

3100098010495

**SISTEM PENGATURAN IMPEDANSI ROBOT  
BERDASARKAN PADA  
PENGATURAN POSISI KOKOH**

**TUGAS AKHIR**

Disusun oleh :

**ADI SARWANTO**

**NRP : 2291.100.094**

RSE  
629.892  
Adi  
s-1  

---

1997



PERPUSTAKAAN	
Tgl. Terima	14 Nov. 97
Tgl. Kembali	H
No. Register	7956

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1997**

**SISTEM PENGATURAN IMPEDANSI ROBOT  
BERDASARKAN PADA  
PENGATURAN POSISI KOKOH**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan**

**Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Teknik Sistem Pengaturan**

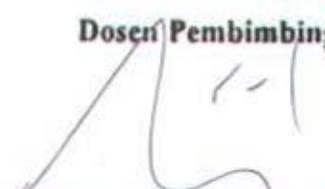
**Jurusan Teknik Elektro**

**Fakultas Teknologi Industri**

**Institut Teknologi Sepuluh Nopember**

**Mengetahui / Menyetujui :**

**Dosen Pembimbing**

  
**DR. Ir. Achmad Jazidie, M.Eng**  
**NIP. 131 647/788**

**SURABAYA**

**AGUSTUS 1997**

## ABSTRAK

---

---

Pengaturan impedansi robot (Impedance Control) adalah salah satu cara yang efektif untuk mengendalikan gaya interaksi antara robot dengan lingkungan. Pada pengaturan impedansi robot konvensional diperlukan data-data robot yang akurat, sehingga robot dapat bekerja dengan baik dan memenuhi hubungan dinamik antara gaya dengan posisi yang diinginkan. Tetapi sangat sulit untuk mendapatkan data robot yang akurat, sehingga akan muncul error yang tidak diharapkan. Hal ini akan membuat robot bergerak dengan impedansi end effector yang tidak sesuai dengan impedansi yang diinginkan bahkan dapat menimbulkan kerusakan pada robot tersebut. Dalam tugas akhir ini dikembangkan suatu sistem pengaturan impedansi yang mampu menekan error yang ditimbulkan oleh ketidaktepatan dari data-data robot dan ketidakpastian gangguan dari luar. Sistem pengaturan impedansi tersebut dikembangkan berdasarkan sistem pengaturan posisi kokoh dimana target posisinya ditentukan melalui target impedansi. Pertama, teori tentang sistem pengaturan impedansi konvensional dibahas secara garis besar dilanjutkan dengan pembahasan tentang sistem pengaturan posisi kokoh. Selanjutnya diberikan pembahasan mengenai implementasi sistem pengaturan posisi kokoh pada sistem pengaturan impedansi. Akhirnya, efektifitas dan validitas dari metode yang dikembangkan diuji dengan simulasi komputer.

---

---

## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur kami panjatkan kepada Allah Yang Maha Pengasih dan Penyayang, yang telah melimpahkan rahmat dan petunjukNya, sehingga penulis dapat menyelesaikan tugas akhir sebagai salah satu syarat memperoleh gelar sarjana Jurusan Teknik Elektro.

Dalam menyusun tugas akhir ini, penulis mendapat bimbingan serta bantuan atau saran dari berbagai pihak. Dan atas terselesaikannya tugas akhir ini penulis ingin menyampaikan rasa terima kasih yang sedalam-dalamnya kepada :

- Bapak Dr. Ir. Achmad Jazidie M.Eng. , selaku Dosen Pembimbing dalam penyusunan tugas akhir ini yang telah banyak sekali memberikan arahan, nasehat, bimbingan dan semangat kepada penulis.
- Bapak Dr. Ir. Moch Rameli , selaku Koordinator Bidang Studi Teknik Sistem Pengaturan yang telah banyak memberikan fasilitas kepada penulis.
- Bapak dan ibu dosen pada Bidang Studi Teknik Sistem Pengaturan pada khususnya dan Jurusan Teknik Elektro pada umumnya.
- Ayah beserta ibu, Tiyok, dan Eyang Kakung yang selalu memberikan dorongan kepada penulis.
- Rekan-rekan di Bidang Studi Teknik Sistem Pengaturan, Wiku, Zul, Kojar, Yusuf, Ancha, Toni, Sahal, Mak, Awik, Ali, Arif, Abed , Okim,

Dodo, Romli, Yoyok, Yana dan para alumni, Johar, Firda, Onni, juga rekan-rekan lainnya yang tidak mungkin disebutkan seluruhnya di sini, atas spirit dan segala dukungannya.

- Rekan-rekan angkatan E-31 yang selalu setia memberikan hiburan, dukungan dan gangguan yang semuanya sangat berarti bagi penulis.
- Rekan-rekan yang telah memberikan kenangan manis maupun kenangan pahit yang tidak akan terlupakan.

Besar harapan penulis semoga tugas akhir ini bermanfaat bagi para pembaca pada umumnya serta mahasiswa Teknik Elektro pada khususnya. Penulis menyadari segala kekurangan yang ada sehingga tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran selalu penulis harapkan.

Akhirnya sekali lagi puji syukur kehadirat Allah SWT.

Wassalamu'alaikum Wr. Wb.

Surabaya, Juli 1997

Penulis.

# DAFTAR ISI

	halaman
Abstrak .....	ii
Kata Pengantar .....	iii
Daftar Isi .....	v
Daftar Gambar .....	vii
Daftar Tabel .....	ix
<b>BAB I</b>	
<b>PENDAHULUAN .....</b>	
<b>1</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Pembatasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Sistematika Pembahasan .....	4
<b>BAB II</b>	
<b>TEORI PENGATURAN IMPEDANSI ROBOT .....</b>	
<b>5</b>	
2.1 Pendahuluan .....	5
2.2 Pengendalian Gabungan ( <i>Hybrid Control</i> ) .....	6
2.3 Pengaturan Impedansi Robot .....	8
2.4 Implementasi Pengaturan Impedansi Robot .....	11

**BAB III****PENGATURAN POSISI KOKOH PADA KOORDINAT CARTESIAN .... 15**

3.1 Pendahuluan ..... 15

3.2 Kontroler Posisi Kokoh Nonlinier ..... 17

**BAB IV****IMPLEMENTASI PENGATURAN POSISI KOKOH PADA  
PENGATURAN IMPEDANSI ROBOT ..... 25**

4.1 Pengaturan Impedansi Robot Berdasarkan Sistem Pengaturan Posisi ..... 25

4.2 Implementasi Pengaturan Posisi Kokoh

Pada Pengaturan Impedansi Robot ..... 26

**BAB V****EKSPERIMEN DENGAN SIMULASI ..... 28**

5.1 Pengaturan Impedansi Konvensional ..... 29

5.2 Pengaturan Posisi Kokoh Pada Pengaturan Impedansi Robot ..... 32

**BAB VI****PENUTUP ..... 41**

Kesimpulan ..... 41

Saran ..... 42

**DAFTAR PUSTAKA ..... 43****LAMPIRAN ..... 46**

## DAFTAR GAMBAR

	halaman
Gambar 2.1 Memasukkan pasak dalam lubang .....	7
Gambar 2.2 Diagram blok pengaturan impedansi konvensional .....	14
Gambar 3.1 Pengaturan posisi servo pada koordinat Cartesian .....	19
Gambar 3.2 Diagram blok pengaturan posisi kokoh nonlinier .....	24
Gambar 4.1 Diagram blok pengaturan posisi robot .....	25
Gambar 4.2 Diagram blok pengaturan impedansi berdasarkan pengaturan posisi robot .....	26
Gambar 4.2 Diagram blok pengaturan posisi kokoh pada pengaturan impedansi .....	27
Gambar 5.1 Posisi awal robot .....	29
Gambar 5.2 Respon kontroler impedansi konvensional di mana data-data robot diketahui secara pasti .....	30
Gambar 5.3 Respon kontroler impedansi konvensional di mana data-data robot diperbesar 80% kecuali panjang <i>link</i> .....	31
Gambar 5.4 Respon kontroler impedansi konvensional di mana panjang <i>link</i> diperbesar 0,005 m dan data-data robot diperbesar 80%.....	31
Gambar 5.5 Respon menggunakan metode yang diusulkan di mana data-data robot diperbesar 80% kecuali panjang <i>link</i> .....	35

Gambar 5.6 Respon menggunakan metode yang diusulkan di mana panjang *link*  
diperbesar 0,005 m dan data-data robot diperbesar 80% ..... 36

## DAFTAR TABEL

	halaman
Tabel 5.1 Data-data <i>manipulator</i> .....	28
Tabel 5.2 Data hasil simulasi di mana data robot diperbesar 80% kecuali panjang <i>link</i> .....	37
Tabel 5.3 Data hasil simulasi di mana panjang tiap <i>link</i> diperbesar 0,005 m dan data robot diperbesar 80% .....	38
Tabel 5.4 Kesalahan data hasil simulasi di mana data robot diperbesar 80% kecuali panjang <i>link</i> .....	39
Tabel 5.5 Kesalahan data hasil simulasi di mana panjang tiap <i>link</i> diperbesar 0,005 m dan data diperbesar 80% .....	40

# BAB I

## PENDAHULUAN

### 1.1. LATAR BELAKANG

Dalam melaksanakan tugas-tugas tertentu yang mengharuskan robot mengadakan interaksi dengan lingkungan telah mendapatkan perhatian yang sangat besar di kalangan ahli ilmu robotika. Di antara beberapa skema pengaturan yang telah diusulkan, salah satunya adalah pengaturan impedansi yang diusulkan oleh Hogan (1985) yang memberikan suatu pendekatan untuk pengaturan posisi dan gaya yang mana variabel-variabel pengaturannya tidak hanya posisi dan gaya melainkan hubungan dinamik antara posisi dengan gaya. Jika kita bisa mendapatkan informasi tentang posisi dan kecepatan *end effector* serta gaya interaksi antara robot dengan lingkungan maka metode ini dapat mengendalikan impedansi *end effector* untuk mengikuti target impedansi yang diinginkan. Secara teoritis telah ditunjukkan bahwa pendekatan tersebut juga menjaga kestabilan gerakan selama terjadi interaksi antara *end effector* dengan lingkungan. Banyak sekali makalah-makalah yang membahas keefektifan dan kekokohan pengaturan impedansi robot yang telah dipublikasikan. Misalnya, metode untuk merealisasikan pengaturan impedansi tanpa menghitung inverse matrik jacobian dan tanpa menggunakan sensor gaya dapat diperoleh pada makalah yang disajikan oleh Hogan (1987) dan Tachi et. al (1989). Analisa kestabilan untuk *manipulator* yang mengadakan

interaksi dengan lingkungan diberikan oleh Colgate *and* Hogan (1989), Luo *and* Ito (1993). Semua hasil penelitian tersebut telah memberikan suatu kerangka kerja untuk mengendalikan interaksi mekanik yang terjadi antara *manipulator* dengan lingkungannya.

Implementasi pengaturan impedansi robot memerlukan data-data robot yang akurat seperti parameter *link* (moment inersia, titik berat), koefisien gesek dari tiap-tiap *joint*, dll, sehingga robot dapat melaksanakan tugas-tugasnya dengan baik. Akan tetapi sangatlah sulit untuk memperoleh data-data robot yang akurat. Hal ini akan menyebabkan timbulnya *error* yang akan menurunkan performansi robot dan akan menyebabkan robot tidak mampu lagi merealisasikan nilai impedansi yang diinginkan bahkan akan merusak robot itu sendiri. Sehingga diperlukan suatu kontroler yang tetap mampu merealisasikan nilai impedansi yang diinginkan meskipun terdapat ketidaktepatan data-data robot dan adanya gangguan dari luar yang tidak tentu.

Dihilami oleh algoritma yang diusulkan oleh Kuo, C. Y., *and* Wang, S. P. T. (1991) untuk pengaturan posisi kokoh dalam koordinat Cartesian, dalam tugas akhir ini akan dibahas suatu pengaturan impedansi robot berdasarkan pada pengaturan posisi kokoh dalam koordinat Cartesian. Pengaturan posisi kokoh ini terdiri dari dua bagian. Bagian pertama adalah bagian nonlinier yang menstabilkan dinamika robot dalam koordinat Cartesian. Bagian kedua adalah bagian linier yang menggunakan teori servomekanisme kokoh untuk menekan efek *error* yang ditimbulkan oleh ketidaktepatan data-data robot dan ketaktentuan gangguan dari

luar. Selanjutnya kontroler ini akan diimplementasikan pada pengaturan impedansi yang telah diusulkan oleh Hogan (1987).

