

PENGENDALIAN MANIPULATOR ROBOT DENGAN FUNGSI BASIS RADIAL

TUGAS AKHIR

Oleh :

LASMAN P. PURBA

2293 100 100

PERPUSTAKAAN ITS	
Terima	31-7-2000
Terima Dari	H
No. Agenda Prp.	21.1424



RSE
629.892
Pur
p-1

2000

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2000

PENGENDALIAN MANIPULATOR ROBOT DENGAN FUNGSI BASIS RADIAL

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro

Pada

Bidang Studi Teknik Sistem Pengaturan

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui,

Dosen Pembimbing



DR. Ir. Achmad Jazidie, M.Eng.

NIP. 131 647 788

**SURABAYA
PEBRUARI 2000**

ABSTRAK

Pergerakan manipulator robot akan mengikuti lintasan yang dikehendaki bila parameter-parameternya diketahui dengan benar. Khususnya parameter massa manipulator diikuti dengan momen inersianya sukar dipastikan besarnya, termasuk dengan struktur / bentuk *link* dan gesekan / friksi. Disamping algoritma Back Propagasi (BP), jaringan syaraf tiruan yang banyak diterapkan adalah Fungsi Basis Radial (RBF). RBF dipakai secara *off-line* untuk meniru parameter manipulator robot tersebut, kemudian parameter hasil tiruan / *learning* dipakai untuk pengendalian yang sama. Hasilnya penggunaan RBF mampu merepresentasikan kembali pengendalian tanpa RBF, matrik massa di *joint space* dengan input vektor lintasan *end-effector* setiap waktu dapat di-*learning* dengan hasil baik, yaitu matrik massa tersebut. Simulasi program komputer dipakai untuk merepresentasikan pengendaliannya.

KATA PENGANTAR

Segala puji, syukur hanya bagi Allah, sebab oleh kehendakNya saya ada. Keadaan saya selalu ada dalam hal yang terbaik didalam pandanganNya. Saya menyelesaikan tugas akhir ini, dengan judul :

PENGENDALIAN MANIPULATOR ROBOT

DENGAN FUNGSI BASIS RADIAL

juga adalah atas pengendalianNya terhadap ruang dan waktu serta lainnya.

Tugas akhir ini disusun untuk memenuhi sebagian persyaratan bagi setiap mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, agar dapat menyandang gelar sarjana.

Tugas akhir ini disusun dari gagasan yang pada awalnya agak tidak jelas. Kemudian berkembang seperti sekarang ini. Matriks massa diikuti momen inersia dan hal nonlinear lainnya dalam pengendalian secara terbatas dipelajari dengan jaringan syaraf tiruan fungsi basis radial. Proses tugas ini membuat saya belajar memberi harga hasil karya sendiri. Saya senang keunikannya dan kekurangan dan kelebihanannya.

Dan akhirnya, penulis berharap karya ini dapat menjadi berkat bagi setiap orang yang akan mengembangkan ilmu robotika. Penulis sangat menyadari setiap kelemahan dan kekurangan serta keterbatasan karya ini, oleh karena itu setiap kritikan dan saran yang sifatnya membangun merupakan penghargaan yang sangat besar maknanya bagi penulis.

Surabaya, Pebruari 2000

Penulis.

UCAPAN TERIMA KASIH

Banyak terima kasih saya sampaikan dari lubuk hati, untuk segala bantuan dan dukungan untuk pengerjaan sampai selesainya tugas akhir ini. Dengan segala keterbatasan dan kemampuan saya, dan juga waktu yang ada, saya ingin sampaikan rasa terima kasih yang dalam kepada :

- Bapak Dr. Ir. Achmad Jazidie M. Eng., selaku dosen pembimbing dalam penyusunan tugas akhir ini yang telah banyak memberikan kritikan yang tajam dan mengesankan. Tidak lupa juga memberi nasehat yang memberi semangat dan bimbingan kepada penulis. Atas dukungannya tugas akhir ini berjalan dengan lancar.
- Bapak Dr. Ir. Moch. Rameli, selaku dosen matakuliah pertama bidang studi sistem pengaturan, yang mengajari saya membuat presentasi yang bagus. Tidak saya lupa berterima kasih akan fasilitas laboratorium yang banyak memberi manfaat.
- Bapak Dr. M. Nuh, selaku dosen penguji. Saya cukup memahami harapan bapak dalam kesempurnaan tugas akhir ini.
- Bapak Ir. Joko Susila, selaku dosen yang sering saya lihat di laboratorium dan membenahi PLC. Dorongan menyelesaikan tugas akhir saya dapatkan juga, terima kasih.
- Segenap staf dosen pengajar Bidang Studi Teknik Sistem Pengaturan dan seluruh keluarga besar Jurusan Teknik Elektro.
- Rekan-rekan di Bidang Studi Teknik Sistem Pengaturan, yang namanya saya tidak sebutkan disini. Bersama kalian saya belajar menikmati hidup keluarga di laboratorium.

- Rekan-rekan Elektro ITS khususnya E-33 dimanapun berada, atas adanya keterlambatan dan kecepatan lulus. Teman-teman setidaknya membuat saya belajar keras.
- Bapa dan Mama, yang oleh kasih yang terbatas telah memelihara saya, membesarkan saya, membawa saya sekolah, dan belajar. Jerih payahmu tak kulupa. Bang Jonni, Kak Masri, Dek Rinando, Adek Vika, juga Keluarga saya lainnya yang tidak saya tau semua.
- Mas Rovy, yang mengajak saya melihat hal yang lebih tinggi nilainya. Nilai kekekalan. Nilai keyakinan.
- Teman-teman RL yang membuat saya belajar secara khusus menjadi team.
- Teman-teman Persekutuan PMK, atas semua hal yang saya terima. Untuk Evi Thelia juga, yang oleh kasih karunia bertemu dengan cara yang berbeda.
- Dan semua pihak yang tidak mungkin saya sebutkan satu demi satu. Benar saya berterimakasih.

DAFTAR ISI

Judul	i
Halaman Pengesahan	ii
Abstrak	iii
Kata Pengantar	iv
Ucapan Terima Kasih	v
Daftar Isi	vii
Daftar Gambar	x
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Permasalahan	3
I.3 Rumusan Masalah	4
I.4 Metode Penelitian	4
I.4 Tujuan	5
I.5 Sistematika Penulisan	5
I.6 Relevansi	6
BAB II DINAMIKA MANIPULATOR ROBOT	7
II.1 Kinematika Manipulator Robot	7
II.2 Dinamika Manipulator Robot	12
II.2.1 Pengendalian Dalam Joint Space	14
II.2.1.1 Persamaan Ruang Keadaan	15
II.2.1.2 Persamaan Ruang Konfigurasi	16

	II.2.2 Pengendalian Dalam Task Space	18
	II.2.2.1 Persamaan Ruang Keadaan	18
	II.2.2.2 Persamaan Ruang Konfigurasi	19
	II.3 Lintasan Manipulator Robot	21
BAB III	KARAKTERISTIK FUNGSI BASIS RADIAL	23
	III.1 Pendahuluan	23
	III.2 Jaringan Syaraf Manusia	23
	III.3 Defenisi Jaringan Syaraf Tiruan	24
	III.4 Jaringan Syaraf Tiruan Fungsi Basis Radial	25
	III.5 Strategi Learning Fungsi Basis Radial	29
	III.5.1 Pusat – Pusat Diseleksi Secara Random	30
	III.5.2 Pusat – Pusat Diseleksi Secara Self-Organized	31
	III.5.2.1 Unsupervised Pada Lapis Input	32
	III.5.2.2 Supervised Pada Lapis Output	33
	III.5.3 Pusat – Pusat Diseleksi Secara Supervised	34
	III.6 Jaringan Syaraf Tiruan Back Propagasi	35
BAB IV	PERENCANAAN PENGENDALIAN	37
	IV.1 Algoritma Konvensional	38
	IV.2 Algoritma Pengendalian Usulan	39
BAB V	ANALISA DENGAN SIMULASI	42
BAB VI	KESIMPULAN	49
	VI.1 Kesimpulan	49

VI.2 Saran	49
DAFTAR PUSTAKA	50
LAMPIRAN A	
LAMPIRAN B	

DAFTAR GAMBAR

Gambar 2.1	Manipulator Planar 2 link	9
Gambar 2.2	Blok Diagram Pengendalian Manipulator Robot di Joint Space	17
Gambar 2.3	Blok Diagram Pengendalian Manipulator Robot di Task Space	20
Gambar 3.1	Suatu Jaringan RBF Dengan m -Input n -Output.....	29
Gambar 4.1	Blok Diagram Pengendalian Manipulator Robot Usulan, dengan RBF	39
Gambar 5.1	Konfigurasi Awal Robot	43
Gambar 5.2	Kecepatan End-Effector Yang Dikehendaki.....	44
Gambar 5.3	Percepatan End-Effector Yang Dikehendaki.....	44
Gambar 5.4	Posisi x End-Effector Hasil Simulasi $l=17, b=1, d=1$	45
Gambar 5.5	Posisi y End-Effector Hasil Simulasi $l=17, b=1, d=1$	46
Gambar 5.6	Respon Lapis Hidden Simulasi $l=7, b=1, d=0.1$	47
Gambar 5.7	Respon Lapis Hidden Simulasi $l=17, b=1, d=1$	47

BAB I

PENDAHULUAN

I.1 LATAR BELAKANG

Teknologi dipakai membantu manusia meningkatkan budaya hidupnya. Robotika sebagai teknologi automasi, untuk membantu manusia dalam melakukan pekerjaannya meningkat. Teknologi direncanakan dan diwujudkan oleh manusia untuk kepentingan manusia. Manusia ada keterbatasan secara kecepatan, ketepatan serta ketahanan dalam melakukan tugas-tugasnya. Manipulator robot menjadi meningkat pentingnya dalam mengatasi adanya keterbatasan manusia. Mengendalikan manipulator-manipulator robot sesuai dengan tugas (*task*) yang harus dilakukannya secara benar/ akurat, sesuai tugas yang diberikan manusia kepada robot tidaklah mudah. Informasi yang tepat berkenaan dengan obyek yang hendak dikendalikan sangatlah dibutuhkan untuk melakukan pengendalian dengan tepat pula. Pemodelan dilakukan untuk mendeskripsikan semua informasi yang diperoleh tentang manipulator robot tersebut dan untuk memenuhi keperluan pengendalian secara otomatis. Model matematis dari pergerakan manipulator robot yang telah ada tidak mampu memodelkan semua parameter yang mempengaruhi pergerakan manipulator robot tersebut. Dengan kata lain, model dinamika yang ada yang dipakai untuk mendesain pengendali otomatis tersebut tidak sesuai dengan pergerakan manipulator robot yang sesungguhnya.

Metoda *Computed Torque* adalah salah satu metoda pengendalian pergerakan manipulator robot yang akan berhasil dengan baik jika parameter-parameternya

diketahui dengan tepat. Kenyataannya, parameter-parameter tersebut sangatlah sulit untuk dapat diketahui secara tepat sehingga menimbulkan ketidakpastian (*uncertainty*) pada pengendalian manipulator robot, yang mengakibatkan kesalahan pada posisi *end-effector* manipulator robot. Ketidakpastian tersebut ada yang statis, dan dinamis. Ketidakpastian yang statis ditemukan pada saat manipulator robot diam, atau masih akan bergerak. Ketidakpastian statis terjadi akibat ketidaktepatan pengetahuan akan harga parameter manipulator robot yang sebenarnya. Ketidakpastian yang dapat terjadi ketika manipulator robot bergerak dalam waktu tertentu dinamakan ketidakpastian dinamis.

Disisi lain, beban dari robot manipulator bervariasi selama pergerakannya, dan variasinya sangat mungkin tidak diketahui secara luas.

Untuk mengatasi permasalahan tersebut strategi kontrol adaptif telah dikembangkan dan menarik banyak peneliti untuk meneliti, yang ditunjukkan oleh Craig *et al.*[1] dan Slotine and Li [2] sebagai contoh. Umumnya strategi ini tidak menghendaki pengetahuan yang diturunkan dari hukum-hukum fisika, persamaan kesetimbangan maupun interaksi struktural (*a priori knowledge*) mengenai parameter-parameter yang tidak diketahui, seperti massa beban. Strategi kontrol adaptif ini dipakai dalam mengatasi perubahan dari parameter-parameter yang ada dengan secepatnya. Intinya strategi kontrol ini dapat beradaptasi dengan perubahan parameter-parameter sistem. Skema-skema *learning control*, yang memperbaiki performansi sistem ketika gerakan yang sama dikerjakan secara berulang-ulang, dapat mengambil tempat disini oleh Arimoto [3], Craig [4]. Hasil simulasi dan percobaan dari sejumlah peneliti seperti, Goldberg and Pearlmuter [5], Miyamoto *et al.* [6], Narendra and Parthasathy [7], telah dikonfirmasi

potensi dari jaringan syaraf tiruan dalam bidang pemodelan dinamika dan pengendalian sistem-sistem nonlinear.

Dewasa ini beberapa perkembangan telah dibuat dengan menggunakan jaringan syaraf tiruan untuk mengendalikan manipulator robot, seperti yang telah dilakukan oleh Miller *et al.* [8], Saad *et al.* [9]. Pada umumnya, desain strategi kontrol jaringan syaraf tiruan dilakukan dalam dua langkah. Pertama, jaringan syaraf tiruan digunakan untuk mendekati model dinamika dari sistem. Pendekatan ini biasanya dibawa keluar, secara *off-line* dan kemudian, langkah kedua mengkonstruksi suatu strategi kontrol setelah pendekatan yang cukup akurat diperoleh. Pendekatan ini telah ditunjukkan bekerja dengan baik untuk beberapa sistem. Namun, masih belum dapat mengatasi perubahan dari sistem. Disinilah sebenarnya perlu mempersatukan strategi kontrol adaptif. Beberapa pekerjaan yang telah dikerjakan dengan baik belakangan ini memakai suatu jaringan syaraf tiruan yang wajar untuk secara langsung melakukan parameterisasi hukum kontrolnya, oleh Tzirkel-Hancock *et al.* [10]. Sistem *loop* tertutup didapati memiliki stabilitas yang baik disini. Strategi kontrol jaringan syaraf tiruan menghendaki evaluasi model *invers* dinamik, menghendaki cukup kecilnya waktu-terpakai selama proses, ini telah didekati dengan usulan Lewis *et al.* [11] dan Ge and Hang [12]. Jaringan syaraf tiruan dapat secara sederhana diinisialisasi nol dengan asumsi tidak ada pengetahuan mengenai sistem. Disamping itu, kontroler kokoh dan mudah diaplikasikan secara *real-time*.

