yang besar berarti dapat menahan perubahan yang besar pula sebelum ketidakstabilan terjadi dalam lingkar tertutupnya. Ingat, karena diagram bode dalam skala logaritmik maka penguatan satu *(magnitude unity gain)* sama dengan ‘zero’ dalam desibel (dB).

Tabel 9.1 Nilai dB untuk beberapa angka

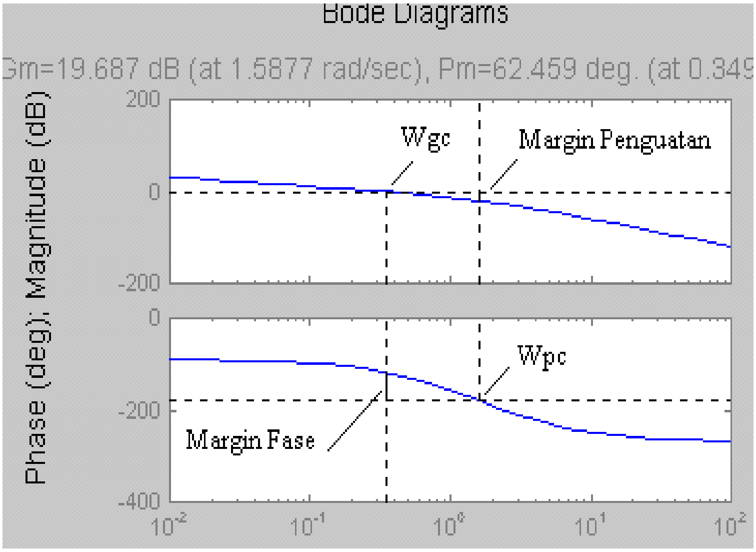
Angka	Desibel (dB)
0,01	-40
0,1	-20
0,5	-6
1	0
2	6
10	20
100	40
200	46

Margin fase didefinisikan sebagai perubahan dalam penggeseran fase lingkar terbuka yang ditetapkan sebelum lingkar tertutupnya tidak stabil. Jadi delay di sini diperhitungkan, dengan toleransi 180/Wpc (dimana Wpc adalah frekuensi dengan pergeseran fase 180 derajat).

Masih melanjutkan instruksi pada command window sebelumnya, masukan perintah sebagai berikut:

» margin(g) % fungsi alih yang dipakai lingkar terbukanya

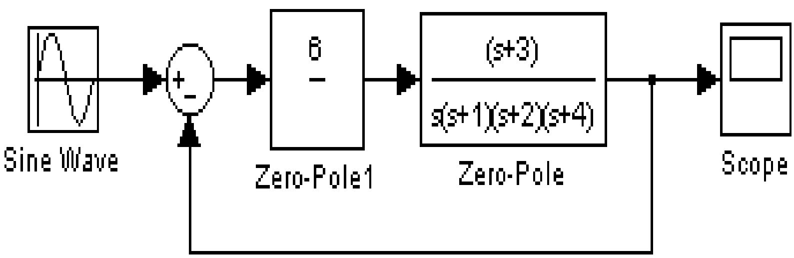
Akan diperoleh grafik seperti di bawah ini.



Gambar 9.4 Margin Penguatan dan Margin Fase Sistem.

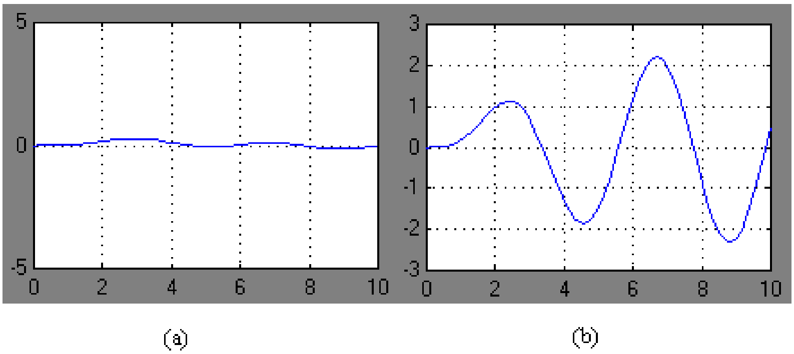
Di atas diagram terdapat informasi mengenai margin penguatan dan fase sistem. Margin penguatan sebesar 19,687 dB pada 1,5877 rad/detik sedangkan margin fase sebesar 61,459 derajat pada 0,34972 rad/detik. Pada Tabel X.1. harga 19,687 dB mendekati angka 1.

Buat simulasi dengan jendela model SIMULINK seperti gambar di bawah ini.



Gambar 9.5 Simulasi dengan SIMULINK

Double klik pada kotak Sine Wave, ambil harga frekuensi 1,5877 rad/detik (frekuensi batas margin penguatan), pertama masukan harga penguat 1, lalu jalankan dan amati dengan cara double klik pada Scope. Hasilnya tampak seperti gambar (a) di bawah ini. Lalu coba Anda ganti penguat dengan harga 6 seperti gambar X.5. Amati kembali keluarannya pada Scope yang dapat Anda lihat pada gambar (b).



Gambar 9.6 Respon Frekuensi Sistem (a). K=1, (b). K=6

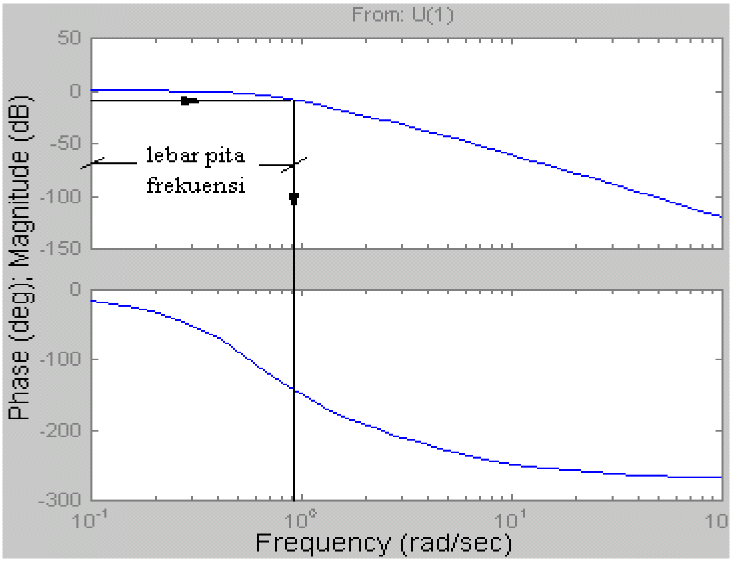
Gambar (a) dengan K=1 ternyata masih stabil, sedangkan (b) dengan penguat 6, sistem membesar terus dan tidak stabil. Pada bab sebelumnya dengan masukan impuls/step, walaupun K=6 (bahkan K=9) sistem tetap sstabil. Jadi kesimpulannya walaupun suatu sistem dengan penguatan tertentu stabil saat diberi masukan impuls/step, belum tentu sistem itu stabil saat diberi masukan berfrekuensi. Sampai di sini Anda harus mulai mengerti pengaruh frekuensi terhadap sistem.