## 1.2. RUMUSAN MASALAH

Pengaturan impedansi robot memerlukan data-data robot yang sangat akurat. Akan tetapi pengaturan impedansi tersebut tidak akan dapat bekerja dengan baik jika timbul *error* yang disebabkan ketidaktepatan data-data robot dan adanya ketaktentuan gangguan dari luar. Sehingga perlu dikembangkan suatu pengaturan impedansi robot yang mampu menekan *error* tersebut. Pengembangan pengaturan impedansi robot ini didasarkan pada sistem pengaturan posisi kokoh yang mana target posisinya ditentukan dari target impedansi.

## 1.3. PEMBatasan MASALAH

Impedansi robot yang dikendalikan adalah impedansi robot yang muncul pada *end effector* saja. Dengan demikian kontroler yang akan dirancang menggunakan posisi robot yang ada pada koordinat Cartesian. *Error* yang muncul sebagai akibat ketidaktepatan permodelan dan ketaktentuan gangguan dari luar dimodelkan sebagai persamaan differensial biasa yang mana kita bisa menentukan sendiri orde dari persamaan deferensial tersebut.

## 1.4. TUJUAN

Tujuan dari Tugas Akhir ini adalah merancang suatu kontroler yang mampu mengendalikan robot sesuai dengan hubungan dinamik antara gaya interaksi

dengan posisi yang diinginkan meskipun terdapat *error* yang timbul sebagai akibat dari ketidaktepatan data-data robot dan ketaktentuan gangguan dari luar.

## 1.5. SISTEMATIKA PEMBAHASAN

Pada Tugas Akhir ini penulis membagi buku dalam beberapa bagian. Urutan pembahasan yang dilakukan adalah sebagai berikut.

Pembahasan latar belakang pengambilan judul, permasalahan yang dihadapi, batasan permasalahan, tujuan yang akan dicapai, dan sistematika pembahasan diberikan pada bab 1.

Bab 2 membahas teori yang berkaitan dengan pengaturan impedansi robot. Pembahasan mengenai pengaturan impedansi robot sampai dengan diperolehnya model hubungan dinamik antara gaya interaksi dengan posisi robot.

Sementara pembahasan tentang teori yang berkaitan dengan permasalahan perancangan kontroler posisi kokoh dibahas pada bab 3. Pada bagian ini diperkenalkan metode dalam memperoleh bagian nonlinier dan linier dari kontroler posisi kokoh yang menjadi inti dari kontroler yang dirancang.

Implementasi dari kontroler posisi kokoh yang diusulkan pada pengaturan impedansi robot dibahas pada bab 4

Hasil simulasi dari pengaturan impedansi yang diusulkan yang kemudian dibandingkan dengan pengaturan impedansi konvensional diberikan dalam bab 5.

Akhirnya, bab 6 memberikan kesimpulan yang didapat setelah semua pembahasan di atas.

## BAB II

# TEORI PENGATURAN IMPEDANSI ROBOT

### 2.1. PENDAHULUAN

Pada bab ini akan dijelaskan suatu pendekatan umum untuk mengendalikan suatu robot *manipulator*. Pendekatan yang dikembangkan meliputi pekerjaan-pekerjaan sederhana seperti memindahkan suatu benda dari suatu titik ke titik lain yang khusus dilakukan oleh robot. Pendekatan ini mengembangkan suatu kemampuan yang mempermudah aplikasi robot dalam melakukan pekerjaannya yang meliputi interaksi dinamik dan statik antara *manipulator* dan lingkungannya.

Pada umumnya, aplikasi robot-robot dalam industri saat ini berdasarkan pada pengaturan posisi, sebagai contoh adalah pengecatan suatu benda. Dalam hal ini, *manipulator* tidak perlu mengadakan hubungan langsung dengan obyek yang sedang dimanipulasi. Sehingga pengendalian yang perlu dilakukan adalah pengendalian gerakannya (*motion*), dengan posisi, kecepatan, percepatan sebagai variabel pengendalian (*free motion*).

Dalam situasi lain *manipulator* menemui batasan-batasan (*constraints*) dalam lingkungannya dan gaya interaksi tidak dapat diabaikan. Dalam hal ini, *manipulator* harus mengadakan hubungan langsung dengan obyek yang sedang dimanipulasi (*constrained motion*). Sebagai contoh dalam kasus di mana *manipulator* diinginkan untuk menuliskan suatu kata atau kalimat di atas kertas

menggunakan pena. Untuk kasus seperti ini pengendalian *manipulator* tidak cukup hanya dengan menggunakan pengendalian posisi, akan tetapi juga diperlukan pengendalian gaya. Sehingga *manipulator* dapat menuliskan kata atau kalimat yang diinginkan dengan jelas dan tidak merusak kertas. Dalam kasus seperti ini, terdapat dua strategi pengendalian yang paling umum digunakan, yaitu *impedance control* yang mana menentukan suatu hubungan linier antara torsi/gaya *end-effector* dengan posisi/orientasi *end-effector* (atau kecepatan *end-effector*) dan *hybrid control* yang mana adalah gabungan dari pengendalian gerakan (posisi/orientasi) pada suatu derajat kebebasan yang telah ditentukan dan pengendalian gaya pada tiap derajat kebebasan selanjutnya.

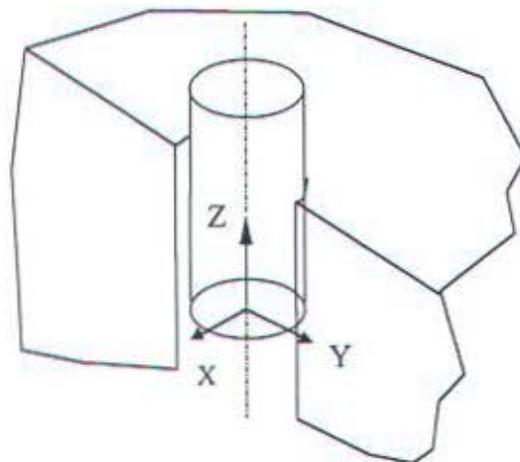
Pengendalian menggunakan *hybrid control* tidak akan dibahas terlalu dalam dalam bab ini, penekanannya akan lebih banyak membahas tentang pengendalian impedansi robot atau *impedance control*.

## 2.2. PENGENDALIAN GABUNGAN (*HYBRID CONTROL*)

Pengendalian gabungan atau *hybrid control* didasari oleh suatu observasi yang menyatakan bahwa kadang-kadang diinginkan untuk memberikan perintah suatu gaya atau torsi dari pada perintah posisi. Pada awalnya, *hybrid control* diusulkan oleh Inoue (1971), yang kemudian pengembangannya dilanjutkan oleh Paul (1972), Silver (1973), Takase, Goto and Inoyama (1974), Paul and Shimano (1976), Mason (1981), Raibert and Craig (1981).

*Hybrid controller* didefinisikan sebagai suatu kontroler yang mengendalikan gaya pada suatu derajat kebebasan tertentu dan mengendalikan

posisi pada derajat kebebasan selanjutnya. Derajat kebebasan dipilih sebagai tiga komponen yang menyatakan translasi dari suatu titik obyek, dan tiga komponen yang memberikan orientasi dari suatu obyek. Biasanya kebebasan ini didefinisikan dalam bentuk *compliance frame*, Paul and Shimano (1976), Raibert and Craig (1981). *Compliance frame* mendefinisikan suatu sistem koordinat tegak lurus dalam ruang. Kebebasan gerak didefinisikan sebagai translasi pada setiap sumbu, dan rotasi terhadap tiap-tiap sumbu. Sehingga dapat ditentukan yang mana saja pengendalian posisi dan yang mana saja pengendalian gaya. Contoh yang paling umum digunakan adalah memasukkan pasak kedalam lubang (Gbr 2.1).



Gambar 2.1. Memasukkan pasak dalam lubang.

Dalam kasus ini, *compliance frame* dipilih sehingga salah satu sumbunya yaitu sumbu-z berimpitan dengan sumbu pasak. Translasi pada sumbu-z dan rotasi terhadap sumbu-z adalah pengendalian posisi, sedangkan kebebasan yang lain adalah pengendalian gaya. Sehingga dapat ditentukan trayektori dari posisi/orientasi dan gaya/torsi pada tiap derajat kebebasan yang sesuai.

### 2.3. PENGATURAN IMPEDANSI ROBOT

Suatu perbedaan antara pengaturan impedansi dengan pendekatan konvensional dalam pengaturan *manipulator* adalah bahwa kontroler berusaha mengimplementasikan suatu hubungan dinamik antara variabel-variabel *manipulator* seperti posisi *end-point* dan gaya antara robot dengan lingkungan daripada hanya mengatur variabel-variabel tersebut secara sendiri-sendiri.

Suku orde terendah dalam setiap impedansi adalah hubungan statik antara gaya output dan input pergeseran, *stiffness*. Dengan asumsi bahwa aktuator mampu menghasilkan gaya atau torsi yang diinginkan,  $\tau$ , sensor mampu memperoleh posisi aktuator atau sudut,  $\theta$ , dan hubungan kinematik murni antara posisi aktuator dengan posisi *end-point*,  $X = L(\theta)$ , maka dapat langsung dirancang suatu pengaturan umpan balik yang diimplementasikan dalam koordinat aktuator suatu hubungan yang diinginkan antara gaya *end point*,  $F_{int}$ , dan posisi,  $X$ . Didefinisikan suatu posisi setimbang yang diinginkan untuk *end point* di mana tidak terdapat gaya interaksi dengan lingkungan (posisi virtual) sebagai  $X_d$ , bentuk umum dari hubungan posisi dan gaya yang diinginkan adalah :

$$F_{int} = K_e [X - X_d] \quad (2.1)$$

di mana  $K_e$  adalah suatu konstanta yang menghubungkan pergeseran dengan gaya interaksi yang lazim disebut sebagai kekakuan atau *stiffness*.

Selanjutnya menghitung matriks *Jacobian*,  $J(\theta)$  :

$$dX = J(\theta)d\theta \quad (2.2)$$

Dari prinsip kerja virtual :

$$\tau = J^T(\theta)F_{int} \quad (2.3)$$

Hubungan yang dibutuhkan dalam koordinat aktuator adalah :

$$\tau = J^T(\theta)K_e[L(\theta) - X_d] \quad (2.4)$$

Jika  $K_e[X - X_d]$  yang dipilih cukup untuk membuat *end point* kaku, maka kontroler yang menggunakan persamaan (2.4) akan menyelesaikan pengaturan posisi *end point* Cartesian dan tidak perlu menyelesaikan masalah “*inverse kinematics*”. Hanya persamaan *forward* kinematik yang perlu dihitung. Ini adalah konsekuensi langsung yang diberikan untuk menjamin bahwa kelakuan yang diinginkan sesuai dengan mekanika dasar manipulasi dan dinyatakan sebagai impedansi.

Hal penting lainnya dalam impedansi *manipulator* adalah hubungan antara gaya dengan kecepatan. Dengan asumsi di atas, maka dapat langsung dirancang suatu pengaturan umpan balik yang diimplementasikan dalam koordinat aktuator suatu hubungan yang diinginkan antara gaya *end point* dengan kecepatan *end point* sebagai berikut :

$$F_{int} = B_e[V - V_d] \quad (2.5)$$

Dari kinematika *manipulator* :

$$V = J(\theta)\omega \quad (2.6)$$

di mana  $B_e$  adalah suatu konstanta yang menghubungkan gaya *end point*  $F_{int}$  dengan kecepatan.  $V$  adalah kecepatan *end point*,  $V_d$  adalah kecepatan *end point* yang diinginkan.

Dari prinsip kerja virtual pada persamaan (2.3) maka hubungan yang diinginkan dalam koordinat aktuator adalah :

$$\tau = J^T(\theta)B_e[J(\theta)\omega - V_d] \quad (2.7)$$

Kelakuan dinamik yang akan dimunculkan oleh *manipulator* harus sesederhana mungkin. Persamaan selanjutnya tidak perlu menghitung inersia, gesekan, gravitasi dinamik *manipulator*. Pada beberapa situasi hal tersebut bisa saja dilakukan, tetapi terdapat banyak sekali situasi di mana parameter-parameter tersebut tidak bisa diabaikan. Untuk menjamin fisibilitas dinamik, pemilihan impedansi yang akan dimunculkan *manipulator* harus didasarkan pada kelakuan dinamik dominan *manipulator*.