Penggunaan pendekatan jaringan syaraf tiruan dalam *task space* atau disebut juga *Cartesian space* atau *operational space* atau koordinat *end-effector* manipulator robot untuk menangani perubahan dari sistem secara keseluruhan

atau '*Built-in-Capability*', telah dibahas [13]. Kontroler yang dikembangkan didasarkan pada teknik pemodelan yang tidak menghendaki evaluasi *invers* dinamika model, demikian juga waktu proses *training*.

I.2 RUMUSAN MASALAH

Dalam tugas akhir ini, metode pengendalian yang diusulkan dalam [13] dikembangkan khusus di koordinat *end-effector*. Perencanaan lintasan halus yang dikehendaki dilewati manipulator robot dinyatakan dengan polinomial orde lima, yang dimulai langsung dari gerakan *end-effector*.

I.3 TUJUAN

Mengetahui pengaruh matrik massa model RBF pada dinamika manipulator robot.

I.4 METODE PENELITIAN

Studi pustaka adalah langkah awal yang diambil untuk memberi suatu kesimpulan dalam pengerjaan sampai penyelesaian tugas akhir ini, kemudian metode pengendalian yang sudah ada dikembangkan secara lebih khusus dalam tugas akhir ini. Hal ini menyangkut keaslian topik yang diusulkan. Pengujian melalui simulasi dimaksudkan untuk mengetahui dan merepresentasikan algoritma yang diusulkan, merupakan langkah selanjutnya sebelum menarik kesimpulan dan rekomendasi terhadap perkembangan penelitian dan pengembangan peneliti kemudian sebagai langkah terakhir.

1.5 SISTEMATIKA PENULISAN

Untuk kemudahan memahami keberadaan karya Tugas Akhir ini secara lengkap, penulis menyajikan dengan sistematika sebagai berikut.

Bab 1. Pendahuluan, mengemukakan latar belakang mengambil judul, yang meliputi sejarah singkat penelitian dan penemuan suatu permasalahan dalam batasan-batasan bahasan yang ada. Metode yang dilakukan dalam proses penyelesaiannya disampaikan disini. Tidak lupa mengenai tujuan yang hendak dicapai dan relevansinya diutarakan dalam bab ini.

Bab 2. Dinamika Manipulator Robot, menguraikan mengenai pemodelan pergerakan manipulator robot serta faktor-faktor yang mempengaruhinya baik di koordinat *joint* maupun koordinat *end-effector*. Algoritma pengendalian manipulator robot yang dipakai didasarkan pada algoritma *invers* dinamika.

Bab 3. Karakteristik Fungsi Basis Radial, berisi teori dasar jaringan syaraf tiruan *radial basis function* disingkat RBF, yang mendeskripsikan mulai dari defenisinya, modelnya, arsitektur, sampai algoritma *learning*-nya. Mengenai RBF dalam tulisan ini lebih banyak bersumber dari Haykin S., [14].

Bab 4. Perencanaan Pengendalian, membahas algoritma pengendalian konvensional yang biasa dipakai, kemudian algoritma pengendalian dengan RBF yang diusulkan diuraikan sedemikian rupa sehingga diperoleh gambaran yang jelas dalam proses *learning* RBF, keduanya dengan maksud membandingkan guna mengetahui performansi yang dihasilkan.

Bab 5. Analisa dengan Simulasi, memberikan uraian secara langsung mengenai hasil simulasi yang dilakukan menyusul analisisnya. Program simulasi dinamika manipulator robot yang telah ada dipakai untuk mengambil data berdasarkan hasil

pemodelan dan menggunakan dua *joint planar*, sedangkan pengendalian yang direncanakan disimulasikan dalam Turbo C++ 4.5 untuk aplikasi window.

Bab 6. Kesimpulan, berisi kesimpulan yang diambil dari keseluruhan proses pengerjaan dan penyelesaian uraian tulisan ini, yang disertai sedikit sumbang saran yang penulis merasa perlu sekali disampaikan.

I.6 RELEVANSI

Tugas akhir ini diharapkan nantinya dapat dipergunakan sebagai pembanding dengan algoritma kontrol yang telah ada, khususnya jenis multi lapis perceptron dengan metode *learning back propagation*, dan juga harapan menerapkan pemakaian teknologi non-konvensional dalam pengendalian manipulator robot.

BAB II

DINAMIKA MANIPULATOR ROBOT

Bagian ini membahas tentang kinematika khususnya kinematika langsung dan kinematika tidak langsung yang dilanjutkan dengan uraian mengenai dinamika khususnya *invers* dinamika manipulator robot. Perencanaan lintasan manipulator robot diuraikan setelah mengetahui dinamika manipulator robot.

II.1 KINEMATIKA MANIPULATOR ROBOT

Sebuah benda dikatakan dalam keadaan bergerak bila kedudukan benda tersebut dari saat ke saat berubah. Ilmu tentang gerakan ini tanpa memperhatikan gaya-gaya yang menyebabkan gerakan itu disebut Kinematika yang berasal dari bahasa Yunani "Kinema". Dalam pergerakannya, suatu benda juga dapat berputar / berotasi atau bergetar. Untuk dapat mengabaikan rotasi atau getaran benda, maka benda-benda dianggap sebagai partikel atau digambarkan sebagai titik. Hal ini tidak selalu berarti bahwa benda tersebut berukuran sangat kecil. Contohnya, bumi dan matahari dapat dianggap sebagai partikel mengingat jarak antara bumi dan matahari yang demikian jauhnya. Dengan menganggap bumi dan matahari sebagai partikel, gerak matahari mengelilingi bumi dipelajari. Ternyata hasilnya sangat mendekati keadaan sebenarnya. Ini berarti kesalahan yang dapat ditimbulkan karena menganggap benda yang bukan titik sebagai benda titik (partikel) dalam hal ini dapat diabaikan.

Bila benda bergerak, berarti benda tersebut mempunyai kecepatan. Dalam beberapa buku, kecepatan sebagai vektor dan besarnya kecepatan sering dibedakan

menjadi *velocity* yang dalam bahasa Indonesia adalah kecepatan, dan *speed* yang di Indonesiakan menjadi kelajuan.

Sekarang benda yang dipandang beralih ke manipulator robot. Secara khusus, manipulator dapat dimodelkan sebagai sebuah rangkaian terbuka benda tegar (*link*) yang dihubungkan secara serial dengan *joint revolute* maupun *joint prismatic* yang digerakkan oleh aktuator. Joint revolute berotasi terhadap suatu sumbu gerakan. Joint prismatic pada sebuah sumbu dapat bergerak secara maju-mundur; naik-turun; keluar-masuk (*roll, pitch, yaw*). Dapat didefinisikan ulang kinematika untuk manipulator robot. Kinematika adalah studi terhadap analisa geometri gerakan robot terhadap sistem koordinat referensi yang tetap tanpa memandang gaya ataupun torsi yang menyebabkan gerakan dari manipulator. Sumber gerakan asli adalah aktuator-aktuator pada *joint*, tetapi secara konseptual sumber gerakan pada *joint* berbeda dengan sumber gerakan pada *end-effector*. Hubungan yang menyatakannya dibagi dua, kinematika langsung (*forward kinematic*) dan kinematika tidak langsung (*invers kinematic*).

Tinjauan pergerakan manipulator robot dilakukan dalam dua jenis koordinat, yakni koordinat *joint* dan koordinat *end-effector*. Disebut koordinat *joint* karena keseluruhan peninjauan melibatkan variabel-variabel *joint*, demikian juga koordinat *end-effector* melakukan peninjauan dalam variabel-variabel *end-effector*. Koordinat *joint* sering disebut *Joint Space*. Koordinat *end-effector* sering disebut *Task Space* atau koordinat Cartesian atau *Operational Space*. Ruang yang dapat dilalui oleh kedua koordinat tersebut sering disebut *Work Space*. Untuk selanjutnya dalam tulisan ini cukup memakai *Joint Space* dan *Operational Space* untuk koordinat *joint* dan koordinat *end-effector*.

Dalam koordinat *joint*, percepatan *joint*, $\ddot{\theta}$ dapat dinyatakan sebagai fungsi posisi *joint*, θ ditulis sebagai :

$$\ddot{\theta} = f(\theta) \dots\dots\dots(2.1)$$

dan juga sebagai fungsi waktu, ditulis :

$$\ddot{\theta} = f(t) \dots\dots\dots(2.2)$$

dimana t = waktu (detik). Pemecahan matematis untuk kedua pernyataan itu berbeda, tetapi inti persoalannya sama-sama dibahas dalam variabel-variabel *joint*.

Sedangkan dalam koordinat Cartesian, percepatan *end-effector*, \ddot{X} dapat dinyatakan sebagai fungsi posisi *end-effector*, X ditulis sebagai :

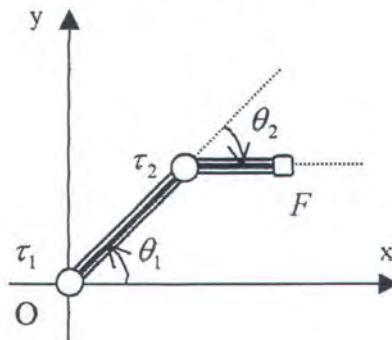
$$\ddot{X} = f(X) \dots\dots\dots(2.3)$$

dan juga sebagai fungsi waktu, ditulis :

$$\ddot{X} = f(t) \dots\dots\dots(2.4)$$

dimana t = waktu (detik). Pemecahan matematis untuk kedua pernyataan itu berbeda, tetapi inti persoalannya sama-sama dibahas dalam variabel-variabel *end-effector*.

Ditunjukkan manipulator robot planar 2 derajat kebebasan pada gambar 2.1.



Gambar 2.1. Manipulator planar 2 link

Persamaan kinematika yang menghubungkan posisi *end-effector* (x, y) dengan posisi *joint* (θ_1, θ_2), diberikan oleh persamaan :

$$x(\theta_1, \theta_2) = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad \dots\dots\dots(2.5)$$

$$y(\theta_1, \theta_2) = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad \dots\dots\dots(2.6)$$

Dua persamaan, persamaan 2.5 dan 2.6 sering disebut persamaan kinematika langsung (*forward kinematics*), yaitu sudut-sudut *joint* (posisi-posisi *joint*) diberikan untuk mencapai posisi *end-effector*. Dengan mendifferensialkan kedua persamaan diatas didapat,

$$dx = \left(\frac{\partial x(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 \right) + \left(\frac{\partial x(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2 \right) \quad \dots\dots\dots(2.7)$$

$$dy = \left(\frac{\partial y(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 \right) + \left(\frac{\partial y(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2 \right) \quad \dots\dots\dots(2.8)$$

dimana : $dX = \begin{bmatrix} dx \\ dy \end{bmatrix}$, adalah pemindahan *end-effector*, perpindahan *virtual* pada *end-effector*, vektor perpindahan *displacement* dari *end-effector*; dan

$$d\theta = \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}, \text{ adalah pemindahan } \textit{joint}, \text{ perpindahan } \textit{virtual} \text{ pada } \textit{joint},$$

vektor perpindahan *displacement* dari *joint*.

Generalisasi persamaan 2.7 dan 2.8 untuk manipulator robot dengan n derajat kebebasan, dan dimensi *task space* adalah m , atau yang menghubungkan kedua *space* diberikan oleh operator Jacobian, dinyatakan,

$$dX = J(\theta)d\theta \quad \dots\dots\dots(2.9)$$

dengan $J(\theta)$ adalah matrik jacobian, yang merepresentasikan hubungan yang sangat kecil antara posisi *joint* dengan posisi dan orientasi *end-effector*, yang merepresentasikan

hubungan yang sangat kecil antara penempatan *joint* dengan posisi *end-effector*.
 Persamaan 2.9 seringkali disebut sebagai persamaan kinematika *instantaneous*.

Dalam memberi perintah manipulator robot untuk bergerak dibutuhkan *invers*, variabel *joint* dicari untuk koordinat *end-effector*, ini dinamakan *invers kinematik* atau kinematika tidak langsung. Memakai ‘Law of Cosines’ dan ‘Elbow Down’,¹ kita dapatkan sudut joint-2 sebagai :

$$\cos \theta_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} = D \quad \dots\dots\dots(2.10)$$

$$\sin \theta_2 = \pm \sqrt{1 - D^2} \quad \dots\dots\dots(2.11)$$

sehingga,

$$\theta_2 = \tan^{-1} \frac{\pm \sqrt{1 - D^2}}{D} \quad \dots\dots\dots(2.12)$$

dan

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1} \left(\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \right) \quad \dots\dots\dots(2.13)$$

Catatan bahwa sudut joint-1, bergantung pada perubahan sudut joint-2. Ini membuat pemahaman secara fisik bahwa untuk adanya perubahan pada sudut joint-1 perlu memilih nilai dari sudut joint-2. Persamaan 2.10, 2.11, 2.12 dan 2.13 disebut persamaan kinematika tidak langsung (*invers kinematics*).

Demikian juga,

$$d\theta = J^{-1}(\theta)dX \quad \dots\dots\dots(2.14)$$

¹Spong, M. W., Vidyasagar, M., "Robot Dynamics and Control", John Wiley & Sons, Canada, 1989, pp.23-

Dua persamaan, persamaan 2.9 dan 2.14 sering disebut persamaan kinematika kecepatan (*velocity kinematics*).

Jadi dibayangkan bahwa ada *virtual work* diantara *joint* dan *end-effector*. Perpindahan aktual tidak sama dengan perpindahan *virtual*. Perpindahan *virtual* hanya memenuhi batasan geometri dan tidak harus memenuhi hukum gerakan yang lain. Prinsip kerja *virtual* adalah rangkaian lengan dalam keadaan setimbang jika dan hanya jika kerja-kerja *virtual* menghilang untuk perpindahan *virtual* sebarang.