3. Lebar Pita Frekuensi

Lebar pita frekuensi *(frequency bandwidth)* di definisikan sebagai batas frekuensi dimana magnitudo yang dihasilkan sekitar –3 dB. Kita ambil contoh sistem yang sudah-sudah. Berapakah lebar pita frekuensi sistem $\dfrac{(s+3)}{s(s+1)(s+2)(s+4)}$?

Berdasarkan definisi di atas, cara mencarinya adalah dengan menarik garis horisontal pada diagram blok gambar X.2

menyentuh kurva lalu turun ke bawah vertikal hingga menyentuh sumbu frekuensi. Lalu baca berapa frekuensinya. Itulah lebar pita frekuensi sistem tersebut.

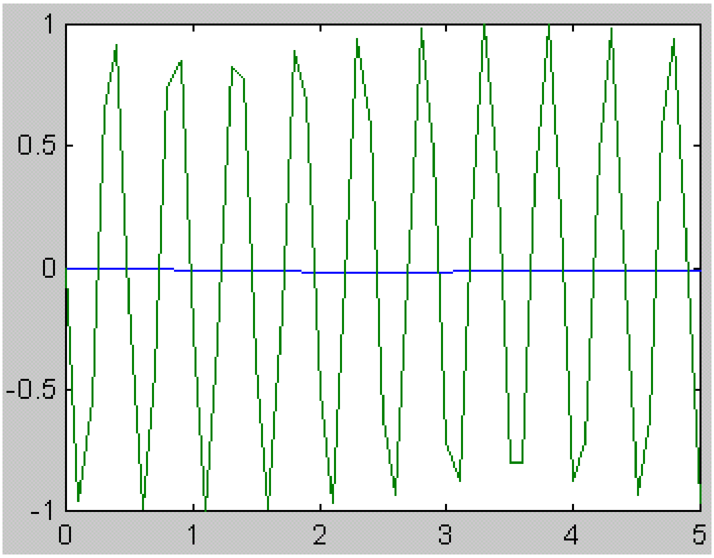


Gambar 9.7 Menentukan Lebar Pita Frekuensi

Apa yang terjadi bila sistem diberi masukan yang frekuensinya melebihi lebar pita? Misalnya kita beri masukan frekuensi 50 rad/detik. Buka command window dan ketikan instruksi sebagai berikut.

```
» w=50;                         % frekuensi 50 rad/detik
» g=zpk([-3],[0 -1 -2 -4],1)    %fungsi alih
     Zero/pole/gain:
          (s+3)
     -----------------
     s (s+1) (s+2) (s+4)

» gc=minreal(g/(1+g))           % fungsi alih lingkar tertutup
     Zero/pole/gain:
                  (s+3)
     ----------------------------------------
     (s+2.176) (s+4.042) (s^2 + 0.7821s + 0.3411)
```

```
» t=0:0.1:100;           % batas waktu
» r=sin(w*t);            % input
» plot(t,y,t,r)
» axis([0, 5,1.1, -1.1])           % range axis
```



Gambar 9.8 Respon pada frekuensi 50 rad/detik.

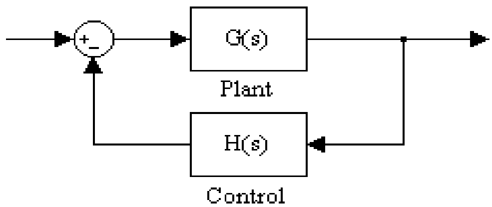
Dan ternyata jika sistem tersebut diberi masukan dengan frekuensi melebihi lebar pita frekuensinya, magnetudo yang dihasilkan kecil sekali (hampir tidak ada respon pada alat).

B. DIAGRAM NYQUIST

Telah disebutkan di awal bab bahwa untuk sistem yang lingkar terbukanya tidak stabil, analisa dengan diagram nyquist adalah pilihan yang tepat. Dasar ilmu yang diperlukan untuk menggambar diagram ini adalah teori variabel kompleks. Tapi dengan MATLAB kita dengan mudah menggambar diagram nyquist.

1. Analisa Kestabilan dengan Diagram Nyquist

Perhatikan sistem berikut.



Gambar 9.9 Blok Diagram dengan umpan balik H(s)

Kriteria Cauchy menyebutkan bahwa jumlah N kali plot G(s)H(s) melingkari –1+ j0 sama dengan jumlah Z zero dari 1+G(s)H(s) frekuensi konturnya dikurangi jumlah P pole dari 1+G(s)H(s) dari frekuensi konturnya (N=Z-P). Kita ketahui bersama bahwa:

a. Zero dari 1+G(s)H(s) adalah pole dari fungsi alih lingkar tertutupnya
b. Pole dari 1+G(s)H(s) adalah pole dari fungsi alih lingkar terbukanya.

Kriteria kestabilan nyquist dinyatakan dengan:

$$Z=P+N$$

dimana:

P = Jumlah pole dari G(s)H(s) di sebelah kanan sumbu khayal bidang s.

N = Berapa kali diagram nyquist mengelilingi titik –1+j0 searah jarum jam (bila searah jarum jam, dihitung sebagai positif dan sebaliknya bila berlawanan dengan arah jarum jam, dianggap negatif).

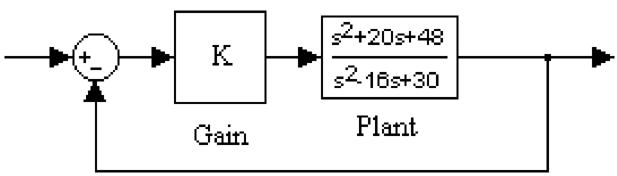
Z = Banyaknya ple (positif, real) di sebelah kanan bidang sistem loop tertutup.

Analisis Kestabilan:

Dalam menguji kestabilan sistem kontrol linear dengan menggunakan kriteria kestabilan Nyquist, kita lihat bahwa ada 3 kemungkinan:

1. Tidak ada pengelilingan titik –1+j0. Ini berarti bahwa sistem stabil jika tidak ada kutub dari G(s)H(s) yang terletak di sebelah kanan sumbu khayal bidang s; jika tidak demikian maka sistem tidak stabil.
2. Ada satu atau lebih pengelilingan titik –1+j0 berlawanan arah jarum jam. dalam hal ini sistem stabil jika banyaknya pengelilingan yang berlawanan arah dengan jarum jam sama dengan banyaknya kutub dari G(s)H(s) yang terletak di sebelah kanan sumbu khayal bidang s, jika tidak maka sistem tidak stabil.
3. Ada satu atau lebih pengelilingan titik –1+j0 yang searah jarum jam. Dalam hal ini sistem tidak stabil.