Tidak terdapat suatu strategi yang mampu mengeliminasi efek inersia dari *manipulator* tetapi inersia *end effector* yang dapat dilihat dengan jelas dapat dimodifikasi. Pendekatan yang digunakan untuk mengatasi kelakuan inersia *manipulator* adalah dengan menutupi dinamika inersia nonlinier yang dimiliki oleh *manipulator* dan memberikan dinamika yang lebih sederhana. Perencanaan dan pelaksanaan tugas dapat disederhanakan jika kelakuan inersia *end point* dimodifikasi dengan suatu inersia tensor yang tetap sama selama translasi dan rotasi pada sumbu koordinat. Hal ini dapat dicapai jika :

$$M_e = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \quad (2.8)$$

Di mana  $M_e$  adalah inersia tensor pada koordinat *end point*,  $m$  adalah massa pada *end point* dan  $I$  adalah inersia pada *end point*. Ini adalah inersia tensor dari benda tegar seperti kubus dengan kerapatan yang sama. Inersia tensor ini mengeliminasi perkalian kecepatan angular dalam persamaan Euler untuk gerakan benda tegar.

Untuk menyatakan kelakuan dominan orde dua dari *manipulator*, maka gaya *interface* noninersial diasumsikan hanya tergantung pada pergeseran, kecepatan dan waktu  $F(X, V)$  :

$$F_{int} = F(X, V) + M_e dV/dt \quad (2.9)$$

Persamaan tersebut di atas dapat ditulis sebagai pergeseran dari posisi yang diinginkan  $X_d$  sebagai berikut :

$$F_{int} = F(X - X_d, V - V_d) + M_e dV/dt \quad (2.10)$$

Untuk penyederhanaan posisi dan kecepatan dapat dipisahkan :

$$F_{int} = K_e [X - X_d] + B_e [V - V_d] + M_e dV/dt \quad (2.11)$$

persamaan (2.11) dapat dinyatakan sebagai persamaan target impedance sebagai berikut :

$$M_e (\ddot{X} - \ddot{X}_d) + B_e (\dot{X} - \dot{X}_d) + K_e (X - X_d) = F_{int} \quad (2.12)$$

Semua parameter dalam persamaan ini diasumsikan sebagai fungsi dari perintah pengaturan dan fungsi waktu, di mana  $M_e$ ,  $B_e$ ,  $K_e$  adalah matriks inersia pada koordinat *end point*, viskositas dan kekakuan yang diinginkan dari *end effector*.

## 2.4. IMPLEMENTASI PENGATURAN IMPEDANSI ROBOT

Untuk mengimplementasikan kelakuan target dari persamaan (2.12), salah satu pendekatan yang digunakan adalah menyatakan impedansi dalam koordinat

Cartesian yang diinginkan dan kemudian menggunakan suatu model dinamika *manipulator* untuk memperoleh persamaan kontroler yang diinginkan. Diasumsikan efek kinematika, inersial, gravitasi dan gesekan memberikan suatu model dinamika *manipulator* yang sesuai sebagai berikut :

$$\mathbf{M}\ddot{\theta} + \mathbf{h}(\theta, \dot{\theta}) + \mathbf{c}(\theta) = \boldsymbol{\tau} + \mathbf{J}^T \mathbf{F}_{int} \quad (2.13)$$

di mana  $\mathbf{F}_{int}$  adalah gaya yang diberikan pada *manipulator* oleh lingkungan;  $\theta$  adalah vektor sudut *joint*;  $\mathbf{M}$  adalah matriks inersia pada koordinat *joint*;  $\mathbf{h}(\theta, \dot{\theta})$  adalah suku nonlinier torsi *joint* sebagai akibat dari sentrifugal, Coriolis,  $\mathbf{c}(\theta)$  adalah gravitasi dan gaya gesek;  $\boldsymbol{\tau}$  adalah vektor torsi *joint*;  $\mathbf{J}$  adalah matriks *Jacobian*.

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{cancel} + \boldsymbol{\tau}_{imp} \quad (2.14)$$

di mana  $\boldsymbol{\tau}_{imp}$  adalah vektor torsi *joint* yang diperlukan untuk menghasilkan impedansi *end effector* yang diinginkan;  $\boldsymbol{\tau}_{cancel}$  adalah vektor torsi *joint* untuk kompensasi nonlinier dengan asumsi bahwa :

$$\boldsymbol{\tau}_{cancel} = \mathbf{h}(\theta, \dot{\theta}) + \mathbf{c}(\theta) \quad (2.15)$$

Persamaan (2.14) dan persamaan (2.15) masuk ke persamaan (2.13), diperoleh :

$$\mathbf{M}\ddot{\theta} = \boldsymbol{\tau}_{imp} + \mathbf{J}^T \mathbf{F}_{int} \quad (2.16)$$

$$\ddot{\theta} = \mathbf{M}^{-1} (\boldsymbol{\tau}_{imp} + \mathbf{J}^T \mathbf{F}_{int}) \quad (2.17)$$

Diketahui pula bahwa :

$$\boldsymbol{\tau}_{imp} = \mathbf{J}^T \mathbf{F}_{imp} \quad (2.18)$$

sehingga persamaan (2.17) menjadi :

$$\ddot{\theta} = \mathbf{M}^{-1} \mathbf{J}^T (\mathbf{F}_{imp} + \mathbf{F}_{int}) \quad (2.19)$$

persamaan (2.19) dikalikan dengan matriks *Jacobi*, diperoleh :

$$\mathbf{J}\ddot{\theta} = \mathbf{W}(\mathbf{F}_{imp} + \mathbf{F}_{int}) \quad (2.20)$$

Di mana  $\mathbf{W} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T$

Dari Hubungan kecepatan *end effector* dengan kecepatan *joint* diperoleh :

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{J}\dot{\theta}, \\ \ddot{\mathbf{X}} &= \mathbf{J}\ddot{\theta} + \dot{\mathbf{J}}\dot{\theta}, \\ \mathbf{J}\ddot{\theta} &= \ddot{\mathbf{X}} - \dot{\mathbf{J}}\dot{\theta} \end{aligned} \quad (2.21)$$

Dari persamaan (2.12) :

$$\ddot{\mathbf{X}} = \ddot{\mathbf{X}}_d - \mathbf{M}_e^{-1} [\mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) + \mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) - \mathbf{F}_{int}] \quad (2.22)$$

Persamaan (2.22) masuk ke persamaan (2.21), diperoleh :

$$\mathbf{J}\ddot{\theta} = \ddot{\mathbf{X}}_d - \mathbf{M}_e^{-1} [\mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) + \mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) - \mathbf{F}_{int}] - \dot{\mathbf{J}}\dot{\theta} \quad (2.23)$$

Persamaan (2.23) masuk ke persamaan (2.20), diperoleh :

$$\mathbf{W}(\mathbf{F}_{imp} + \mathbf{F}_{int}) = \ddot{\mathbf{X}}_d - \mathbf{M}_e^{-1} [\mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) + \mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) - \mathbf{F}_{int}] - \dot{\mathbf{J}}\dot{\theta}$$

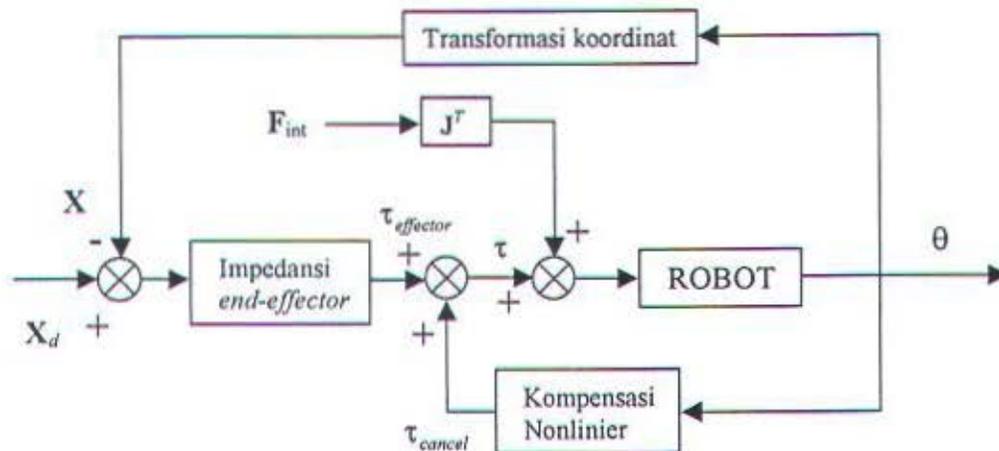
$$\mathbf{F}_{imp} + \mathbf{F}_{int} = \mathbf{W}^{-1} \{ \ddot{\mathbf{X}}_d - \mathbf{M}_e^{-1} [\mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) + \mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) - \mathbf{F}_{int}] - \dot{\mathbf{J}}\dot{\theta} \}$$

$$\mathbf{F}_{imp} = \mathbf{W}^{-1} \{ \ddot{\mathbf{X}}_d - \mathbf{M}_e^{-1} [\mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) + \mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) - \mathbf{F}_{int}] - \dot{\mathbf{J}}\dot{\theta} \} - \mathbf{F}_{int} \quad (2.24)$$

persamaan (2.24) dikalikan dengan *Jacobi transpose*, diperoleh :

$$\begin{aligned} \tau_{imp} &= \mathbf{J}^T \mathbf{W}^{-1} \{ \ddot{\mathbf{X}}_d - \mathbf{M}_e^{-1} [\mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) + \mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) - \mathbf{F}_{int}] \\ &\quad - \dot{\mathbf{J}}\dot{\theta} \} - \mathbf{J}^T \mathbf{F}_{int} \end{aligned} \quad (2.25)$$

Persamaan (2.25) digunakan sebagai kontroler impedansi konvensional tanpa menghitung *inverse Jacobi*. Diagram blok pengaturan impedansi konvensional ditunjukkan oleh **Gbr 2.2**.



Gambar 2.2. Diagram blok pengaturan impedansi konvensional.

Persamaan (2.25) adalah persamaan yang menyatakan kelakuan yang diinginkan yang harus disediakan oleh kontroler sebagai impedansi nonlinier dalam koordinat Cartesian. Hal tersebut dapat dipandang sebagai pengaturan umpan balik nonlinier yang menghubungkan torsi aktuator dengan posisi, kecepatan dan gaya *interface* aktuator. Variabel-variabel input adalah posisi Cartesian yang diinginkan dan kelakuan dinamik Cartesian yang diinginkan diberikan oleh  $M_e$ ,  $B_e$ , dan  $K_e$ .

# BAB III

## PENGATURAN POSISI KOKOH

### PADA KOORDINAT CARTESIAN

#### 3.1. PENDAHULUAN

Tujuan dari suatu robot adalah mengatur gerakan *gripper*-nya untuk melaksanakan berbagai macam pekerjaan seperti merakit, mengecat, mengelas. Sistem pengaturan dari robot harus memenuhi beberapa syarat seperti waktu *cycle* yang singkat, akurasi *tracking* yang tinggi, *flexibel* dan mampu beradaptasi terhadap perubahan tugas.

Berdasarkan variabel yang digunakan dalam formulasi sistem pengaturan, suatu skema pengaturan robot dapat dikategorikan sebagai pengaturan *joint* dan pengaturan Cartesian. Pengaturan *joint* menggunakan variabel *joint* dan pengaturan Cartesian menggunakan variabel dalam koordinat Cartesian. Pada umumnya robot-robot yang ada menggunakan kontroler dalam koordinat *joint*, dengan asumsi bahwa tiap *link* tidak tergantung satu sama lain dan menggunakan kontroler proposional dan derivatif pada tiap *joint*-nya. Tetapi pada saat kecepatan robot bertambah akan timbul efek *coupling* diantara *link*-nya, sehingga performansi kontroler PD akan menurun. Suatu kontroler *decoupling* nonlinier dapat menghilangkan efek *coupling* ini. Tetapi performansinya akan menjadi jelek jika timbul *error* permodelan.