Sementara *joint* bergeser sebesar $d\theta_i$, torsi *joint*, τ_i dan gaya serta moment pada ujung manipulator (*end-effector*), F_i bekerja pada manipulator. Hubungan antara torsi yang diberikan pada tiap *joint* dengan gaya dan moment pada *end effector* diberikan sebagai :

$$\tau(\theta) = J^T(\theta)F \quad \dots\dots\dots(2.15)$$

Persamaan 2.15 sering disebut sebagai hubungan gaya statis manipulator robot.

II.2 DINAMIKA MANIPULATOR ROBOT

Dinamika adalah studi mengenai analisa gerakan robot terhadap sistem koordinat referensi yang ada dengan memandang gaya ataupun torsi yang menyebabkan gerakan. Pada kinematika tidak mendiskusikan penyebab dari gerakan, karena hal posisi, kecepatan dan percepatan adalah suatu aspek dari mekanika. Hukum-hukum gerakan yang dibicarakan adalah hukum-hukum yang dirumuskan Sir Isaac Newton (1642 – 1727) sekalipun sudah diperkenalkan sebelumnya oleh kerja Galileo Galilei (1564 – 1642).

Gerakan terjadi akibat adanya gaya yang dikerjakan pada obyek. Gaya timbul akibat energi yang ada di alam. Mekanisme gerakan adalah suatu bidang yang dalam banyak buku telah ditulis. Sungguh, mempelajari bidang ini menghabiskan banyak tahun. Tulisan ini secara pasti tidak dapat merangkumkan materi-materi yang berharga itu secara lengkap.

Namun, formulasi-formulasi permasalahan dinamika yang dapat dipercaya terasa sama, teristimewa dalam hal untuk aplikasi manipulator robot.² Ada dua permasalahan yang dihubungkan dengan dinamika manipulator robot yang akan dicoba diselesaikan. Dalam permasalahan yang pertama diberikan suatu lintasan, dalam θ , $\dot{\theta}$ dan $\ddot{\theta}$, dan diminta untuk menemukan vektor torsi *joint* τ . Permasalahan yang kedua adalah untuk menghitung bagaimana mekanisme akan berpindahannya manipulator robot pada aplikasi dari suatu himpunan torsi-torsi *joint*. Yakni, diberikan suatu vektor torsi, τ , kemudian menghitung hasil gerakan dari manipulator robot dalam θ , $\dot{\theta}$ dan $\ddot{\theta}$.

Bila dalam koordinat *end-effector* lintasan diberikan dalam X , \dot{X} dan \ddot{X} dan diminta untuk menemukan vektor gaya dan moment, F pada ujung manipulator, adalah permasalahan pertama, atau sering disebut *invers dynamics problem*. Permasalahan yang kedua adalah menemukan lintasan X , \dot{X} dan \ddot{X} , dengan memberikan gaya dan moment, F pada ujung manipulator robot, ini sering disebut *forward dynamics problem*.

Pernyataan dinamika langsung (*forward dynamic*), bagaimana mengetahui percepatan manipulator robot yang mempunyai massa tertentu bila dikerjakan suatu gaya.

²Craig J. J., "Introduction to ROBOTICS Mechanics & Control", Addison-Wesley Publishing Company, Canada, 1988, pp.159

Pernyataan dinamika tidak langsung (*invers dynamic*), bagaimana mengetahui gaya yang bekerja pada manipulator robot yang bergerak dengan percepatan yang sudah diberikan.

Menguraikan pernyataan dinamika ini dilakukan berikut ini. Untuk membentuk sebuah lintasan, diperlukan sebuah perencanaan lintasan dengan memperhitungkan posisi-posisi yang dilewati - termasuk didalamnya kecepatan dan percepatan - dalam waktu tertentu. Perencanaan lintasan akan diuraikan secara lengkap pada bagian akhir bab ini.

II.2.1 Pengendalian dalam Joint Space

Beberapa dari struktur umum persamaan-persamaan untuk mengekspresikan persamaan dinamika dari suatu manipulator dalam suatu persamaan yang secara detail adalah persamaan ruang keadaan dan persamaan ruang konfigurasi.³

II.2.1.1 Persamaan ruang-keadaan (*state space*)

Ketika persamaan Newton-Euler⁴ dievaluasi secara simbolik untuk beberapa manipulator robot, persamaan dinamika dapat ditulis dalam bentuk :

$$M(\theta) \cdot \ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) = \tau \quad \dots\dots\dots(2.16)$$

dimana:

$M(\theta)$ adalah matrik massa dari manipulator robot dengan dimensi $n \times n$,

$V(\theta, \dot{\theta})$ adalah suatu vektor istilah Centrifugal dan Coriolis dengan dimensi $n \times 1$,

$G(\theta)$ adalah suatu vektor istilah gravitasi dengan dimensi $n \times 1$,

³Ibid pp.176-183

⁴Ibid pp.167-173

θ , $\dot{\theta}$ dan $\ddot{\theta}$ adalah vektor pergeseran posisi sudut joint, vektor kecepatan sudut joint dan vektor percepatan sudut joint, masing-masing dengan dimensi $n \times 1$, dan τ adalah matrik torsi input yang diberikan ke tiap joint dengan dimensi $n \times 1$.

Menggunakan istilah persamaan ruang-keadaan karena istilah $V(\theta, \dot{\theta})$, yang muncul dalam persamaan diatas bergantung kepada variabel posisi dan kecepatan (Khatib, 1983).

Setiap element dari $M(\theta)$ dan $G(\theta)$ adalah suatu fungsi yang rumit yang mana tergantung pada besarnya θ , posisi dari semua joint dari manipulator. Setiap elemen dari $V(\theta, \dot{\theta})$ adalah suatu fungsi yang rumit dari θ , dan $\dot{\theta}$.

Untuk jumlah link adalah 2 ($n=2$) maka, dalam *joint space* akan didapat bagian-bagian pembentuk dinamika manipulator robot planar, sebagai berikut :

$$M(\theta) = \begin{bmatrix} l_2^2 m_2 + 2l_1 l_2 m_2 c_2 + l_1^2 (m_1 + m_2) & l_2^2 m_2 + l_1 l_2 m_2 c_2 \\ l_2^2 m_2 + l_1 l_2 m_2 c_2 & l_2^2 m_2 \end{bmatrix}$$

$$V(\theta, \dot{\theta}) = \begin{bmatrix} -m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 l_2 s_2 \dot{\theta}_1^2 \end{bmatrix}$$

$$G(\theta) = \begin{bmatrix} m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ m_2 l_2 g c_{12} \end{bmatrix}$$

$$s_1 = \sin \theta_1; s_{12} = \sin(\theta_1 + \theta_2)$$

$$c_1 = \cos \theta_1; c_{12} = \cos(\theta_1 + \theta_2)$$

.....(2.17)

Yang mempengaruhi matrik massa dan vektor gravitasi adalah θ saja bukan $\dot{\theta}$.

Tetapi yang mempengaruhi istilah kecepatan (Coriolis dan Centrifugal) adalah θ dan $\dot{\theta}$.

II.2.1.2 Persamaan ruang-konfigurasi (*configuration space*)

Dengan menuliskan istilah kecepatan, $V(\theta, \dot{\theta})$ dalam suatu bentuk yang berbeda, kita dapat menuliskan persamaan dinamika sebagai :

$$M(\theta) \ddot{\theta} + B(\theta) \begin{bmatrix} \dot{\theta} \dot{\theta} \end{bmatrix} + C(\theta) \begin{bmatrix} \dot{\theta}^2 \end{bmatrix} + G(\theta) = \tau \quad \dots\dots\dots(2.18)$$

dimana:

$B(\theta)$ adalah suatu matriks koefisien-koefisien Coriolis dimensi $n \times n(n-1)/2$,

$\begin{bmatrix} \dot{\theta} \dot{\theta} \end{bmatrix}$ adalah suatu vektor perkalian kecepatan joint, diberikan oleh :

$$\begin{bmatrix} \dot{\theta} \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \dot{\theta}_2 & \dot{\theta}_1 \dot{\theta}_3 & \dots & \dot{\theta}_{n-1} \dot{\theta}_n \end{bmatrix}^T \quad \dots\dots\dots(2.19)$$

$C(\theta)$ adalah suatu koefisien-koefisien matrik Centrifugal dimensi $n \times n$, dan

$\begin{bmatrix} \dot{\theta}^2 \end{bmatrix}$ adalah suatu vektor dimensi $n \times 1$, diberikan oleh :

$$\begin{bmatrix} \dot{\theta}^2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1^2 & \dot{\theta}_2^2 & \dots & \dot{\theta}_n^2 \end{bmatrix}^T \quad \dots\dots\dots(2.20)$$

Disebut persamaan ruang-konfigurasi karena matrik-matriknya adalah fungsi dari posisi manipulator robot saja.

Persamaan dinamika dalam bentuk ini, kompleksitas komputasi terlihat dalam bentuk parameter-parameter yang bervariasi dari posisi manipulator robot saja, θ . Ini penting dalam aplikasi komputer sebagai kendali yang menghendaki *update* persamaan dinamika gerakannya. Persamaan diatas memberikan suatu bentuk yang dapat meng-

update kecepatan yang berhubungan dengan seberapa cepat perubahan konfigurasi manipulator robot.

Blok diagram pengendalian dalam *joint space* ditunjukkan dalam gambar 2.2.

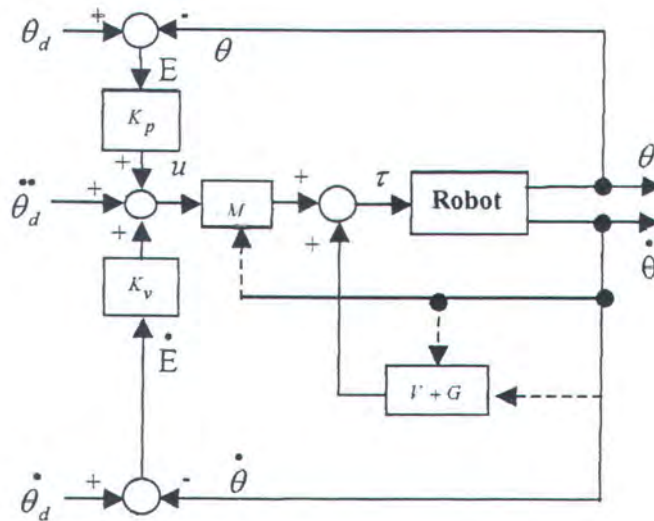
Hukum pengendalian yang berlaku adalah :

$$u = \ddot{\theta}_d + K_v \dot{E} + K_p E \tag{2.20}$$

dimana fungsi obyektif pengendalian adalah dipenuhinya $\hat{M} = M$, dan $\hat{V} = V$ serta $\hat{G} = G$, dimana \hat{M} , \hat{V} dan \hat{G} adalah pendekatan dari masing-masing parameter M , V dan G .

Sehingga dengan memasukkan persamaan (2.16) didapatkan,

$$M(\theta)[\ddot{\theta}_d + K_v(\dot{\theta}_d - \dot{\theta}) + K_p(\theta_d - \theta)] = 0 \tag{2.21}$$



Gambar 2.2 Blok diagram pengendalian manipulator robot di Joint Space

Berikut vektor error didefenisikan :

$$E = \theta_d - \theta ; \dot{E} = \dot{\theta}_d - \dot{\theta} ; \text{ dan } \ddot{E} = \ddot{\theta}_d - \ddot{\theta} \tag{2.22}$$

Ini dihubungkan oleh persamaan berikut ini, karena $M(\theta)$ adalah definit positif :

$$\ddot{E} + K_v \dot{E} + K_p E = 0 \dots\dots\dots(2.23)$$

Nilai $K_v > 0$ dan $K_p > 0$ adalah diagonal dari matrik konstanta dimensi $n \times n$.

II.2.2 Pengendalian dalam Task Space

Formulasi persamaan dinamika yang berhubungan dengan percepatan dari *end-effector* dinyatakan dalam *task space*, untuk mengetahui gaya-gaya Cartesian dan momen-momen yang bekerja pada *end effector* manipulator robot.

II.2.2.1 Persamaan ruang-keadaan (*state space*)

Persamaan ruang-keadaan manipulator robot nonredundant n -derajat kebebasan adalah :

$$M_x(\theta) \ddot{X} + V_x(\theta, \dot{\theta}) + G_x(\theta) = F_x \dots\dots\dots(2.24)$$

dengan

$$\begin{aligned} M_x(\theta) &= J^{-T}(\theta)M(\theta)J^{-1}(\theta) \\ V_x(\theta, \dot{\theta}) &= J^{-T}(\theta)[V(\theta, \dot{\theta}) - M(\theta)J^{-1}(\theta)\dot{J}(\theta)]J^{-1}(\theta) \\ G_x(\theta) &= J^{-T}(\theta)G(\theta) \\ F_x &= J^{-T}(\theta)\tau \end{aligned} \dots\dots\dots(2.25)$$

Subscript 'x' menunjukkan posisi dan orientasi koordinat *end-efector* dalam $x \in R^n$.

II.2.2.2 Persamaan ruang-konfigurasi (*configuration space*)

Dengan menggabungkan persamaan (2.24) dengan (2.25) dapat ditulis torsi-torsi penggerak joint dinyatakan dalam *task space* :

$$\tau = J^T(\theta)(M_x(\theta)\ddot{X} + V_x(\theta, \dot{\theta}) + G_x(\theta)) \dots\dots\dots(2.25)$$

Persamaan (2.25) akan lebih berguna bila dituliskan dalam bentuk :

$$\tau = J^T(\theta)M_x(\theta)\ddot{X} + B_x(\theta)\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} + C_x(\theta)\begin{bmatrix} \dot{\theta}^2 \end{bmatrix} + G(\theta) \dots(2.26)$$

dimana :

$B_x(\theta)$ adalah suatu matrik koefisien-koefisien Coriolis dimensi $n \times n(n-1)/2$,

$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix}$ adalah suatu vektor perkalian kecepatan joint, diberikan oleh :

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \ddot{\theta}_1 & \ddot{\theta}_2 & \ddot{\theta}_1 & \ddot{\theta}_3 & \dots & \ddot{\theta}_{n-1} & \ddot{\theta}_n \end{bmatrix}^T \dots\dots\dots(2.27)$$

$C_x(\theta)$ adalah suatu koefisien-koefisien matrik Centrifugal dimensi $n \times n$, dan

$\begin{bmatrix} \dot{\theta}^2 \end{bmatrix}$ adalah suatu vektor dimensi $n \times 1$, diberikan oleh :

$$\begin{bmatrix} \dot{\theta}^2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1^2 & \dot{\theta}_2^2 & \dots & \dot{\theta}_n^2 \end{bmatrix}^T \dots\dots\dots(2.28)$$

Catatan bahwa dalam persamaan (2.26), $G(\theta)$ adalah sama dengan persamaan dinamika dalam *joint space*, tetapi pada umumnya $B_x(\theta) \neq B(\theta)$ dan juga $C_x(\theta) \neq C(\theta)$.