Jika Anda masih belum memahami, perhatikan contoh ilustrasi di bawah ini.



Gambar 9.10 Merancang Sistem Kontrol dengan Diagram Nyquist

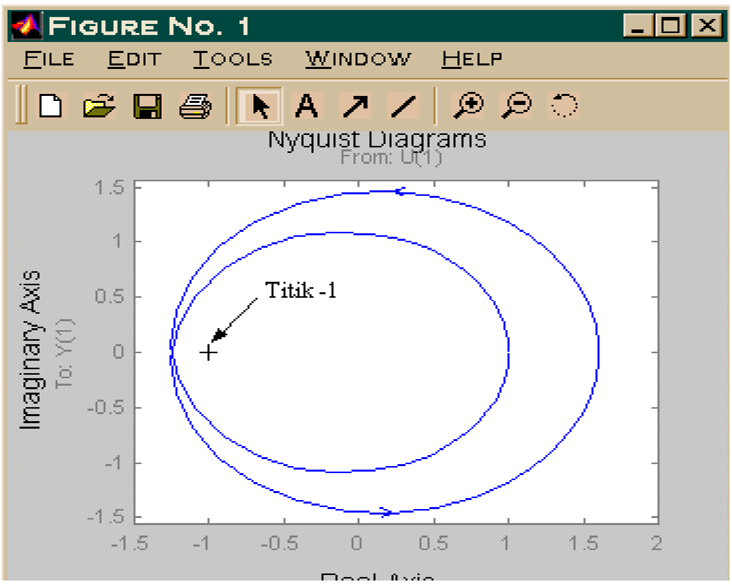
Bab sebelumnya Anda sudah dijelaskan cara menentukan batas K agar sistem stabil dengan teknik tempat kedudukan akar. Tentu saja diagram bode tidak bisa digunakan di sini karena lingkar terbuka sistem di atas tidak stabil, dimana ada dua pole di sebelah kanan sumbu imajiner.

```
» roots([1 -16 30])
ans =
          13.8310
           2.1690
```

Pole lingkar terbukanya yaitu 13,831 dan 2,169 yang semuanya positif. Menurut teori agar stabil kita butuh sistem yang diagram nyquist-nya mengelilingi titik –1+j0 yang berlawanan arah jarum jam sebanya dua kali (N = - 2) agar diperoleh Z = P + N. Pertama, kita ambil K=1.

```
» g=tf([1 20 48],[1 -16 30])          % lingkar terbuka dengan K=1
  Transfer function:
  s^2 + 20 s + 48
  ---------------
  s^2 - 16 s + 30

» nyquist(g)
```

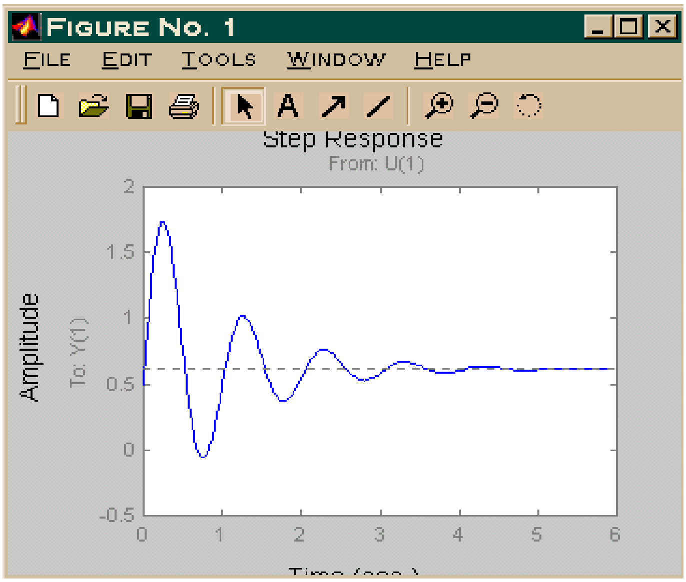
Diperoleh diagram Nyquist sebagai berikut:



Gambar 9.11 Diagram Nyquist dengan K=1

Pada gambar Anda dapat melihat bahwa titik diagram nyquist sistem dengan K=1 mengelilingi titik –1+j0 berlawanan dengan arah jarum jam (lihat panah pada grafik) sebanyak dua kali. Menurut H. Nyquist (penggagas teori kestabilan nyquist) sistem itu stabil dengan syarat penguatan tertentu (lihat point 2 pada analisa kestabilan). Untuk membuktikan kebenaran teori tersebut, lanjutkan program pada command window Anda.

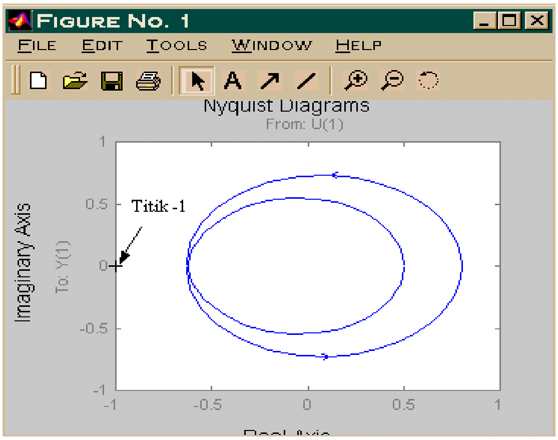
```
» gc=minreal(g/(1+g))        % lingkar tertutup dengan K=1
   Transfer function:
   0.5 s^2 + 10 s + 24
   -------------------
   s^2 + 2 s + 39

» step(gc)
```



Gambar 9.12 Respon Step Sistem dengan K=1

Anda dapat melihat pada grafik di atas bahwa sistem stabil. Dan sekarang ambil harga K=0,5. Apakah sistem lingkar tertutupnya stabil seperti pada K=1? Mari kita coba.