Untuk menggunakan pengaturan *joint*, trayektori yang diinginkan dalam koordinat Cartesian, pertama-tama harus diubah ke trayektori *joint*. Dalam hal lain, perubahan *joint*-Cartesian dapat diabaikan jika digunakan pengaturan Cartesian. Pembahasan selanjutnya akan memusatkan perhatian pada koordinat Cartesian berdasarkan aspek pengaturan posisi. Penekanan khusus akan diberikan pada robustness dari skema yang diusulkan untuk mengatasi *error* permodelan dan gangguan dari luar yang mungkin timbul pada saat robot bekerja.

Sejumlah metode pengaturan posisi dalam koordinat Cartesian telah diusulkan seperti *resolved motion rate control* (RMRC), *resolved motion acceleration control* (RMAC), *resolved motion force control* (RMFC), pengaturan adaptif, pengaturan nonlinier. RMRC dan RMAC tidak memiliki ketentuan untuk menghitung *error* permodelan dan gangguan dari luar. RMFC hanya bergantung pada umpan balik gaya untuk mengkompensasi *error* permodelan dan gangguan. Input kendali dari algoritma pengatur adaptif terdiri dari bagian umpan maju dan bagian umpan balik. Bagian umpan maju mengkompensasi efek kopling diantara seluruh koordinat sedangkan bagian umpan balik mengurangi *error* posisi sepanjang trayektori nominal. Tetapi, seringkali cukup sulit untuk membuktikan kestabilan algoritma pengatur adaptif, khususnya jika ada dinamika yang tak termodelkan.

Pengatur nonlinier menggunakan suatu algoritma koreksi *error optimal* untuk mengurangi *error tracking* yang berawal dari *error* permodelan. Algoritma ini bergantung pada pemilihan matriks pembobot yang tepat yang secara efektif

dapat mengurangi *error tracking*. Tetapi tidak ada cara analitis untuk mendapatkan matriks pembobot, dan algoritma pengatur tidak dapat mengatasi gangguan yang tidak diketahui.

### 3.2. KONTROLER POSISI KOKOH NONLINIER

Persamaan dinamik robot *n-joint* dalam koordinat Cartesian dapat dinyatakan sebagai berikut (Lampiran A):

$$\mathbf{D}_x \ddot{\mathbf{X}} + \mathbf{h}_x + \mathbf{c}_x = \mathbf{F} + \mathbf{F}_{int} \quad (3.1)$$

di mana

$$\mathbf{D}_x = (\mathbf{J}^T)^{-1} \mathbf{M} \mathbf{J}^{-1} \quad (3.2)$$

$$\mathbf{h}_x = (\mathbf{J}^T)^{-1} \mathbf{h} - \mathbf{D}_x \dot{\mathbf{J}} \dot{\boldsymbol{\theta}} \quad (3.3)$$

$$\mathbf{c}_x = (\mathbf{J}^T)^{-1} \mathbf{c} \quad (3.4)$$

$\mathbf{J}$  adalah matriks *Jacobian*,  $\mathbf{X}$  dan  $\mathbf{F}$  adalah vektor posisi dan gaya input *end effector* dalam koordinat Cartesian,  $\mathbf{F}_{int}$  adalah vektor gaya interaksi lingkungan.  $\ddot{\boldsymbol{\theta}}, \dot{\boldsymbol{\theta}}, \boldsymbol{\theta}$  adalah vektor  $n \times 1$  percepatan angular, kecepatan, posisi *joint*.  $\mathbf{M}$ ,  $\mathbf{h}$ , dan  $\mathbf{c}$  menyatakan inersia, Coriolis, sentrifugal, gravitasi dalam dalam koordinat *joint*.  
Jika persamaan dinamika robot yang digunakan dalam kontroler adalah :

$$\mathbf{D}'_x \mathbf{P}' + \mathbf{h}'_x + \mathbf{c}'_x = \mathbf{F} + \mathbf{F}_{int} \quad (3.5)$$

di mana  $\mathbf{D}'_x, \mathbf{h}'_x$ , dan  $\mathbf{c}'_x$  mengandung parameter robot yang digunakan dalam merancang kontroler, yang hampir selalu terdapat perbedaan yang penting dari nilai

sebenarnya. Jika persamaan (3.5) disubstitusikan ke persamaan (3.1) maka diperoleh persamaan sebagai berikut (Kuo, C. Y. and Wang, S. P. T, 1991):

$$\ddot{\mathbf{y}} = \ddot{\mathbf{X}} = \mathbf{P}' + \mathbf{e} \quad (3.6)$$

di mana vektor kesalahan  $\mathbf{e}$  yang diberikan oleh persamaan berikut ini,

$$\mathbf{e} = \mathbf{D}_x^{-1}(\mathbf{h}'_x - \mathbf{h}_x) + \mathbf{D}_x^{-1}(\mathbf{c}'_x - \mathbf{c}_x) + (\mathbf{D}_x^{-1}\mathbf{D}'_x - \mathbf{I})\mathbf{P}' \quad (3.7)$$

menggabungkan semua efek *error* parameter ini. Persamaan ini juga menyatakan efek gangguan dari luar. Dalam (3.7),  $\mathbf{I}$  adalah matriks satuan  $n \times n$  dan dalam (3.6),  $\mathbf{y}$  adalah vektor output yang dinyatakan dengan vektor posisi  $\mathbf{X}$  dalam koordinat Cartesian. Perlu dicatat bahwa  $\mathbf{y}$  dapat diperoleh secara langsung dari vektor posisi *joint*  $\theta$  melalui transformasi *homogenous* berantai.

Persamaan (3.6) dapat distabilkan dengan menggunakan pengaturan servo berikut (Kuo, C. Y. and Wang, S. P. T, 1989) :

$$\mathbf{P}' = \begin{bmatrix} \mathbf{P}'_1 \\ \mathbf{P}'_i \\ \mathbf{P}'_n \end{bmatrix} \quad (3.8)$$

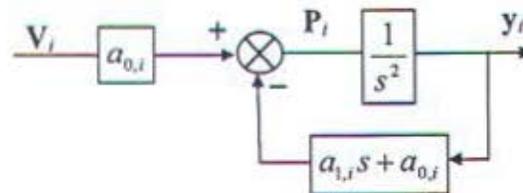
di mana

$$\mathbf{P}'_i = a_{0,i} \mathbf{V}_i - a_{1,i} \dot{\mathbf{y}}_i - a_{0,i} \mathbf{y}_i \quad (3.9)$$

dan  $\mathbf{V}_i$  adalah input linier untuk koordinat ke- $i$ . Substitusi persamaan (3.8) ke persamaan (3.6) didapat persamaan dinamik untuk tiap-tiap koordinat sebagai berikut :

$$\ddot{y}_i + a_{1,i}\dot{y}_i + a_{0,i}y_i = a_{0,i}V_i + e_i \quad (3.10)$$

Gambar 3.1 menunjukkan pengaturan posisi servo pada koordinat Cartesian yang diberikan oleh persamaan (3.9).



Gambar 3.1. Pengaturan posisi servo pada koordinat Cartesian

Terbukti bahwa performansi dari sistem pengaturan ini akan menurun jika efek dari *error* permodelan dan gangguan (yang muncul dalam (3.10) sebagai  $e_i$ ) tidak diperhitungkan. Akan dikembangkan suatu algoritma untuk mengatur  $V_i$  sehingga *error tracking* dapat diminimumkan. Pertama ditentukan output referensi ke- $i$   $y_i^d$  sebagai

$$\ddot{y}_i^d + a_{1,i}\dot{y}_i^d + a_{0,i}y_i^d = a_{0,i}V_i^d, \quad i = 1, 2, 3, \dots, n \quad (3.11)$$

di mana  $V_i^d$  output referensi linier untuk koordinat ke- $i$ . Dengan pemilihan yang sesuai untuk koefisien-koefisien ( $a_{0,i}$  dan  $a_{1,i}$ ) pada persamaan (3.11),  $y_i^d$  akan mengikuti  $V_i^d$ . Sehingga  $V_i^d$  dapat digunakan untuk menentukan trayektori *end effector* yang diinginkan dalam koordinat Cartesian. Dengan mengurangkan (3.11) dari (3.10), didapatkan persamaan *error tracking* sebagai berikut :

$$\ddot{y}_i' + a_{1,i}\dot{y}_i' + a_{0,i}y_i' = a_{0,i}V_i' + e_i, \quad i = 1, 2, 3, \dots, n \quad (3.12)$$

$$y_i' = y_i - y_i^d \quad (3.13)$$

$$\mathbf{V}_i' = \mathbf{V}_i - \mathbf{V}_i^d \quad (3.14)$$

Persamaan (3.12) dapat ditulis dalam bentuk persamaan *state* sebagai berikut :

$$\dot{\mathbf{x}}_i' = \mathbf{T}_i \mathbf{x}_i' + \mathbf{G}_i \mathbf{V}_i' + \mathbf{G}_i' \mathbf{e}_i \quad (3.15)$$

$$\mathbf{y}_i' = \mathbf{H}_i' \mathbf{x}_i' \quad (3.16)$$

di mana

$$\mathbf{x}_i' = \begin{bmatrix} \mathbf{y}_i' \\ \dot{\mathbf{y}}_i' \end{bmatrix} \quad (3.17)$$

$$\mathbf{T}_i = \begin{bmatrix} 0 & 1 \\ -a_{0,i} & -a_{1,i} \end{bmatrix} \quad (3.18)$$

$$\mathbf{G}_i = \begin{bmatrix} 0 \\ a_{0,i} \end{bmatrix} \quad (3.19)$$

$$\mathbf{G}_i' = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.20)$$

$$\mathbf{H}_i' = [1 \quad 0] \quad (3.21)$$

Untuk mendapatkan akurasi *tracking* yang tinggi, diinginkan untuk membangkitkan suatu sinyal kendali linier tambahan  $\mathbf{V}_i'$  agar supaya *error tracking*  $\mathbf{x}_i'$  yang berarti juga  $\mathbf{y}_i'$  secara asimtotik menuju nol. Untuk mencapai tujuan ini, diasumsikan  $\mathbf{e}_i$  dapat dimodelkan oleh persamaan diferensial biasa orde  $p$  (Franklin, G. F. and Naeni, E. A, 1983) :

$$\mathbf{e}_i^{(p)} = \sum_{j=1}^p b_j \mathbf{e}_i^{(p-j)} \quad (3.22)$$

di mana superscript  $(p - j)$  berarti turunan ke- $(p - j)$ . Orde  $p$  dari persamaan diferensial ini menyatakan struktur dinamik dari  $\mathbf{e}_i$ , dengan koefisien  $b_j$  seorang perancang dapat menggunakan kebebasannya dalam menentukan nilai-nilainya. Selanjutnya didefinisikan kuantitas-kuantitas berikut (Franklin, G. F. and Naeni, E. A, 1983) :

$$\mathbf{z}_i = \mathbf{x}_i^{(p)} - \sum_{j=1}^p b_j \mathbf{x}_i^{(p-j)} \quad (3.23)$$

$$\mathbf{s}_i = \mathbf{V}_i^{(p)} - \sum_{j=1}^p b_j \mathbf{V}_i^{(p-j)} \quad (3.24)$$

Persamaan (3.23) didiferensialkan terhadap waktu dan dengan menggunakan persamaan (3.15), persamaan (3.16), dan persamaan (3.24) didapat bahwa :

$$\dot{\mathbf{z}}_i = \mathbf{T}_i \mathbf{z}_i + \mathbf{G}_i \mathbf{s}_i \quad (3.25)$$

$$\mathbf{y}_i^{(p)} = \mathbf{H}_i \mathbf{x}_i^{(p)} = \sum_{j=1}^p b_j \mathbf{y}_i^{(p-j)} + \mathbf{H}_i \mathbf{z}_i \quad (3.26)$$

Misalkan  $\mathbf{Z}_i$  didefinisikan oleh :

$$\mathbf{Z}_i = [\mathbf{y}_i' \quad \dot{\mathbf{y}}_i' \quad \bullet \quad \bullet \quad \bullet \quad \mathbf{y}_i^{(p-j)} \quad \mathbf{z}_i]^\top \quad (3.27)$$