Blok diagram pengendalian manipulator robot dalam *task space* ditunjukkan dalam gambar 2.3. Gambar 2.3 disebut juga perencanaan lintasan dalam *task space*.

Didefinisikan sinyal pengendali :

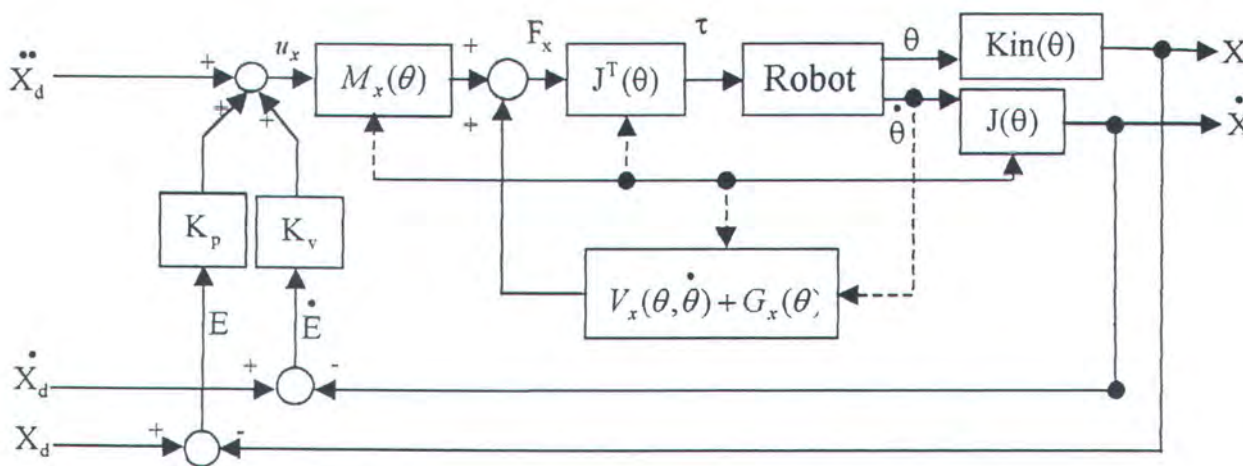
$$u_x = \ddot{X}_d + K_v(\dot{X}_d - \dot{X}) + K_p(X_d - X) \dots\dots\dots(2.29)$$

dengan fungsi obyektif $\hat{M}_x = M_x$ dan $\hat{V}_x = V_x$ serta $\hat{G}_x = G_x$. Sehingga dengan memasukkan persamaan (2.24) didapatkan,

$$M_x(\theta)[\ddot{X}_d + K_v(\dot{X}_d - \dot{X}) + K_p(X_d - X)] = 0 \dots\dots\dots(2.30)$$

Berikut vektor error didefinisikan :

$$E = X_d - X ; \dot{E} = \dot{X}_d - \dot{X} ; \text{ dan } \ddot{E} = \ddot{X}_d - \ddot{X} \dots\dots\dots(2.31)$$



Gambar 2.3 Blok diagram pengendalian manipulator robot di Task Space

Ini dihubungkan oleh persamaan berikut ini, karena $M_x(\theta)$ adalah definit positif :

$$\ddot{E} + K_v \dot{E} + K_p E = 0 \dots\dots\dots(2.32)$$

Nilai $K_v > 0$ dan $K_p > 0$ adalah diagonal dari matrik konstanta dimensi $n \times n$. Yang mempengaruhi inerti matrik dalam *task space* adalah θ .

Tetapi yang mempengaruhi istilah kecepatan Coriolis dan Centrifugal adalah θ dan $\dot{\theta}$.

II.3 LINTASAN *END-EFFECTOR* MANIPULATOR ROBOT

Gerakan robot yang paling sederhana adalah gerakan titik ke titik (*point to point*). Gerakan ini tanpa memandang lintasan yang akan dilalui *end-effector*. Gerakan ini memadai bila ruang kerja (*work space*) manipulator robot bebas dari rintangan. Bila *end-effector* melewati rintangan seringkali harus ditambahkan sebuah titik antara (*via point*) sehingga *end-effector* tidak menabrak rintangan yang ada. Gerakan manipulator robot yang halus didapatkan dengan jenis gerakan yang disebut *continuous path*. Secara umum gerakan manipulator robot dalam tulisan berikut ini dipakai kata lintasan untuk menyatakan titik-titik yang akan dilewati manipulator robot dimaksudkan, baik yang dilewati *joint-joint* maupun yang dilewati *end-effector* dari manipulator robot.

Perencanaan lintasan dari manipulator robot secara sistematis dibagi dua bagian yaitu perencanaan dalam *joint space*, dan perencanaan dalam *task space*. Pada perencanaan lintasan dalam *joint space*, kendala-kendala ditentukan dalam kordinat *joint*. Keuntungan penggunaan pendekatan lintasan ini adalah bahwa perencanaan dapat langsung dilakukan pada tiap *joint*, yang dapat secara langsung digunakan sebagai acuan untuk tiap link sehingga dapat menghemat waktu komputasi yang diperlukan. Kekurangan pendekatan ini adalah bahwa lintasan yang dilalui *end-effector* tidak dapat diketahui dengan pasti.

Perencanaan lintasan dalam *task space* (gambar 2.3) digunakan bila gerakan *end-effector* diinginkan untuk melewati lintasan tertentu dalam *task space*. Untuk beberapa tujuan seringkali manipulator robot dirancang untuk dapat menghindari rintangan yang ada, dengan cara menambahkan titik-titik diantara titik awal dan titik akhir lintasan yang dikehendaki. Dalam mencapai titik-titik itu harus dijaga kecepatan dan juga percepatan

joint-joint maupun *end-effector* sehingga diperoleh gerakan yang halus. Titik-titik yang dilewati oleh *joint* dan *end-effector* secara umum dapat didekati dengan polinomial orde ke- n .

BAB III

KARAKTERISTIK FUNGSI BASIS RADIAL

III.1 PENDAHULUAN

Jaringan Syaraf Tiruan dibentuk berdasarkan kerja jaringan syaraf otak manusia. Otak manusia tersusun dari banyak sel syaraf yang disebut neuron dan saling terhubung, yang diperkirakan mengandung sekitar 100 milyar sel neuron. Satu neuron dapat membuat 10 ribu sambungan sinapsis, sehingga otak manusia dapat memberikan sekitar 10¹⁵ sambungan sinapsis. Otak manusia juga mempunyai karakteristik mengolah informasi yang masuk secara paralel, mempunyai kemampuan belajar, kemampuan mengelola dirinya dan mengingat informasi yang masuk terdahulu. Kenyataannya karakteristik otak tersebut tidak seluruhnya dapat direpresentasikan dalam jaringan syaraf tiruan. Jaringan yang dibuat kemudian membutuhkan sejumlah asumsi anatomi dan fungsi otak yang mungkin sekali tidak sama dengan fungsi otak sebenarnya.

III.2 JARINGAN SYARAF MANUSIA

Sistem syaraf manusia tidak mengacu pada otak manusia, tetapi juga meliputi sistem sensoris tubuh, yang menerima rangsangan atau informasi dari luar tubuh, dan sistem motoris yang berhubungan dengan pergerakan atau reaksi tubuh terhadap rangsangan tadi.

Sistem sensoris tubuh akan menjalankan informasi sensoris dari reseptor yang ada hampir di seluruh tubuh ke otak, yang kemudian diolah untuk menentukan reaksi motoris tubuh yang tepat. Reaksi tubuh terjadi berdasarkan pada pengalaman sensoris dari

reseptor-reseptor tersebut. Sedangkan reaksi sistem motoris tubuh dilakukan dengan mengatur kontraksi otot rangka, otot organ internal tubuh serta fungsi kelenjar tubuh. Misalnya ketika kulit lengan menyentuh bahan panas, rangsangan dari luar tersebut diterima reseptor syaraf, dialirkan oleh sistem sensoris tubuh ke susunan syaraf pusat untuk di olah, hasil olahan tersebut dikirim ke sistem motoris tubuh untuk menentukan reaksi tubuh yang perlu diambil. Dua komponen penting yang berperanan dalam proses kerja sistem syaraf tersebut adalah sel syaraf (neuron) dan sambungan antar syaraf (sinapsis).

III.3 DEFENISI JARINGAN SYARAF TIRUAN

Terminologi Jaringan Syaraf Tiruan (*Artificial Neuron Network*), juga dikenal dengan istilah *Connectionist Model*, *Parallel Distributed Processing Models*, *Neuromorphic Systems*, atau hanya ditulis *Neural Networks*. Dalam pembahasan selanjutnya akan ditulis jaringan syaraf tiruan (JST) atau jaringan syaraf untuk membedakannya dengan jaringan syaraf biologis.

JST menurut Kohonen, didefinisikan sebagai suatu jaringan terhubung paralel yang biasanya terdiri dari elemen-elemen adaptif dan terorganisasi hirarkisnya, yang diharapkan dapat berinteraksi dengan benda atau keadaan nyata sama seperti sistem syaraf otak biologis melakukannya. JST juga dapat dipandang sebagai suatu sistem yang terdiri dari elemen-elemen pengolah terdistribusi secara paralel dengan kemampuan untuk memperbaiki kinerjanya melalui proses belajar.

Dari definisi diatas secara umum suatu JST terdiri dari elemen-elemen pengolah (*processing elements* atau *neurons*), disusun membentuk suatu struktur jaringan tertentu, dan berusaha meniru sistem syaraf biologis.

Suatu JST biasanya tersusun dari banyak lapisan neuron. Satu lapisan mengandung neuron-neuron yang tersusun paralel. Neuron-neuron ini dapat saling tersambung atau tidak, bergantung pada struktur jaringan yang dipilih. Jika JST tersebut memiliki banyak lapisan maka neuron-neuron dalam satu lapisan tersambung dengan neuron-neuron pada lapisan yang lain. Lapisan yang berada antara lapisan input dan lapisan output disebut lapisan *hidden*.

Sambungan-sambungan tersebut diberi harga yang dapat diatur dan menyatakan kekuatan sambungannya, disebut pembobot jaringan (*weights*). Proses untuk mendapatkan harga pembobot yang diinginkan ini disebut belajar (*learning*) dan dikerjakan menurut algoritma belajar tertentu.

III.4 JARINGAN SYARAF TIRUAN FUNGSI BASIS RADIAL

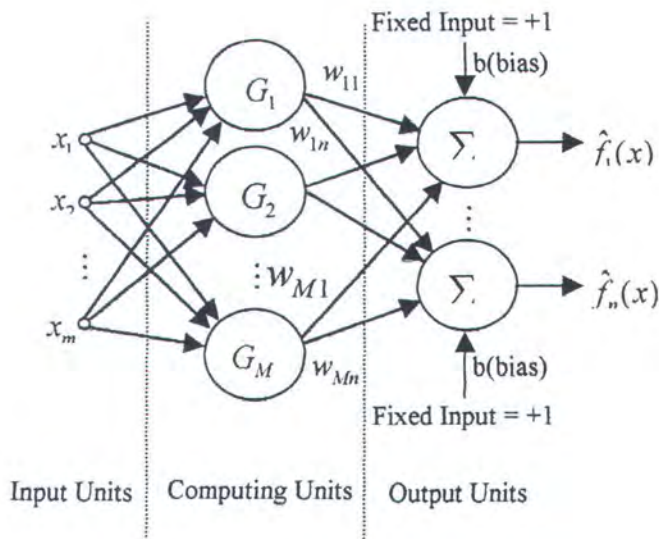
Desain dari suatu metode supervised mungkin dilakukan dengan cara bervariasi. Algoritma belajar BP untuk desain suatu multilapis perceptron dengan metode supervised adalah suatu aplikasi dari metode optimasi yang dikenal dalam statistika sebagai pendekatan stokastik (*stochastic approximation*). JST Fungsi Basis Radial (*Radial Basis Function Networks*) atau disingkat menjadi RBF adalah pendekatan yang berbeda dengan BP. Desain dari RBF adalah sebagai suatu *curve-fitting (approximation) problem* dalam suatu ruang dimensi-tinggi (*high-dimension space*). Mengacu kepada pendekatan ini, algoritma belajar ekuivalen dengan menemukan suatu permukaan dalam

suatu ruang dimensi banyak yang menyediakan suatu yang paling cocok dengan data latihan. Kriteria yang 'paling cocok' diukur dalam statistik. Bersamaan dengan itu, generalisasi ekuivalen dengan menggunakan permukaan dimensi banyak ini untuk interpolasi data. Dengan demikian, tentu saja motivasi disamping metode RBF adalah bahwa generalisasi memposisikan suatu tarikan meneliti interpolasi tradisional dalam suatu ruang dimensi banyak. Dalam konteks JST, unit-unit *hidden* menyediakan suatu himpunan 'fungsi-fungsi' yang merupakan suatu 'basis' sebarang untuk pola-pola vektor input ketika unit-unit tersebut diperluas ke unit *hidden-space*, fungsi-fungsi inilah yang dinamakan *radial basis functions*. RBF diperkenalkan pertama kali dalam menyelesaikan permasalahan dari interpolasi nyata variasi banyak (*real multivariate interpolation*). Hal mengerjakan dalam obyek ini disurvei oleh Powell (1985). Sekarang RBF adalah satu dari *field* utama dari penelitian-penelitian dalam analisa numerik, merupakan satu dari dua paradigma belajar yang paling populer dalam literatur JST, selain BP.