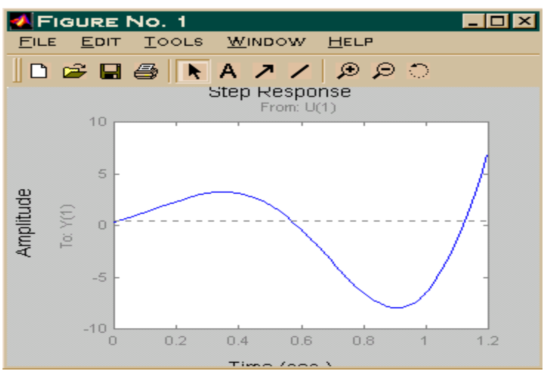
```
» g=tf(0.5*[1 20 48],[1 -16 30])        % lingkar terbuka
  dengan K=0,5
   Transfer function:
   0.5 s^2 + 10 s + 24
   -------------------
   s^2 - 16 s + 30
» nyquist(g)
```



Gambar 9.13 Diagram Nyquist Sistem dengan K=0,5

Stabilkah sisstem dengan grafik di atas? Ternyata tidak ada pengelilingan titik –1+j0, sehingga sistem tidak stabil. Dengan cara yang sama seperti sebelumnya, kita coba buktikan dengan masukan step. Lanjutkan program sebelumnya pada command window Anda.

```
» gc=minreal(g/(1+g))          % lingkar tertutup dengan K=0,5
     Transfer function:
     0.3333 s^2 + 6.667 s + 16
     -------------------------
         s^2 - 4 s + 36
» step(gc)
```

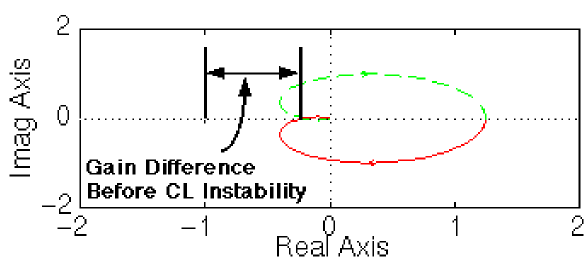


Gambar 9.14 Respon Step Sistem dengan K=0,5

Dan terbukti bahwa sistem tidak stabil, dimana keluaran terus membesar tak hingga. Mudah bukan? Untuk mencari harga K kritis Anda dapat mengulang langkah-langkah di atas untuk K antara 0,5 dan 1.

2. Margin Penguatan dan Margin Fase Pada Diagram Nyquist

Pengertian margin penguatan dan margin fase pada diagram nyquist tidak berbeda dengan diagram bode/margin. Sistem lingkar terbuka akan menjadi tak stabil dalam lingkar tertutup jika penguatan ditingkatkan melewati batas yang telah ditentukan.

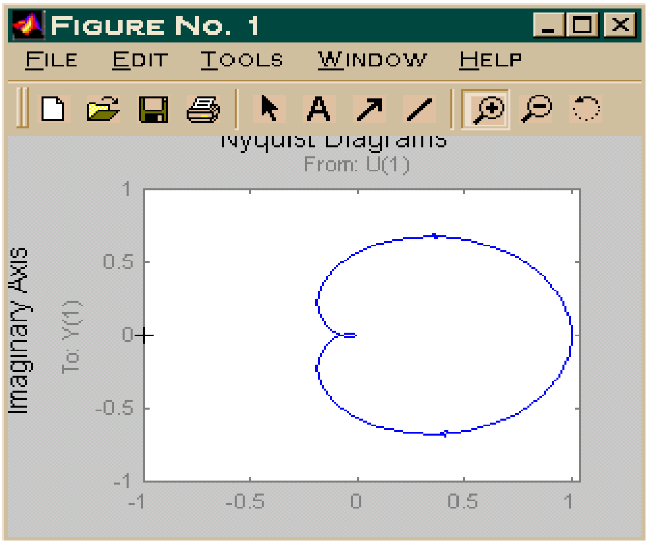


Gambar 9.15 Menentukan Margin Penguatan dengan Diagram Nyquist

Hanya karena grafiknya berbeda, cara membacanya pun berbeda. Kita ambil contoh sistem dengan fungsi alih lingkar terbukanya : $G(s) = \dfrac{10}{s^2 + 2s + 5}$ yang dengan MATLAB (atau Anda bisa menghitung dengan rumus abc untuk akar kuadrat) akar karakteristiknya negatif, sehingga sistem stabil pada lingkar terbukanya.

```
» g=zpk([],[-1 -2 -5],10)          % Fungsi alih dengan penguatan K=10
    Zero/pole/gain:
         10
   -----------------
   (s+1) (s+2) (s+5)
```

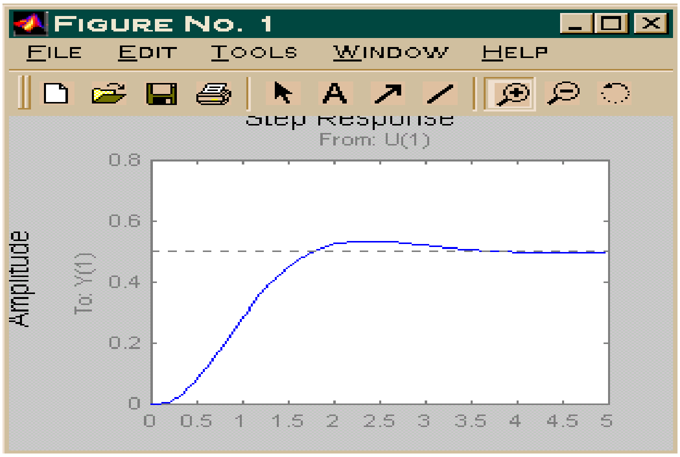
» nyquist(g)



Gambar 9.16 Diagram Nyquist dengan K=10

Perhatikan arah panah pada diagram nyquist di atas yang searah dengan arah jarum jam berbeda dengan sistem sebelumnya. Lihat kriteria kestabilan nyquist point 3. Tidak ada pengelilingan pada titik –1+j0 sehingga sistem stabil pada lingkar tertutupnya. Gunakan perintah step untuk membuktikannya.

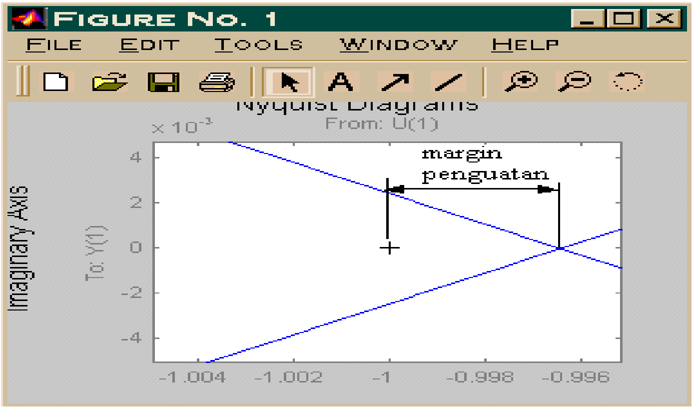
```
» gc=minreal(g/(1+g))                    % Lingkar tertutup
  sistem K=10
      Zero/pole/gain:
               10
     ------------------------------
     (s+5.603) (s^2 + 2.397s + 3.57)

» step(gc)                          % Masukan Step
```



Gambar 9.17 Respon Step Sistem K=10 (stabil)