Sehingga dapat diperoleh persamaan *state* sebagai berikut :

$$\dot{\mathbf{Z}}_i = \mathbf{A}_i \mathbf{Z}_i + \mathbf{B}_i \mathbf{s}_i \quad (3.28)$$

$$\mathbf{y}_i = \mathbf{C}_i \mathbf{Z}_i \quad (3.29)$$

di mana

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \bullet & \bullet & \bullet & 0 & 0 \\ 0 & 0 & 1 & \bullet & \bullet & \bullet & 0 & 0 \\ \bullet & \bullet \\ \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & 1 & \\ b_p & b_{p-1} & \bullet & \bullet & \bullet & \bullet & b_1 & H_i \\ 0 & 0 & \bullet & \bullet & \bullet & \bullet & 0 & F_i \end{bmatrix} \quad (3.30)$$

$$B_i = \begin{bmatrix} 0 \\ 0 \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ 0 \\ G_i \end{bmatrix} \quad (3.31)$$

$$C_i = [1 \ 0 \ 0 \ \bullet \ \bullet \ \bullet \ \bullet \ 0] \quad (3.32)$$

Perlu dicatat bahwa jika  $(T_i, G_i)$  adalah *controllable*, maka  $(A_i, B_i)$  juga *controllable*. Sehingga, selalu mungkin untuk mencari suatu *gain feedback*  $K_i$  dengan

$$s_i = -K_i Z_i \quad (3.33)$$

untuk memastikan  $Z_i$  dan  $y_i'$  mendekati nol secara asimtotik. Dalam paragraf, berikut dapat ditunjukkan bahwa  $s_i$  dan  $V_i$  dapat dinyatakan sebagai fungsi eksplisit dari  $K_i$  dan *error tracking*  $y_i'$ . Persamaan (3.33) dapat dikembangkan sebagai berikut :

$$s_i = -K_{p,i}y_i' - K_{p-1,i}\dot{y}_i' - \dots - K_{1,i}y_i'^{(p-1)} - K_{0,i}z_i \quad (3.34)$$

substitusi persamaan (3.23) dan persamaan (3.24) ke persamaan (3.34) diperoleh

$$(V_i' + K_{0,i}x_i')^{(p)} = -\sum_{j=1}^p K_{j,i}y_i'^{(p-j)} + \sum_{j=1}^p b_j(V_i' + K_{0,i}x_i')^{(p-j)} \quad (3.35)$$

Karena  $x_i'$  adalah fungsi dari *error tracking*  $y_i'$  saja, maka nilai dari input linier tambahan  $V_i'$  hanya bergantung pada *error tracking*.

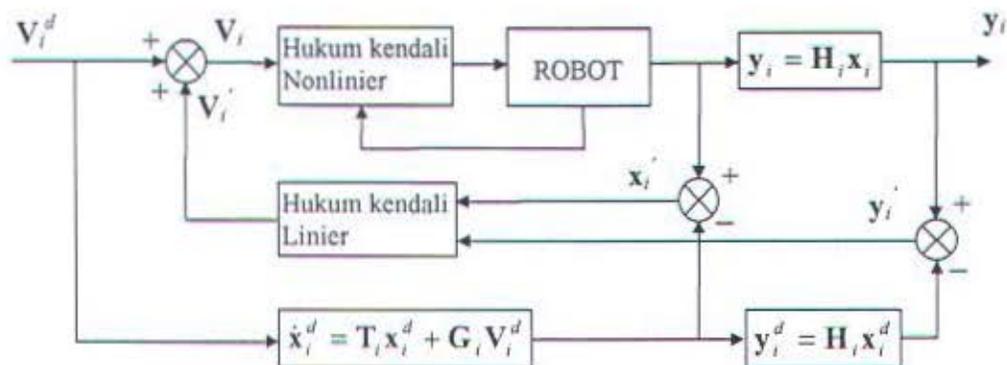
Langkah terakhir adalah mengatur input kendali linier  $V_i$  pada persamaan (3.10) untuk mengatasi efek *error* permodelan dan gangguan sebagai berikut :

$$V_i = V_i^d + V_i' \quad (3.36)$$

Nilai dari  $K_i$  dalam (3.33) dapat dihitung dengan memaksa *error tracking* mengikuti output dari suatu model implicit; tetapi metode ini biasanya digunakan dalam domain waktu kontinyu dan tidak praktis dalam waktu diskrit. Cara yang lebih cocok untuk menghitung nilai  $K_i$  adalah memilih pole-pole *closed-loop* dari  $(A_i, B_i)$  yang sesuai, kemudian menghitung nilai  $K_i$  yang sesuai. Untuk melakukannya, dua himpunan pole-pole sistem *closed-loop* harus ditentukan :

- 1) Bagian kendali nonlinier : menempatkan pole-pole pada tiap arah koordinat dekopel; persamaan (3.10).
- 2) Bagian kendali linier : menempatkan pole-pole *closed-loop* sistem  $(A_i, B_i)$ ; persamaan (3.28).

Untuk menentukan pole-pole *closed-loop* yang sesuai dengan kedua sistem, harus diketahui tujuan dari bagian pengaturan linier dan nonlinier. Tujuan dari pengaturan nonlinier adalah memisahkan dinamika robot dan output kendali  $y_i$  untuk mengikuti input referensi  $V_i$ ; lihat persamaan (3.10). Sedangkan, tujuan dari pengaturan linier adalah untuk menghilangkan efek tak tentu (*error* permodelan dan gangguan). Perlu dicatat bahwa secara matematis, *error tracking*  $y_i'$  akan mendekati nol sebagaimana dibuktikan di atas hanya jika vektor tak tentu  $e$  yang tergabung didalam *error* permodelan dan gangguan sesuai dengan model yang diasumsikan. Tetapi, akurasi *tracking* yang bagus dapat dicapai dengan orde model yang relatif rendah ( $p = 1$  atau  $2$ ) selama kita dapat memilih vektor *gain feedback*  $K_i$  dan koefisien model tak tentu  $b_i$  sehingga *bandwidth* dari *loop robustness* linier (lihat (3.28) dan (3.29)) mencakup *range* frekuensi operasi dari sistem terpisah. Biasanya dipilih  $p = 1$  atau  $2$  dan menempatkan pole-pole *closed-loop* sistem  $(A_i, B_i)$  jauh disebelah kiri sumbu imajiner dalam bidang kompleks  $s$  daripada pole-pole *closed loop decouple* (lihat (3.10)). Diagram blok pengaturan posisi kokoh nonlinier ditunjukkan oleh Gbr 3.2.



Gambar 3.2. Diagram blok pengaturan posisi kokoh nonlinier

## BAB IV

# IMPLEMENTASI PENGATURAN POSISI KOKOH PADA PENGATURAN IMPEDANSI ROBOT

### 4.1. PENGATURAN IMPEDANSI ROBOT BERDASARKAN SISTEM PENGATURAN POSISI

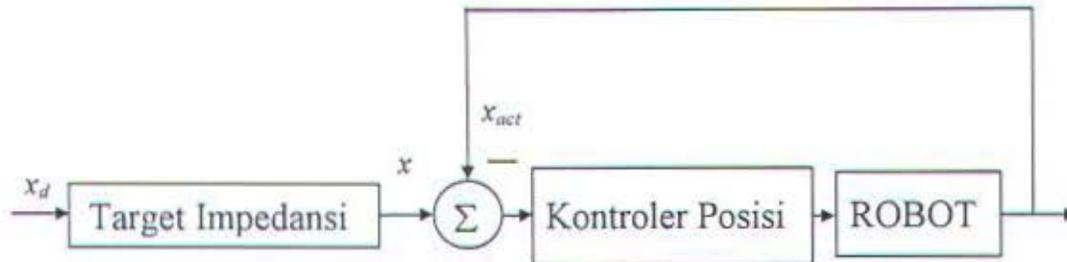
Pengendalian robot dapat dikategorikan sebagai pengendalian *joint* dan pengendalian Cartesian. Pengendalian *joint* menggunakan variabel-variabel *joint* dan pengendalian Cartesian menggunakan variabel-variabel Cartesian. Untuk menggunakan pengendalian *joint*, trayektori yang diinginkan dalam koordinat Cartesian harus diubah menjadi trayektori *joint*. Akan tetapi, perubahan *joint* ke Cartesian dapat diabaikan jika digunakan pengendalian Cartesian. Secara umum diagram blok pengaturan posisi robot diberikan oleh **Gbr 4.1**.



Gambar 4.1. Diagram blok pengaturan posisi robot

Esensi dari pengaturan impedansi adalah membentuk kembali hubungan dinamik antara lingkungan dengan *end-point* sehingga *end-point* dapat bergerak sepanjang trayektori referensi target impedansi. Sehingga dapat digunakan

pengaturan posisi seperti diagram blok di atas untuk mengendalikan impedansi robot agar mengikuti trayektori referensi target impedansi. Diagram blok pengaturan impedansi berdasarkan pengaturan posisi diberikan oleh **Gbr 4.2**.



*Gambar 4.2.* Diagram blok pengaturan impedansi berdasarkan pengaturan posisi robot

## 4.2. IMPLEMENTASI PENGATURAN POSISI KOKOH PADA PENGATURAN IMPEDANSI ROBOT

Dari gambaran pengaturan impedansi berdasarkan sistem pengaturan posisi secara umum di atas, akan diaplikasikan pengatur posisi kokoh pada pengaturan impedansi robot. Untuk selanjutnya akan ditulis kembali persamaan target impedansi yang telah dijelaskan pada **BAB II**, persamaan (2.12) sebagai berikut :

$$\mathbf{M}_e(\ddot{\mathbf{X}} - \ddot{\mathbf{X}}_d) + \mathbf{B}_e(\dot{\mathbf{X}} - \dot{\mathbf{X}}_d) + \mathbf{K}_e(\mathbf{X} - \mathbf{X}_d) = \mathbf{F}_{int} \quad (4.1)$$

Persamaan di atas dapat dikembangkan sebagai berikut :

$$\mathbf{M}_e\ddot{\mathbf{X}} + \mathbf{B}_e\dot{\mathbf{X}} + \mathbf{K}_e\mathbf{X} = \mathbf{F}_{int} + \mathbf{M}_e\ddot{\mathbf{X}}_d + \mathbf{B}_e\dot{\mathbf{X}}_d + \mathbf{K}_e\mathbf{X}_d \quad (4.2)$$

Pada bab sebelumnya telah disebutkan bahwa harus menentukan trayektori output referensi dalam hal ini persamaan (3.11).

$$\ddot{y}_i^d + a_{1,i}\dot{y}_i^d + a_{0,i}y_i^d = a_{0,i}V_i^d, \quad i = 1, 2, 3, \dots, n \quad (4.3)$$

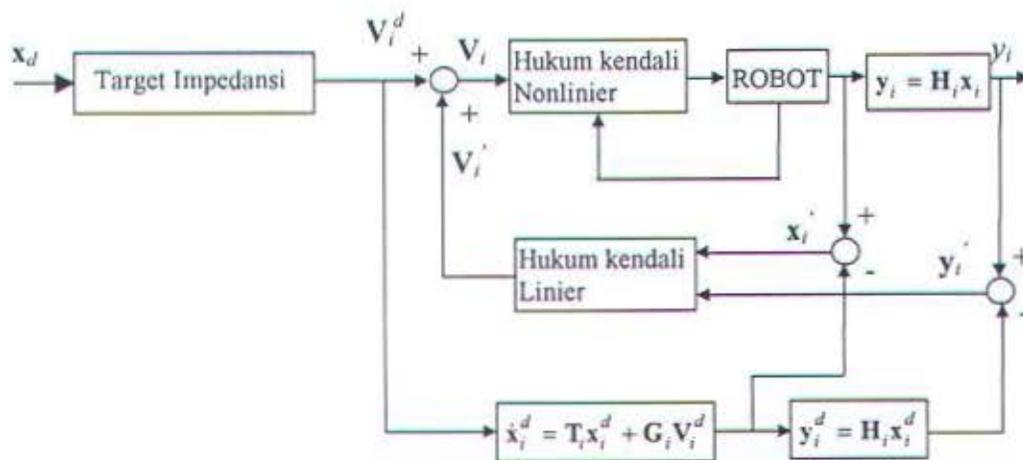
Persamaan (4.2) akan digunakan sebagai trayektori output referensi. Dengan menyamakan persamaan (4.2) dan (4.3) diperoleh :

$$a_{0,i} = \frac{K_e}{M_e} \quad (4.4)$$

$$a_{1,i} = \frac{B_e}{M_e} \quad (4.5)$$

$$V_i^d = \frac{(F_{int} + M_e \ddot{X}_d + B_e \dot{X}_d + K_e X_d) / M_e}{a_{0,i}} \quad (4.6)$$

Variabel-variabel yang telah ditentukan pada persamaan (4.4), persamaan (4.5), persamaan (4.6) dan **Gbr 3.2** akan digunakan pada skema pengaturan impedansi berdasarkan pengaturan posisi kokoh yang ditunjukkan oleh **Gbr 4.3**.