Broomhead and Lowe (1988) adalah orang yang pertama memanfaatkan RBF dalam desain JST. Kontribusi yang banyak lainnya terhadap teori, desain, dan penerapan dari RBF termasuk dalam paper Moody and Darken (1989), Renals (1989), dan Poggio and Girosi (1990a). Paper dari Poggio and Girosi menegaskan menggunakan penerapan teori regulasi untuk kelas dari JST ini sebagai suatu metode untuk dimanfaatkan secara umum terhadap data baru.

Konstruksi dari RBF pada dasarnya ada tiga lapis yang berbeda. Lapis input yang berhubungan dengan sumber data (*sensory unit*). Lapis kedua adalah *hidden* dengan dimensi cukup tinggi, yang memiliki perbedaan maksud daripada yang ada pada multi lapis perceptron. Lapis output menyediakan respon jaringan terhadap pola-pola aktivasi

yang diterapkan dari lapis input. Transformasi dari *input-space* ke unit *hidden-space* adalah nonlinear, sedangkan transformasi dari unit *hidden-space* ke *output-space* adalah linear. Suatu dasar matematika untuk menyatakan sebagai rasional mungkin bisa dipandang kembali ke paper dari Cover (1965). Paper ini mencatat bahwa suatu permasalahan linearitas klasifikasi pola dalam ruang dimensi tinggi adalah lebih seperti pemisahan secara linear dalam suatu ruang dimensi rendah, sehingga dengan alasan ini memuat dimensi dari unit *hidden-space* dalam suatu RBF cukup tinggi. Melalui desain yang hati-hati, adalah mungkin untuk mereduksi dimensi dari unit *hidden-space*, khususnya bila pusat-pusat dari unit-unit *hidden* dibuat adaptif. Arsitektur RBF ditunjukkan dalam gambar 3.1.



Gambar 3.1 Suatu jaringan RBF dengan m -input n -output⁵

⁵Behera L., Gopal M., and Chaudhury S., "On Adaptive Trajectory Tracking of Robot Manipulator Using Inversion of Its Neural Emulator," IEEE Trans. On Neural Networks, vol. 7, no. 6, November 1996

Beberapa tipe fungsi basis atau fungsi aktivasi RBF⁶ adalah :

1. Fungsi *Thin-Plate-Spline* :

$$\varphi(z,1) = z^2 \log(z) \dots\dots\dots(3.1)$$

2. Fungsi *Multiquadratic* :

$$\varphi(z, \sigma) = (z^2 + \sigma^2)^{1/2} \dots\dots\dots(3.2)$$

3. Fungsi *Inverse-Multiquadratic* :

$$\varphi(z, \sigma) = \frac{1}{(z^2 + \sigma^2)^{1/2}} \dots\dots\dots(3.3)$$

4. Fungsi Gauss :

$$\varphi(z, \sigma) = \exp(-z^2 / \sigma^2) \dots\dots\dots(3.4)$$

Paling umum fungsi basis dipilih suatu fungsi Gauss. Hasil-hasil secara teori menyatakan bahwa seleksi dari keempat fungsi nonlinear ini tidak menentukan untuk performansi. Meskipun lebar pada setiap unit neuron dapat bernilai berbeda, lebar yang sama untuk setiap unit neuron adalah cukup untuk pendekatan secara universal (Park and Sanberg, 1991). Ini berarti bahwa semua lebar-lebar dapat *fixed* pada sebuah nilai σ , untuk memperlengkapi suatu strategi *training* yang lebih sederhana.

Persamaan pemetaan / *mapping* dari lapis input-*hidden*-output adalah :

$$\hat{f}_n(x_j) = \sum_{j=1}^M wG(\|x_j - c_i\|^2) + b \dots\dots\dots(3.5)$$

dimana x_j adalah vektor input dengan dimensi $m \times 1$, sebanyak- j pola, w adalah bobot-bobot penggabung linear,

⁶Chen S., Billing S. A., "Neural Networks for Nonlinear Dynamic System Modeling and Identification," Int. J. Control, vol. 56, no. 2, pp. 329-330

$\|\bullet\|$ menyatakan *norm euclidean*, dan b adalah bias yang biasanya bernilai 1.

III.5 Strategi *Learning* RBF

Proses *learning* oleh RBF divisualisasikan berikut ini. Pembobot linear yang dihubungkan dengan unit output dari jaringan membentang secara lambat laun pada suatu “*time scale*” yang berbeda dibandingkan dengan fungsi aktivasi nonlinear dari unit-unit neuron lapis *hidden*. Fungsi aktivasi dari lapis *hidden* secara lambat laun akan menuruti beberapa strategi optimasi nonlinear, bobot lapis output disetel secara pribadi secara cepat melewati suatu strategi optimasi linear. *Point* yang penting dicatat disini adalah bahwa lapis yang berbeda dari suatu RBF mengerjakan tugas yang berbeda, dan itulah sebabnya maka cukup beralasan untuk memisahkan optimisasi lapis *hidden* dan lapis output dari jaringan dengan memakai teknik yang tidak sama, dan kadang-kadang mengoperasikannya dalam perbedaan penskalaan waktu (Lowe, 1991a).

Ada perbedaan strategi *learning* yang akan diikuti berikut ini dalam mendesain suatu RBF, bergantung pada bagaimana pusat-pusat dari RBF dari jaringan dispesifikasi. Secara esensi, kita mengidentifikasi tiga pendekatan, sebagai yang didiskusikan berikut ini.

III.5.1 Pusat-Pusat Diseleksi Secara Random.

Pendekatan yang pertama mengasumsikan fungsi-fungsi aktivasi dari unit-unit lapis *hidden* adalah *fixed*. Secara khusus, lokasi-lokasi dari pusat-pusat RBF dipilih secara random dari himpunan *training* data. Ini dipertimbangkan menjadi suatu pendekatan “*sensible*”, yang diperlengkapi bahwa *training* data didistribusikan dalam

suatu cara yang mewakili permasalahan ditangan (Lowe, 1989). Untuk itu, digunakan suatu fungsi Gaus *isotropik* yang standard deviasinya *fixed* mengikuti penyebaran dari pusat-pusat. Secara spesifik, suatu RBF (ternormalisasi) yang berpusat di c_i didefenisikan sebagai :

$$G(\|x - c_i\|^2) = \exp\left(-\frac{M}{d^2} \|x - c_i\|^2\right) \dots\dots\dots(3.6)$$

dimana M : jumlah dari pusat-pusat, dan

d : jarak maksimum antara pusat-pusat yang dipilih.

$i=1,2,\dots,M$

Dalam hal ini, standard deviasi (yakni lebar atau *width*) dari RBF Gaus *fixed* pada :

$$\sigma = \frac{d}{\sqrt{2M}} \dots\dots\dots(3.7)$$

Sebagai suatu pilihan untuk standard deviasi, σ adalah menjamin bahwa fungsi-fungsi Gaus tidak begitu tajam atau begitu datar; kedua ekstrim ini dihindari. Bentuk lain dari persamaan (3.4) adalah persamaan (3.6) dan persamaan (3.7).

Parameter yang dibutuhkan untuk *di-learning* dalam pendekatan ini hanya pembobot-pembobot linear pada lapis output dari jaringan. Suatu metode / prosedur yang secara langsung untuk melakukan ini adalah menggunakan metode *pseudoinverse* (Broomhead and Lowe, 1988). Secara khusus, kita mempunyai :

$$w = G^+ D \dots\dots\dots(3.8)$$

dimana D adalah vektor tanggapan yang dikehendaki dalam himpunan *training* data, G^+ adalah *pseudoinverse* dari matrik G , yang mana didefenisikan sebagai :

$$G = \{ g_{ji} \} \dots\dots\dots(3.9)$$

dengan :

$$g_{ji} = \exp\left(-\frac{M}{d^2} \|x_j - c_i\|^2\right) \dots\dots\dots(3.10)$$

dimana x_j adalah vektor input ke- j dari himpunan *training*, $j=1, 2, \dots, N$, dan $i=1, 2, \dots, M$.

Dasar dari semua algoritma untuk komputasi dari *pseudoinverse* dari suatu matrik adalah *singular-value decomposition* (Golub and Van Loan, 1989; Haykin, 1991).

III.5.2 Pusat-Pusat Diseleksi Secara Metode Self-Organized.

Dalam pendekatan yang kedua, fungsi-fungsi radial-basis adalah diijinkan untuk memindahkan lokasi-lokasi dari pusat-pusat dalam suatu *self-organized*, dimana bobot-bobot linear dari lapis output dihitung memakai aturan *supervised learning*. Dengan kata lain, jaringan mengalami proses *hybrid learning* (Moody and Darken, 1989; Lippmann, 1989b). Jaringan *hybrid* adalah jaringan yang mengkombinasikan paradigma *learning supervised* dan *unsupervised*. Dimana sebagian dari bobot ditentukan dengan *supervised learning* dan sebagian lain diperoleh dari *unsupervised learning*. Komponen *self-organized* dari proses *learning* mengalokasikan sumber-sumber jaringan dalam suatu cara yang sangat berarti dengan menempatkan pusat-pusat dari RBF hanya dalam daerah-daerah dari input *space* dimana data yang signifikan / penting muncul. Dengan kata lain jenis RBF di-*training* dengan aturan *unsupervised learning* pada lapis input, dan aturan *supervised learning* pada lapis output.

III.5.2.1 *Unsupervised Learning* Pada Lapis Input

Unsupervised learning adalah *learning* dimana tidak dibutuhkan output yang benar (output yang dituju tidak diberikan) untuk setiap pola input yang diberikan (yang ada). Tetapi dia menjelajahi struktur pada data, dan mengorganisasi pola itu kedalam suatu kategori dari hubungan tersebut. *Unsupervised learning* dikembangkan dari pemikiran bahwa algoritma *supervised* tidak masuk akal secara biologis, dalam artian susah untuk memahami mekanisme belajar dalam otak yang membandingkan output yang diinginkan dengan output aktual dan mengumpan balikkan faktor koreksi yang terjadi, karena dalam mekanisme otak tidak diketahui asal dari *desire* output.

Contoh *Unsupervised Learning* : *Self-Organizing Future Maps* (Kohonen).

Unsupervised learning ini akan menentukan pusat-pusat *field receptive* Gauss, c_i dan menentukan lebar-lebar, σ menurut persamaan (3.6). Makin kecil varians maka kesalahan pengukuran makin kecil, dan inilah yang diinginkan. Jika varians minimum berarti membuat kurva distribusi data menjadi semakin sempit. Teknik minimum varians adalah salah satu dari dua strategi teknik perancangan sistem kendali *prediktive*.

III.5.2.2 *Supervised Learning* Pada Lapis Output

Supervised learning adalah *learning* dimana output yang benar disediakan (output yang dituju diberikan) untuk setiap pola input yang diberikan (yang ada). Disini bobot ditentukan sedemikian rupa sehingga output jaringan mendekati output sebenarnya (output yang benar) yang telah diketahui tersebut. variasi dari proses belajar ini adalah *reinforcement learning* dimana hanya disediakan tinjauan benar-tidaknya output jaringan.

Contoh *Supervised learning* : *Back Propagation*

Algoritma untuk pendekatan kedua ini adalah :

1. Inisialisasi bobot

Bobot-bobot pada lapis *hidden* ditentukan dengan algoritma pengelompokan (*clustering algorithm*).

Bobot-bobot pada lapis output diinisialisasikan sebagai sebarang nilai yang kecil.

2. Penghitungan Aktivasi

1. Level aktivasi O_j hidden unit ke j dihitung dengan :

$$O_j = \exp[-(X - W_j).(X - W_j) / 2\sigma_j^2] \dots\dots\dots(3.11)$$

2. Level aktivasi O_j output unit ditentukan dengan :

$$O_j = \sum w_{ji} O_i \dots\dots\dots(3.12)$$

3. *Learning* bobot

-Pada lapis *hidden*, menggunakan *k-nearest-neighbor rule*.

-Pada lapis output :

1. Menyetel bobot-bobot dengan :

$$w_{ji}(t + 1) = w_{ji}(t) + \Delta w_{ji} \dots\dots\dots(3.13)$$

dimana $w_{ji}(t)$ adalah bobot dari unit ke- i ke unit ke- j pada waktu ke- t (atau iterasi ke- t), dan Δw_{ji} adalah bobot yang disetel.

2. Perubahan bobot dihitung dengan :

$$\Delta w_{ji} = \eta \delta_j O_i \dots\dots\dots(3.14)$$

dimana η adalah suatu *learning rate trial-independent*, dan δ_j adalah error pada unit ke- j :

$$\delta_j = T_j - O_j \dots\dots\dots(3.15)$$

dimana T_j adalah aktivasi output yang diinginkan (target), dan O_j adalah aktivasi output aktual pada output unit ke- j .

4. Mengulangi iterasi sampai konvergen.

III.5.3 Pusat-Pusat Diseleksi Secara Supervised

Pendekatan ketiga, pusat-pusat dari RBF dan semua parameter-parameter bebas dari jaringan mengalami suatu proses *supervised learning*, dengan kata lain, RBF mengambil bentuk yang lebih umum. Suatu proses yang alami dipakai adalah *error-corection learning*, yang mana pemakaiannya diimplementasikan memakai prosedur *gradient-descent* yang merepresentasikan suatu bentuk umum dari algoritma *least-mean-square* (LMS).

Langkah pertama dalam pengembangan suatu proses *learning* ini adalah mendefenisikan harga fungsi *cost* :

$$E = \frac{1}{2} \sum_{j=1}^N e_j^2 \quad \dots\dots\dots(3.16)$$

dimana N adalah jumlah contoh *training* yang digunakan dalam proses *learning*, dan e_j adalah sinyal error, yang didefenisikan dengan :

$$\begin{aligned} e_j &= d_j - y(x_j) \\ e_j &= d_j - \sum_{i=1}^M w_i G(\|x_j - t_i\|) \quad \dots\dots\dots(3.17) \end{aligned}$$

Yang dibutuhkan adalah menemukan parameter-parameter bebas, w_i , t_i dan \sum_i^{-1} (dihubungkan ke *norm* matriks-pembobot), yang meminimisasi E .