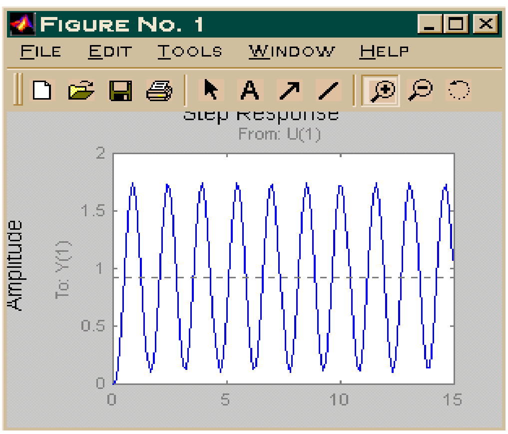
Grafik di atas menunjukan sistem yang stabil pada penguatan K=10. Coba Anda naikan terus penguatannya, misalnya K=200 (ternyata sistem tidak stabil). Berikut ini harga batas K dimana sistem mulai tidak stabil.

```
» g=zpk([],[-1 -2 -5],126)          % Fungsi Alih dengan
  K=126
        Zero/pole/gain:
              126
       ------------------
       (s+1) (s+2) (s+5)

» nyquist(g)                        % Diagram Nyquist
» gc=minreal(g/(1+g))               % Lingkar tertutup
  K=126
        Zero/pole/gain:
             126
       -----------------
       (s+8) (s^2 + 17)
» step(gc)                          % Masukan Step
```



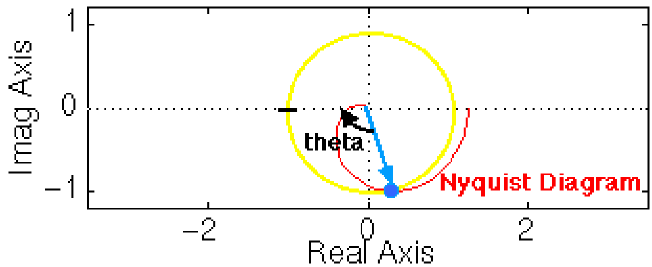
Gambar 9.18 Diagram Nyquist, Diperbesar pada titik –1 +j0

Grafik di atas adalah pembesaran di sekitar titik –1+j0 yang dapat Anda buat dengan mengklik tanda yang berbentuk seperti kaca pembesar. Tarik garis pada titik –1+j0 dengan cara seperti gambar di atas, maka Anda akan memperoleh margin penguatan. Berikut grafik respon stepnya yang merupakan batas, bila K diperbesar lagi sistem akan tidak stabil. Dan harga di atas harus diubah dalam desibel (dB).



Gambar 9.19 Respon Step Sistem K=126 (batas kestabilan)

Margin Fase didefinisikan sebagai perubahan pergeseran fase yang dikehendaki dalam lingkar terbuka pada penguatan terpadu untuk membuat sisstem lingkar tertutup jadi tidak stabil.



Gambar 9.20 Menentukan Margin Fase

Pada contoh sistem sebelumnya kita ketahui bahwa sistem tersebut akan tak stabil pada lingkar tertutup jika diagram nyquist mengelilingi titik –1 + j0.

Akan tetapi kita harus sadar bahwa jika diagram bergeser sebesar theta derajat maka akan menyentuh titik –1 + j0 pada sumbu nyata negatif, membuat sistem stabil pada lingkar tertutup. Sudut yang dikehendaki untuk membuat batas sistem stabil pada lingkar tertutupnya dinamakan margin fase (diukur dalam derajat).

Sebagai latihan, Anda dapat mencoba mencari margin penguatan dan margin fase dengan diagram bode/margin. Berikut ini tabel yang memuat instruksi-instruksi pada MATLAB untuk analisa respon frekuensi.

Tabel 9.2 Fasilitas MATLAB untuk Analisa Respon Frekuensi

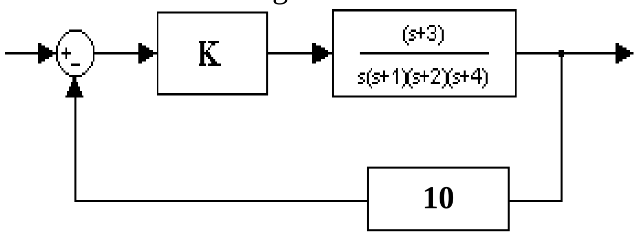
Bode	grafik Bode respon frekuensi
Sigma	grafik nilai frekuensi singular
nyquist	grafik Nyquist
nichols	grafik Nichols
Ltiview	analisa respon berbasis GUI
Evalfr	mengevaluasi respon frekuensi pada frekuensi tertentu
freqresp	respon frekuensi dalam suatu grid frekuensi
margin	perolehan dan margin fase

SOAL LATIHAN

Bagaimana representasi grafik:

1. Diagram bode
2. Respon frekuensi, dan
3. Diagram nyquist

Dari persamaan dan blok diagram sistem kendali berikut ini.



DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

BAB 10

ANALISA SISTEM KONTROL DALAM RUANG KEADAAN

Sejauh ini kita telah menganalisa sistem kontrol dengan teknik konvensional atau sering disebut metode klasik yang cirinya adalah satu masukan dan satu keluaran *(SISO).* Ternyata metode klasik tidak memadai untuk sistem-sistem kontrol yang banyak dirancang saat ini. Misalnya sistem pemanas air, ternyata tidak hanya dipengaruhi oleh laju pemanasan saja, melainkan juga oleh laju aliran air. Pada AC, ternyata pendinginan tidak hanya dipengaruhi sistem pendinginnya melainkan juga oleh kecepatan kipas dan gerak sirip.

Pada bab ini kita akan membahas sistem dengan masukan dan keluaran yang lebih dari satu *(MIMO)* dengan metode yang disebut metode ruang keadaan.

A. MEMBENTUK PERSAMAAN RUANG KEADAAN

Ada baiknya Anda buka kembali buku tentang matriks dan vektor, karena persamaan keadaan adalah persamaan dalam bentuk matriks. Jika x adalah variabel keadaan suatu sistem, maka persamaan keadaannya didefinisikan sebagai berikut:

$$
\begin{aligned}
\dot{x}_1 &= X_1(x_1, x_2, ..., x_n) \\
\dot{x}_2 &= X_2(x_1, x_2, ..., x_n) \\
&\vdots \\
\dot{x}_n &= X_n(x_1, x_2, ..., x_n)
\end{aligned}
$$

Dengan syarat bahwa untuk setiap himpunan nilai $\dot{x}_1, \dot{x}_2, ..., \dot{x}_n$ terdapat suatu himpunan nilai x_1, x_2, x_3, ..., x_n yang unik dan sebaliknya. Jadi jika x merupakan vektor keadaan maka : $\dot{x} = Px$ juga merupakan vektor keadaan, dengan syarat bahwa matriks P nonsingular. Vektor-vektor keadaan yang berbeda membawa informasi yang sama mengenai perilaku sistem.