Gambar 4.3. Diagram blok pengaturan posisi kokoh pada pengaturan impedansi

## BAB V

### EKSPERIMEN DENGAN SIMULASI

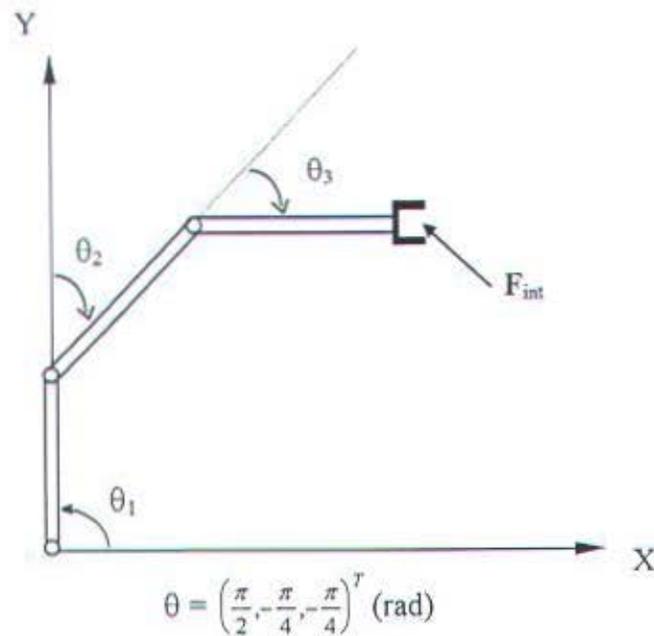
Untuk mengevaluasi validitas dari kontroler yang dikembangkan, contoh numerik diberikan untuk kasus robot planar. Data hasil simulasi pengaturan impedansi robot yang diusulkan pada tugas akhir ini akan dibandingkan dengan data hasil simulasi pengaturan impedansi konvensional.

Kasus yang diberikan adalah untuk lengan robot planar dengan tiga *joint*. Untuk mempermudah perhitungan, digunakan jenis dan karakteristik *link* yang sama. Parameter-parameter yang dimiliki oleh tiap *link* diberikan pada Tabel 5.1. Posisi awal lengan robot ditunjukkan oleh **Gbr 5.1**. Kemudian robot diberikan suatu gaya eksternal di mana posisi robot akan terus bergerak selama gaya eksternal itu diberikan padanya. Gerakan posisi robot ini harus memenuhi karakteristik target impedansi yang diinginkan.

Tujuan ataupun harapan yang diinginkan adalah posisi awal *end effector* bergerak ke posisi akhir sesuai dengan karakteristik target impedansi yang diinginkan.

*Tabel 5.1. Data-data manipulator*

Data manipulator	<i>link</i> $i$ ( $i=1,2,3$ )
Panjang(m)	0.2
Massa(kgm)	1.57
Moment Inersia(kgm <sup>2</sup> )	0.8



Gambar 5.1. Posisi awal robot

## 5.1. PENGATURAN IMPEDANSI KONVENSIONAL

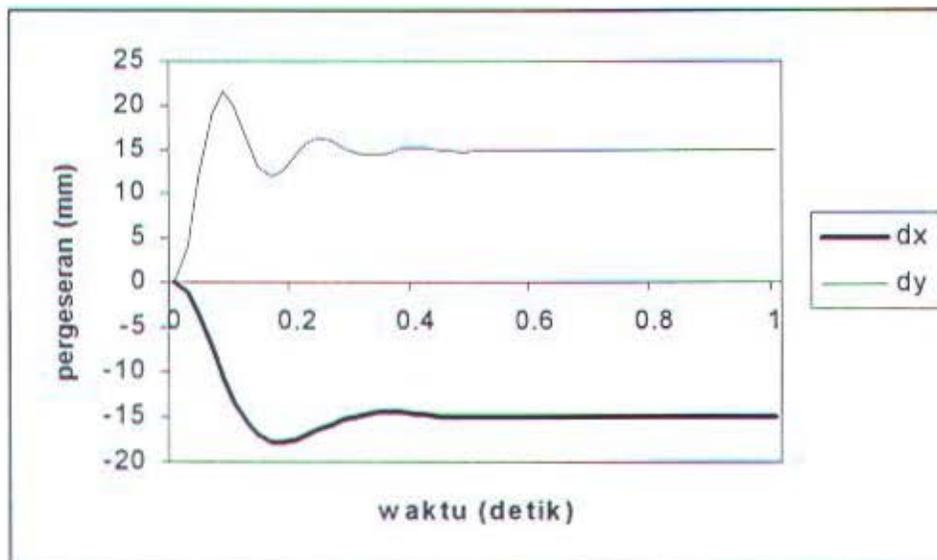
Dalam simulasi ini akan ditunjukkan kinerja dari pengaturan impedansi konvensional. Pada simulasi yang pertama akan digunakan data-data robot yang diketahui secara pasti. Pada simulasi kedua, data-data robot yang digunakan dalam kontroler impedansi konvensional diperbesar 80% dari data-data yang ditunjukkan oleh Tabel 5.1 kecuali panjang *link*. Pada simulasi ketiga, panjang *link* diperbesar 0,005 m dan data-data robot diperbesar 80%. Parameter-parameter dari target impedansi adalah sebagai berikut :

- a. Gaya eksternal yang diberikan pada robot adalah :  $F_{int} = [-6, 24]^T$  (N).
- b.  $M_e = \text{diag.}[1, 1]$  (kg).
- c.  $B_e = \text{diag.}[20, 20]$  (N/(m/s)).
- d.  $K_e = \text{diag.}[400, 1600]$  (N/m).

Dengan parameter target impedansi di atas dan menggunakan persamaan (2.12) dapat diperoleh  $\xi = 0.5$  untuk sumbu X dan  $\xi = 0.25$  untuk sumbu Y.

#### 1. Data-data robot diketahui secara pasti

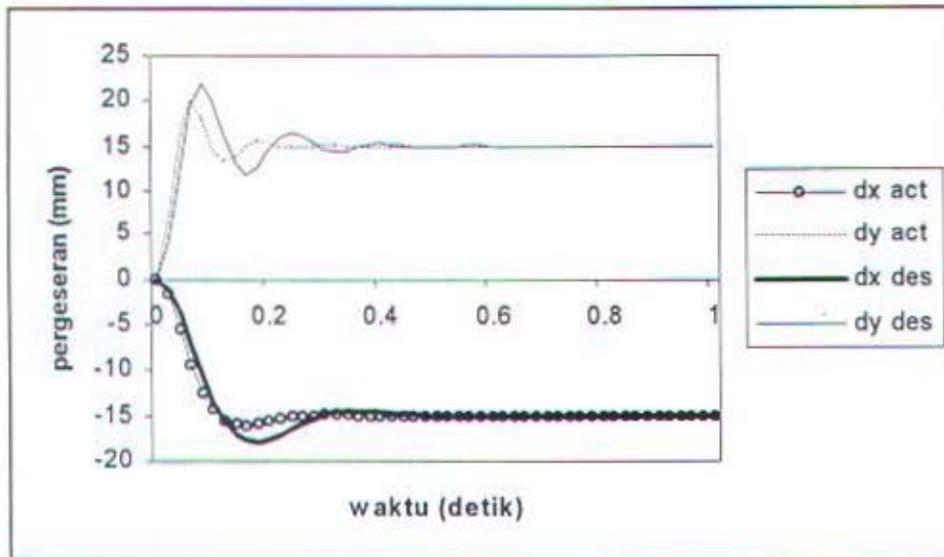
Hasil simulasi untuk pengaturan impedansi konvensional di mana data-data robot diketahui secara pasti ditunjukkan oleh **Gbr 5.2**. Ini adalah respon yang diinginkan di mana robot bergerak sesuai dengan target impedansi yang telah ditentukan.



Gambar 5.2. Respon kontroler impedansi konvensional di mana data-data robot diketahui secara pasti

#### 2. Data-data robot diperbesar 80% kecuali panjang *link*

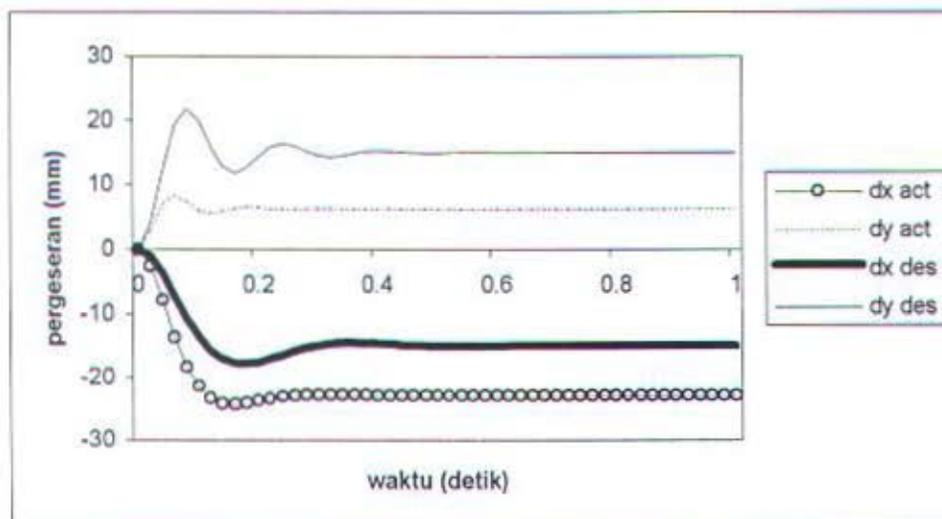
Hasil simulasi pengaturan impedansi konvensional di mana terdapat kesalahan pada data-data robot yang digunakan dalam kontroler ditunjukkan oleh **Gbr 5.3**. Dari gambar tersebut terlihat bahwa pengaturan impedansi konvensional sudah tidak mampu lagi mengendalikan robot agar bergerak sesuai dengan target impedansi yang diinginkan seperti yang ditunjukkan oleh **Gbr 5.2**.



Gambar 5.3. Respon kontroler impedansi konvensional di mana data-data robot diperbesar 80% kecuali panjang *link*.

3. Data-data robot diperbesar 80% dan panjang *link* diperbesar 0,005 m

Gambar 5.4 menunjukkan hasil simulasi pengaturan impedansi konvensional di mana panjang *link* diperbesar 0,005 m dari panjang sebenarnya dan data-data robot diperbesar 80%.



Gambar 5.4. Respon kontroler impedansi konvensional di mana panjang *link* diperbesar 0,005 m dan data-data robot diperbesar 80%.

Dengan adanya kesalahan pada panjang tiap *link* menyebabkan *steady state* pergeseran atau *displacement* menjadi tidak sama dengan *steady state* pergeseran yang diinginkan.

Dari ketiga simulasi di atas terlihat bahwa pengaturan impedansi konvensional sudah tidak mampu lagi memenuhi karakteristik target impedansi yang diinginkan jika terdapat ketidaktepatan data-data robot dan adanya gangguan dari luar.