III.6 JARINGAN SYARAF TIRUAN BACK PROPAGASI

Algoritma belajar JST Back Propagasi (*Back Propagation Neural Networks*) selanjutnya akan disingkat BP, pertama diformulasikan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada jaringan syaraf, didesain untuk beroperasi pada jaringan feedforward multi lapis dengan metode supervised.

Algoritma ini banyak dipakai pada aplikasi pengendalian karena prosedur belajarnya didasarkan pada hubungan yang sederhana : jika output memberikan hasil yang salah maka pembobot dikoreksi supaya error dapat diperkecil dan respon jaringan selanjutnya diharapkan akan mendekati harga yang benar. Prosedur ini analog dengan proses pengaturan pada sistem pengaturan. BP juga berkemampuan untuk menangani pelatihan pembobot pada lapisan *hidden*. Tetapi kekurangan utama pada BP adalah konvergensi hasil yang tidak dijamin, tetapi dengan perhitungan matematisnya yang sederhana algoritma ini tetap banyak dipakai.

Fungsi aktivasi output yang biasa dipergunakan adalah jenis fungsi *sigmoid* atau fungsi *hiperbolik*, karena metode BP memerlukan jenis fungsi yang dapat diturunkan (*differentiable*) dan bersifat *monoton nondecreasing*. Penambahan harga aktivasi hanya dapat menambah output atau tetap, tanpa pernah berkurang.

Penguatan pada fungsi ini merupakan kemiringan kurva pada suatu tingkat eksitasi tertentu, dan nilainya berubah dari harga yang kecil pada eksitasi negatif besar menjadi harga yang besar pada eksitasi 0 dan kemudian kembali mengecil pada harga eksitasi yang semakin besar dan positif. Output lapisan ini disebar maju ke lapisan selanjutnya secara berurutan, untuk kemudian mengalami proses yang sama sampai pada

lapisan output. Lapisan output jaringan secara keseluruhan. Jadi arah sebaran informasi adalah lapisan input-*hidden*-output.

Tahapan propagasi balik dimulai dengan membandingkan respon jaringan keseluruhan dengan output yang diinginkan. Perbedaan yang terjadi atau errornya, kemudian dipergunakan untuk memperbaiki harga pembobot jaringan. Mekanisme koreksi berawal dari lapisan output dan menyebar balik ke lapisan *hidden* menuju ke lapisan input.

BAB IV

PERENCANAAN PENGENDALIAN

Diberikan dalam koordinat end-effector $X_d(t)$, lintasan posisi, $\dot{X}_d(t)$, kecepatan yang dikehendaki, dan $\ddot{X}_d(t)$, percepatannya. Lintasannya dinyatakan dalam persamaan orde-5, [28] sebagai :

$$X_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad \dots\dots\dots(4.1)$$

dengan menurunkan persamaan diatas terhadap waktu, didapat persamaan kecepatan dan percepatan sebagai berikut :

$$\dot{X}_d(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad \dots\dots\dots(4.2)$$

$$\ddot{X}_d(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad \dots\dots\dots(4.3)$$

Kendalanya diberikan oleh :

$$\begin{aligned} X_0 &= a_0, \\ X_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5, \\ \dot{X}_0 &= a_1, \\ \dot{X}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4, \\ \ddot{X}_0 &= 2a_2, \\ \ddot{X}_f &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{aligned} \quad \dots\dots\dots(4.4)$$

Penyelesaian untuk enam persamaan kendala (persamaan 4.4) tersebut, dengan enam koefisien yang tidak diketahui adalah :

$$\begin{aligned}
 a_0 &= X_0, \\
 a_1 &= \dot{X}_0, \\
 a_2 &= \frac{\ddot{X}_0}{2}, \\
 a_3 &= \frac{20X_f - 20X_0 - (8\dot{X}_f + 12\dot{X}_0)t_f - (3\ddot{X}_0 - \ddot{X}_f)t_f^2}{2t_f^3}, \\
 a_4 &= \frac{30X_0 - 30X_f + (14\dot{X}_f + 16\dot{X}_0)t_f + (3\ddot{X}_0 - 2\ddot{X}_f)t_f^2}{2t_f^4}, \\
 a_5 &= \frac{12X_f - 12X_0 - (6\dot{X}_f + 6\dot{X}_0)t_f - (\ddot{X}_0 - \ddot{X}_f)t_f^2}{2t_f^5}.
 \end{aligned}
 \tag{4.5}$$

Harga-harga koefisien ini selanjutnya digunakan sebagai referensi posisi, kecepatan dan percepatan. Dalam algoritma pengendalian berikut akan dipilih harga K_v dan K_p masing-masing adalah 10.0 dan 5.0.

IV.1 MEMAKAI ALGORITMA KONVENSIONAL

Pengendalian memakai blok diagram seperti telah ditunjukkan dalam gambar 2.3.

Didefinisikan sinyal pengendali :

$$u_x = \ddot{X}_d + K_v(\dot{X}_d - \dot{X}) + K_p(X_d - X)
 \tag{4.6}$$

dengan fungsi obyektif $\hat{V}=V$ dan $\hat{G}=G$. Sehingga dengan memasukkan persamaan (2.24) didapatkan,

$$M_x(\theta)[\ddot{X}_d + K_v(\dot{X}_d - \dot{X}) + K_p(X_d - X)] = 0 \quad \dots\dots\dots(4.7)$$

Berikut vektor error didefinisikan :

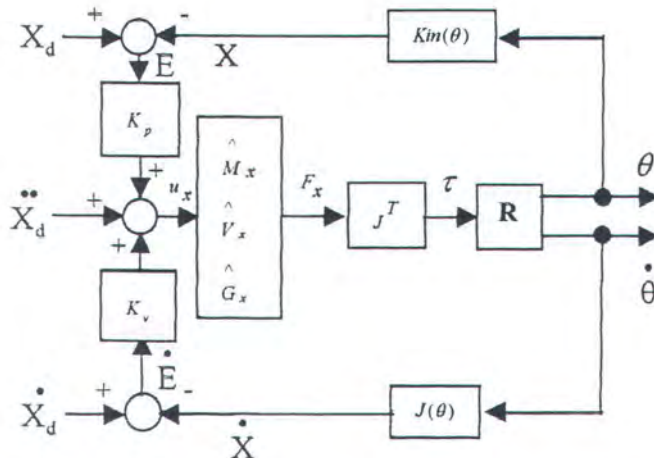
$$E = X_d - X ; \dot{E} = \dot{X}_d - \dot{X} ; \text{ dan } \ddot{E} = \ddot{X}_d - \ddot{X} \quad \dots\dots\dots(4.8)$$

Ini dihubungkan oleh persamaan berikut ini, karena $M_x(x)$ adalah definit positif :

$$\ddot{E} + K_v \dot{E} + K_p E = 0 \quad \dots\dots\dots(4.9)$$

IV.2 ALGORITMA PENGENDALIAN USULAN

Diperlihatkan dalam gambar 4.1 blok diagram pengendalian manipulator robot yang diusulkan. Dalam tugas akhir ini pemilihan pusat-pusat unit lapis *hidden* dilakukan secara random, yang adalah pendekatan pertama dari tiga pendekatan yang didiskusikan dalam bab III. Sehingga algoritma pengendalian yang diusulkan adalah sebagai berikut :



Gambar 4.1 Blok diagram pengendalian manipulator robot usulan, dengan RBF

1. Mengambil data input posisi yang dikehendaki, yakni vektor X_d dengan dimensi 2×1 , setiap waktu sample, menurut persamaan (4.1).
2. Pusat-pusat, c_i setiap unit neuron lapis *hidden* dipilih secara random, menurut '*sensible approach*'⁷. Masing-masing vektor dengan dimensi 2×1 . Jarak tiap elemen antar pusat disesuaikan dengan jumlah data, dengan demikian akan didapat harga standard deviasi, menurut persamaan (3.7).
3. Menghitung respon tiap unit neuron lapis *hidden* terhadap vektor input yang dimasukkan dari unit-unit neuron lapis input, dengan persamaan (3.6). Disini harga bias (b) bisa tidak sama dengan 1.
4. Menghitung harga pembobot, w beserta representasi dari error, yakni bias (b) yang memodelkan matrik massa manipulator robot, dengan persamaan (3.8). Disini $D = M$. Bila matriks hasil persamaan (3.6) pada langkah 3 tidak bujur sangkar atau tidak mempunyai *invers* (matrik G singular) maka algoritma Moore-Penrose *pseudoinverse* atau *singular value decomposition* dipakai untuk menyelesaikannya.
5. Menghitung harga-harga fungsi pendekatan atau fungsi *mapping* dengan persamaan (3.5). Persamaan (3.5) adalah fungsi hasil pendekatan atau fungsi hasil pemodelan oleh RBF terhadap D atau M . Harga bias (b) disini adalah harga model pendekatan error.
6. Matrik massa yang telah didapatkan berdasarkan langkah-langkah seperti diatas, kemudian dimasukkan atau dipakai sebagai matriks massa untuk mengetahui respon manipulator robot.

⁷Haykin S., "Neural Networks : A Comprehensive Foundation," Macmillan College Publishing Company Inc., 1994, pp. 264

Perlu diberikan catatan bahwa vektor-vektor pola input sebarang, seperti posisi, ataupun vektor-vektor pola input lainnya pasti selalu akan dihamburkan dalam bidang atau fungsi keanggotaan Gaus, dengan nilai antara 0 dan 1, tidak lebih. Ketajaman atau kelandaian ditentukan dengan harga varians atau kwadrat dari standard deviasi.

BAB V

ANALISA DENGAN SIMULASI

Untuk maksud ilustrasi, dibahas penggunaan suatu manipulator robot planar 2 link secara luas dalam simulasi ini. Simulasi yang akan dilakukan adalah menggerakkan *end-effector* manipulator robot dari satu posisi ke posisi tujuan tertentu melewati suatu lintasan halus yang dinyatakan dengan persamaan polinomial derajat lima. Simulasi dilakukan selama waktu yang telah ditentukan.

Parameter-parameter manipulator robot⁸ antara lain adalah :

- Arm length, L
- Link center of gravity, k
- Mass, m
- Inertia, I
- Gear ratio, N
- Coulomb friction coefficient, T
- Motor inertia, J_m
- Motor damping coefficient, D_m

Misalkan diberikan parameter yang benar dari manipulator robot yang digunakan dalam simulasi, ditunjukkan pada tabel.5.1.

Seperti diperlihatkan pada gambar 5.1, konfigurasi awal lengan manipulator robot tersebut adalah, $\theta_1 = 45^\circ$ dan $\theta_2 = -45^\circ$.

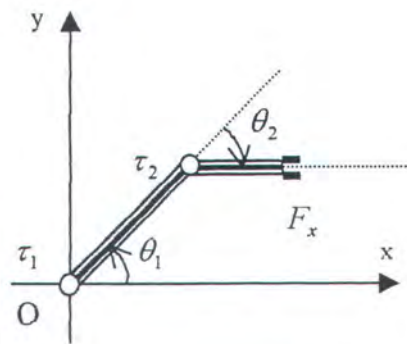
⁷Ishiguro A., Furuhashi T., Okuma S., Yoshiki U., "A Neural Network Compensator for Uncertainties of Robotics Manipulators," IEEE Trans. On Industrial Electronics, vol. 39, no. 6, Dec 1992, pp. 568

Tabel 5.1 Parameter Manipulator Robot

Link	Panjang	Massa	Inersia
(i)	(m)	(Kg)	(Kg.m.m)
1	0.25	9.5	4.3
2	0.16	5.0	1.6

End-effectornya berada pada koordinat :

$$X_d(0) = [0.3368(m) \quad 0.1768(m)]^T \quad \dots\dots\dots(5.1)$$



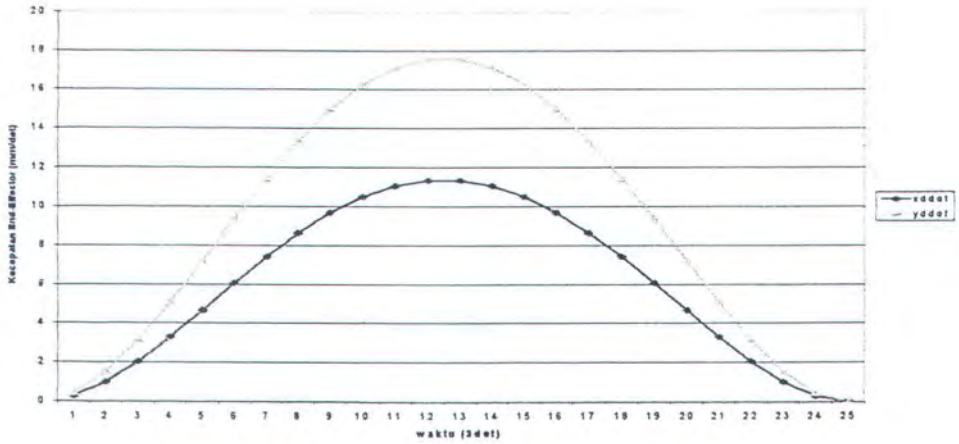
Gambar 5.1. Konfigurasi awal robot

Setelah 3 detik manipulator robot tersebut diharapkan berada pada konfigurasi akhir, yakni pada koordinat :

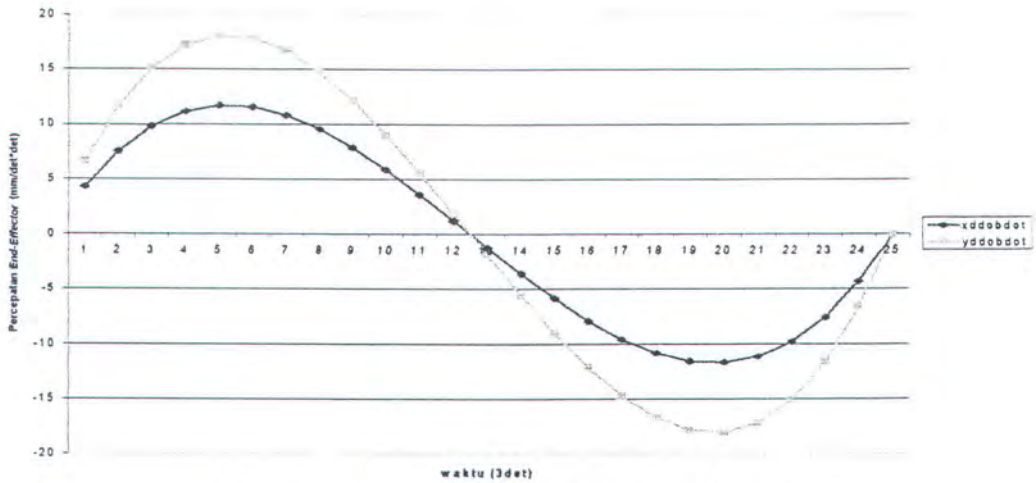
$$X_d(3) = [0.35507(m) \quad 0.205(m)]^T \quad \dots\dots\dots(5.2)$$