Biasanya kita lebih mudah memahami sesuatu dari contoh. Misal sistem dengan persamaan diferensial :

$$\dddot{y} = 6\ddot{y} + 11\dot{y} + 6y = 6u \qquad (11.1)$$

dengan y adalah keluaran dan u adalah masukan sistem. Kita pilih variabel baru sebagai berikut:

$$
\begin{aligned}
x_1 &= y \\
x_2 &= \dot{y} \\
x_3 &= \ddot{y}
\end{aligned}
$$

Kemudian kita dapatkan:

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= -6x_1 - 11x_2 - 6x_3 + 6u
\end{aligned}
$$

Lihat persamaan terakhir di atas adalah hasil penyelesaian $\dddot{y}$ dari persamaan diferensial pada soal. Kemudian kita buat matriks ruang keadaannya dengan 3 persamaan diferensial.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} u \qquad (10.2)$$

Persamaan keluaran diberikan oleh:

$$y = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad (10.3)$$

Persamaan (11.2) dan (11.3) dapat dituliskan dalam bentuk standar sebagai berikut:

$$\begin{aligned} y &= Ax + Bu \\ y &= Cx \end{aligned} \qquad (10.4)$$

dengan

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

B. KETIDAKUNIKAN PERSAMAAN KEADAAN

Persamaan keadaan (11.4) ternyata bukan satu satunya persamaan yang merepresentasikan sistem tersebut. Kita akan coba membentuk persamaan yang lain dari sistem itu. Bentuk fungsi alihnya dalam laplace.

$$\frac{Y(s)}{U(s)} = \frac{6}{s^3 + 6s^2 + 11s + 6} = \frac{6}{(s+1)(s+2)(s+3)}$$

Coba Anda cari pecahan parsialnya dengan MATLAB, Anda akan memperoleh:

$$\frac{Y(s)}{U(s)} = \frac{3}{s+1} + \frac{-6}{s+2} + \frac{3}{s+3}$$

Sehingga didapat

$$Y(s)=\frac{3}{s+1}U(s)+\frac{-6}{s+2}U(s)+\frac{3}{s+3}U(s) \qquad (10.5)$$

Marilah kita definisikan

$$X_1(s)=\frac{3}{s+1}U(s) \qquad (10.6)$$

$$X_2(s)=\frac{3}{s+2}U(s) \qquad (10.7)$$

$$X_3(s)=\frac{3}{s+3}U(s) \qquad (10.8)$$

Dengan membalik transformasi laplace persamaan (10.6), (10.7) dan (10.8) kita peroleh:

$$\begin{aligned}\dot{x}_1 &= -x_1 + 3u\\ \dot{x}_2 &= -2x_2 + 6u\\ \dot{x}_3 &= -3x_3 + 3u\end{aligned}$$

Dalam bentuk notasi matriks vektor kita dapatkan:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}-1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & -3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}3\\ -6\\ 3\end{bmatrix}u \qquad (11.9)$$

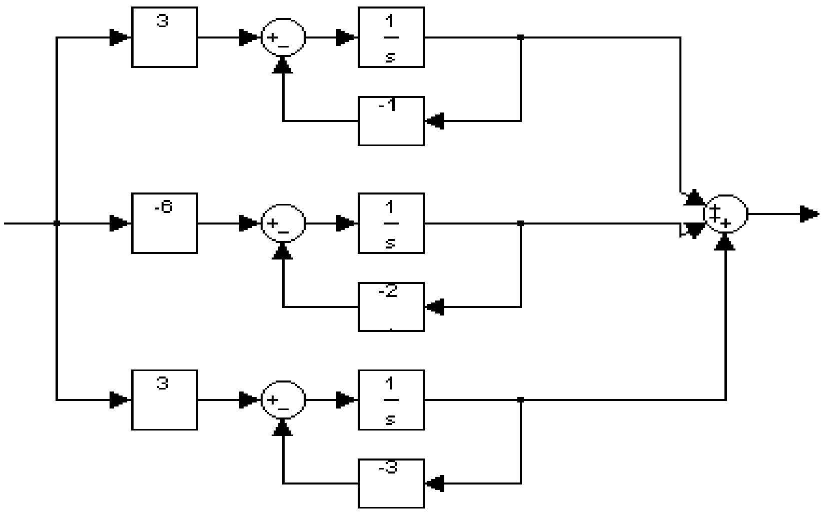
Berdasarkan persamaan (10.5), diperoleh persamaan keluarannya:

$$y = x_1 + x_2 + x_3$$

atau:

$$y=\begin{bmatrix}1 & 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \qquad (10.9)$$

Gambar diagram bloknya adalah sebagai berikut:



Gambar 10.1 Diagram Blok Sistem dengan Persamaan Keadaan (10.9)

Mengapa persamaan (10.9) dan persamaan (10.2) berbeda padahal keduanya merupakan representasi dari sistem yang sama? Hal ini terjadi karena cara membentuk persamaan ruang keadaan kedua persamaan itu berbeda. Sehingga tiap persamaan ruang keadaan bukan merupakan persamaan yang unik. Kasus tersebut terjadi karena invariansi nilai eigen.

Tetapi walaupun tiap persamaan keadaan tidak unik, baik persamaan (10.2) maupun (10.9) memiliki nilai eigen yang sama yaitu –1, -2 dan –3. Buka kembali command window MATLAB anda.

```
» A1=[0 1 0; 0 0 1; -6 -11 -6]          % Matriks A
  persamaan (10.2)
     A1 =
        0     1     0
        0     0     1
       -6   -11    -6
» eig(A1)
```

```
ans =
   -1.0000
   -2.0000
   -3.0000
» A2=[-1 0 0; 0 -2 0; 0 0 -3]        % Matriks A persamaan (10.9)
A2 =
   -1   0   0
    0  -2   0
    0   0  -3
» eig(A2)
ans =
   -1
   -2
   -3
```

Lihat, terbukti bahwa nilai eigen kedua persamaan sama. Bagaimana jika nilai eigen kedua persamaan itu berbeda? Bila hal ini terjadi berarti kedua persamaan yang nilai eigennya berbeda itu bukan representasi sistem yang sama.