## 5.2. PENGATURAN POSISI KOKOH PADA PENGATURAN IMPEDANSI ROBOT

Dalam simulasi ini akan digunakan pengaturan posisi kokoh yang telah dibahas pada bab sebelumnya. Parameter-parameter target impedansi yang digunakan dalam simulasi ini sama dengan simulasi sebelumnya. Dari parameter-parameter target impedansi di atas dan persamaan (2.12) dapat diperoleh persamaan target impedansi sebagai berikut :

Sumbu X :

$$(\ddot{x} - \ddot{x}_d) + 20(\dot{x} - \dot{x}_d) + 400(x - x_d) = -6 \quad (5.1)$$

Sumbu Y :

$$(\ddot{y} - \ddot{y}_d) + 20(\dot{y} - \dot{y}_d) + 1600(y - y_d) = 24 \quad (5.2)$$

Kemudian akan dijelaskan perancangan kontroler posisi kokoh pada target impedansi sumbu X. Dengan mengikuti prosedur yang telah dijelaskan pada **BAB III** dapat ditentukan persamaan dekopel dalam koordinat Cartesian berikut ini :

$$\ddot{x} + 20\dot{x} + 400x = 400V + e \quad (5.3)$$

di mana  $e$  menyatakan efek dari *error* permodelan dan gangguan dari luar. Output referensi ditentukan oleh

$$\ddot{x}_d + 20\dot{x}_d + 400x_d = 400V_d \quad (5.4)$$

Persamaan (5.4) dikurangkan dengan persamaan (5.3), diperoleh persamaan *error tracking* sebagai berikut

$$\ddot{x}' + 20\dot{x}' + 400x' = 400V' + e \quad (5.5)$$

Persamaan di atas dapat dinyatakan dalam bentuk *state space* sebagai berikut :

$$\dot{x}' = \mathbf{T}x' + \mathbf{G}V' + \mathbf{G}'e \quad (5.6)$$

$$y' = \mathbf{H}x' \quad (5.7)$$

di mana

$$x' = \begin{bmatrix} y' \\ \dot{y}' \end{bmatrix} \quad (5.8)$$

$$\mathbf{T} = \begin{bmatrix} 0 & 1 \\ -400 & -20 \end{bmatrix} \quad (5.9)$$

$$\mathbf{G} = \begin{bmatrix} 0 \\ 400 \end{bmatrix} \quad (5.10)$$

$$\mathbf{G}' = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5.11)$$

$$\mathbf{H}' = [1 \ 0] \quad (5.12)$$

Jika dipilih  $p = 1$  untuk ketaktentuan (lihat (3.22)) dan persamaan ketaktentuan tersebut disamadengankan nol atau dengan kata lain

$$e_i^{(1)} = b_i e_i = 0 \quad (5.13)$$

maka

$e_i =$  konstanta sembarang (fungsi *step*)

$b_i = 0$

Dengan mengikuti prosedur yang dijelaskan dalam persamaan (3.23) - (3.32),

diperoleh

$$\mathbf{Z} = [\mathbf{y}' \quad \mathbf{z}]^T \quad (5.14)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -400 & -20 \end{bmatrix} \quad (5.15)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 400 \end{bmatrix} \quad (5.16)$$

$$\mathbf{C} = [1 \quad 0 \quad 0] \quad (5.17)$$

Karena  $(\mathbf{T}, \mathbf{G})$  *controllable* maka  $(\mathbf{A}, \mathbf{B})$  juga *controllable*. Umpan balik adalah sebagai berikut :

$$\mathbf{s} = -\mathbf{KZ} \quad (5.18)$$

dan

$$\mathbf{K} = [K_1 \quad K_2 \quad K_3] \quad (5.19)$$

Sehingga dapat ditentukan pole-pole sistem *loop* tertutup  $(\mathbf{A}, \mathbf{B})$  untuk menghitung vektor gain umpan balik  $\mathbf{K}$ . Pole-pole sistem *loop* tertutup  $(\mathbf{A}, \mathbf{B})$  harus diletakkan jauh di sebelah kiri sumbu imajiner bidang  $s$  dari pada pole-pole pada persamaan (5.3) untuk mendapatkan sifat-sifat kekokohan (*robustness*) yang bagus. Ditentukan pole-pole *loop* tertutup berikut pada sistem  $(\mathbf{A}, \mathbf{B})$  :

$$p_1, p_2, p_3 = -100, -55, -50$$

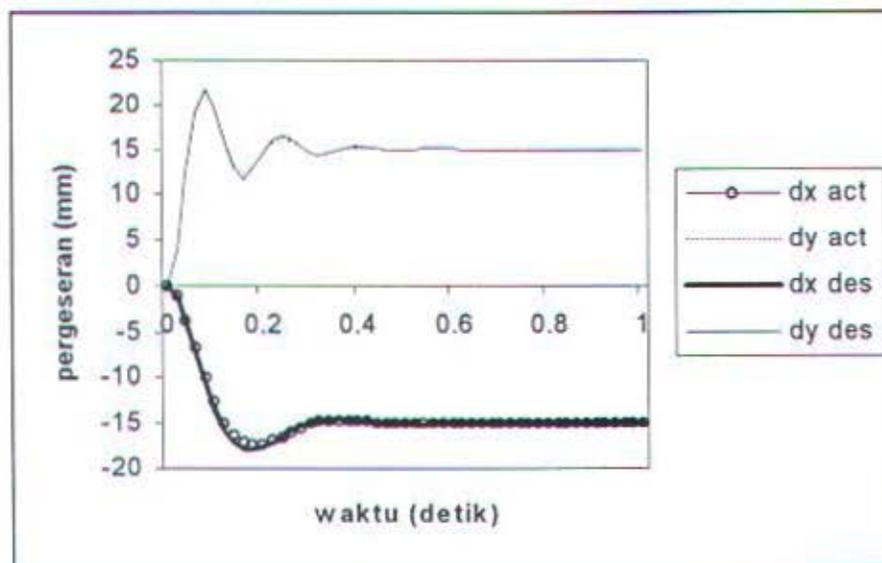
Berdasarkan pada pole-pole *loop* tertutup ini, diperoleh *gain* umpan balik sebagai berikut :

$$K_1 = 687.5, K_2 = 32.125, K_3 = 0.4125$$

Prosedur perhitungan yang sama juga dilakukan pada target impedansi sumbu Y. Tetapi parameter target impedansi diganti dengan  $M_e = 1$ ,  $B_e = 20$ ,  $K_e = 1600$  dan  $F_{int} = 24$ .

1. Data-data robot diperbesar 80% kecuali panjang *link*.

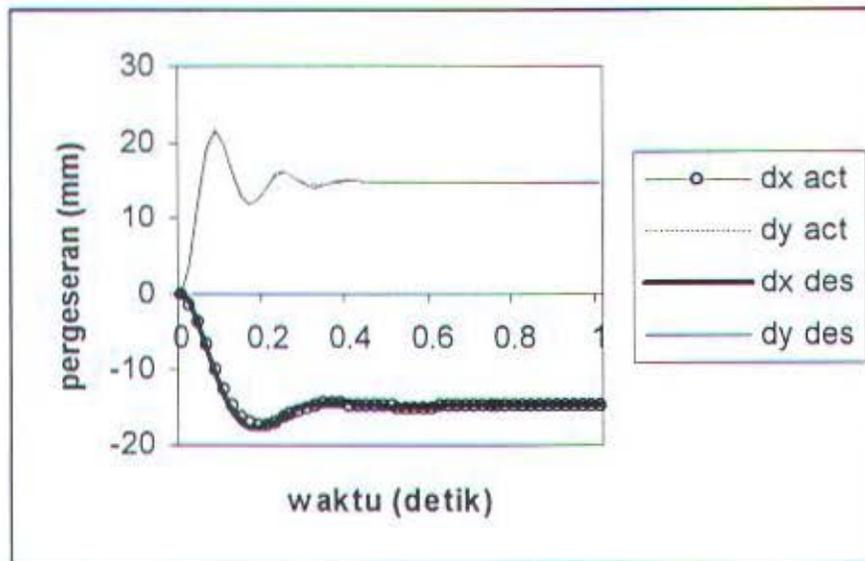
Dari simulasi yang telah dilakukan diperoleh hasil seperti pada **Gbr 5.5**. Dari gambar tersebut terlihat bahwa robot dapat bergerak sesuai dengan target impedansi yang diinginkan.



Gambar 5.5. Respon menggunakan metode yang diusulkan di mana data-data robot diperbesar 80% kecuali panjang *link*.

2. Panjang *link* diperbesar 0,005 m dan data-data robot diperbesar 80%.

Hasil simulasi ditunjukkan oleh Gbr 5.6. Terlihat bahwa pengaturan impedansi yang diusulkan dalam tugas akhir ini mampu menjaga *steady state* pergeseran tetap sama dengan *steady state* pergeseran yang diinginkan.



Gambar 5.6. Respon menggunakan metode yang diusulkan di mana panjang *link* diperbesar 0,005 m dan data-data robot diperbesar 80%.

Dari hasil simulasi di atas terlihat bahwa pengaturan impedansi yang diusulkan dalam tugas akhir ini mampu memenuhi karakteristik target impedansi meskipun terdapat ketidaktepatan pada data-data robot yang digunakan dalam kontroler dan adanya gangguan dari luar. Perbandingan data hasil simulasi pengaturan impedansi konvensional dengan metode yang diusulkan ditunjukkan oleh Tabel 5.2 dan Tabel 5.3. Tabel 5.4 dan Tabel 5.5 menunjukkan tabel kesalahan data hasil simulasi. Dari tabel tersebut dapat hitung *root mean square error* dari hasil simulasi pengaturan impedansi konvensional dan metode yang diusulkan.

Tabel 5.2. Data hasil simulasi di mana data robot diperbesar 80% kecuali panjang link

Waktu (detik)	Tanpa kesalahan (mm)		Pengaturan Impedansi Konvensional (mm)		Metode yang diusulkan (mm)	
	dx	dy	dx	dy	dx	dy
0	0	0	0	0	0	0
0.1	-13.443	20.029	-14.435	14.752	-12.65	20.01
0.2	-17.561	14.371	-15.545	15.419	-17.30	14.25
0.3	-14.930	14.606	-14.940	15.065	-15.04	14.73
0.4	-14.666	15.307	-14.994	14.996	-14.68	15.24
0.5	-15.053	14.887	-15.002	14.997	-15.03	14.90
0.6	-15.039	15.017	-15.000	14.999	-15.03	15.02
0.7	-14.988	15.006	-15.000	15.000	-14.99	15.00
0.8	-14.996	14.993	-15.000	15.000	-14.99	14.99
0.9	-15.002	15.002	-15.000	15.000	-15.00	15.00
1	-15.000	14.999	-15.000	15.000	-15.00	14.99

Tabel 5.3. Data hasil simulasi di mana panjang tiap *link* diperbesar 0,005 m dan data robot diperbesar 80%

Waktu (detik)	Tanpa kesalahan (mm)		Pengaturan Impedansi Konvensional (mm)		Metode yang diusulkan (mm)	
	dx	dy	dx	dy	dx	dy
0	0	0	0	0	0	0
0.1	-13.443	20.029	-21.552	6.0674	-12.69	20.10
0.2	-17.561	14.371	-23.787	6.4664	-17.35	14.18
0.3	-14.930	14.606	-22.884	6.3362	-15.04	14.75
0.4	-14.666	15.307	-22.950	6.3061	-14.67	15.24
0.5	-15.053	14.887	-22.965	6.3058	-15.03	14.89
0.6	-15.039	15.017	-22.961	6.3067	-15.04	15.02
0.7	-14.988	15.006	-22.961	6.3068	-14.99	15.00
0.8	-14.996	14.993	-22.961	6.3068	-14.99	14.99
0.9	-15.002	15.002	-22.961	6.3068	-15.00	15.00
1	-15.000	14.999	-22.961	6.3068	-15.00	14.99

Tabel 5.4. Kesalahan data hasil simulasi di mana data robot diperbesar 80% kecuali panjang *link*

Waktu (detik)	Pengaturan Impedansi Konvensional (mm)		Metode yang diusulkan (mm)	
	$ dx - dx_d $	$ dy - dy_d $	$ dx - dx_d $	$ dy - dy_d $
0	0	0	0	0
0.1	0.992	5.277	0.793	0.019
0.2	2.016	1.048	0.261	0.121
0.3	0.01	0.459	0.11	0.2
0.4	0.328	0.311	0.014	0.067
0.5	0.051	0.11	0.023	0.153
0.6	0.039	0.018	0.009	0.019
0.7	0.012	0.006	0.002	0.012
0.8	0.004	0.007	0.006	0.006
0.9	0.002	0.002	0.002	0.002
1	0	0.001	0	0

Dari tabel di atas dapat diperoleh *root mean square error* sebagai berikut :

Pengaturan impedansi konvensional :

*Root mean square error* sumbu X = 0.6849

*Root mean square error* sumbu Y = 1.6311

Metode yang diusulkan :

*Root mean square error* sumbu X = 0.25404

*Root mean square error* sumbu Y = 0.08709

Tabel 5.5. Kesalahan data hasil simulasi di mana panjang *link* diperbesar 0.005 m dan data robot diperbesar 80%