Untuk memenuhi lintasan yang dinyatakan persamaan 4.1, 4.2, dan 4.3 yaitu lintasan halus seperti diperlihatkan pada gambar 5.2 dan 5.3, matriks massa hasil *learning* harus sama dengan matriks massa ketika parameter seperti pada tabel 5.1 dicoba diterapkan dalam pengendalian manipulator robot, mengikuti persamaan (2.16), sehingga ditulis :

$$\begin{aligned}
 M(\theta) &= \hat{M}(\theta) \\
 V(\theta, \dot{\theta}) &= \hat{V}(\theta, \dot{\theta}) \\
 G(\theta) &= \hat{G}(\theta)
 \end{aligned}
 \tag{5.3}$$



Gambar 5.2 Kecepatan *end-effector* yang dikehendaki

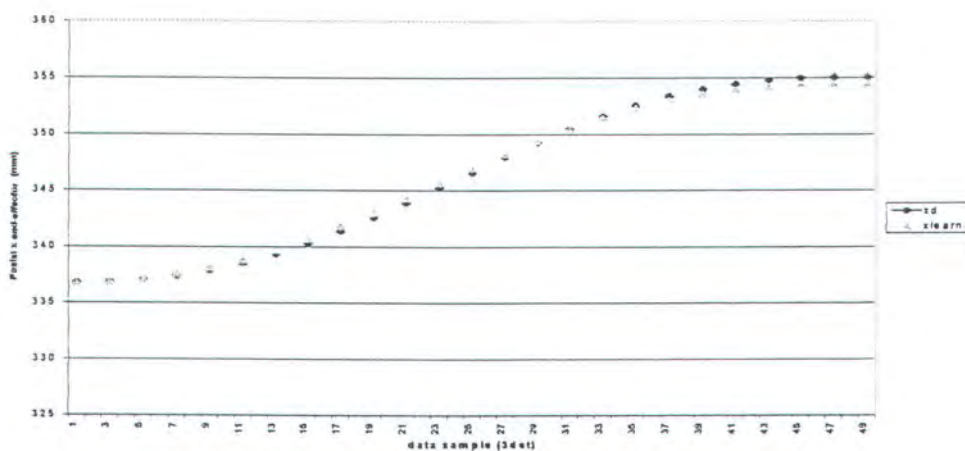


Gambar 5.3 Percepatan *end-effector* yang dikehendaki

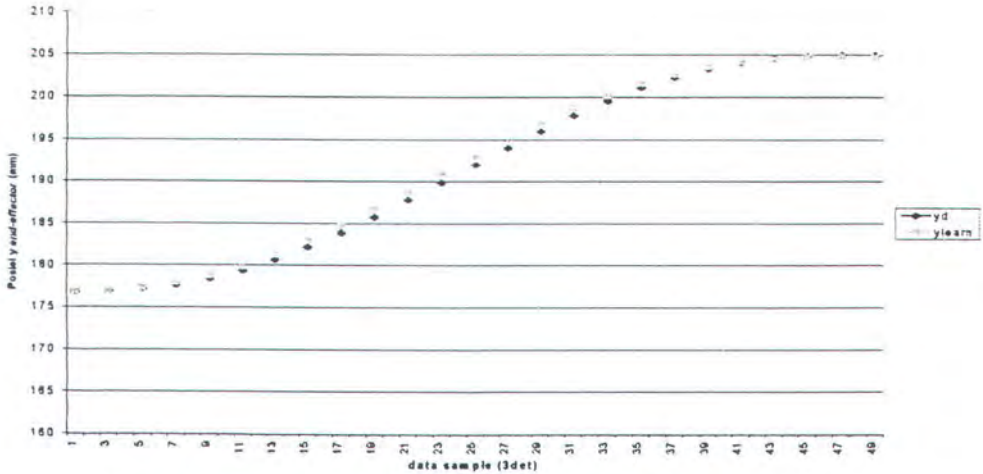
Pada titik kecepatan maksimum terdapat percepatan yang paling minimum yakni 0 (gambar 5.3). Karena dalam pergerakannya *end-effector* dikehendaki untuk berhenti tanpa osilasi pada koordinat tujuan.

Untuk mencapai hasil seperti gambar 5.2 dan 5.3 kemudian dilakukan simulasi dengan memakai RBF dengan ketentuan yang dipakai adalah sebagai berikut. Elemen lapis input ada dua buah. Karena ada dua link maka matrik massa mempunyai empat elemen, sebagai jumlah unit lapis output. Sedangkan jumlah unit lapis *hidden* dipakai dua kali, yaitu 7 buah elemen untuk simulasi pertama, dan 17 buah elemen untuk simulasi kedua. Sekali lagi disebutkan bahwa pusat dari setiap unit lapis *hidden* dipilih secara random, dengan jarak yang sama.

Dengan memakai persamaan (2.25) dengan kondisi persamaan (5.3), maka akan didapatkan hukum pengendalian dalam *operational space* seperti yang ditulis dengan persamaan (2.24). Respon pengendalian di *operational space* ditunjukkan gambar 5.4, 5.5. Gambar 5.4 dan gambar 5.5 adalah hasil simulasi posisi *end-effector* manipulator robot yang diuraikan dalam sumbu x dan sumbu y, keduanya fungsi waktu. Ini didapat



Gambar 5.4 Posisi x *end-effector* hasil simulasi $l=17$, $b=1$, $d=1$



Gambar 5.5 Posisi y end-effector hasil simulasi $l=17, b=1, d=1$

dengan memberikan matrik massa dengan pengerjaan algoritma *Moore-Penrose pseudoinvers* dalam RBF. Matrik massa yang di-*learning* dalam simulasi ini adalah :

$$M = \begin{bmatrix} 8.7804 & 4.3320 \\ 4.3320 & 4.3320 \end{bmatrix} \dots\dots\dots(5.4)$$

Setelah melakukan algoritma pengendalian usulan pada bab IV, maka didapat matrik massa hasil *learning* dengan parameter tepat adalah :

$$\hat{M} = \begin{bmatrix} 8.7804 & 4.3320 \\ 4.3320 & 4.3320 \end{bmatrix} \dots\dots\dots(5.5)$$

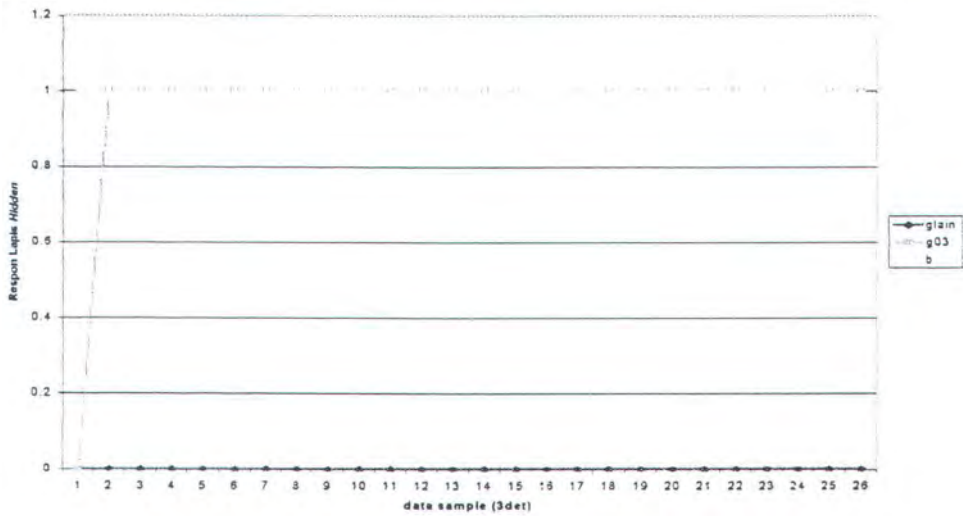
sedangkan untuk parameter yang telah diubah besarnya (M_{11}) -yang ditampilkan hasilnya dalam gambar 5.4 dan 5.5 untuk lintasan- adalah :

$$\hat{M} = \begin{bmatrix} 8.7484 & 4.3320 \\ 4.3320 & 4.3320 \end{bmatrix} \dots\dots\dots(5.6)$$

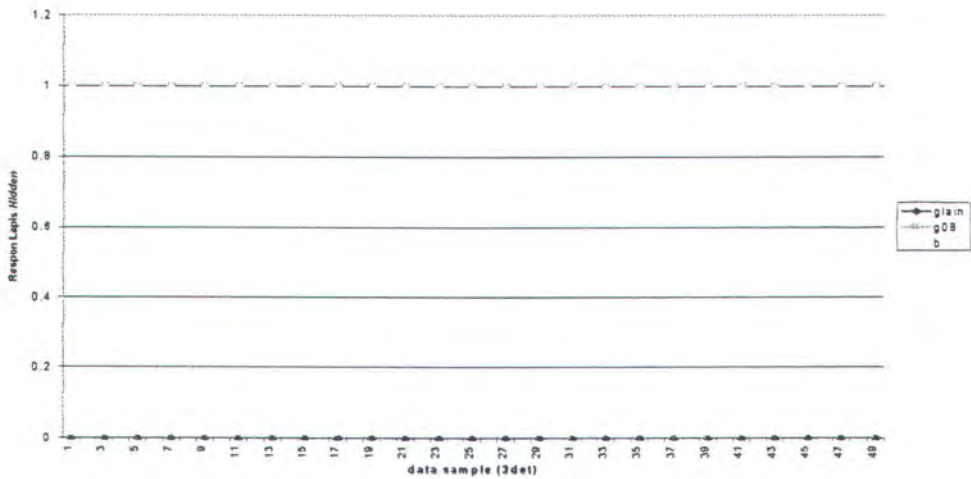
Persamaan (5.4) dan persamaan (5.5) menunjukkan bahwa proses *learning* dengan memakai RBF berfungsi dengan benar, logis. Persamaan (5.4) dan persamaan

(5.6) menunjukkan proses *learning* terhadap adanya data penyimpangan seperti diberikan dalam dalam lampiran tulisan ini, yaitu pada salah satu elemen matrik M yaitu M_{11} .

Respon unit-unit neuron lapis *hidden* diperlihatkan pada gambar 5.6 dan 5.7, dimana dengan jarak antar pusat unitnya dipilih secara '*sensible approach*' yakni mengambil nilai disekitar posisi yang dikehendaki. Hal itu diperlihatkan dalam fungsi Gaus yang menempatkannya pada harga keanggotaan disekitar maksimum yaitu 1.



Gambar 5.6 Respon Lapis *Hidden* Simulasi $l=7$, $b=1$, $d=0.1$



Gambar 5.7. Respon Lapis *Hidden*, $l=17$, $b=1$, $d=1$

Gambar respon lapis *hidden* (5.6 dan 5.7) keduanya mengambil parameter regulasi nol, karena suatu jaringan RBF yang di-*training* dengan metode seleksi pusat dengan cara random relatif tidak sensitif terhadap parameter regulasi⁸. Pada gambar 5.6, g_{l0} ada pada titik nol, artinya respon lapis *hidden* untuk unit-unit selain unit ke-4 lapis *hidden* (g_{03}) dan bias (b) memiliki fungsi keanggotaan (respon lapis *hidden*) sama dengan nol. Respon lapis *hidden* unit ke- l , ditulis g_l , $l=0, 1, \dots, 6$. Unit ke-8 –dan juga unit terakhir dari respon lapis *hidden*- adalah nilai bias (b) yang ditentukan, disini satu. Untuk bias yang tidak sama dengan satu dapat lebih jauh dilihat dalam lampiran A maupun lampiran B. Sedangkan pada gambar 5.7, g_{l0} sama seperti gambar 5.6 memiliki respon lapis *hidden* nol, tetapi untuk respon unit ke-9 lapis *hidden* (g_{08}) adalah satu artinya pemetaan matrik yang di-*learning* dilakukan dengan tepat.

⁸Haykin S., "Neural Networks : A Comprehensive Foundation," Macmillan College Publishing Company Inc., 1994, pp. 269

BAB VI

KESIMPULAN

VI.1 KESIMPULAN

Kesimpulan yang dapat diambil dari data-data hasil simulasi dan analisisnya :

1. Unit-unit neuron lapis *hidden* makin banyak akan makin teliti dalam menyebarkan pola input kedalam nilai keanggotaan sama dengan satu dalam fungsi basis Gauss. Ini sudah terlihat dalam hasil simulasi antara 7 dan 17 unit neuron lapis *hidden*.
2. Kelemahan pemilihan pusat-pusat unit lapis hidden secara random –sebagaimana telah dikerjakan dalam tugas akhir ini- yang adalah dilakukan secara *off-line*, tidak bisa mempelajari parameter-parameter yang tidak diketahui dari manipulator robot.
3. Matrik massa manipulator robot hasil *learning* dapat menyamai matrik massa yang *di-learning*, demikian juga fungsi lintasan yang didekati dengan RBF.

VI.2 SARAN

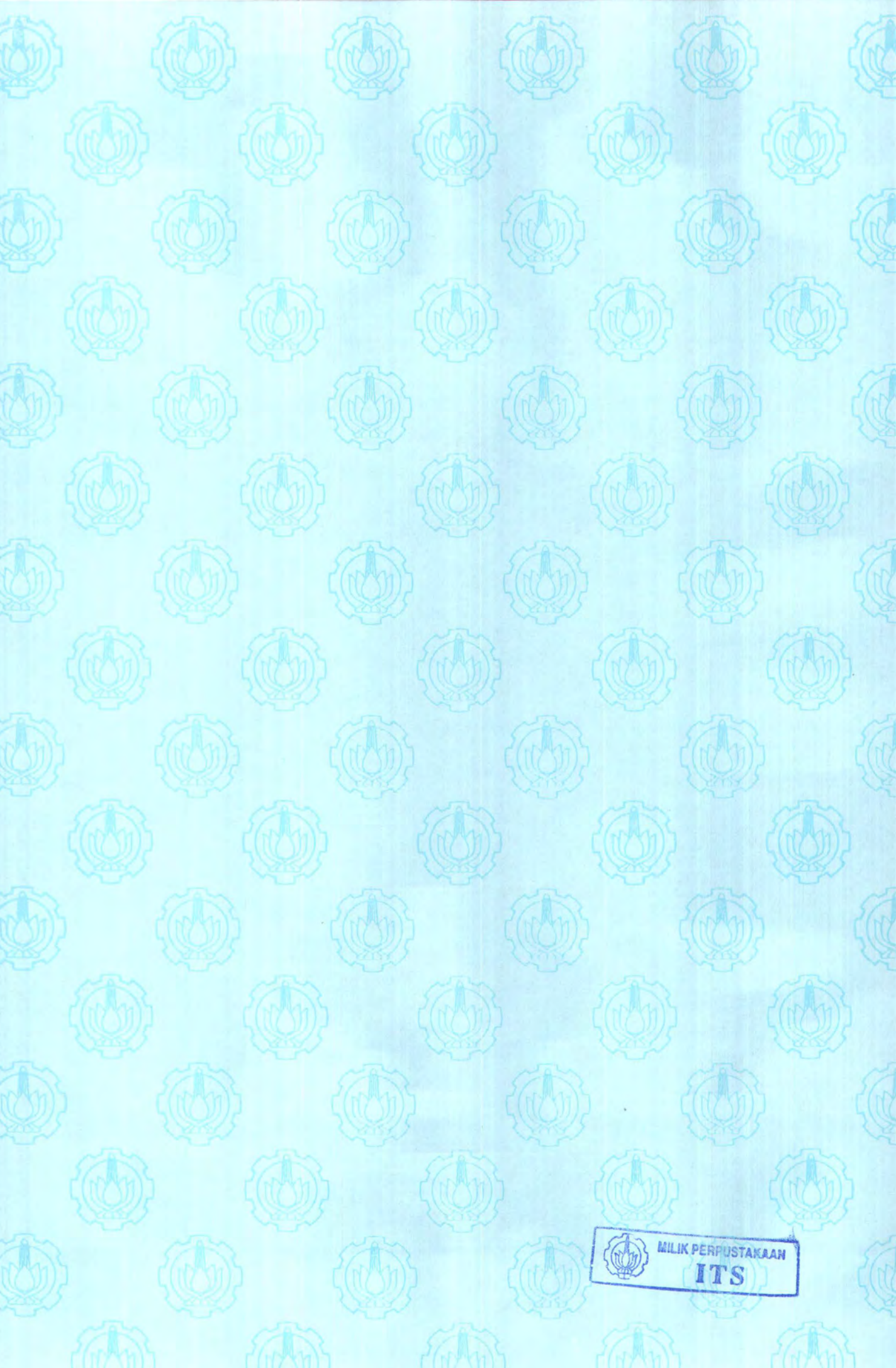
Pengerjaan untuk pemilihan pusat unit-unit neuron lapis hidden telah dilakukan. Realisasi pengendalian manipulator robot dengan pengendali modern memakai massa yang dianggap tidak diketahui secara pasti, perlu penelitian lebih lanjut untuk pendekatan pemilihan pusat dengan *self-organized* dan dengan metode supervised, seperti telah diuraikan dalam bab III, tulisan ini.

DAFTAR PUSTAKA

- [1] Craig J. J., Hsu P., and Sastry S. S., "Adaptive Control of Mechanical Manipulators," *Int. J. Robotics Res...*, vol. 6, no. 2, pp. 16-28, 1987.
- [2] Slotine J. J. E. and Li W., "On the Adaptive Control of Robot Manipulators," vol. 6, no. 3, pp. 49-59, 1987.
- [3] Arimoto S., "Learning Control Theory for Robot Motion," *Int. J. Adapt. Control Signal Process.*, vol. 4, no. 6, pp. 543-564, 1990.
- [4] Craig J. J., "Adaptive Control of Mechanical Manipulators," Addison-Wesley Publishing Company, Canada, 1988.
- [5] Goldberg K. Y., and Pearlmutter B. A., "Using Backpropagation with Temporal Windows to Learn the Dynamics of CMU Direct Drive Arm II," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed., San Mateo, CA: Morgan Kaufmann, 1989.
- [6] Miyamoto H., Kawato M., Setoyama T., and Suzuki R., "Feedback Error Learning Neural Network for Trajectory Control of A Robotic Manipulator," *Neural Networks*, vol. I, no. 3, pp. 251-265, 1988
- [7] Narendra K. S. and Partasarathy K., "Identification and Control of Dynamic Systems Using Neural Networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, 1990.
- [8] Miller W. T., Glanz F. H., and Kraft L. G. III, "Application of A General Learning Algorithm to the Control of Robot Manipulators," *Int. J. Robot Res.*, pp. 84-98, 1987

- [9] Saad M., Dessaint L. A., Bigras P., and Al-haddad K., "Adaptive versus Neural Network Adaptive Control : Application to Robotics," *Int. J. Adapt. Control Signal Process.*, vol. 8, no. 3, pp. 223-236, 1994
- [10] Tzirkel-Hancock E. and Fallside F., "A Direct Control Method for A Class of Nonlinear Systems Using Neural Network," Cambridge Univ., Cambridge, U.K., Rep. CUED/F-INFENG/TR65, 1991
- [11] Lewis F. L., Liu K., and Yesildirek A., "Neural Net Robot Controller with Guaranteed Tracking Performance," *IEEE Trans. Neural Networks*, vol. 6, pp. 703-715, May 1995
- [12] Ge S. S., and Hang C. C., "Direct Adaptive Neural Network Control of Robots," *Int. J. Sys. Sci.*, vol. 27, no. 6, pp.533-542, 1996
- [13] Ge S. Shuzhi, Hang C. C., Woon L. C., "Adaptive Neural Network Control of Robot Manipulators in Task Space," *IEEE Trans. on Industrial Electronics*, vol. 44, no. 6, pp. 746-752, Desember 1997.
- [14] Haykin, S., " *Neural Networks : A Comprehensive Foundation*," Macmillan College Publishing Company Inc., 1994.
- [15] Ishiguro Akio, Furuhashi Takeshi, Okuma Shigeru, Yoshiki Uchikawa, "A Neural Network Compensator for Uncertainties of Robotics Manipulators", *IEEE Trans. on Industrial Electronics*, vol.39, No. 6, Dec 1992.
- [16] Fu, LiMin, "Neural Networks in Computer Intelligence," McGraw-Hill Inc., Singapore, 1994.
- [17] Hortigan J. A., "Clustering Algorithms," John Wiley & Sons, Canada, 1975
- [18] Zalzal A. M. S. and Morris A. S., " *Neural Networks for Robotic Control : Theory and Applications*," Ellis Horwood, London, 1996.

- [19] Spong M. W., Vidyasagar M., "Robot Dynamics and Control," John Wiley & Sons, Canada, 1989.
- [20] Fu K. S., Gonzalez R. C., Lee C. S. G., "Robotics : Control, Sensing, Vision, and Intelligence," McGraw-Hill Book Company, Singapore, 1987.
- [21] Patterson D. W., "Artificial Neural Networks : Theory and Applications," Prentice Hall, Singapore, 1996.
- [22] Asada H., and Slotine J. E., "Robot Analysis and Control," John Willey and Sons, Massachusettes, 1986.
- [23] Setiawan Sandi, "Mengenal Network Saraf," Andi Offset, Yogyakarta, 1991.
- [24] Yamada T., and Yabuta T., "Adaptive Neural Network Controllers for Dynamics Systems," J. of Robotics and Mechatronics, vol. 2, no. 4, pp. 29-41.
- [25] Wiratno T., "Perancangan Kompensator Jaringan Syaraf Tiruan di Operational Space untuk Mengatasi Ketidakpastian pada Manipulator Robot," Tugas Akhir, Teknik Elektro ITS, 1998.
- [26] Chen S., Billings S. A., "Neural Networks for Nonlinear Dynamic System Modelling and Identification," Int. J. Control, vol. 56, no. 2, pp. 319-346, 1992.
- [27] Behera L., Gopal M., and Chaudhury S., "On Adaptive Trajectory Tracking of a Robot Manipulator Using Inversion of Its Neural Emulator," IEEE Trans. on Neural Networks, vol. 7, no. 6, November 1996.
- [28] Craig J. J., "Introduction to ROBOTICS Mechanics & Control," Addison-Wesley Publishing Company, Canada, 1986



 MILIK PERPUSTAKAAN
ITS

gb1 de4 = Respon Lapis Hidden dengan dimensi (25x8)

[0	0	0.0005	0.3927	0	0	0	0.1
0	0	0	0	0.7202	0	0	0	0.1
0	0	0	0.0024	0.0928	0	0	0	0.1
0	0	0	0	0	0	0.0297	0	0.1
0	0	0	0	0	0	0	0.0002	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1
0	0	0	0	0	0	0	0	0.1]

wb1de4 = Matrik bobot setelah melalui algoritma pseudoinvers dengan dimensi (8x3)

[0	0	0
0.0000	0.0000	0.0000	
-0.0000	-0.0000	-0.0000	
0.0000	0.0000	0.0000	
-0.0000	0.0000	0.0000	
0.0000	0.0000	0.0000	
-0.0000	-0.0000	-0.0000	
87.8040	43.3200	43.3200]	

- Untuk $h=7$, $b=1$, $d=0.1$, M_q dibuat tidak lurus (massa berubah-ubah)

$M_q =$

[8.7804	4.3320	4.3320
8.6804	4.3320	4.3320
8.5804	4.3320	4.3320
8.4804	4.3320	4.3320
8.5804	4.3320	4.3320
8.6804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320]

$w_{bd1} =$

[0	0	0
-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000
-179.0999	-0.0000	-0.0000
0	0	0
0	0	0
0	0	0
187.7841	4.3320	4.3320]

biasbd1 =

[187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320
187.7841	4.3320	4.3320]

Mactbd1 =

[8.6842	4.3320	4.3320
8.6842	4.3320	4.3320
8.6842	4.3320	4.3320
8.7021	4.3320	4.3320
8.7021	4.3320	4.3320
8.7200	4.3320	4.3320
8.7380	4.3320	4.3320
8.7738	4.3320	4.3320
8.7917	4.3320	4.3320
8.7917	4.3320	4.3320
8.8096	4.3320	4.3320
8.8275	4.3320	4.3320
8.8096	4.3320	4.3320
8.7917	4.3320	4.3320
8.7738	4.3320	4.3320
8.7559	4.3320	4.3320
8.7380	4.3320	4.3320
8.7200	4.3320	4.3320
8.7200	4.3320	4.3320
8.7200	4.3320	4.3320
8.7200	4.3320	4.3320
8.7380	4.3320	4.3320
8.7380	4.3320	4.3320
8.7380	4.3320	4.3320
8.7380	4.3320	4.3320]

- Untuk $hidden=7$, $b=1$, $d=0.1$

gbd1=Respon Lapis Hidden dengan dimensi (25x8)

[0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1
0	0	0	0.9999	0	0	0	1
0	0	0	0.9999	0	0	0	1
0	0	0	0.9998	0	0	0	1
0	0	0	0.9997	0	0	0	1
0	0	0	0.9995	0	0	0	1
0	0	0	0.9994	0	0	0	1
0	0	0	0.9994	0	0	0	1
0	0	0	0.9993	0	0	0	1
0	0	0	0.9992	0	0	0	1
0	0	0	0.9993	0	0	0	1
0	0	0	0.9994	0	0	0	1
0	0	0	0.9995	0	0	0	1
0	0	0	0.9996	0	0	0	1
0	0	0	0.9997	0	0	0	1
0	0	0	0.9998	0	0	0	1
0	0	0	0.9998	0	0	0	1
0	0	0	0.9998	0	0	0	1
0	0	0	0.9998	0	0	0	1
0	0	0	0.9997	0	0	0	1
0	0	0	0.9997	0	0	0	1
0	0	0	0.9997	0	0	0	1
0	0	0	0.9997	0	0	0	1]

wbd1 = Matrik bobot setelah melalui algoritma pseudoinvers dengan dimensi (8x3)

[0	0	0
-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000
0	0	0
0	0	0
0	0	0
8.7804	4.3320	4.3320]

LAMPIRAN B

(hidden 17)

- untuk $hidden=17$, $b=1$, $d=0.1$, Mq dibuat tidak lurus (massa berubah-ubah)

$Mq =$

[8.7804	4.3320	4.3320
8.6804	4.3320	4.3320
8.5804	4.3320	4.3320
8.4804	4.3320	4.3320
8.5804	4.3320	4.3320
8.6804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320
8.7804	4.3320	4.3320]

$w17bd1 =$

[0	0	0
-0.0000	-0.0000	-0.0000
0.0000	0.0000	0.0000
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-77.3965	0.0000	0.0000
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
86.0786	4.3320	4.3320]

bias17bd1 =

[86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320
86.0786 4.3320 4.3320]

M17bd1 =

[8.6821 4.3320 4.3320
8.6821 4.3320 4.3320
8.6821 4.3320 4.3320
8.6899 4.3320 4.3320
8.7053 4.3320 4.3320
8.7131 4.3320 4.3320
8.7363 4.3320 4.3320
8.7750 4.3320 4.3320
8.7905 4.3320 4.3320
8.7982 4.3320 4.3320
8.8137 4.3320 4.3320
8.8292 4.3320 4.3320
8.8137 4.3320 4.3320
8.7905 4.3320 4.3320
8.7750 4.3320 4.3320
8.7595 4.3320 4.3320
8.7363 4.3320 4.3320
8.7208 4.3320 4.3320
8.7131 4.3320 4.3320
8.7208 4.3320 4.3320
8.7286 4.3320 4.3320
8.7286 4.3320 4.3320
8.7363 4.3320 4.3320
8.7440 4.3320 4.3320
8.7440 4.3320 4.3320]