C. MERUBAH FUNGSI ALIH MENJADI RUANG KEADAAN

Masih sistem dengan persamaan (11.1) dengan fungsi alih:

$$\frac{Y(s)}{U(s)} = \frac{6}{s^3 + 6s^2 + 11s + 6} = \frac{6}{(s+1)(s+2)(s+3)}$$

Berikut instruksi pada command window guna merubah fungsi alih menjadi persamaan ruang keadaan.

```
» g=tf([6],[1 6 11 6])
 Transfer function:        6
                   ----------------------
                   s^3 + 6 s^2 + 11 s +
» ss(g)
 a =
```

	x1	x2	x3
x1	-6	-2.75	-0.75
x2	4	0	0
x3	0	2	0

b =

	u1
x1	1
x2	0
x3	0

c =

	x1	x2	x3
y1	0	0	0.75

d =

	u1
y1	0

Anda lihat, ternyata matriks A berbeda sehingga persamaan ruang keadaannya berbeda baik dengan persamaan (10.2) maupun (10.3). Tetapi sesungguhnya persamaan itu sama karena nilai eigen matriks a di atas:

```
» A3=[-6 -2.75 -0.75; 4 0 0; 0 2 0]
   A3 =
   -6.0000   -2.7500   -0.7500
    4.0000         0         0
         0    2.0000         0
» eig(A3)
   ans =
   -3.0000
   -2.0000
   -1.0000
```

Ternyata nilai eigen persamaan ruang keadaan yang baru kita bentuk lewat MATLAB sama dengan nilai eigen kedua persamaan keadaan sebelumnya. Nilai eigen sering disebut akar karakteristik sistem.

Bentuk standar MATLAB dalam merubah fungsi alih menjadi ruang keadaan adalah sebagai berikut:

```
» num=[6];                        % Pembilang Fungsi Alih
» den=[1 6 11 6];                 % Penyebut Fungsi Alih

» [A,B,C,D]=tf2ss(num,den)        % Merubah Fungsi Alih
  menjadi Ruang Keadaan
    A =
       -6  -11   -6
        1    0    0
        0    1    0
    B =
        1
        0
        0
    C =
        0    0    6
    D =
        0
```

D. MERUBAH PERSAMAAN RUANG KEADAAN MENJADI FUNGSI ALIH

Tentu saja kita dapat merubah persamaan keadaan menjadi fungsi alih dalam variabel s.

Format penulisan:

[n,d]=ss2tf(A,B,C,D,iu)

dengan iu merupakan masukan yang lebih dari satu.

Masih dengan contoh sebelumnya, coba Anda rubah kembali ke fungsi alihnya dengan instruksi sebagai berikut:

```
» [num,den]=ss2tf(A,B,C,D)

    num =
           0    0.0000    0.0000    6.0000
    den =
```

1.0000   6.0000   11.0000   6.0000

Ternyata hasilnya tepat sesuai dengan fungsi alih semula. Kita ambil contoh yang lebih kompleks.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -25 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Mengikuti format penulisannya, masukkan instruksi pada command window sebagai berikut:

```
» A=[0 1; -25 -4];
» B=[1 1; 0 1];
» C=[1 0; 0 1];
» D=[0 0; 0 0];
» [n,d]=ss2tf(A,B,C,D,1)

    n =
        0     1     4
        0     0   -25
    d =
        1     4    25

» [n,d]=ss2tf(A,B,C,D,2)
    n =
            0    1.0000    5.0000
            0    1.0000  -25.0000
    d =
        1     4    25
```

Apa maksud hasil keluaran pada MATLAB di atas?. Simbol n berarti pembilang dan d berarti penyebut (singkatan dari numerator dan denumerator). Berikut adalah pembacaan dari hasil keluaran MATLAB tersebut.

Terhadap masukan pertama u_1:

$$\frac{Y_1(s)}{U_1(s)} = \frac{s+4}{s^2+4s+25}$$

$$\frac{Y_2(s)}{U_1(s)} = \frac{-25}{s^2+4s+25}$$

Terhadap masukan kedua u_2:

$$\frac{Y_1(s)}{U_2(s)} = \frac{s+5}{s^2+4s+25}$$

$$\frac{Y_2(s)}{U_2(s)} = \frac{s-25}{s^2+4s+25}$$

E. MENGGAMBAR DIAGRAM BLOK PERSAMAAN RUANG KEADAAN DENGAN SIMULINK

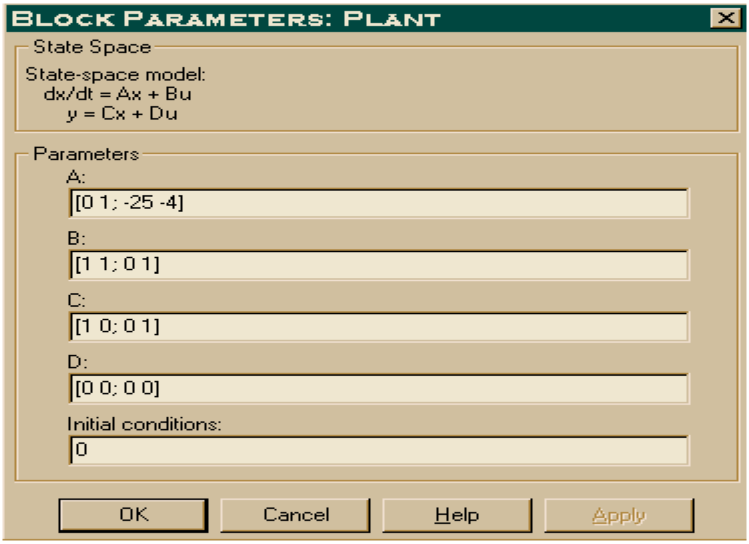
Masih ingat cara membuka jendela model? Pada bab terdahulu Anda telah diperkenalkan cara menggambar diagram blok dengan fasilitas yang ada pada MATLAB yaitu SIMULINK. Versi terakhir SIMULINK saat buku ini ditulis adalah versi 6. Seperti biasa, klik **File** – **New** – **Model** akan membawa Anda ke jendela model. Misal kita ingin melihat respon sistem dengan persamaan ruang keadaan sistem contoh bab 11.4.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -25 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Pertama-tama kita ingin menggambar blok diagram persamaan ruang keadaan di atas. Pada Simulink Library Browser, klik tanda ‘+’ **SIMULINK** dan **CONTINUOUS**, yang akan memunculkan menu-menu yang termasuk di dalamnya

STATE-SPACE. Klik dan drag menu state-space ke jendela model. Double klik pada state-space saat Anda ingin memasukkan persamaan ruang keadaan. Masukan harganya seperti di bawah ini. Perhatikan spasi antar elemen matriks parameter.

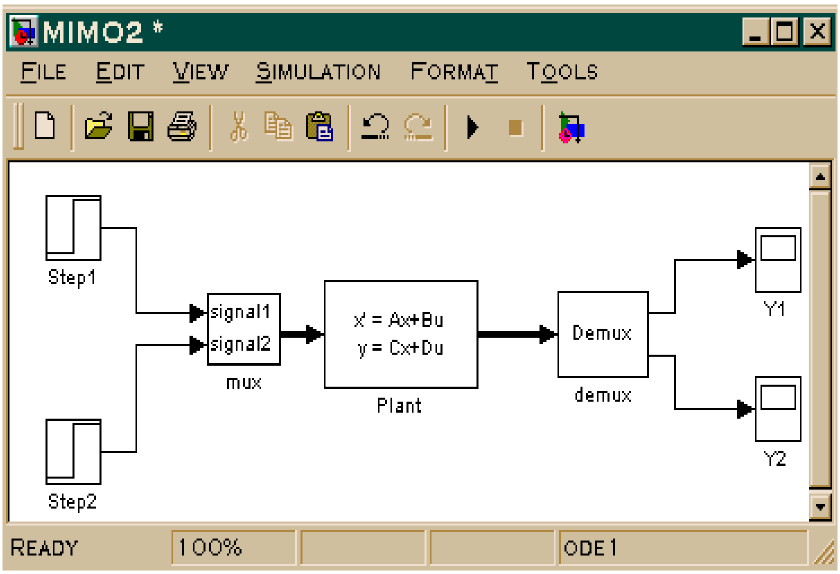


Gambar 10.2 Menu Input Persamaan Ruang Keadaan Pada SIMULINK

Sistem dengan dua masukan dan dua keluaran mengharuskan kita menerapkan blok khusus Multiplexing dan Demultiplexing. Klik tanda '+' pada **SIMULINK** dan **SIGNALS AND SYSTEMS**. Tampak di sana blok **MUX** dan **DEMUX**. Klik dan drug ke jendela model, dimana MUX di bagian input Plant sedangkan DEMUX di bagian output. Double klik pada blok MUX dan DEMUX rubah sesuai dengan sistem di atas dimana jumlah input dan output kita pilih 2.

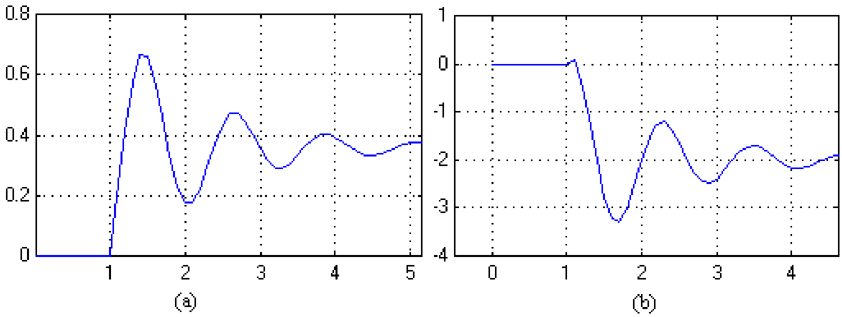
Mungkin saat Anda mengklik dan drug MUX tampak bentuk yang berbeda dengan bentuk di gambar XI.3. Oleh karena itu Anda harus memperbesar ukuran dengan cara mengklik blok tersebut lalu menggeret ujungnya agar membesar.

Sistem kita memerlukan dua buah masukan step, dan kita ingin mengetahui responnya pada dua terminal keluaran. Buka Simulink Library Browser lagi, Klik tanda '+' pada **SIMULINK** dan **SOURCES** lalu pilih **STEP**. Setelah klik dan drug, rubah setting dengan double klik, dan isilah agar lebih akurat **sample time** pada menu properties harga yang kecil, misalnya 0,1. Buat setting untuk kedua masukan. Untuk melihat keluaran buat blok scope dengan klik tanda '+' pada **SIMULINK** dan **SINKS** lalu klik dan drug **SCOPE** ke jendela model Anda. Buat seperti pada gambar berikut.



Gambar 10.3 Penggambaran dengan SIMULINK

Lalu jalankan dengan mengklik icon ► dan bila ingin berhenti klik icon ■. Lalu untuk melihat hasilnya double klik blok SCOPE **Y1** dan SCOPE **Y2**. Untuk memperjelas dan memperbesar grafik klik pada bagian yang ingin Anda lihat. Hasilnya akan tampak seperti gambar di bawah ini (Warna putih pada background dibuat dengan invert color).



Gambar 10.4 Keluaran pada SCOPE

Gambar (a) adalah keluaran pada SCOPE Y1 sedangkan gambar (b) keluaran pada SCOPE Y2. Cobalah Anda ganti masukan dengan input yang lain, misalnya: sinus, random, ramp dan lain-lain. Berlatihlah dengan mencoba menggambar blok sistem multi masukan dan multi keluaran yang lain.

Tabel berikut memuat fungsi lain yang berguna dalam analisa sistem kontrol dalam ruang keadaan.

Tabel 10.1 Model Status-Ruang

ras,drss	model status-ruang stabil acak
ss2ss	transformasi koordinat status
Canon	bentuk kanonik status-ruang
ctrb,obsv	matriks pengontrolan dan pengamatan
Gram	gramian pengontrolan dan pengamatan
Ssbal	realisasi penyeimbangan diagonal status-ruang
Balreal	penyeimbangan input-output berbasis gramian
Modred	reduksi model status
Mineral	realisasi minimal dan pembatalan kutub/nol
Augstate	penambahan output dengan menambahkan status

SOAL LATIHAN

Tulis dan gambarkan kondisi dan kinerja sistem kontrol dalam ruang keadaan dari persamaan berikut:

$$\frac{R(s)}{C(s)} = \frac{2{,}5}{s^3 + 2s^2 + 8s + 6} = \frac{5}{(s+1)(s+2)(s+4)}$$

DAFTAR PUSTAKA

[2] Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

[3] D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

[4] Dorf. 2003. Modern Control System. Addison Wesley

[6] Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

[8] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

DAFTAR PUSTAKA

Anoname. 2004. Matlab Tutorial. Diambil dari: http://www.engin.umich.edu/ group/ctm.

Bryan, G. T. 1977. Control System for Technicians. Penerbit Hodder and Stoughton. London.

D'Azzo, John J., Constantine H. Houpis. 1981. Linear Control System Analysis and Design. Penerbit Mc Graw Hill. London.

Dorf. 2003. Modern Control System. Addison Wesley

Eko Mursito Budi, Manase Sitorus, Estiyanti Ekawati. 2006. Simulator Untuk Pengajaran Sistem Kontrol. Kelompok Keahlian Instrumentasi & Kontrol ITB, Bandung

Gajic, C, M. Lelic. 1996. Modern Control System Engineering. Prentice Hall. London.

Hanselman, Duane., Bruce Littlefield. 1997. Matlab, Bahasa Komputasi Teknis (terj.). Penerbit Andi. Yogyakarta.

Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik (terj.) jilid 1 & 2. Penerbit Erlangga. Jakarta.

Pakpahan, Sahat. (1994). ***Kontrol Otomatik (Teori Dan Penerapan).*** Erlangga, Jakarta.

Siswosudarmo, Muhammadi., R. Gatot Prio Utomo. 1995. Dasar Sistem Kendali (terj.). Penerbit Universitas Indonesia. Jakarta.

Suhendar, 2005, Programmable Logic Controller dalam Dasar-Dasar Sistem

Kendali Motor Listrik Induksi, Graha Ilmu, Yogyakarta