Waktu (detik)	Pengaturan Impedansi Konvensional (mm)		Metode yang diusulkan (mm)	
	$ dx - dx_d $	$ dy - dy_d $	$ dx - dx_d $	$ dy - dy_d $
0	0	0	0	0
0.1	8.109	13.9616	0.753	0.071
0.2	6.226	7.9046	0.211	0.191
0.3	7.954	7.2698	0.11	0.144
0.4	8.284	9.0009	0.004	0.067
0.5	7.912	8.5812	0.023	0.003
0.6	7.922	8.7103	0.001	0.003
0.7	7.973	8.6992	0.002	0.006
0.8	7.965	8.6862	0.006	0.003
0.9	7.959	8.6952	0.002	0.002
1	7.961	8.6922	0	0

Dari tabel di atas dapat diperoleh *root mean square error* sebagai berikut :

Pengaturan impedansi konvensional :

$$\text{Root mean square error sumbu X} = 7.4803$$

$$\text{Root mean square error sumbu Y} = 8.7546$$

Metode yang diusulkan :

$$\text{Root mean square error sumbu X} = 0.2382$$

$$\text{Root mean square error sumbu Y} = 0.0779$$

## BAB VI

### PENUTUP

#### 6.1. KESIMPULAN

Dari pembahasan yang telah dilakukan pada tugas akhir ini, ada beberapa kesimpulan yang dapat diambil. Kesimpulan-kesimpulan tersebut adalah :

1. Pada tugas akhir ini telah dikembangkan suatu pengaturan impedansi robot yang mampu mengatasi timbulnya *error* yang disebabkan oleh ketidaktepatan data-data robot dan adanya gangguan dari luar, sehingga robot mampu bergerak sesuai dengan target impedansi yang diinginkan.
2. Dari hasil simulasi dan eksperimen yang telah dilakukan menunjukkan bahwa pengaturan impedansi konvensional sudah tidak mampu lagi untuk mengatasi *error* yang muncul akibat dari ketidaktepatan data-data *manipulator* dan gangguan dari luar.
3. Pengaturan impedansi robot yang dikembangkan dalam tugas akhir ini telah menunjukkan kinerjanya dengan baik. Dari hasil simulasi dan eksperimen menunjukkan bahwa kontroler ini mampu menekan *error* yang muncul sebagai akibat dari ketidaktepatan data-data manipulator dan gangguan dari luar. Sehingga robot mampu bergerak sesuai dengan karakteristik target impedansi yang telah ditentukan.

## 6.2. SARAN

Ada beberapa saran yang dirasa perlu yaitu antara lain:

1. Perlu dilakukan simulasi dan eksperimen yang lebih banyak lagi dengan aplikasi-aplikasi yang berbeda untuk menguji sampai sejauh mana kemampuan pengaturan impedansi yang diusulkan dalam tugas akhir ini dalam mengatasi *error* atau kesalahan yang disebabkan oleh ketidaktepatan data-data robot dan adanya gangguan dari luar.
2. Perlu dikembangkan suatu metode pengaturan impedansi robot yang tidak hanya membahas impedansi di *end-effector*, akan tetapi juga memperhitungkan impedansi *joint*. Sehingga interaksi robot dengan lingkungan dapat dilakukan dengan lebih baik.

## DAFTAR PUSTAKA

- Asada, H. and J.J.E. Slotine. (1986). *Robot Analysis and Control*. Cambridge, Massachuttes: John Wiley and Sons, Inc.
- Colgate, E., and Hogan, N. (1989). An analysis of contact instability in terms of passive physical equivalents. *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pp.404-409.
- Craig, J. J. (1989). *Introduction to Robotics*, 2<sup>nd</sup> ed. Reading MA : Addison-Wesley.
- Franklin, G. F. and Naeni, E. A. (1983). A foundation of the multivariable information servomachanism problem. *Int. Rep., Stanford Univ., Information Sciences Lab*.
- Hogan, N. (1985). Impedance Control: An Aproach to Manipulation Part I, II, III. *ASME J. Dyn. Syst., Meas., Control*, vol. 107, pp. 1-24.
- Hogan, N. (1987). Stable execution of contact tasks using impedance control. *Proc. IEEE Conf. Robotics Automat.*, pp. 1047-1054.
- Inoue, H. (1971). Computer controlled bilateral manipulator. *Bulletin, Japan Soc. Mech. Eng.*, vol. 14, no. 69, pp.199-207.
- Jazidie, A. (1995). *Modeling and Simulation impedance Control for Redundant Manipulator Systems*. Doctoral Dissertation. Faculty of Engineering, Hiroshima University.

- Kuo, C. Y. and Wang, S. P. T. (1989). Nonlinear robust industrial robot control. *ASME Trans., J. Dynam. Syst. Measurement Contr.*, vol. 111, no. 1, pp. 24-30.
- Kuo, C. Y. and Wang, S. P. T. (1991). Robust Position Control of Robotic Manipulator in Cartesian Coordinate. *IEEE Trans. On Robotics and Automation*, vol. 7, no. 5, pp. 653-659.
- Kwakernaak, H., and Sivan, R. (1972) *Linear Optimal Control Systems*. New York: Wiley-Interscience.
- Luo, Z. W., and Ito, M. (1993). Control design of robot for compliant manipulation on dynamic environments. *IEEE Transactions on Robotics and Automation*, RA-9-3, pp.286-296.
- Mason, M. T. (1981). Compliance and Force Control for Computer Controlled Manipulators. *IEEE Trans. On Syst, Man, and Cybernetics*, vol. SMC-11, no. 6, pp. 418-432.
- Paul, R. (1972). Modelling, trajectory calculation and servoing of a computer controlled arm. Stanford Artificial Intelligence Lab., Stanford University, Stanford, CA, Memo. AIM-177.
- Paul, R. and Shimano, B. (1976). Compliance and Control. presented at *Proc., 1976 Joint Automat. Contr. Conf.*
- Raibert, H. M. and Craig, J. J. (1981). Hybrid force/position control of manipulators. Presented at Amer. Soc. Mech. Eng. Winter Annu. Meeting, Chicago, IL.

- Silver, D. (1973). The little robot system. Artificial Intelligence Lab. Massachusetts Institute of Technology, Cambridge, MA, Memo. No. 273.
- Tachi, S., Sakaki, T., Arai, H., Nishizawa, S., and Pelaez-Polo, J. F. (1989). Impedance Control of a direct-drive manipulator without using force sensors. *Journal of Robotics Society of Japan*, 7-3, pp.172-184 (in Japanese).
- Wibowo, A. B. (1996). *Perancangan dan Pembuatan Prgram Simulasi Dinamik Lengan Robot dengan Menggunakan Metode Appel-Gibbs*. Tugas Akhir, Teknik Elektro ITS, 1996.

## LAMPIRAN A

### TRANFORMASI DARI *JOINT SPACE* KE *CARTESIAN SPACE*

---

Dari persamaan dinamika robot di *joint space* :

$$M\ddot{\theta} + h + c = \tau + J^T F_{int} \quad (A.1)$$

Dari hubungan kecepatan *end effector* dengan kecepatan *joint* diperoleh :

$$\dot{X} = J\dot{\theta}$$

$$\ddot{X} = J\ddot{\theta} + \dot{J}\dot{\theta}$$

$$\ddot{\theta} = J^{-1}(\ddot{X} - \dot{J}\dot{\theta}) \quad (A.2)$$

$$\tau = J^T F \quad (A.3)$$

persamaan (A.2) dan persamaan (A.3) masuk ke persamaan (A.1) diperoleh :

$$MJ^{-1}(\ddot{X} - \dot{J}\dot{\theta}) + h + c = J^T F + J^T F_{int}$$

$$MJ^{-1}\ddot{X} + h - MJ^{-1}\dot{J}\dot{\theta} + c = J^T F + J^T F_{int}$$

$$(J^T)^{-1}MJ^{-1}\ddot{X} + (J^T)^{-1}h - (J^T)^{-1}MJ^{-1}\dot{J}\dot{\theta} + (J^T)^{-1}c = F + F_{int} \quad (A.4)$$

Dari persamaan (A.4) diperoleh :

$$D_x = (J^T)^{-1}MJ^{-1}$$

$$h_x = (J^T)^{-1}h - D_x\dot{J}\dot{\theta}$$

$$c_x = (J^T)^{-1}c$$

sehingga persamaan dinamika robot dalam koordinat Cartesian adalah :

$$D_x\ddot{X} + h_x + c_x = F + F_{int} \quad (A.5)$$

**LAMPIRAN B**  
**ALGORITMA PROGRAM SIMULASI**  
**KONTROLER IMPEDANSI KONVENSIONAL**

---

1. Memasukkan harga-harga parameter lengan robot :  
massa, panjang, titik berat, momen inersia, konfigurasi awal dari lengan robot,  
jumlah *link* dan waktu iterasi ( $\Delta t$ ).
2. Menentukan parameter-parameter target impedansi.
3. Dengan harga matriks  $\theta$  mendapatkan posisi *end-effector*.
4. Menghitung harga matriks Jacobian.
5. Menghitung harga-harga matriks  $\Theta$ ,  $\beta$ ,  $\phi$ ,  $\tau$ ,  $\Omega$ ,  $D$ ,  $V$ ,  $M$  dan  $Y$ .
6. Menghitung  $\tau_{imp}$ ,  $\tau_{cancel}$ ,  $\tau_{int}$ .
7. Menghitung harga matriks percepatan sudut  $\ddot{\theta}$ , dengan  
$$M\ddot{\theta} = \tau_{imp} + \tau_{cancel} + \tau_{int}$$
8. Menggunakan metode Runge-Kutta untuk mencari harga matriks  $\dot{\theta}$  dan  $\theta$ .
9. Mengulangi langkah 4 ~ 9 sampai waktu yang ditentukan.
10. Dengan harga matriks  $\theta$  mendapatkan posisi akhir dari *end-effector*.
11. Menyimpan data jumlah *link*, panjang, dan harga-harga parameter lengan robot,  
serta waktu tiap iterasi, sudut tiap *joint*, posisi *end-effector* hasil simulasi dalam  
*file*.

LAMPIRAN C

ALGORITMA PROGRAM SIMULASI

MENGUNAKAN METODE YANG DIUSULKAN

---

1. Memasukkan harga-harga parameter lengan robot :  
massa, panjang, titik berat, momen inersia, konfigurasi awal dari lengan robot, jumlah *link* dan waktu iterasi ( $\Delta t$ ).
2. Memasukkan harga-harga parameter lengan robot yang digunakan dalam kontroler.
3. Menentukan parameter-parameter target impedansi.
4. Dengan harga matriks  $\theta$  mendapatkan posisi *end-effector*.
5. Menghitung harga matriks Jacobian.
6. Menghitung harga-harga matriks  $\Theta$ ,  $\beta$ ,  $\phi$ ,  $\tau$ ,  $\Omega$ ,  $D$ ,  $V$ ,  $M$  dan  $Y$ .
7. Menghitung  $\tau_{imp}$ ,  $\tau_{cancel}$ ,  $\tau_{int}$ .
8. Menghitung harga matriks percepatan sudut  $\ddot{\theta}$ , dengan
$$M\ddot{\theta} = \tau_{imp} + \tau_{cancel} + \tau_{int}.$$
9. Menggunakan metode Runge-Kutta untuk mencari harga matriks  $\dot{\theta}$  dan  $\theta$ .
10. Menentukan trayektori impedansi yang diinginkan.
11. Menentukan input kendali linier tambahan  $V_i'$ .
12. Mengulangi langkah 4 ~ 11 sampai waktu yang ditentukan.
13. Dengan harga matriks  $\theta$  mendapatkan posisi akhir dari *end-effector*.

14. Menyimpan data jumlah *link*, panjang, dan harga-harga parameter lengan robot, serta waktu tiap iterasi, sudut tiap *joint*, posisi *end-effector* hasil simulasi dalam *file*.

## RIWAYAT HIDUP

---



Adi Sarwanto dilahirkan di Kediri 23 Februari 1973, sebagai anak pertama dari dua bersaudara keluarga Bapak Saridjo. Bertempat tinggal di Jl. Kawi no 51 Kediri. Sejak tahun 1991 terdaftar sebagai mahasiswa Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya.

### Riwayat Pendidikan :

- TK Pertiwi Bojonegoro lulus tahun 1979
- SDN Kadipaten 1 Bojonegoro lulus tahun 1985.
- SMPN 1 Bojonegoro lulus tahun 1988.
- SMAN 1 Bojonegoro lulus tahun 1991.

Pada bulan Juli tahun 1997 mengikuti seminar dan ujian Tugas Akhir di Bidang Studi Sistem Pengaturan, Jurusan Teknik Elektro FTI-ITS sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektro.