



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599

**PENGATURAN GERAK *MANIPULATOR PUMA 560*
MENGUNAKAN *OPTIMAL COMPUTED TORQUE*
*CONTROL (OPTIMAL CTC)***

Aditya Rahmat Abdillah
NRP 2213105064

Dosen Pembimbing
Dr. Trihastuti Agustinah, S.T., M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE141599

**PUMA 560 MANIPULATOR MOTION CONTROL USING
OPTIMAL COMPUTED TORQUE CONTROL
(OPTIMAL CTC)**

Aditya Rahmat Abdillah
NRP 2213105064

Supervisor
Dr. Trihastuti Agustinah, S.T., M.T.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

**PENGATURAN GERAK MANIPULATOR PUMA 560
MENGUNAKAN OPTIMAL COMPUTED TORQUE CONTROL
(OPTIMAL CTC)**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing



Dr. Trihastuti Agustinah, S.T., M.T.
NIP. 196808121994032001

**SURABAYA
JULI, 2015**

PENGATURAN GERAK MANIPULATOR PUMA 560 MENGUNAKAN *OPTIMAL COMPUTED TORQUE CONTROL* (*OPTIMAL CTC*)

Nama : Aditya Rahmat Abdillah
Pembimbing : Dr. Trihastuti Agustinah, S.T., M.T.

ABSTRAK

Penggunaan *manipulator* di dunia industri berkembang pesat. Hal tersebut terkait dengan peningkatan performa kerja dan berkurangnya biaya produksi. Pada umumnya *manipulator* digunakan untuk menyelesaikan tugas seperti *material handling*, *assembling*, *welding*, dan sebagainya. Penugasan tersebut direpresentasikan melalui suatu trayektori dengan nilai referensi sudut yang berubah sesuai dengan titik-titik pada koordinat kartesian. Nonlinearitas sistem dan gaya friksi pada tiap *joint* dapat mengurangi akurasi dari *manipulator* sehingga hasil dari penugasan tidak sesuai. Untuk mengatasi permasalahan tersebut digunakan metode kontrol *Optimal CTC*. Pada metode tersebut, sinyal kontrol u untuk CTC dihitung menggunakan *linear quadratic regulator* (LQR). Metode kontrol tersebut menghasilkan solusi dalam ruang *joint*. Penyelesaian *inverse kinematic* menggunakan *jacobian manipulator* diterapkan pada sistem untuk menghasilkan solusi dalam ruang kartesian, sehingga penugasan dapat diberikan pada *manipulator*. Penugasan yang diberikan pada *manipulator* berupa trayektori berbentuk lingkaran dengan radius 10 hingga 50 cm. Hasil simulasi menunjukkan bahwa rata-rata nilai RMSE dari posisi *end-effector* sebesar 0,0065 cm pada sumbu x, 0,0099 cm pada sumbu y, dan 0,0082 cm pada sumbu z.

Kata Kunci : Akurasi, *Inverse Kinematic*, *Optimal CTC*

PENGATURAN GERAK *MANIPULATOR* PUMA 560 MENGUNAKAN *OPTIMAL COMPUTED TORQUE CONTROL* (*OPTIMAL CTC*)

Nama : Aditya Rahmat Abdillah
Pembimbing : Dr. Trihastuti Agustinah, S.T., M.T.

ABSTRAK

Penggunaan *manipulator* di dunia industri berkembang pesat. Hal tersebut terkait dengan peningkatan performa kerja dan berkurangnya biaya produksi. Pada umumnya *manipulator* digunakan untuk menyelesaikan tugas seperti *material handling*, *assembling*, *welding*, dan sebagainya. Penugasan tersebut direpresentasikan melalui suatu trayektori dengan nilai referensi sudut yang berubah sesuai dengan titik-titik pada koordinat kartesian. Nonlinearitas sistem dan gaya friksi pada tiap *joint* dapat mengurangi akurasi dari *manipulator* sehingga hasil dari penugasan tidak sesuai. Untuk mengatasi permasalahan tersebut digunakan metode kontrol *Optimal CTC*. Pada metode tersebut, sinyal kontrol u untuk CTC dihitung menggunakan *linear quadratic regulator* (LQR). Metode kontrol tersebut menghasilkan solusi dalam ruang *joint*. Penyelesaian *inverse kinematic* menggunakan *jacobian manipulator* diterapkan pada sistem untuk menghasilkan solusi dalam ruang kartesian, sehingga penugasan dapat diberikan pada *manipulator*. Penugasan yang diberikan pada *manipulator* berupa trayektori berbentuk lingkaran dengan radius 10 hingga 50 cm. Hasil simulasi menunjukkan bahwa rata-rata nilai RMSE dari posisi *end-effector* sebesar 0,0065 cm pada sumbu x, 0,0099 cm pada sumbu y, dan 0,0082 cm pada sumbu z.

Kata Kunci : Akurasi, *Inverse Kinematic*, *Optimal CTC*



PUMA 560 MANIPULATOR MOTION CONTROL USING OPTIMAL COMPUTED TORQUE CONTROL (OPTIMAL CTC)

Name : Aditya Rahmat Abdillah
Supervisor : Dr. Trihastuti Agustinah, S.T., M.T.

ABSTRACT

There is a rapid development of manipulator application in industrial environment. It was concerned with performance upgrading and cost reduction. Most of manipulators are used for solving tasks such material handling, assembling, welding, etc. The tasks given contain some trajectories with vary joint reference values. Nonlinearity of system and friction for each joint cause the accuracy of manipulator decrease. Optimal CTC is used to solve the problem with solutions in joint space. Inverse kinematic solutions is applied to the system for obtaining solutions in cartesian space, so the task can be applied to manipulator. A circular trajectory with radius 10 to 50 cm are given for the manipulator as a task. The simulation results show that RMSE values average of x-axis is 0.0065 cm, 0.0099 cm at y-axis, and 0.0082 cm at z-axis.

Keywords : Accuracy, Inverse Kinematic, Optimal CTC



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah mencurahkan segala rahmat dan hidayahnya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul *Pengaturan Gerak Manipulator PUMA 560 Menggunakan Optimal Computed Torque Control (Optimal CTC)* tepat waktu.

Tugas Akhir ini merupakan syarat untuk menyelesaikan studi dan memperoleh gelar Sarjana di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih banyak atas bantuan dan motivasi yang diberikan baik secara langsung ataupun tidak langsung dalam pelaksanaan Tugas Akhir ini kepada:

1. Ibu Dr. Trihastuti Agustinah, S.T., M.T., selaku Dosen Pembimbing dalam Tugas Akhir ini
2. Orang tua dan keluarga yang telah mencurahkan doa, kasih sayang dan semangat,
3. Serta semua pihak yang tidak bisa disebutkan satu-persatu yang turut membantu dan memperlancar jalankan Tugas Akhir ini.

Penulis menyadari bahwa penyusunan buku ini masih belum sempurna. Untuk itu, penulis mengharapkan adanya kritik dan saran demi kebaikan di kemudian hari. Akhir kata, penulis mohon maaf jika ada kekurangan selama pengerjaan Tugas Akhir ini.

Surabaya, Juni 2015

Penulis

Aditya Rahmat A



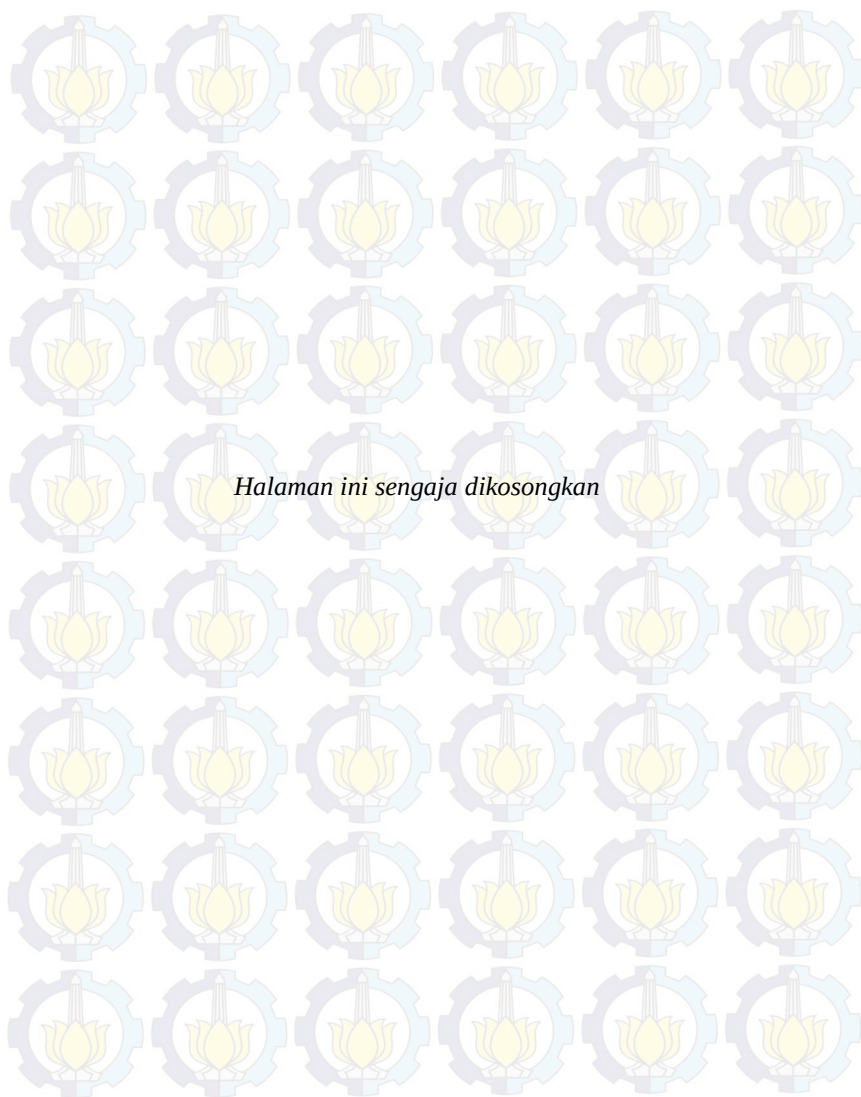
DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Metodologi Penelitian.....	2
1.6. Sistematika Penulisan.....	3
1.7. Relevansi.....	4
BAB II TEORI PENUNJANG.....	5
2.1. <i>Manipulator</i>	5
2.1.1. Komponen dan Struktur dari <i>Manipulator</i>	5
2.1.2. Klasifikasi dari <i>Manipulator</i>	6
2.2. Kinematika <i>Manipulator</i>	8
2.2.1. <i>Forward Kinematic</i>	8
2.2.2. <i>Inverse Kinematic</i>	11
2.3. <i>Jacobian Manipulator</i>	12
2.4. Model Dinamika PUMA 560.....	13
2.5. <i>Computed Torque Control (CTC)</i>	16
2.6. <i>Linear Quadratic Regulator (LQR)</i>	17
2.7. Perencanaan Trayektori.....	18
2.8. Spesifikasi PUMA 560.....	19
BAB III PERANCANGAN SISTEM.....	21
3.1. Perancangan Sistem Keseluruhan.....	21
3.2. Perancangan <i>Forward Kinematic</i> PUMA 560.....	22
3.3. Perancangan <i>Inverse Kinematic</i> PUMA 560.....	23

3.4. Perancangan Model Dinamika PUMA 560.....	26
3.5. Perancangan Kontroler <i>Linear Quadratic Regulator</i>	28
3.6. Perancangan Kontroler <i>Computed Torque Control</i>	30
3.7. Perancangan Trayektori Kartesian.....	31
3.8. Perancangan Sistem Keseluruhan Menggunakan Matlab dengan <i>Toolbox Robot</i> Peter I Corke.....	32
BAB IV PENGUJIAN DAN ANALISIS SISTEM.....	33
4.1. Pengujian dan Analisis Respons <i>Joint</i>	33
4.1.1. Pengujian dan Analisis Menggunakan Sinyal <i>Step</i>	33
4.1.2. Pengujian dan Analisis Menggunakan Sinyal <i>Cubical Polynomial</i>	38
4.2. Pengujian dan Analisis <i>Inverse Kinematic</i> PUMA 560 Menggunakan <i>Jacobian Manipulator</i>	41
4.3. Pengujian dan Analisis Akurasi Posisi Aktual <i>End-Effector</i> Melalui Suatu Penugasan.....	42
BAB V PENUTUP.....	47
5.1. Kesimpulan.....	47
5.2. Saran.....	47
DAFTAR PUSTAKA.....	49
LAMPIRAN A.....	51
LAMPIRAN B.....	55
LAMPIRAN C.....	59
RIWAYAT PENULIS.....	69

DAFTAR TABEL

Tabel 2.1 Batas Gerak <i>Joint</i> PUMA 560.....	19
Tabel 2.2 Torsi Maksimum PUMA 560.....	19
Tabel 3.1 Parameter D-H Standar PUMA 560.....	22
Tabel 3.2 Konstanta Inersia PUMA 560.....	27
Tabel 3.3 Konstanta Gravitasi PUMA 560.....	27
Tabel 4.1 Rata-rata Nilai RMSE dan Waktu <i>Steady State</i> Respons Sistem.....	36
Tabel 4.2 Perbandingan Respons CTC Terhadap <i>Optimal</i> CTC....	38
Tabel 4.3 Nilai RMSE dari Respons Tiap <i>Joint</i> Terhadap Sinyal <i>Cubical Polynomial</i>	39
Tabel 4.4 Hasil Perhitungan <i>Inverse Kinematic</i>	41
Tabel 4.5 Pengujian Validitas Hasil <i>Inverse Kinematic</i>	42
Tabel 4.6 Nilai RMSE Posisi <i>End-Effector</i> 1.....	42
Tabel 4.7 Nilai RMSE Posisi <i>End-Effector</i> 2.....	43



DAFTAR GAMBAR

Gambar 2.1	Bagian-bagian <i>Manipulator</i>	6
Gambar 2.2	Jenis-jenis <i>Manipulator</i> Berdasarkan Bentuk Geometri.....	7
Gambar 2.3	Sumbu <i>Joint Manipulator</i>	9
Gambar 2.4	Acuan Pengukuran θ_i	9
Gambar 2.5	Acuan Pengukuran α_i	10
Gambar 2.6	Gaya Sentripetal.....	14
Gambar 2.7	Gaya Koriolis.....	14
Gambar 3.1	Diagram Blok Sistem Keseluruhan.....	21
Gambar 3.2	<i>Flowchart</i> Perancangan <i>Forward Kinematic</i> PUMA 560 Menggunakan Matlab.....	24
Gambar 3.3	Perancangan <i>Jacobian</i> PUMA 560.....	25
Gambar 3.4	Perancangan <i>Inverse Kinematic</i> PUMA 560.....	26
Gambar 3.5	<i>Flowchart</i> Perancangan Model Dinamik PUMA 560.....	28
Gambar 3.6	Diagram Blok Kontrol LQR.....	29
Gambar 3.7	<i>Flowchart</i> Perancangan Kontrol CTC.....	30
Gambar 3.8	Perancangan Trayektori Kartesian.....	31
Gambar 3.9	Animasi <i>Manipulator</i> Pada <i>Software</i> Matlab.....	32
Gambar 4.1	Respons <i>Joint</i> 1 CTC Terhadap Sinyal Uji <i>Step</i>	34
Gambar 4.2	Respons <i>Joint</i> 2 CTC Terhadap Sinyal Uji <i>Step</i>	34
Gambar 4.3	Respons <i>Joint</i> 3 CTC Terhadap Sinyal Uji <i>Step</i>	34
Gambar 4.4	Respons <i>Joint</i> 4 CTC Terhadap Sinyal Uji <i>Step</i>	35
Gambar 4.5	Respons <i>Joint</i> 5 CTC Terhadap Sinyal Uji <i>Step</i>	35
Gambar 4.6	Respons <i>Joint</i> 6 CTC Terhadap Sinyal Uji <i>Step</i>	35
Gambar 4.7	Respons <i>Joint</i> 1 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	36
Gambar 4.8	Respons <i>Joint</i> 2 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	36
Gambar 4.9	Respons <i>Joint</i> 3 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	37
Gambar 4.10	Respons <i>Joint</i> 4 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	37
Gambar 4.11	Respons <i>Joint</i> 5 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	37
Gambar 4.12	Respons <i>Joint</i> 6 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	38

Gambar 4.13	Respons <i>Joint</i> 1 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	39
Gambar 4.14	Respons <i>Joint</i> 2 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	39
Gambar 4.15	Respons <i>Joint</i> 3 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	40
Gambar 4.16	Respons <i>Joint</i> 4 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	40
Gambar 4.17	Respons <i>Joint</i> 5 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	40
Gambar 4.18	Respons <i>Joint</i> 6 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	41
Gambar 4.19	Hasil Penugasan Trayektori Lingkaran Untuk <i>Manipulator</i>	43
Gambar 4.20	Respons <i>Joint</i> 1 Untuk Penugasan Trayektori Lingkaran.....	44
Gambar 4.21	Respons <i>Joint</i> 2 Untuk Penugasan Trayektori Lingkaran.....	44
Gambar 4.22	Respons <i>Joint</i> 3 Untuk Penugasan Trayektori Lingkaran.....	44
Gambar 4.23	Respons <i>Joint</i> 4 Untuk Penugasan Trayektori Lingkaran.....	45
Gambar 4.24	Respons <i>Joint</i> 5 Untuk Penugasan Trayektori Lingkaran.....	45
Gambar 4.25	Respons <i>Joint</i> 6 Untuk Penugasan Trayektori Lingkaran.....	45

RIWAYAT PENULIS



Aditya Rahmat Abdillah, dilahirkan di Surabaya, 19 Maret 1993 sebagai putra sulung dari Bapak Panca Suluh DS dan Ibu Aisyah Nikma S. Penulis sempat merasakan beberapa tempat selama menempuh masa pendidikan. Sekolah dasar di SD Negeri Bangselok 1 Sumenep, sempat pindah ke SD Negeri Pucang Anom 1 Sidoarjo saat kelas 2 hingga kelas 3, melanjutkan sekolah menengah di SMP Negeri 1 Sumenep, dan SMA Negeri 1 Sumenep. Setelah lulus sekolah menengah dilanjutkan ke Diploma 3 Teknik Elektro ITS dan lulus pada Tahun 2013, kemudian melanjutkan ke S1 Teknik Elektro di bidang studi teknik sistem pengaturan. Pada bulan juni 2015 penulis melaksanakan seminar dan sidang Tugas Akhir untuk memperoleh gelar Sarjana S1 Teknik Elektro.

Email: aditya.rahmat10@mhs.ee.its.ac.id



BAB I

PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, perumusan masalah, batasan masalah dan tujuan dari tugas akhir yang dilakukan. Metodologi penelitian dan sistematika penulisan dalam penyelesaian tugas akhir juga disertakan. Relevansi diberikan sebagai langkah untuk penelitian selanjutnya.

1.1. Latar Belakang

Penggunaan robot di dunia industri berkembang pesat, khususnya *manipulator* [1]. Peningkatan performa kerja dari *manipulator* seperti efektivitas kerja, akurasi, dan presisi merupakan faktor utama yang memicu hal tersebut. Seiring dengan meningkatnya performa kerja tersebut, jumlah produksi menjadi lebih banyak, dan sebaliknya biaya produksi menjadi berkurang. *Manipulator* juga memiliki keunggulan berupa kemampuan menyelesaikan pekerjaan yang berbahaya untuk manusia seperti tempat radioaktif dan penjelajahan luar angkasa [1].

Pada umumnya *manipulator* melakukan tugas seperti *material handling*, *assembling*, *welding* dan sebagainya. Akurasi pada *manipulator* akan berkurang akibat adanya nonlinearitas dari sistem dan gaya friksi pada tiap *joint*, sehingga hasil dari tugas yang diberikan tidak sesuai. Oleh karena permasalahan tersebut, dilakukan pengaturan gerak untuk *manipulator* agar hasil dari penugasan sesuai. Pengaturan gerak yang digunakan menghasilkan penyelesaian dalam ruang *joint* sehingga digunakan *inverse kinematic* untuk memperoleh solusi pengaturan gerak *manipulator* dalam ruang kartesian.

Metode pengaturan gerak yang digunakan berupa *model-based control* yang membutuhkan model dinamika dari *manipulator* untuk proses pengaturannya. Metode *model-based control* yang umum digunakan yaitu *feedforward* dan *computed torque control*. Pada tugas akhir ini digunakan *computed torque control* sebagai metode untuk pengaturan gerak *manipulator*. Metode tersebut dikombinasikan dengan kontrol optimal untuk perhitungan *gain feedback outer loop*, sehingga akurasi dari *manipulator* dapat ditingkatkan. Perhitungan *inverse kinematic* dilakukan menggunakan *Jacobian manipulator*. *Jacobian manipulator* mendeskripsikan hubungan antara kecepatan *end-effector* dengan kecepatan *joint* dari *manipulator*.

1.2. Perumusan Masalah

Permasalahan yang muncul dalam penggunaan *manipulator* yaitu sifat nonlinear dan besaran seperti gaya friksi pada tiap *joint* yang menyebabkan akurasi *manipulator* berkurang, sehingga hasil dari penugasan tidak sesuai dengan referensi. Suatu kontroler nonlinear dirancang untuk mengatasi permasalahan akurasi tersebut. Penugasan untuk *manipulator* dilakukan dengan merencanakan suatu trayektori dalam ruang *joint* atau kartesian. Trayektori yang direncanakan dalam ruang *joint* lebih sulit dilakukan dibandingkan trayektori dalam ruang kartesian karena kemungkinan nilai-nilai sudut yang mewakili posisi dalam ruang kartesian sangat banyak. Penyelesaian *inverse kinematic* digunakan agar perencanaan trayektori kartesian dapat dilakukan. Trayektori yang digunakan sebagai penugasan berisi nilai referensi *joint* yang bervariasi dalam waktu yang singkat sehingga dapat meningkatkan *error tracking* jika respons dari tiap *joint* lambat. Oleh karena itu digunakan metode *optimal* untuk meningkatkan kecepatan respons tiap *joint*.

1.3. Batasan Masalah

Pada tugas akhir ini *manipulator* yang dikontrol adalah PUMA 560. Pengaturan gerak PUMA 560 meliputi posisi, kecepatan, dan percepatan dari sudut *joint*. Oleh karena itu pengaturan gerak *manipulator* pada tugas akhir ini dikhususkan pada penugasan *material handling*. Penyelesaian *inverse kinematic* yang digunakan menggunakan *jacobian manipulator*. Diasumsikan *workspace* bebas dari penghalang.

1.4. Tujuan Penelitian

Penelitian pada tugas akhir ini memiliki tujuan yaitu merancang kontroler *optimal* CTC untuk mengatur gerak *manipulator* PUMA 560 dalam ruang *joint* dan merancang *inverse kinematic* menggunakan *jacobian manipulator* untuk merubah nilai-nilai koordinat dari trayektori kartesian menjadi nilai-nilai sudut tiap *joint* sehingga perencanaan trayektori dalam ruang kartesian dapat dilakukan untuk penugasan *manipulator* PUMA 560.

1.5. Metodologi Penelitian

Metodologi yang dilakukan pada Tugas Akhir ini yaitu :

a. Studi Literatur

Pada tahap ini dilakukan kajian teori yang berkaitan dengan pemodelan kinematika dan dinamika dari *manipulator* khususnya PUMA 560. Sumber yang digunakan berupa buku referensi, *paper* penelitian terkait, dan jurnal ilmiah.

b. Pemodelan Sistem

Pemodelan sistem terdiri dari *kinematic* dan dinamika dari PUMA 560 yang diperoleh dari jurnal ilmiah terkait.

c. Perancangan Kontroler

Pada tahap ini dilakukan perancangan kontroler *computed torque control* yang dikombinasikan dengan *linear quadratic regulator*.

d. Perancangan Simulasi

Hasil perancangan sistem pengaturan gerak *manipulator* disimulasikan menggunakan *software* Matlab 2010a dengan *toolbox* robot Peter I Corke versi 8.0.

e. Penulisan Buku

Hasil dari tahap studi literatur hingga perancangan simulasi dituliskan pada buku laporan tugas akhir.

1.6. Sistematika Penulisan

Penulisan tugas akhir ini disusun dalam lima bab yang masing-masing membahas permasalahan yang berhubungan dengan tugas akhir yang telah dibuat dengan sistematika penulisan sebagai berikut:

a. BAB I PENDAHULUAN

Berisi tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan, metodologi, sistematika penulisan dan relevansi pembahasan tugas akhir ini.

b. BAB II TEORI PENUNJANG

Memberi gambaran secara umum mengenai konsep teori yang mendasari perancangan tugas akhir ini, meliputi teori *manipulator*, model kinematika *manipulator*, model dinamika *manipulator*, *jacobian manipulator*, *computed torque control*,

linear quadratic regulator, dan perencanaan trayektori untuk *manipulator*.

c. BAB III PERANCANGAN SISTEM

Dijelaskan secara menyeluruh tentang perancangan simulasi sistem yang terdiri dari perancangan *forward kinematic*, perancangan *inverse kinematic*, perancangan model dinamika, perancangan *computed torque control*, perancangan *linear quadratic regulator*, dan perancangan trayektori *manipulator*.

d. BAB IV PENGUJIAN DAN ANALISIS SISTEM

Berisi hasil pengujian dan analisis dari sistem yang telah dirancang. Pengujian yang dilakukan meliputi respons *joint* terhadap sinyal uji, *inverse kinematic*, dan penugasan trayektori lingkaran. Hasil dari tiap pengujian dianalisis dengan cara melihat *settling time* untuk respons sinyal tiap sinyal uji, validitas solusi nilai *joint* untuk *inverse kinematic*, dan akurasi dari hasil penugasan yang diberikan pada *manipulator*.

e. BAB V PENUTUP

Berisi tentang kesimpulan dan saran untuk pengembangan tugas akhir ini.

1.7. Relevansi

Tugas akhir ini diharapkan dapat digunakan sebagai referensi pengaturan gerak pada *manipulator* jenis lainnya, dilakukan implementasi pada *plant manipulator*, serta pengembangan untuk perancangan metode kontrol selanjutnya.

BAB II

TEORI PENUNJANG

Pada bab ini dibahas teori-teori yang berkaitan dengan topik penelitian yang dilakukan. Dasar teori pada bab ini meliputi pendahuluan mengenai *manipulator*, komponen dan struktur dari *manipulator*, kinematika dan dinamika *manipulator*, metode *linear quadratic regulator* (LQR), *computed torque control* (CTC), perencanaan trayektori, dan spesifikasi PUMA 560 yang akan diterapkan pada *manipulator*.

2.1. Manipulator

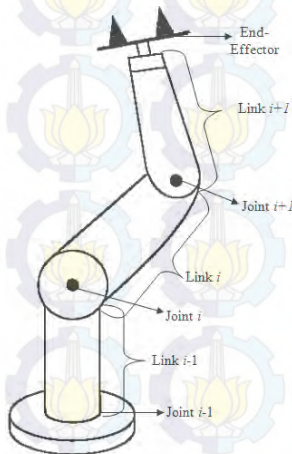
Robot Institute of America (RIA) mendefinisikan robot sebagai perangkat multifungsi terprogram yang dirancang untuk memindahkan material, atau alat khusus dengan perencanaan pergerakan untuk menyelesaikan tugas tertentu [2]. Secara struktur mekanik, robot diklasifikasikan menjadi *manipulator (fixed base frame)* dan *mobile robot (mobile base frame)*. Kedua jenis robot tersebut dibedakan berdasarkan *base frame*-nya. *Base frame* adalah sistem koordinat yang dijadikan sebagai acuan dari *manipulator* dan segala objek pada *workspace manipulator*. *Base frame* dapat bersifat tetap (*fixed*) atau bergerak (*mobile*). Penjelasan selanjutnya akan dikhususkan untuk *manipulator*.

Manipulator merupakan jenis robot yang memiliki *base frame* tetap. Tujuan dari penggunaan *manipulator* untuk membantu pekerjaan yang dilakukan oleh manusia khususnya untuk pekerjaan yang tidak memungkinkan seperti eksplorasi bawah laut dan luar angkasa. Penggunaan *manipulator* di dunia industri juga mampu mengurangi biaya produksi, serta meningkatkan kualitas dan kapasitas produksi.

2.1.1. Komponen dan Struktur dari Manipulator

Manipulator terdiri dari beberapa komponen antara lain *joint*, *link*, dan *end-effector*. *Joint* adalah komponen yang menggerakkan *link* dan *end-effector*. Struktur mekanik *manipulator* terdiri dari rangkaian *link* yang terhubung melalui *joint*. Rangkaian dari *joint*, *link*, dan *end-effector* disebut *kinematic chain*. *Kinematic chain* dapat berupa *serial/open* atau *parallel/close* [3]. Struktur dasar dari *manipulator* sebagian besar berupa *serial/open kinematic chain*. Pergerakan *manipulator* ditentukan oleh adanya *joint* yang dapat berupa *prismatic* atau *revolute joint*. Untuk menggunakan *manipulator* dalam berbagai tugas, diperlukan spesifikasi

workspace. *Workspace* merupakan area yang dapat dijangkau oleh *manipulator* berdasarkan nilai variabel *joint* yang memungkinkan. *Workspace* dibagi menjadi *reachable* dan *dextrous* [2]. Bagian-bagian dari *manipulator* ditunjukkan pada Gambar 2.1.



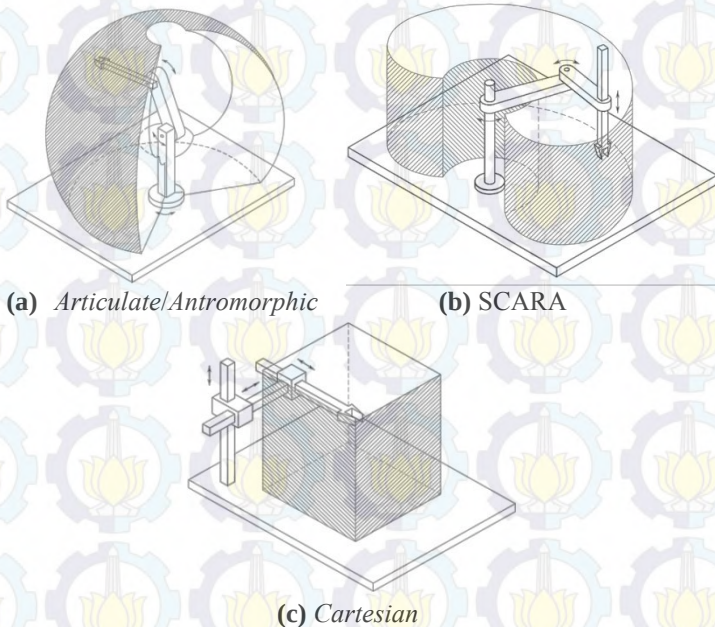
Gambar 2.1 Bagian-bagian *Manipulator*

Dalam suatu *open kinematic chain*, tiap *joint* memberikan pergerakan sebanyak satu derajat kebebasan (DOF). Derajat kebebasan merupakan suatu ukuran yang menunjukkan kemampuan *manipulator* untuk menuju ke titik-titik kerja dalam *workspace*. Pada umumnya *manipulator* memiliki enam derajat kebebasan yang berfungsi untuk pengerjaan tugas dengan posisi dan orientasi dalam ruang tiga dimensi. Sistem robot selain dari komponen mekanik yang telah dijelaskan antara lain *power supply*, lengan mekanik, sensor, *teach pendant*, kontroler, penyimpanan program, dan *tool end-effector* [3].

2.1.2. Klasifikasi dari *Manipulator*

Manipulator diklasifikasikan berdasarkan sumber daya, area aplikasi, dan geometrinya [2]. Sumber daya *manipulator* dapat berupa hidrolik, pneumatik, atau motor servo. Berdasarkan area aplikasi *manipulator* dibagi menjadi *assembly* dan *non-assembly*. Ukuran *manipulator* untuk area *assembly* umumnya berbentuk kecil. Area *non-*

assembly meliputi pengecatan, pengelasan, dan *material handling*. Berdasarkan geometri *manipulator* terbagi menjadi lima antara lain *articulate/antromorphic*, *spherical*, SCARA, *cylindrical*, dan *cartesian*. Beberapa jenis *manipulator* berdasarkan bentuk geometrinya antara lain, *Articulate/Antromorphic*, *Cartesian*, dan SCARA [2]. Beberapa Bentuk fisik dari *manipulator* tersebut ditunjukkan pada Gambar 2.2.



Gambar 2.2 Jenis-jenis *Manipulator* Berdasarkan Bentuk Geometri [2]

Dari ketiga *manipulator* tersebut memiliki perbedaan dari konfigurasi *joint*. *Articulate/antromorphic* terdiri atas enam *revolute joint*, SCARA terdiri atas dua atau tiga *revolute joint* dan satu *prismatic joint* yang terletak pada *end-effector*. *Cartesian* memiliki tiga sampai lima *prismatic joint*. Dalam melakukan tugas yang diprogramkan, *manipulator* harus memenuhi tiga syarat yaitu:

- *Manipulator* tidak boleh membahayakan manusia.
- *Manipulator* harus mematuhi perintah yang diberikan oleh manusia, kecuali perintah yang melanggar syarat pertama.

- *Manipulator* harus menjaga kondisinya sendiri selama proteksi tersebut tidak bertentangan dengan syarat pertama dan kedua.

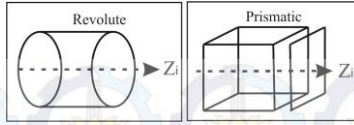
Penyelesaian tugas dengan persyaratan di atas dapat dipenuhi dengan perancangan dan perencanaan terkait kontrol dan *trajectory planning* dari *manipulator* tersebut. Kontrol *manipulator* berhubungan dengan model dinamika dari *manipulator*. Hasil dari kontrol *manipulator* berhubungan dengan akurasi dan presisi. Sebagian besar *manipulator* saat ini memiliki presisi yang tinggi, namun tidak terlalu akurat. Pengukuran nilai variabel *joint* umumnya menggunakan *encoder*. *Encoder* diletakkan pada *joint* atau lebih tepatnya pada *shaft* dari motor servo. Posisi dan orientasi dari *end-effector* umumnya tidak diukur secara langsung. Oleh karena itu pengukuran dari posisi dan orientasi *end-effector* menggunakan kinematika berdasarkan nilai variabel *joint*.

2.2. Kinematika *Manipulator* [2]

Kinematika merupakan studi tentang gerak dari suatu benda tanpa menghiraukan gaya yang menyebabkannya. Dalam penerapannya untuk *manipulator*, kinematika digunakan untuk mengetahui posisi dan orientasi dari *end-effector* terhadap *base frame manipulator* serta benda lain yang ada dalam lingkup *workspace manipulator*. Posisi dan orientasi ini dapat direpresentasikan dalam ruang kartesian maupun ruang *joint*. Untuk memperoleh koordinat kartesian berdasarkan nilai-nilai sudut tiap *joint* digunakan *forward kinematic* sedangkan sebaliknya digunakan *inverse kinematic*.

2.2.1. *Forward Kinematic* [2]

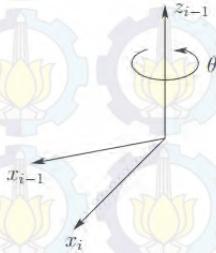
Perhitungan *forward kinematic* menggunakan metode yang biasa disebut dengan kaidah Denavit-Hartenberg. Kaidah ini merepresentasikan tiap transformasi homogen ' A_i ' sebagai perkalian dari empat transformasi dasar yang ditunjukkan pada Persamaan 2.1. Matriks transformasi homogen A_i memiliki empat parameter yaitu θ_i , a_i , d_i , dan α_i . Untuk *prismatic joint* nilai d_i berupa variabel, sedangkan untuk *revolute joint* nilai θ_i berupa variabel. Perhitungan kuantitas tersebut dilakukan dengan cara menentukan *frame* untuk tiap *joint manipulator* yang terbagi menjadi sembilan langkah. Langkah pertama dalam pembentukan *frame* yaitu menentukan sumbu *joint* $z_0, \dots, z_{n-1}$ dimana n adalah jumlah *joint manipulator*. Sumbu *joint manipulator* ditunjukkan pada Gambar 2.3.



Gambar 2.3 Sumbu *Joint Manipulator*

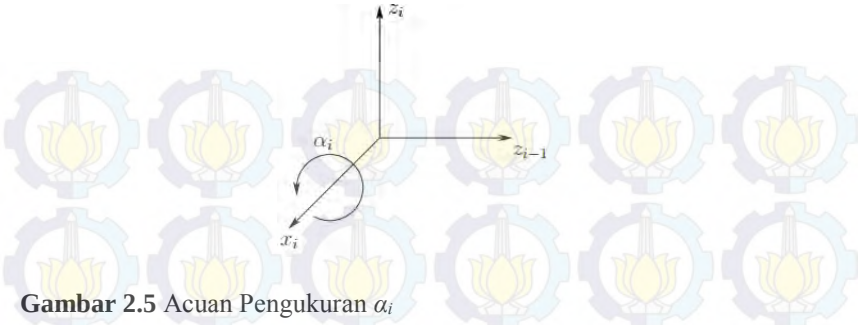
Langkah kedua yaitu menentukan *base frame*. Titik *origin* o_0 diletakkan bebas di sepanjang sumbu z_0 . Sumbu x_0 dan y_0 dibentuk berdasarkan kaidah *right-hand frame*. Langkah ketiga yaitu dibentuk *origin* o_i dimana $i = 1, \dots, n-1$ tegak lurus terhadap z_i dan z_{i-1} memotong z_i . Untuk z_i berpotongan dengan z_{i-1} , o_i diletakkan pada titik perpotongan tersebut, sedangkan jika z_i paralel dengan z_{i-1} , o_i diletakkan pada posisi yang sesuai di sepanjang z_i . Langkah keempat yaitu dibentuk sumbu x_i sepanjang tegak lurus antara z_{i-1} dan z_i melalui o_i atau searah tegak lurus terhadap bidang $z_{i-1} - z_i$ jika z_{i-1} dan z_i berpotongan. Langkah kelima yaitu dibentuk sumbu y_i sesuai dengan kaidah *right-hand frame*. Langkah keenam yaitu dibentuk *end-effector frame* $o_n x_n y_n z_n$. Diasumsikan *joint* ke- n adalah *revolute*, bentuk z_n sepanjang arah z_{n-1} . Dibentuk titik *origin* o_n sepanjang z_n , sebaiknya ditengah *gripper* atau *tool end-effector*. Dibentuk y_n searah penutupan *gripper* dan $x_n = y_n \times z_n$. Langkah ketujuh yaitu membuat tabel parameter θ_i , a_i , d_i , dan α_i dengan ketentuan sebagai berikut:

- θ_i : Sudut antara x_{i-1} dan x_i diukur terhadap z_{i-1} .



Gambar 2.4 Acuan Pengukuran θ_i

- α_i : Sudut antara z_{i-1} dan z_i diukur terhadap x_i .



Gambar 2.5 Acuan Pengukuran α_i

- a_i : Jarak sepanjang x_i dari o_i hingga perpotongan sumbu x_i dan z_{i-1} .
- d_i : Jarak sepanjang z_{i-1} dari o_{i-1} hingga perpotongan sumbu x_i dan z_{i-1} .

Langkah kedelapan yaitu dibentuk matriks transformasi homogen A_i dengan mensubstitusi parameter pada langkah ketujuh ke Persamaan 2.1.

$$\begin{aligned}
 A_i &= \text{Rotasi}_{z, \theta_i} \text{Translasi}_{z, d_i} \text{Translasi}_{x, a_i} \text{Rotasi}_{x, \alpha_i} \\
 &= \begin{bmatrix} R_i^{i-1} & \mathbf{p}_i^{i-1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)
 \end{aligned}$$

dengan matriks rotasi $R_3 = [n \ s \ a]$, dan matriks posisi $\mathbf{p}_{3 \times 1} = [p_x \ p_y \ p_z]^T$. Langkah kesembilan yaitu membentuk matriks T_n^{i-1} untuk *manipulator* dengan n -DOF yang ditunjukkan pada Persamaan 2.2.

$$T_n^{i-1} = A_i A_{i+1} \dots A_n, \quad i = 1 \quad (2.2)$$

2.2.2. Inverse Kinematic [2]

Inverse kinematic merupakan suatu perhitungan nilai-nilai variabel tiap *joint manipulator* berdasarkan posisi dan orientasi yang telah

diketahui sebelumnya. Pada umumnya metode aljabar digunakan untuk menyelesaikan perhitungan ini. Perhitungan tersebut dilakukan dengan cara menyelesaikan persamaan-persamaan yang berisi variabel *joint* pada elemen dari matriks transformasi homogen berdasarkan matriks rotasi dan posisi yang telah diketahui. Perhitungan metode aljabar menjadi sulit dilakukan untuk *manipulator* yang memiliki *joint* banyak. Contoh persoalan *inverse kinematic* untuk *Stanford Manipulator*[3] ditunjukkan pada Persamaan 2.3.

$$n_x = C_1 [C_2 (C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] - S_1 (S_4 C_5 C_6 + C_4 S_6) \quad (2.3.a)$$

$$n_y = S_1 [C_2 (C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] + C_1 (S_4 C_5 C_6 + C_4 S_6) \quad (2.3.b)$$

$$n_z = -S_2 (C_4 C_5 C_6 - S_4 S_6) - C_2 S_5 S_6 \quad (2.3.c)$$

$$s_x = C_1 [-C_2 (C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] - S_1 (-S_4 C_5 S_6 + C_4 C_6) \quad (2.3.d)$$

$$s_y = S_1 [-C_2 (C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] + C_1 (-S_4 C_5 S_6 + C_4 C_6) \quad (2.3.e)$$

$$s_z = S_2 (C_4 C_5 S_6 + S_4 C_6) + C_2 S_5 S_6 \quad (2.3.f)$$

$$a_x = C_1 (C_2 C_4 S_5 + S_2 C_5) - S_1 S_4 S_5 \quad (2.3.g)$$

$$a_y = S_1 (C_2 C_4 S_5 + S_2 C_5) + C_1 S_4 S_5 \quad (2.3.h)$$

$$a_z = -S_2 C_4 S_5 + C_2 C_5 \quad (2.3.i)$$

$$p_x = C_1 S_2 d_3 - S_1 d_2 + d_6 (C_1 C_2 C_4 S_5 + C_1 C_5 S_2 - S_1 S_4 S_5) \quad (2.3.j)$$

$$p_y = S_1 S_2 d_3 + C_1 d_2 + d_6 (C_1 S_4 S_5 + C_2 C_4 S_1 S_5 - C_5 S_1 S_2) \quad (2.3.k)$$

$$p_z = C_2 d_3 + d_6 (C_2 C_5 - C_2 S_4 S_5) \quad (2.3.l)$$

Keterangan:

$C_i = \cos\theta_i$, $S_i = \sin\theta_i$, d_i = Parameter D-H untuk panjang *link* ke- i

Pada Persamaan 2.3 tampak bahwa penyelesaian tiap variabel *joint* harus memenuhi semua persamaan. Metode lain yang dapat digunakan untuk persoalan *inverse kinematic* antara lain pendekatan geometri, dan melalui *jacobian manipulator*.

2.3. Jacobian Manipulator [2]

Pada penjelasan *forward kinematic* diketahui bahwa dari nilai-nilai variabel *joint* yang diketahui, dapat peroleh nilai posisi dan orientasi dari

end-effector manipulator tersebut. Untuk menentukan hubungan antara kecepatan linear dengan kecepatan sudut dari *end-effector*, digunakan suatu fungsi yang disebut *Jacobian*. Melalui *Jacobian* dapat pula diperoleh nilai-nilai variabel *joint* berdasarkan posisi dan orientasi dari *end-effector* yang telah diketahui.

Hubungan antara kecepatan linear dengan kecepatan sudut dari *end-effector* ditunjukkan pada Persamaan 2.4.

$$v = J_{n \times n} \dot{\theta} \quad (2.4)$$

dengan:

v : Vektor kecepatan *end-effector*, $v \in R^n$

$J_{n \times n}$: *Jacobian manipulator*, n adalah jumlah DOF manipulator

$\dot{\theta}$: Kecepatan sudut *joint*, $\dot{\theta} \in R^n$

Untuk Manipulator n -DOF variabel *joint* $\theta_1, \theta_2, \dots, \theta_n$ memiliki matriks transformasi homogen seperti yang ditunjukkan pada Persamaan 2.5.

$$T_n^0(\theta) = \begin{bmatrix} R_{n \times 3}^0(\theta) & p_{n \times 1}^0(\theta) \\ 0 & 1 \end{bmatrix} \quad (2.5)$$

Dari matriks transformasi homogen pada Persamaan 2.5 diperoleh kecepatan linear $\dot{v}_n^0$ dari *end-effector* melalui perpindahan posisi *end-effector* seperti yang ditunjukkan pada Persamaan 2.6.

$$\dot{v}_{n \times 1}^0 = \dot{p}_{n \times 1}^0 \quad (2.6)$$

Untuk kecepatan sudut dari *end-effector* $\omega_{n \times 1}^0$ diperoleh melalui representasi sudut euler pada Persamaan 2.6.

$$\omega_{n \times 1}^0 = \begin{bmatrix} 1 & 0 & s\theta_p \\ 0 & c\theta_r & -c\theta_p s\theta_r \\ 0 & s\theta_r & c\theta_p s\theta_r \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_p \\ \dot{\theta}_y \end{bmatrix} \quad (2.7)$$

Berdasarkan Persamaan 2.4, diperoleh hubungan antara vektor kecepatan *end-effector* dengan kecepatan sudut *joint* melalui Persamaan 2.8 berikut

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{\theta} = \begin{bmatrix} J_v \dot{\theta} \\ J_\omega \dot{\theta} \end{bmatrix} \quad (2.8)$$

Dengan J_v dan J_ω berupa matriks $3 \times n$ merupakan *jacobian* untuk kecepatan linear dan kecepatan sudut dari *end-effector*. Oleh karena semua *joint* berupa *revolute*, maka *jacobian* untuk kecepatan sudut *end-effector* diperoleh melalui Persamaan 2.9

$$J_\omega = [z_0 \ z_1 \ \dots \ z_s] \quad (2.9)$$

dengan $z_{i-1}^0 = R_{i-1}^0 k$ untuk $i = 1, \dots, n$. $k = [0 \ 0 \ 1]^T$ adalah vektor koordinat satuan sumbu z dan $z_0^0 = k$. Untuk *jacobian* kecepatan linear *end-effector* diperoleh melalui Persamaan 2.10

$$J_{vi} = z_{i-1}^0 \times (p_n^0 - p_{i-1}^0) \quad (2.10)$$

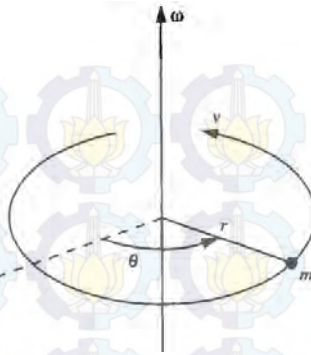
dengan $\times$ adalah notasi *cross product* dan $p_0^0 = [0 \ 0 \ 0]^T$.

2.4. Model Dinamika PUMA 560 [4]

Secara umum pemodelan *manipulator* diturunkan menggunakan dinamika euler-lagrange. Dinamika tersebut berisi perhitungan energi kinetik dan potensial dari tiap *link manipulator*. Perhitungan energi kinetik dan potensial berhubungan dengan gaya, momen inersia dan energi yang mempengaruhi tiap *link manipulator*.

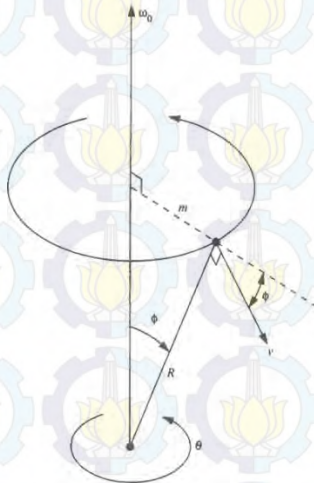
Gaya sentripetal dari suatu massa m yang mengitari titik dengan radius r dan kecepatan sudut ω yang ditunjukkan pada Gambar 2.6 diperoleh berdasarkan Persamaan 2.11.

$$F_{sen} = \frac{mv^2}{r} = m\omega^2 r = m\dot{\theta}^2 r \quad (2.11)$$



Gambar 2.6 Gaya Sentripetal

Diasumsikan suatu lingkaran berputar terhadap pusatnya dengan kecepatan sudut ω_0 seperti yang ditunjukkan pada Gambar 2.7.



Gambar 2.7 Gaya Koriolis

Gaya koriolis pada suatu benda dengan massa m yang bergerak dengan kecepatan v pada permukaan lingkaran tersebut ditunjukkan pada Persamaan 2.12 dengan menggunakan kaidah putaran tangan kanan.

$$F_{kor} = -2m\omega_0 \times v \quad (2.12)$$

Energi kinetik suatu massa m yang bergerak dengan kecepatan linear v ditunjukkan pada Persamaan 2.13.

$$K = \frac{1}{2}mv^2 \quad (2.13)$$

Energi kinetik rotasi $K_{rot} = \frac{1}{2}I\omega^2$ dengan $I = \int_{vol} \rho(r)r^2 dr$, dimana $\rho(r)$ adalah distribusi massa pada radius r dalam suatu volume. Untuk m yang diasumsikan sebagai suatu titik massa maka $I = mr^2$. Oleh karena itu energi kinetik rotasi ditunjukkan pada Persamaan 2.14.

$$K_{rot} = \frac{1}{2}mr^2\dot{\theta}^2 \quad (2.14)$$

Energi potensial dari suatu massa m pada ketinggian h pada suatu medan gravitasi g yang konstan diperoleh melalui Persamaan 2.15.

$$P = mgh \quad (2.15)$$

Berdasarkan gaya dan energi tersebut, persamaan gerak lagrange diperoleh dari selisih antara energi kinetik terhadap energi potensial. Secara umum persamaan lagrange tersebut ditunjukkan pada Persamaan 2.16.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \quad (2.16)$$

Pada Persamaan 2.16, q adalah variabel *joint manipulator* dalam satuan derajat untuk *revolute* atau meter untuk *prismatic* dan τ adalah torsi dengan satuan newton-meter.

Berdasarkan penjabaran energi kinetik dan potensial sebelumnya, diperoleh suatu model dinamika *manipulator* secara umum yang dinotasikan pada Persamaan 2.17.

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) = \tau \cong M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (2.17)$$

dengan:

$$M(q) = [m_{ij}(q)]_{n \times n}, i, j = 1, \dots, n$$

$$m_{ij}(q) = m_{ji}(q)$$

$$V(q, \dot{q}) + F(\dot{q}) + G(q) \cong N(q, \dot{q}) = [n_i(q, \dot{q})]_{n \times 1}, i = 1, \dots, n$$

Model dinamika PUMA 560 direpresentasikan melalui matriks M dan vektor N yang ditunjukkan pada Lampiran B.

2.5. Computed Torque Control (CTC) [4]

Metode kontrol ini merupakan salah satu cara untuk mengatur gerak *manipulator* dalam ruang *joint*. Pada metode ini digunakan *feedback linearization* untuk menghilangkan nonlinearitas dari model *error* variabel *joint*. Torsi yang dibutuhkan agar *manipulator* bergerak menuju nilai *joint* yang diinginkan dihitung menggunakan *inverse dynamic* yang diperoleh dari model dinamika *manipulator*.

Nilai variabel *joint* yang diinginkan $q_d(t)$ ditentukan sebagai referensi *manipulator*. *Error* dari variabel *joint* didefinisikan sebagai selisih nilai variabel *joint* yang diinginkan terhadap nilai variabel *joint* aktual $q(t)$ yang ditunjukkan pada Persamaan 2.18. Untuk memperoleh relasi antara torsi dengan *error* dilakukan turunan sebanyak dua kali dari *error* terhadap waktu t .

$$e_i(t) = q_{di}(t) - q_i(t), i = 1, \dots, n \quad (2.18)$$

$$\ddot{e}_i(t) = \ddot{q}_{di}(t) - \ddot{q}_i(t), i = 1, \dots, n \quad (2.19)$$

Pada Persamaan 2.19, $\ddot{q}_i(t)$ disubstitusikan berdasarkan model dinamika pada Persamaan 2.17, sehingga diperoleh hubungan antara torsi dan *error* yang ditunjukkan pada Persamaan 2.20.

$$\ddot{e}_i(t) = \ddot{q}_{di}(t) + M^{-1}(N - \tau_i), i = 1, \dots, n \quad (2.20)$$

Didefinisikan suatu fungsi masukan kontrol yaitu $u_i = \ddot{q}_{di}(t) + M^{-1}(N - \tau_i)$ dan *state variable* $x = [e_i \dot{e}_i]^T$, diperoleh suatu model linear dari *error* variabel *joint* yang ditunjukkan pada Persamaan 2.21.

$$\frac{d}{dt} \begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (2.21)$$

Sistem linear dari Persamaan 2.21 disebut bentuk *Brunovsky canonical* dengan nilai *error* dan *delta error* sebagai *state variable* untuk tiap *joint* dari *manipulator*. Metode *feedback linearization* dilakukan pada pendefinisian fungsi masukan kontrol u . Torsi masukan untuk *manipulator* ditunjukkan pada Persamaan 2.22.

$$\tau_i = M(\ddot{q}_{di} - u_i) + N \quad (2.22)$$

Masukan kontrol dapat dilakukan dengan metode kontrol yang dapat membuat *error* dan *delta error* menjadi nol.

2.6. Linear Quadratic Regulator (LQR) [4]

Metode *linear quadratic regulator* merupakan salah satu metode kontrol optimal yang digunakan untuk membuat *state variable* yang telah didefinisikan pada Subbab 2.5 menuju ke nol.

Suatu sistem LTI dalam bentuk *state space* ditunjukkan pada Persamaan 2.23.

$$\dot{x} = Ax + Bu \quad (2.23)$$

dengan *state feedback gain* K dipilih untuk memperoleh *closed-loop* stabil yang ditunjukkan pada Persamaan 2.24.

$$\dot{x} = (A - BK)x \quad (2.24)$$

Didefinisikan suatu indeks performansi PI yang ditunjukkan pada Persamaan 2.25.

$$PI = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.25)$$

Q adalah matriks semidefinit positif pembobot *state* dan R adalah matriks pembobot kontrol. Matriks Q dipilih dengan cara membentuk matriks diagonal dengan nilai elemen diagonal lebih dari nol. Nilai R akan mempengaruhi besarnya sinyal kontrol yang dikeluarkan. Untuk kebutuhan sistem dengan energi minimal, maka nilai R yang dipilih dibuat besar.

Nilai K diperoleh dengan menyelesaikan aljabar Riccati ($A^T P + PA + Q - PBR^{-1}B^T P = 0$) dengan $K = R^{-1}B^T P$.

2.7. Perencanaan Trayektori

Perencanaan trayektori dilakukan dengan menentukan titik-titik pada koordinat kartesian. Pergerakan dari titik ke titik selanjutnya dibuat dengan menentukan posisi awal, posisi tujuan, kecepatan awal, kecepatan tujuan, percepatan awal, percepatan tujuan, waktu awal, dan waktu tujuan. Dari beberapa ketentuan tersebut dihitung persamaan polinomial yang disebut *quintic cubical polynomial* yang ditunjukkan pada Persamaan 2.26 [2].

$$p_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \quad (2.26.a)$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \quad (2.26.b)$$

$$ac_0 = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \quad (2.26.c)$$

$$p_d = a_0 + a_1 t_d + a_2 t_d^2 + a_3 t_d^3 + a_4 t_d^4 + a_5 t_d^5 \quad (2.26.d)$$

$$v_d = a_1 + 2a_2 t_d + 3a_3 t_d^2 + 4a_4 t_d^3 + 5a_5 t_d^4 \quad (2.26.e)$$

$$ac_d = 2a_2 + 6a_3 t_d + 12a_4 t_d^2 + 20a_5 t_d^3 \quad (2.26.f)$$

Dari Persamaan 2.26 diubah ke bentuk matriks menjadi Persamaan 2.27.

$$Da = b$$

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_d & t_d^2 & t_d^3 & t_d^4 & t_d^5 \\ 0 & 1 & 2t_d & 3t_d^2 & 4t_d^3 & 5t_d^4 \\ 0 & 0 & 2 & 6t_d & 12t_d^2 & 20t_d^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} p_0 \\ v_0 \\ ac_0 \\ p_d \\ v_d \\ ac_d \end{bmatrix} \quad (2.27)$$

Koefisien a_0 a_1 a_2 a_3 a_4 a_5 pada matriks a dihitung dengan cara $a = D^{-1}b$.

2.8. Spesifikasi PUMA 560

Spesifikasi PUMA 560 digunakan untuk perancangan simulasi sistem. Spesifikasi terdiri dari batas gerak *joint*, dan torsi maksimum yang mampu diterapkan pada PUMA 560. Batas gerak *joint* PUMA 560 ditunjukkan pada Tabel 2.1 dan torsi maksimum PUMA 560 ditunjukkan pada Tabel 2.2. Bentuk fisik dari PUMA 560 ditunjukkan pada Gambar 2.8.

Tabel 2.1 Batas Gerak *Joint* PUMA 560 [5]

<i>Joint</i>	<i>Range</i> (°)
1	-160 – +160
2	-245 – +45
3	-45 – +225
4	-110 – +170
5	-100 – +100
6	-266 – +266

Tabel 2.2 Torsi Maksimum PUMA 560 [6]

<i>Joint</i>	Torsi Maksimum (N-m)
1	97,6
2	186,4
3	89,4
4	24,2
5	20,1
6	21,3

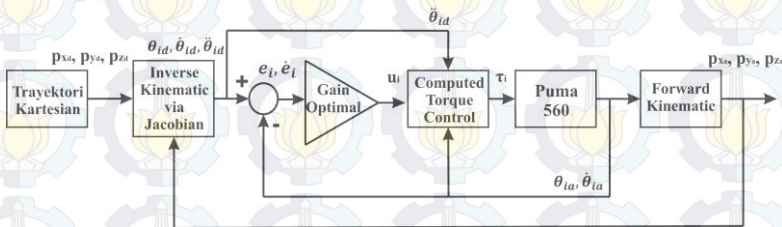


BAB III PERANCANGAN SISTEM

Pada bab ini dibahas perancangan sistem secara lengkap meliputi model kinematika dan dinamika dari *manipulator* PUMA 560, *Jacobian manipulator*, trayektori koordinat kartesian, perancangan metode *optimal computed torque control*, dan perancangan *software* yang akan digunakan untuk simulasi.

3.1. Perancangan Sistem Keseluruhan

Perancangan sistem pengaturan gerak *manipulator* secara keseluruhan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem Keseluruhan

Secara umum pergerakan *manipulator* dikontrol menggunakan *optimal computed torque control* untuk mengikuti referensi yang diberikan yaitu posisi dari *end-effector* (p_{xd}, p_{yd}, p_{zd}). Posisi yang diberikan diubah menjadi sudut *joint* menggunakan *inverse kinematic*. Nilai sudut *joint* diturunkan terhadap waktu sebanyak dua kali sehingga diperoleh percepatan sudut *joint* yang digunakan untuk perhitungan *computed torque control*. Keluaran dari perhitungan tersebut berupa torsi untuk tiap *joint* dimasukkan ke *manipulator* PUMA 560 sehingga terjadi perubahan nilai *joint* sampai nilai *joint* aktual sama dengan nilai *joint* hasil perhitungan *inverse kinematic*. Posisi aktual dari *manipulator* diperoleh melalui perhitungan *forward kinematic* berdasarkan nilai *joint* aktual. Hasil perhitungan berupa transformasi homogen yang berisi matriks rotasi dan posisi aktual dari *end-effector* digunakan untuk perhitungan *Jacobian manipulator*.

3.2. Perancangan *Forward Kinematic* PUMA 560

Perhitungan *forward kinematic* menggunakan parameter D-H standar hasil penelitian R.P Paul yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Parameter D-H Standar PUMA 560 [6]

Joint ke- <i>i</i>	θ_i (0)	α_i (0)	a_i (mm)	d_i (mm)
1	*	90	0	0
2	*	0	431,8	0
3	*	-90	19,1	125,4
4	*	90	0	431,8
5	*	-90	0	0
6	*	0	0	0

Keterangan: * Menandakan Nilai Parameter Berupa Variabel Sesuai Spesifikasi PUMA 560

Penurunan matriks transformasi yang digunakan untuk menghitung posisi dan orientasi dari *end-effector* yaitu $T_6^0 = A_1 A_2 A_3 A_4 A_5 A_6$ dengan nilai-nilai A_i untuk $i = 1, 2, \dots, 6$ ditunjukkan pada Persamaan 3.1 hingga Persamaan 3.6.

$$A_1 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$A_2 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 431,8c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & 431,8s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$A_3 = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & 19,1c\theta_3 \\ s\theta_3 & 0 & c\theta_3 & 19,1s\theta_3 \\ 0 & -1 & 0 & 125,4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$A_4 = \begin{bmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & 431,8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$A_5 = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$A_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Perancangan *forward kinematic* pada software Matlab 2010a ditunjukkan oleh *flowchart* pada Gambar 3.2. *Listing program* ditunjukkan pada Lampiran C.

3.3. Perancangan *Inverse Kinematic* PUMA 560

Perhitungan *inverse kinematic* dilakukan menghitung matriks *jacobian* manipulator dengan masukan berupa posisi *end-effector* dari manipulator. Orientasi dari *end-effector* dibuat tetap dengan matriks rotasi $R_3 = I_3$. Perubahan sudut *joint* diperoleh dengan melakukan perkalian matriks *jacobian* “ J ” dengan vektor $v_{6 \times 1} = [v_n \ \omega_n]^T$ dimana:

$$v_{n3 \times 1} = [v_x \ v_y \ v_z]^T \quad (3.7)$$

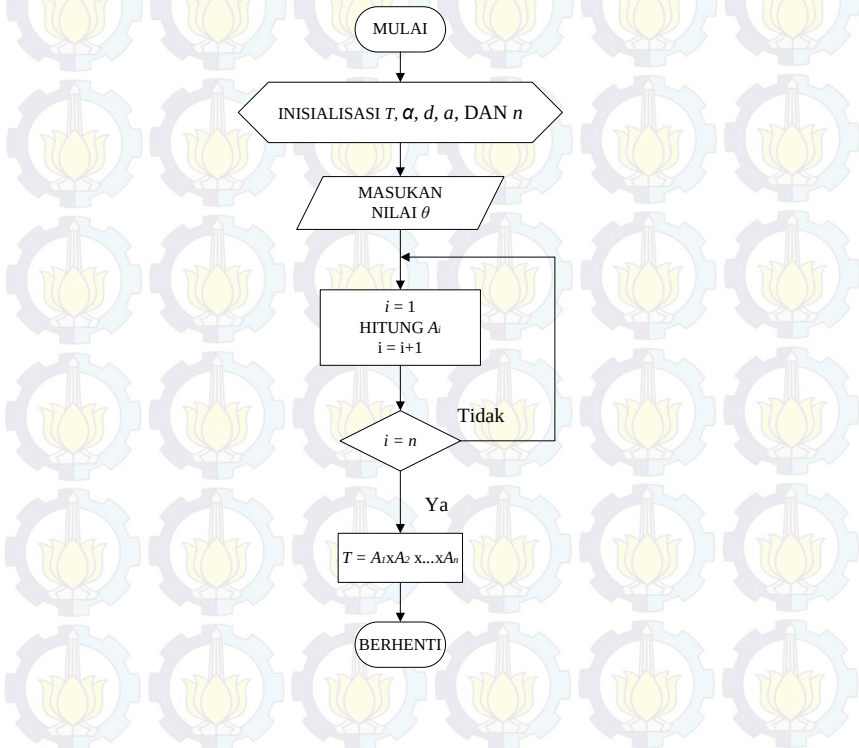
$$\omega_{n3 \times 1} = \begin{bmatrix} 1 & 0 & s\theta_p \\ 0 & c\theta_r & -c\theta_p s\theta_r \\ 0 & s\theta_r & c\theta_p s\theta_r \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_p \\ \dot{\theta}_y \end{bmatrix} \quad (3.8)$$

Dengan matriks *jacobian* manipulator yang ditunjukkan pada Persamaan 3.9 dapat dihitung nilai perubahan sudut *joint* berdasarkan perpindahan titik pada koordinat kartesian dengan menggunakan Persamaan 3.10.

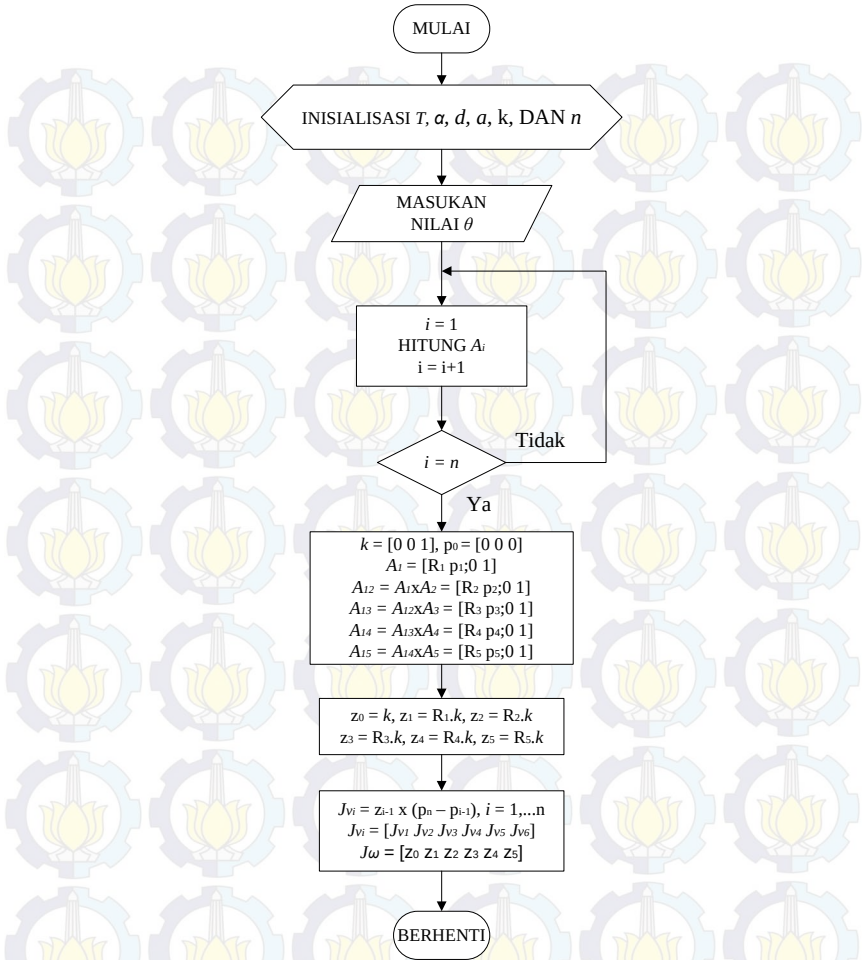
$$J = \begin{bmatrix} J_{v0} & J_{v1} & J_{v2} & J_{v3} & J_{v4} & J_{v5} \\ J_{\omega1} & J_{\omega2} & J_{\omega3} & J_{\omega4} & J_{\omega4} & J_{\omega5} \end{bmatrix} \quad (3.9)$$

$$\dot{\theta} = J^{-1}v = J^{-1} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_{n \times 1} \end{bmatrix} \quad (3.10)$$

Perancangan *jacobian* pada Matlab ditunjukkan melalui *flowchart* pada Gambar 3.3. *Listing program* ditunjukkan pada Lampiran C.

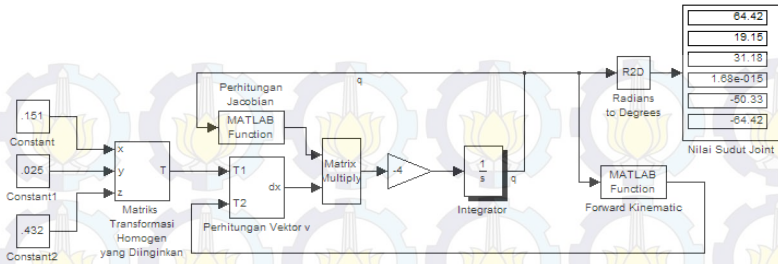


Gambar 3.2 *Flowchart* Perancangan *Forward Kinematic* PUMA 560 Menggunakan Matlab



Gambar 3.3 Perancangan *Jacobian* PUMA 560

Perubahan sudut *joint* tersebut diintegrasikan untuk memperoleh nilai sudut *joint* yang akan dibentuk untuk mencapai posisi *end-effector* yang diinginkan. Perancangan keseluruhan *inverse kinematic* menggunakan *jacobian* pada Matlab ditunjukkan pada Gambar 3.4.



Gambar 3.4 Perancangan *Inverse Kinematic* PUMA 560

3.4. Perancangan Model Dinamika PUMA 560

Pemodelan dinamika dari PUMA 560 terdiri dari matriks $M_{6 \times 6}$ yang berisi konstanta inersia dan matriks $N_{6 \times 1} \equiv V + F + G$ dengan V berupa vektor Koriolis, F berupa vektor friksi viskos dan dinamik, serta G berupa matriks gravitasi. Berdasarkan Persamaan 2.16, diperoleh elemen-elemen matriks M dan N ditunjukkan pada Persamaan 3.9 dan 3.10 [4].

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 & 0 & 0 \\ m_{12} & m_{22} & m_{23} & 0 & 0 & 0 \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} & 0 \\ 0 & 0 & m_{34} & m_{44} & 0 & 0 \\ 0 & 0 & m_{35} & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} \quad (3.9)$$

$$N = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ 0 \end{bmatrix} \quad (3.10)$$

Beberapa konstanta yang digunakan sebagai penyusun model dinamik PUMA 560 ditunjukkan pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.2 Konstanta Inersia PUMA 560 [7]

$I_1 = 1,43 \pm 0,05$	$I_{13} = -3,79 \times 10^{-3} \pm 0,9 \times 10^{-3}$
$I_2 = 1,75 \pm 0,05$	$I_{14} = 1,64 \times 10^{-3} \pm 0,07 \times 10^{-3}$
$I_3 = 1,38 \pm 0,05$	$I_{15} = 1,25 \times 10^{-3} \pm 0,3 \times 10^{-3}$
$I_4 = 6,9 \times 10^{-1} \pm 0,2 \times 10^{-1}$	$I_{16} = 1,24 \times 10^{-3} \pm 0,3 \times 10^{-3}$
$I_5 = 3,72 \times 10^{-1} \pm 0,31 \times 10^{-1}$	$I_{17} = 6,42 \times 10^{-4} \pm 3 \times 10^{-4}$
$I_6 = 3,33 \times 10^{-1} \pm 0,16 \times 10^{-1}$	$I_{18} = 4,31 \times 10^{-4} \pm 1,3 \times 10^{-4}$
$I_7 = 2,98 \times 10^{-1} \pm 0,29 \times 10^{-1}$	$I_{19} = 3 \times 10^{-4} \pm 14 \times 10^{-4}$
$I_8 = -1,34 \times 10^{-1} \pm 0,14 \times 10^{-1}$	$I_{20} = -2,02 \times 10^{-4} \pm 8 \times 10^{-4}$
$I_9 = 2,38 \times 10^{-2} \pm 1,2 \times 10^{-2}$	$I_{21} = -1 \times 10^{-4} \pm 6 \times 10^{-4}$
$I_{10} = -2,13 \times 10^{-2} \pm 0,22 \times 10^{-2}$	$I_{22} = -5,8 \times 10^{-5} \pm 1,5 \times 10^{-5}$
$I_{11} = -1,42 \times 10^{-2} \pm 0,7 \times 10^{-2}$	$I_{23} = 4 \times 10^{-5} \pm 2 \times 10^{-5}$
$I_{12} = -1,1 \times 10^{-2} \pm 0,11 \times 10^{-2}$	
$I_{m1} = 1,14 \pm 0,27$	$I_{m4} = 2 \times 10^{-1} \pm 0,16 \times 10^{-1}$
$I_{m2} = 4,71 \pm 0,54$	$I_{m5} = 1,79 \times 10^{-1} \pm 0,14 \times 10^{-1}$
$I_{m3} = 8,27 \times 10^{-1} \pm 0,93 \times 10^{-1}$	$I_{m6} = 1,93 \times 10^{-1} \pm 0,16 \times 10^{-1}$

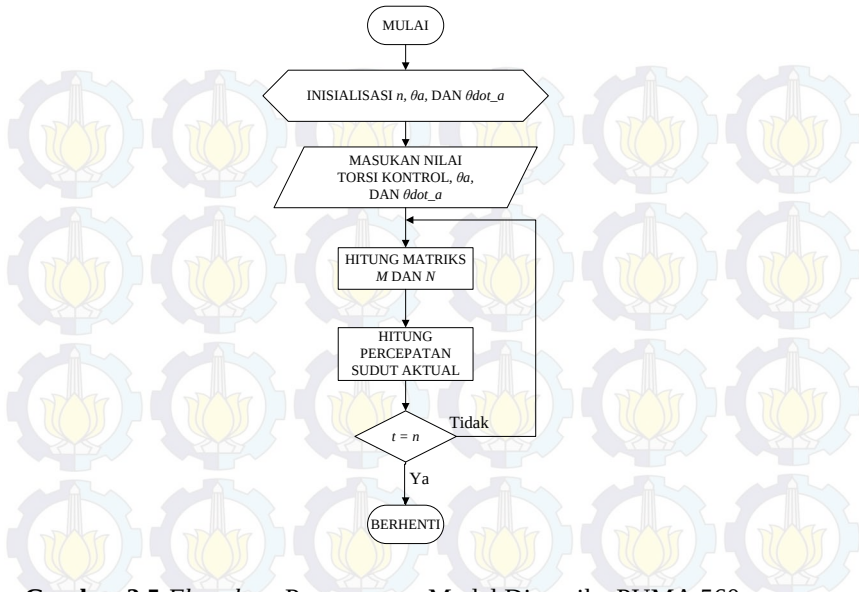
Keterangan: Satuan dari konstanta inersia adalah kg/m^2 .

Tabel 3.3 Konstanta Gravitasi PUMA 560 [7]

$g_1 = -37,2 \pm 0,05$	$g_4 = 2,49 \times 10^{-1} \pm 0,25 \times 10^{-1}$
$g_2 = -8,44 \pm 0,2$	$g_5 = -2,82 \times 10^{-2} \pm 0,56 \times 10^{-2}$
$g_3 = 1,02 \pm 0,5$	

Keterangan: Satuan dari konstanta gravitasi adalah newton-meter.

Berdasarkan konstanta tersebut, dibentuk nilai-nilai untuk elemen matriks pada model dinamika *manipulator*. Nilai dari tiap elemen ditunjukkan pada Lampiran B. Pada simulasi semua elemen didefinisikan dengan masukan antara lain torsi kontrol, posisi sudut *joint* aktual, dan kecepatan sudut *joint* aktual. Perancangan model dinamika PUMA 560 melalui *flowchart* pada Gambar 3.5. *Listing program* ditunjukkan pada Lampiran C.



Gambar 3.5 Flowchart Perancangan Model Dinamika PUMA 560

Keluaran dari model dinamika berupa percepatan aktual yang kemudian diintegrasikan dua kali untuk memperoleh posisi dan kecepatan *joint*. Nilai posisi *joint* dibatasi sesuai dengan spesifikasi sudut maksimum pergerakan tiap *joint*.

3.5. Perancangan Kontroler *Linear Quadratic Regulator* (LQR)

Pada perancangan ini ditentukan matriks Q sebagai pembobot *state variable* dari sistem yang akan dikontrol dan konstanta R untuk pembobot sinyal kontrol. Matriks A dan B diperoleh dari model manipulator setelah dilakukan linearisasi pada Persamaan 2.21. Matriks A dan B dibuat untuk masing-masing *joint*. Matriks P , solusi dari aljabar Riccati dihitung menggunakan fungsi matlab “[K P e] = lqr (A , B , Q , R , N)” dimana $K = [K_1 K_2]$ digunakan sebagai *gain* untuk *error* dan *delta error* pada tiap *joint* dan nilai $N = 0$. Untuk perancangan didefinisikan A , B , Q , dan R sebagai berikut:

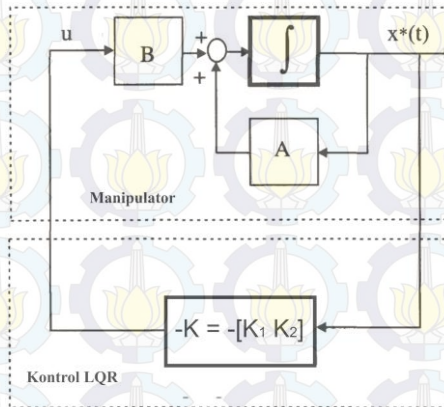
$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (3.11)$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.12)$$

$$Q = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix}, \text{ dengan } Q_{11}, Q_{22} > 0 \quad (3.13)$$

$$R = c, \text{ dengan } c > 0 \quad (3.14)$$

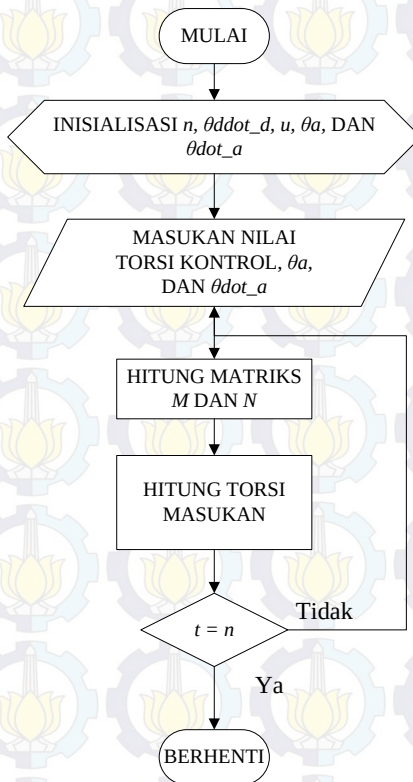
Dari Persamaan 3.11 hingga Persamaan 3.14, dihitung matriks solusi aljabar Ricatti dengan keluaran yaitu *gain* K , solusi Riccati P , dan nilai eigen e dari sistem baru. Nilai dari *state variable* $x = [e_i \dot{e}_i]$ akan semakin cepat menuju ke nol jika elemen matriks pembobot Q_{11} dan Q_{22} semakin besar. Pada simulasi ini nilai R dibuat tetap dan pembobot Q diubah-ubah untuk memperoleh respons yang diinginkan. Diagram blok dari LQR ditunjukkan pada Gambar 3.6. Hasil perkalian dari *state variable* dan gain K didefinisikan sebagai u dan digunakan sebagai masukan pada perhitungan *computed torque control*.



Gambar 3.6 Diagram Blok Kontrol LQR

3.6. Perancangan Kontroler *Computed Torque Control* (CTC)

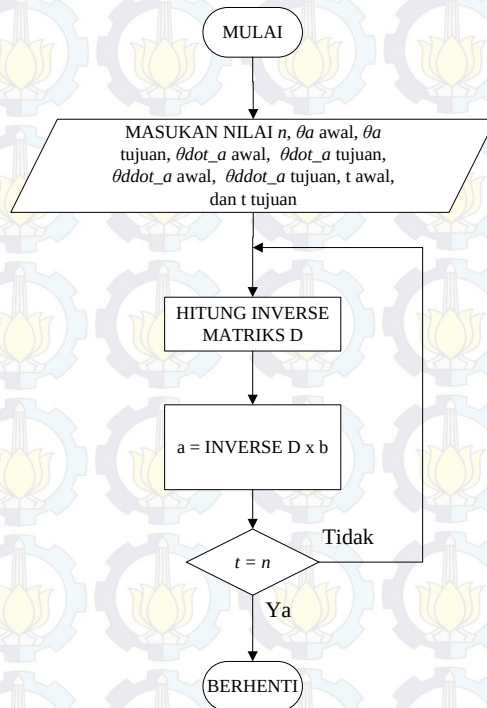
Pada bab 2 Subbab 2.5 telah dijelaskan perhitungan kontrol ini menggunakan model dinamika dari PUMA 560 untuk menghitung torsi yang dibutuhkan untuk membentuk nilai sudut *joint* yang diinginkan. Perancangan kontroler *computed torque control* melalui flowchart pada Gambar 3.7. Listing program ditunjukkan pada Lampiran C.



Gambar 3.7 Flowchart Perancangan Kontroler CTC

3.7. Perancangan Trayektori Kartesian

Pada perancangan ini dilakukan penentuan jalur berdasarkan titik-titik yang berada dalam ruang kartesian. Jalur yang dibentuk berbentuk lingkaran dengan radius 10 hingga 50 cm. Perancangan trayektori lingkaran pada Matlab ditunjukkan melalui *flowchart* pada Gambar 3.8. *Listing program* ditunjukkan pada Lampiran C.



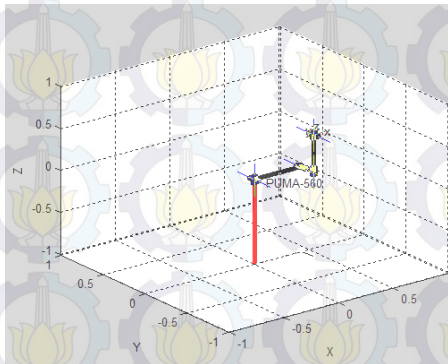
Gambar 3.8 Perancangan Trayektori Kartesian

Tujuan dari perancangan untuk penugasan *manipulator* dengan lintasan berbentuk lingkaran yaitu untuk mengetahui kemampuan PUMA 560 dalam menyelesaikan tugas dengan trayektori yang relatif sulit. Koordinat dari trayektori tersebut diubah ke bentuk sudut *joint* melalui perhitungan *inverse kinematic* yang telah dirancang pada Subbab 3.3.

3.8. Perancangan Sistem Keseluruhan Menggunakan Matlab dengan *Toolbox Robot Peter I Corke* [6]

Sistem secara keseluruhan disusun berdasarkan beberapa subsistem yang telah dijelaskan sebelumnya hingga sesuai dengan diagram blok secara keseluruhan yang telah dijelaskan pada Subbab 3.1. Diagram blok simulasi sistem secara keseluruhan ditunjukkan pada Lampiran A.

Dalam simulasi tersebut hasil dari sistem ditunjukkan dalam suatu animasi gerak manipulator. Animasi tersebut dibentuk menggunakan perintah *plot* dari Matlab berdasarkan parameter D-H PUMA 560. Bentuk animasi manipulator tersebut ditunjukkan pada Gambar 3.9.



Gambar 3.9 Animasi Manipulator pada *Software Matlab*

BAB IV

PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini dibahas hasil pengujian simulasi sistem secara lengkap meliputi respons *joint* terhadap dua sinyal uji, *inverse kinematic* dari PUMA 560, dan pengujian akurasi dari posisi *end-effector* melalui penugasan tertentu.

4.1. Pengujian dan Analisis Respons *Joint*

Pada pengujian ini dipilih beberapa sinyal uji antara lain *step* dan *cubical polynomial*. Untuk sinyal uji *step*, diberikan pada *manipulator* yang telah dikontrol menggunakan CTC dan *Optimal* CTC. Untuk metode *Optimal* CTC digunakan nilai pembobot Q dan R yang sesuai untuk mendapatkan respons dengan *settling time* paling kecil.

4.1.1. Pengujian dan Analisis Menggunakan Sinyal *Step*

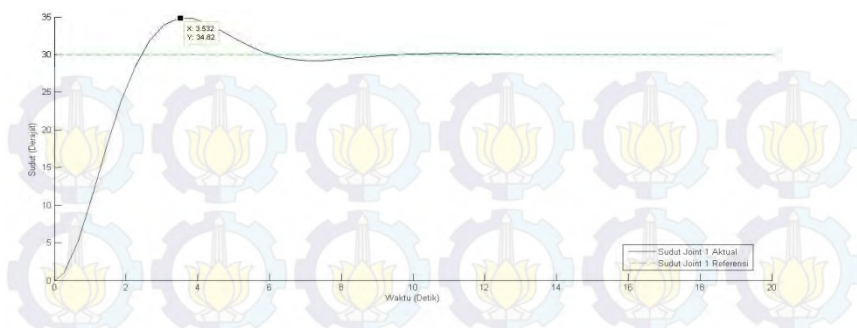
Sinyal *step* yang diberikan berupa enam nilai sudut referensi untuk masing-masing *joint* yang ditunjukkan pada Persamaan 4.1. Untuk pengujian sinyal *step* dipilih kondisi awal (IC) dari sudut *joint* yang ditunjukkan pada Persamaan 4.2. Pengujian dilakukan dengan membandingkan sistem yang dikontrol menggunakan CTC tanpa sinyal kontrol optimal terhadap CTC dengan sinyal kontrol *optimal*. Kontroler CTC tanpa sinyal kontrol *optimal* ditunjukkan pada Persamaan 4.3, sedangkan CTC dengan sinyal kontrol *optimal* ditunjukkan pada Persamaan 4.4. Nilai K_1 dan K_2 pada Persamaan 4.4 dihitung menggunakan metode LQR. Pemilihan matriks pembobot Q dan R untuk perhitungan metode LQR dilakukan berdasarkan *settling* yang paling kecil dari respons sistem. Rata-rata RMSE dan *settling time* dari respons tersebut ditunjukkan pada Tabel 4.1. Respons dari tiap *joint* yang dikontrol menggunakan CTC ditunjukkan pada Gambar 4.1 hingga Gambar 4.6.

$$\theta_d = [30^\circ \ 30^\circ \ 30^\circ \ 45^\circ \ 100^\circ \ 166^\circ]^T \quad (4.1)$$

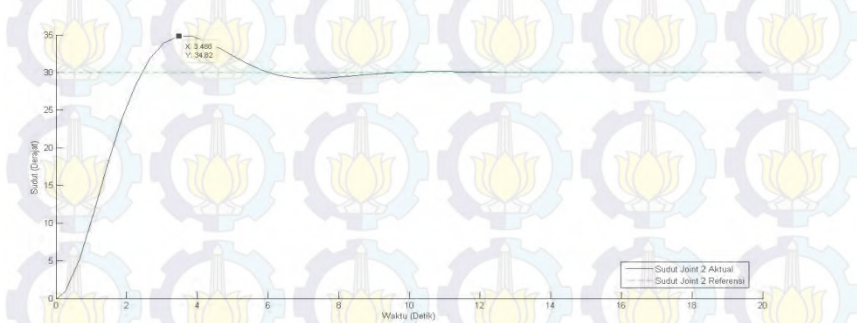
$$IC = [0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ]^T \quad (4.2)$$

$$u = -(e_i + \dot{e}_i) \quad (4.3)$$

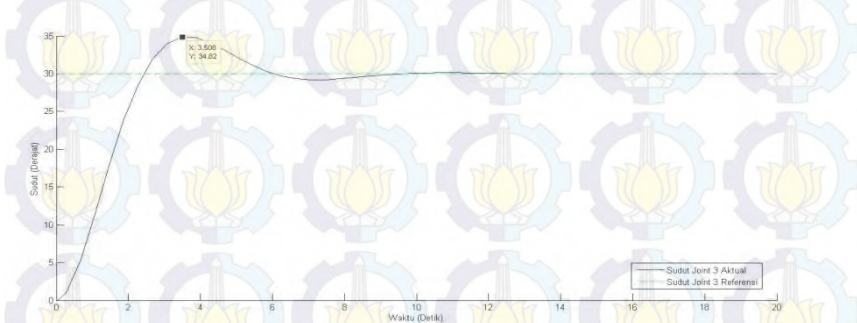
$$u = -(K_1 e_i + K_2 \dot{e}_i) \quad (4.3)$$



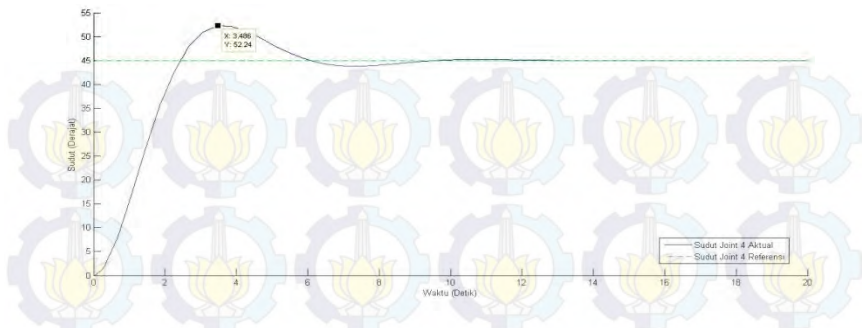
Gambar 4.1 Respons *Joint 1* CTC Terhadap Sinyal Uji Step



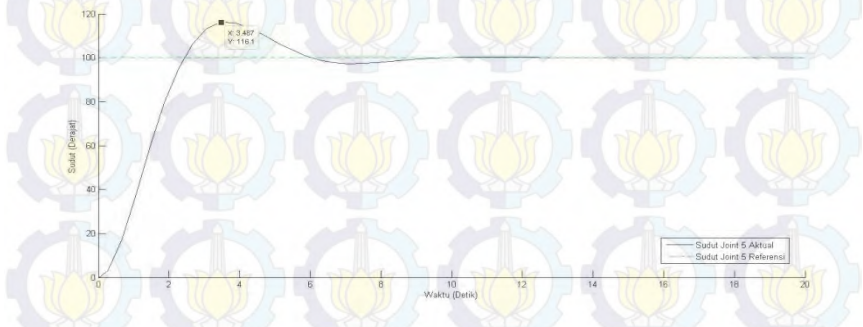
Gambar 4.2 Respons *Joint 2* CTC Terhadap Sinyal Uji Step



Gambar 4.3 Respons *Joint 3* CTC Terhadap Sinyal Uji Step



Gambar 4.4 Respons *Joint 4* CTC Terhadap Sinyal Uji Step



Gambar 4.5 Respons *Joint 5* CTC Terhadap Sinyal Uji Step



Gambar 4.6 Respons *Joint 6* CTC Terhadap Sinyal Uji Step

Tabel 4.1 Rata-rata Nilai RMSE dan *Settling Time* Respons Sistem

Sistem	1	2	3	4
Rata-rata RMSE (°)	6,001	5,884	6,101	7,567
Rata-rata Ts(Detik)	0,336	0,375	0,472	0,564

dengan:

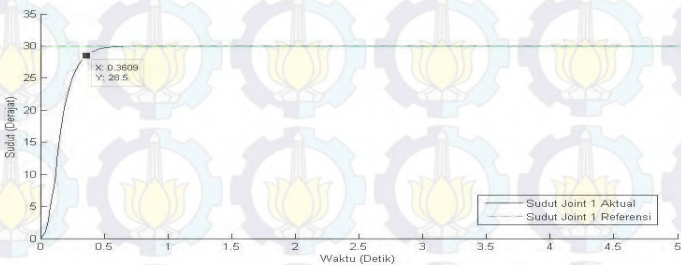
Sistem 1: $Q = [1200000 \ 0; 0 \ 10000]$; $R = 2$; $K = [774,59 \ 80,9271]$

Sistem 2: $Q = [900000 \ 0; 0 \ 10000]$; $R = 2$; $K = [670,82 \ 79,6344]$

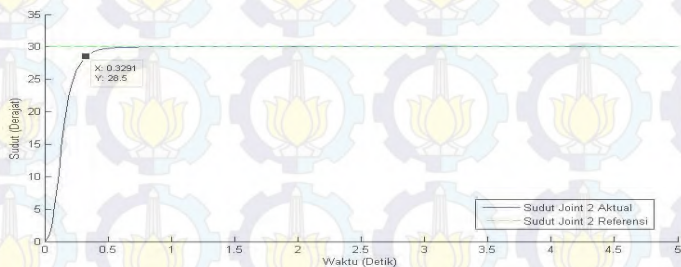
Sistem 3: $Q = [300000 \ 0; 0 \ 9000]$; $R = 2$; $K = [387,298 \ 72,6264]$

Sistem 4: $Q = [100000 \ 0; 0 \ 3000]$; $R = 2$; $K = [223,6068 \ 44,1272]$

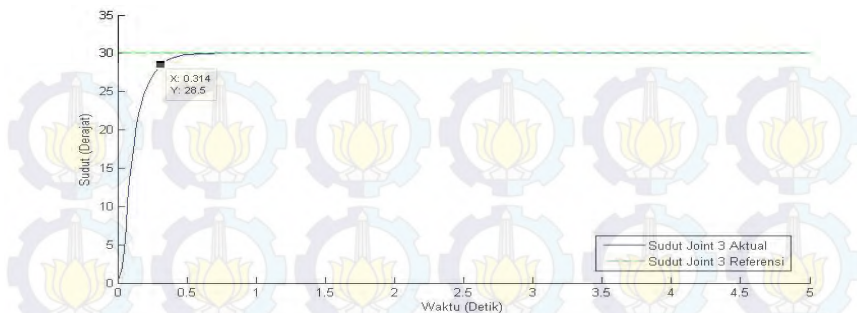
Berdasarkan kondisi tersebut, dipilih sistem 1 dengan *settling time* yang relatif kecil dibandingkan sistem lainnya. Dari Sistem 1, diperoleh nilai *gain* $K = [774,59 \ 80,9271]$. Respons dari tiap *joint* yang dikontrol menggunakan CTC dengan sinyal kontrol *optimal* ditunjukkan pada Gambar 4.7 hingga Gambar 4.12.



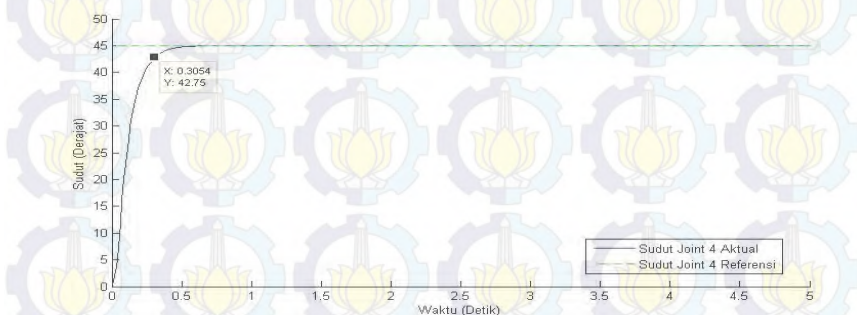
Gambar 4.7 Respons *Joint 1* *Optimal* CTC Terhadap Sinyal Uji Step



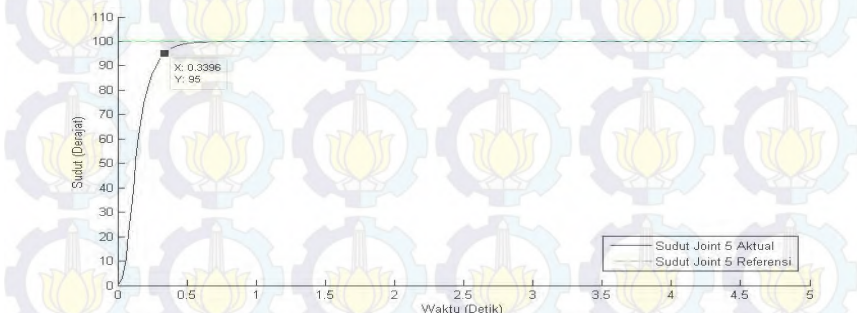
Gambar 4.8 Respons *Joint 2* *Optimal* CTC Terhadap Sinyal Uji Step



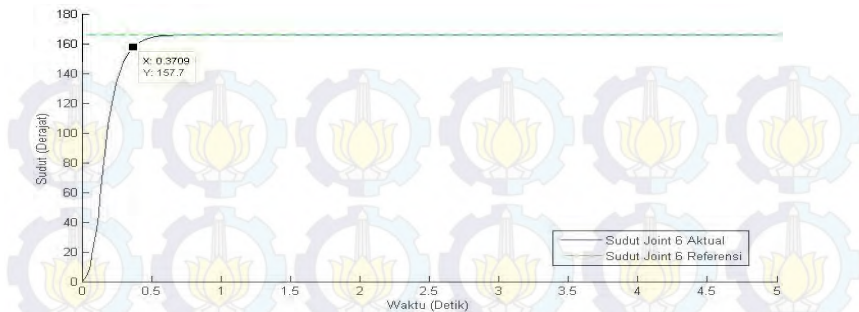
Gambar 4.9 Respons *Joint 3 Optimal CTC* Terhadap Sinyal Uji Step



Gambar 4.10 Respons *Joint 4 Optimal CTC* Terhadap Sinyal Uji Step



Gambar 4.11 Respons *Joint 5 Optimal CTC* Terhadap Sinyal Uji Step



Gambar 4.12 Respons *Joint 6 Optimal CTC* Terhadap Sinyal Uji *Step*

Berdasarkan pengujian menggunakan sinyal *step*, diperoleh *settling time* di sekitar $< 0,4$ detik dan tanpa *overshoot* untuk tiap *joint*. Pengujian dengan nilai *joint* statis menunjukkan bahwa *manipulator* mampu mengikuti referensi *joint* yang diberikan dengan toleransi perubahan nilai sudut referensi kurang maksimal 0,4 detik. Perbandingan respons dari CTC terhadap *Optimal CTC* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Perbandingan Respons CTC Terhadap *Optimal CTC*

	Joint	RMSE ($^{\circ}$)		Joint	RMSE ($^{\circ}$)
	1	3,9392		1	2,6941
CTC	2	3,9392	Optimal CTC	2	2,6941
	3	3,9392		3	2,6941
	4	5,9088		4	4,0411
	5	13,1308		5	8,9804
	6	21,7969		6	14,9072

Berdasarkan Tabel 4.2, diketahui *optimal CTC* mampu memperbaiki respons sudut dari tiap *joint* maksimal sebesar $6,8897^{\circ}$.

4.1.2. Pengujian dan Analisis Menggunakan Sinyal *Cubical Polynomial*

Sinyal *cubical polynomial* digunakan untuk mengetahui respons *joint* terhadap perubahan referensi dalam rentang waktu tertentu. Sinyal *cubical polynomial* dibuat dengan ketentuan yang telah dijelaskan pada Subbab 2.7. Nilai *joint* referensi $\theta_d = [90^{\circ}40^{\circ} - 25^{\circ} - 100^{\circ}35^{\circ}150^{\circ}]^T$ dan

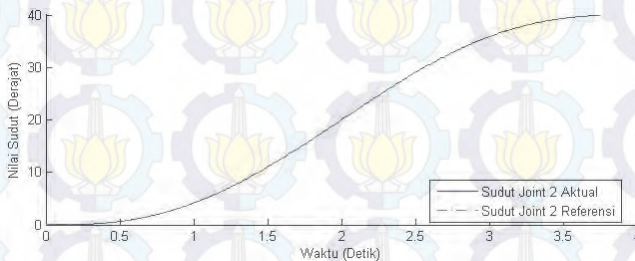
kondisi awal dari sistem dibuat sama dengan pengujian menggunakan sinyal *step* seperti yang ditunjukkan pada Persamaan 4.1 dan Persamaan 4.2. Nilai matriks pembobot *state* $Q = \text{diagonal}(1200000, 10000)$ dan pembobot sinyal kontrol $R = 2$. Respons dari tiap *joint* terhadap kondisi yang telah ditentukan sebelumnya ditunjukkan pada Gambar 4.13 hingga Gambar 4.18. Berdasarkan pengujian yang dilakukan menggunakan sinyal uji *cubical polynomial*, disimpulkan bahwa respons sistem dengan kontroler *optimal* CTC mampu mengikuti perubahan referensi dengan nilai RMSE dari tiap respons *joint* yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Nilai RMSE dari Respons Tiap *Joint* Terhadap Sinyal *Cubical Polynomial*

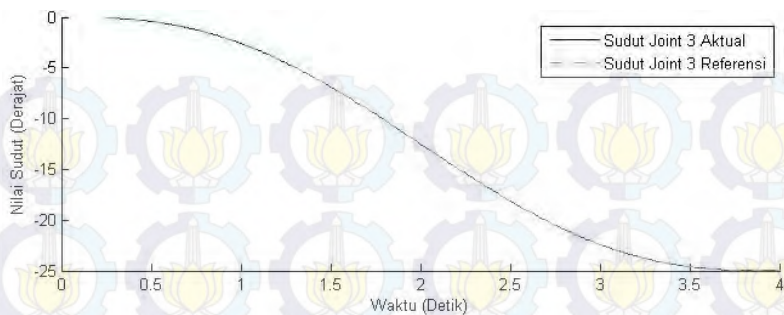
<i>Joint</i>	1	2	3	4	5	6
RMSE ($^{\circ}$)	0,0108	0,0048	0,003	0,012	0,0042	0,0192



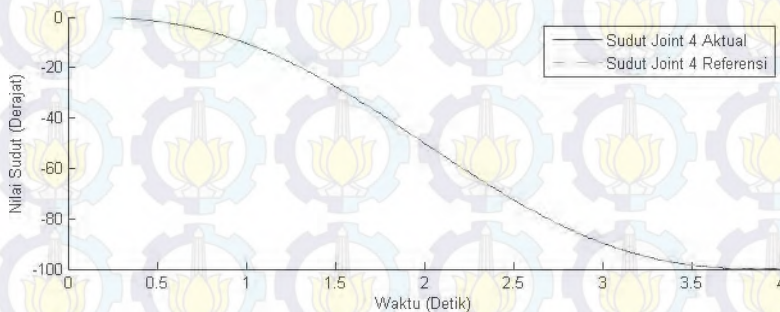
Gambar 4.13 Respons *Joint* 1 Terhadap Sinyal Uji *Cubical Polynomial*



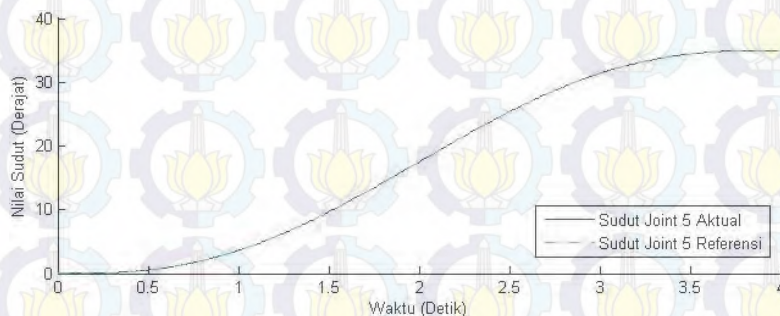
Gambar 4.14 Respons *Joint* 2 Terhadap Sinyal Uji *Cubical Polynomial*



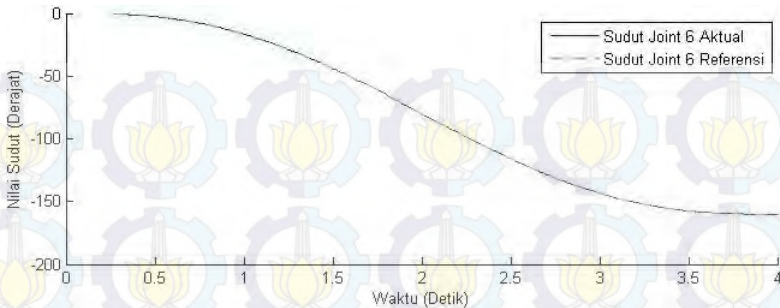
Gambar 4.15 Respons *Joint 3* Terhadap Sinyal Uji *Cubical Polynomial*



Gambar 4.16 Respons *Joint 4* Terhadap Sinyal Uji *Cubical Polynomial*



Gambar 4.17 Respons *Joint 5* Terhadap Sinyal Uji *Cubical Polynomial*



Gambar 4.18 Respons *Joint 6* Terhadap Sinyal Uji *Cubical Polynomial*

4.2. Pengujian dan Analisis *Inverse Kinematic* PUMA 560 Menggunakan *Jacobian Manipulator*

Pengujian *inverse kinematic* dilakukan dengan diberikan masukan posisi *end-effector* dalam ruang kartesian dan menghasilkan keluaran berupa nilai sudut *joint*. Hasil dari pengujian tersebut ditunjukkan pada Tabel 4.4. Untuk validitas hasil dari nilai sudut *joint*, dilakukan perhitungan *forward kinematic* berdasarkan keluaran nilai sudut *joint* sehingga diperoleh posisi dari *end-effector*. Hasil dari pengujian tersebut ditunjukkan pada Tabel 4.5.

Tabel 4.4 Hasil Perhitungan *Inverse Kinematic*

No.	Masukan (cm)			Keluaran ($^{\circ}$)					
	p_x	p_y	p_z	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	0	-50	1	-75,48	-54,78	24,38	-180	-30,4	255,5
2	12,36	-40	1	-55,4	-61,07	37,44	0	23,63	55,4
3	32,36	-20	1	-12,47	-63,89	43,39	0	20,51	12,47
4	40	0	1	18,27	-62,46	40,35	0	22,11	-18,27
5	32,36	20	1	50,96	-63,89	43,39	0	20,51	-50,96
6	12,36	40	1	90,26	-61,07	37,44	0	23,63	-90,26
7	0	50	1	104,5	-54,78	24,38	0	30,4	-104,5

Tabel 4.5 Pengujian Validitas Hasil *Inverse Kinematic*

No.	Masukan (derajat)						Keluaran (cm)		
	1	2	3	4	5	6	px	py	pz
1	-75,48	-54,78	24,38	-180	-30,4	255,5	0	-50	1
2	-55,4	-61,07	37,44	0	23,63	55,4	12,36	-40	1
3	-12,47	-63,89	43,39	0	20,51	12,47	32,36	-20	1
4	18,27	-62,46	40,35	0	22,11	-18,27	40	0	1
5	50,96	-63,89	43,39	0	20,51	-50,96	32,36	20	1
6	90,26	-61,07	37,44	0	23,63	-90,26	12,36	40	1
7	104,5	-54,78	24,38	0	30,4	-104,5	0	50	1

Dari Tabel 4.2 dan Tabel 4.3, diketahui hasil perhitungan *inverse kinematic* menghasilkan posisi *end-effector* yang sama dengan masukan posisi yang diinginkan, sehingga perhitungan *inverse kinematic* sesuai.

4.3. Pengujian dan Analisis Akurasi Posisi Aktual *End-Effector* Melalui Suatu Penugasan

Penugasan untuk *manipulator* berupa trayektori kartesian dengan lintasan berbentuk lingkaran dengan radius 10 hingga 50 cm. *Error* dari posisi *end-effector* dihitung untuk menentukan akurasi dari *manipulator*. Hasil dari penugasan ditunjukkan pada Gambar 4.19. Nilai RMSE dari posisi *end-effector* ditunjukkan pada Tabel 4.6.

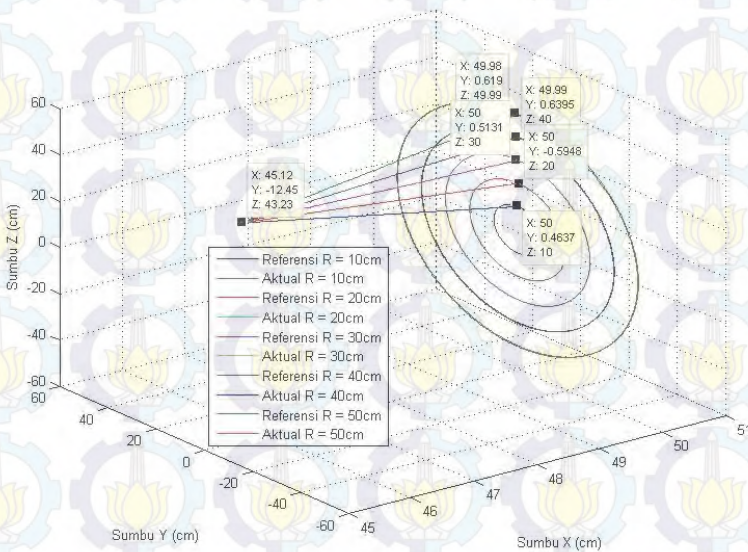
Tabel 4.6 Nilai RMSE Posisi *End-Effector* 1

Metode	Penugasan	RMSE Posisi End-Effector (cm)		
		X	Y	Z
Optimal CTC	Lingkaran Radius 10 cm	0,0016	0,0036	0,0075
	Lingkaran Radius 20 cm	0,0028	0,0061	0,0065
	Lingkaran Radius 30 cm	0,0053	0,0116	0,0071
	Lingkaran Radius 40 cm	0,0088	0,0117	0,0085
	Lingkaran Radius 50 cm	0,0142	0,0165	0,0112

Tabel 4.7 Nilai RMSE Posisi *End-Effector 2*

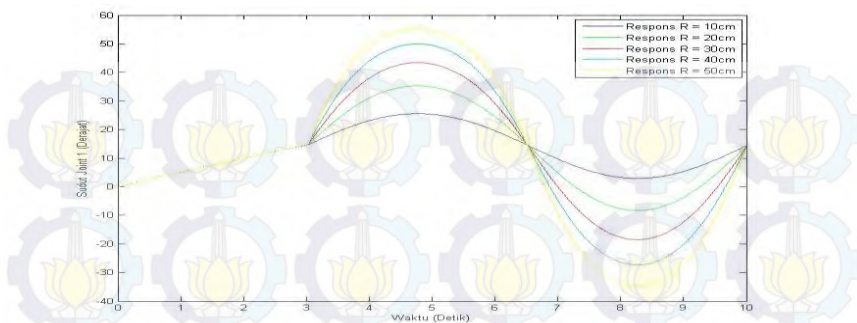
Rata-rata RMSE		0,0065	0,0099	0,0082
CTC	Lingkar Radius 10 cm	0,5593	2,1514	2,4294
	Lingkar Radius 20 cm	0,5778	3,7329	3,6488
	Lingkar Radius 30 cm	1,2075	5,7475	5,7471
	Lingkar Radius 40 cm	1,785	8,4642	9,0944
	Lingkar Radius 50 cm	8,852	12,336	11,068
Rata-rata RMSE		2,5963	6,4864	6,3975

Berdasarkan Tabel 4.6, diketahui metode *Optimal* CTC menghasilkan akurasi posisi yang lebih tinggi.

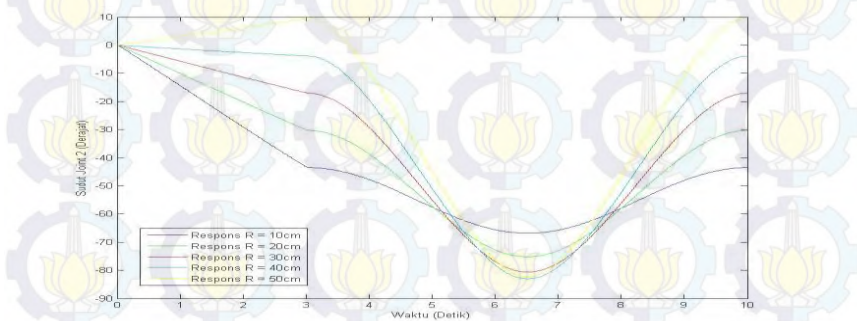


Gambar 4.19 Hasil Penugasan Trayektori Lingkaran Untuk *Manipulator*

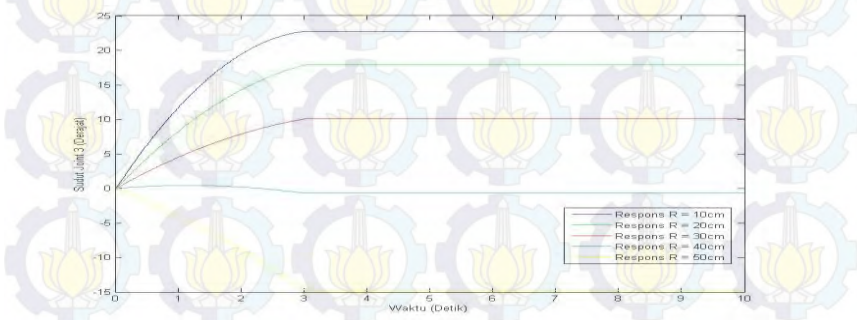
Untuk respons tiap *joint* dari penugasan trayektori lingkaran ditunjukkan pada Gambar 4.20 sampai Gambar 4.25.



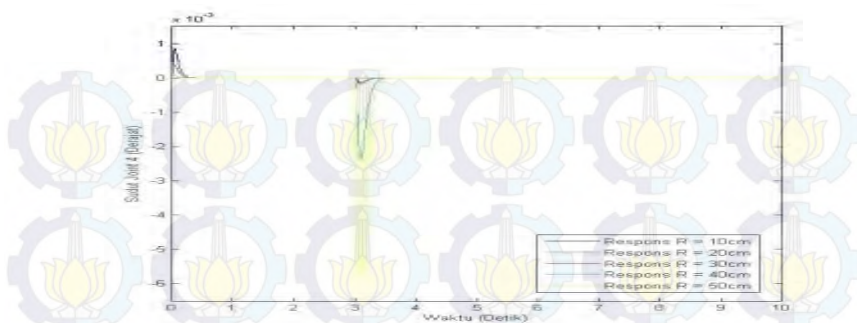
Gambar 4.20 Respons *Joint 1* Untuk Penugasan Trayektori Lingkaran



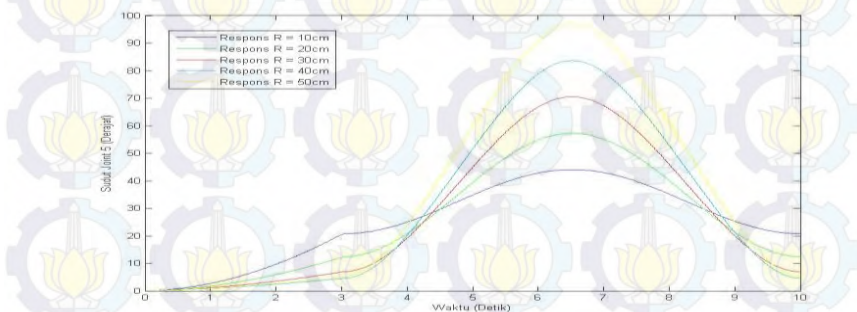
Gambar 4.21 Respons *Joint 2* Untuk Penugasan Trayektori Lingkaran



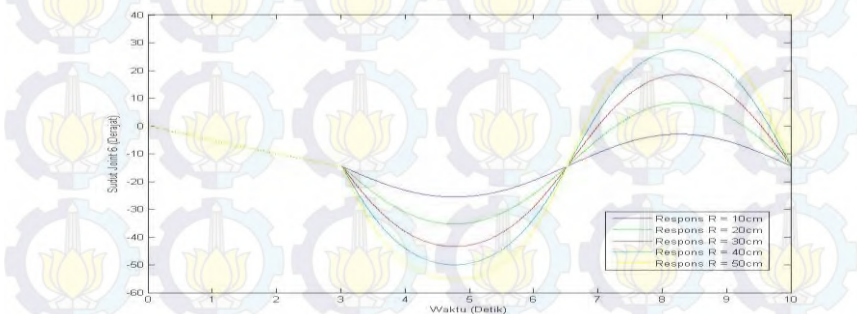
Gambar 4.22 Respons *Joint 3* Untuk Penugasan Trayektori Lingkaran



Gambar 4.23 Respons *Joint 4* Untuk Penugasan Trayektori Lingkaran

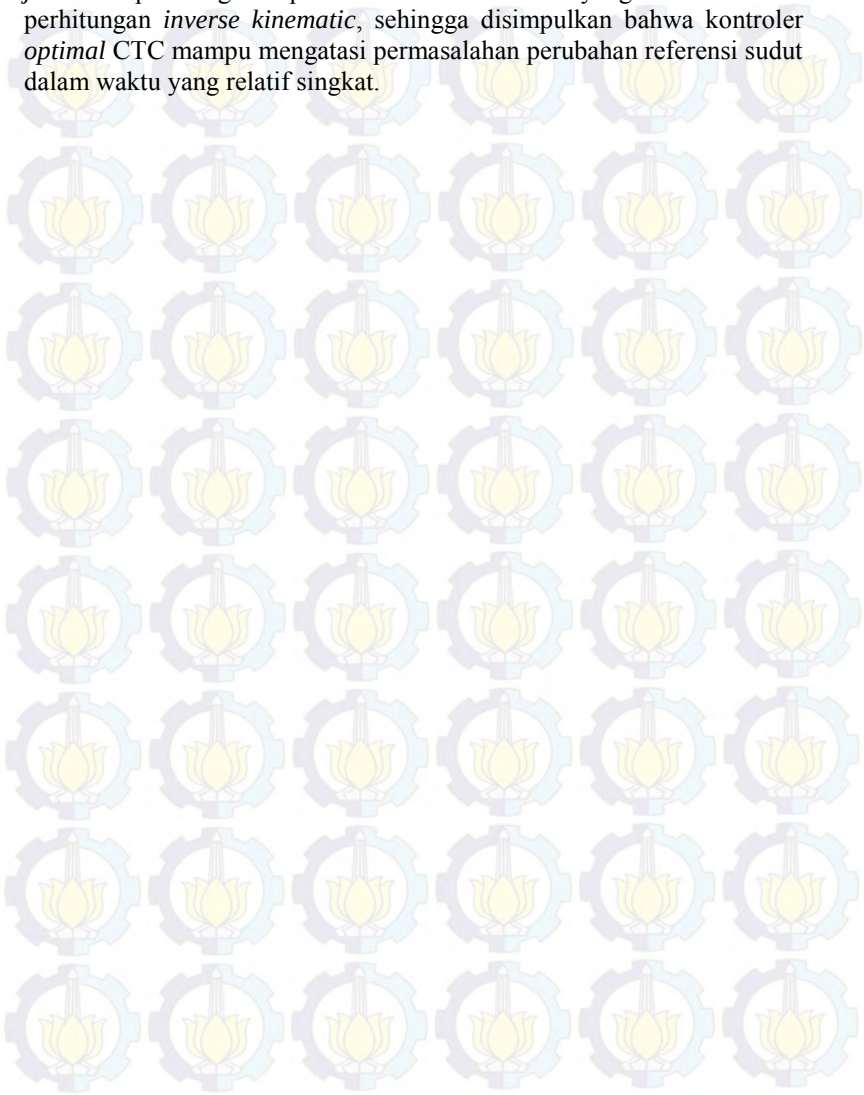


Gambar 4.24 Respons *Joint 5* Untuk Penugasan Trayektori Lingkaran



Gambar 4.25 Respons *Joint 6* Untuk Penugasan Trayektori Lingkaran

Berdasarkan Gambar 4.19 hingga Gambar 4.25, diketahui bahwa respons *joint* mampu mengikuti perubahan referensi sudut yang dihasilkan dari perhitungan *inverse kinematic*, sehingga disimpulkan bahwa kontroler *optimal* CTC mampu mengatasi permasalahan perubahan referensi sudut dalam waktu yang relatif singkat.



DAFTAR PUSTAKA

- [1] Craig, John J., "*Introduction to Robotics Mechanics and Control – Third Edition*", Prentice Hall, USA, 2005.
- [2] Spong, Mark W., Hutchinson, S. dan Vidyasagar, M., "*Robot Dynamics and Control – Third Edition*", John Wiley & Sons, 2004.
- [3] Siciliano, Bruce, "*Robotics Modelling Planning and Control*", Springer, UK, 2009.
- [4] Lewis, Frank L., Dawson, Darren M., Abdallah, Chaouki T., "*Robot Manipulator Control Theory and Practice - Second Edition*", Marcel Dekker, USA, 2004.
- [5] Benitez, Antonio., "PUMA 560: Robot Prototype with Graphic Simulation Environment", *Advanced In Mechanical Engineering* ISSN: 2160-0619 vol. 2 no. 1, Mexico, 2012.
- [6] Corke, Peter I., Armstrong-Helouvry, Brian., "A Search for Consensus Among Model Parameters Reported for the PUMA 560 Robot", *Proc. Int. Conf. Robotics and Automation.*, pp1608-1613, San Diego, 1994.
- [7] Armstrong-Helouvry, Brian., Khatib, Oussama., Burdick, Joe., "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", *Proc. 1986 IEEE Conf. Robot Autom.*, pp 510-518, San Fransisco, April 1986.
- [8] Corke, Peter I., "*Robotics, Vision, and Control Fundamental Algorithms in Matlab*", Springer, Australia, 2011.
- [9] Naidu, Desineni S., "*Optimal Control Systems*", CRC Press, Idaho, 2002.



BAB V

PENUTUP

Pada bagian ini akan diberikan kesimpulan dan saran berdasarkan hasil penelitian yang telah dilakukan.

5.1. Kesimpulan

Berdasarkan analisis dan hasil percobaan dari penelitian yang dilakukan, dapat disimpulkan bahwa :

1. Penyelesaian *inverse kinematic* menghasilkan nilai sudut *joint* yang akurat terbukti melalui uji validitas dari hasil penyelesaian tersebut.
2. Kontroler *optimal* CTC mampu mengurangi nilai RMSE dari respons tiap sudut *joint* dibandingkan dengan CTC sebesar $6,8897^0$.
3. Kontroler *optimal* CTC mampu mengatur gerak PUMA 560 lebih akurat dari CTC dengan rata-rata nilai RMSE untuk pergerakan sumbu $x = 0,0065$ cm; sumbu $y = 0,0099$ cm; sumbu $z = 0,0082$ cm untuk pengugasan trayektori lingkaran dengan radius 10 hingga 50 cm.

5.2. Saran

Berdasarkan hasil dari penelitian yang dilakukan, disarankan agar penelitian selanjutnya ditambahkan pengaturan gaya *end-effector* agar *manipulator* dapat melakukan tugas seperti *welding*, *assembling*, dan *milling*.



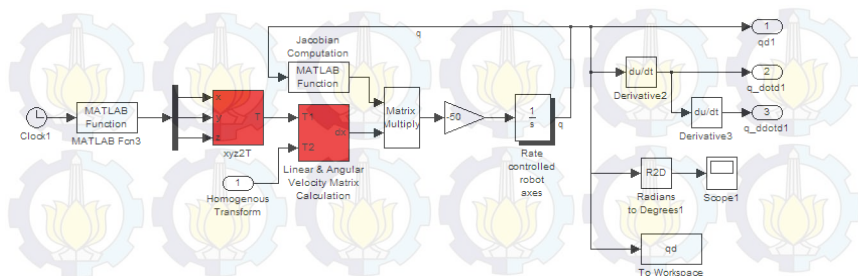
Halaman ini sengaja dikosongkan

A.1. Diagram Blok Simulasi Sistem Keseluruhan Pada Software Matlab 2010a

A.1.Diagram Blok Simulasi Sistem Keseluruhan Pada *Software* Matlab 2010a

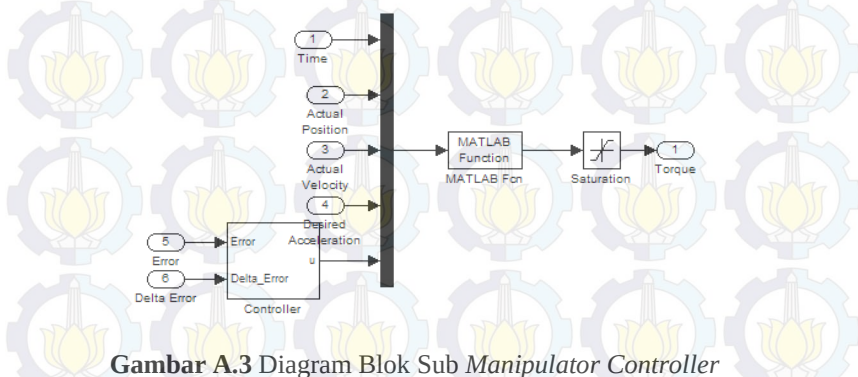


A.2. Diagram Blok Sub *Cartesian Trajectory Planning*



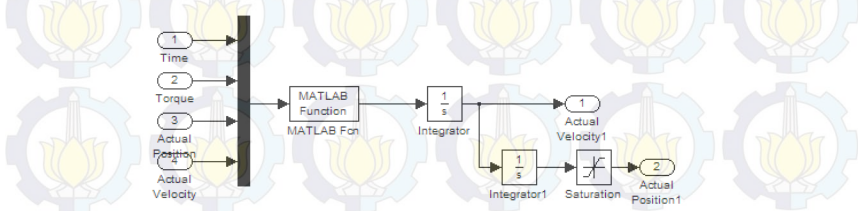
Gambar A.2 Diagram Blok Sub *Cartesian Trajectory Planning*

A.3. Diagram Blok Sub *Manipulator Controller*



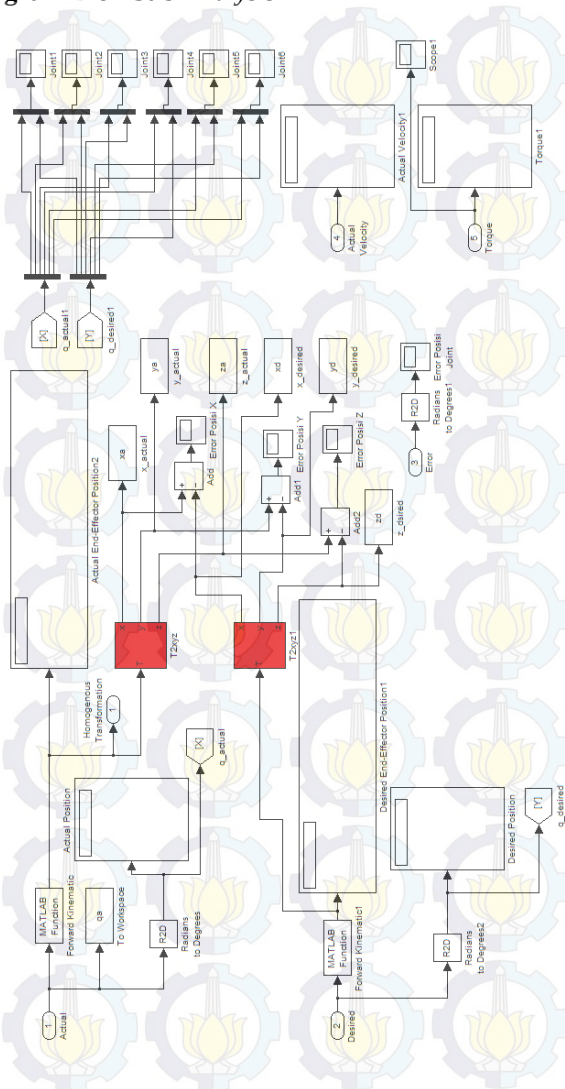
Gambar A.3 Diagram Blok Sub *Manipulator Controller*

A.4. Diagram Blok Sub PUMA 560



Gambar A.4 Diagram Blok Sub PUMA 560

A.5. Diagram Blok Sub Analysis



Gambar A.5 Diagram Blok Sub Analysis



LAMPIRAN B

B.1. Komponen Matriks *Jacobian* untuk *Manipulator 6-DOF*

$$J = \begin{bmatrix} J_{V0} & J_{V1} & J_{V2} & J_{V3} & J_{V4} & J_{V5} \\ J_{\omega0} & J_{\omega1} & J_{\omega2} & J_{\omega3} & J_{\omega4} & J_{\omega5} \end{bmatrix}$$

$$J_{V0} = z_0^0 x(p_6^0 - p_0^0); \quad J_{V1} = z_1^0 x(p_6^0 - p_1^0); \quad J_{V2} = z_2^0 x(p_6^0 - p_2^0)$$

$$J_{V3} = z_3^0 x(p_6^0 - p_3^0); \quad J_{V4} = z_4^0 x(p_6^0 - p_4^0); \quad J_{V5} = z_5^0 x(p_6^0 - p_5^0)$$

$$J_{\omega0} = z_0^0; \quad J_{\omega1} = z_1^0; \quad J_{\omega2} = z_2^0$$

$$J_{\omega3} = z_3^0; \quad J_{\omega4} = z_4^0; \quad J_{\omega5} = z_5^0$$

$z_0^0 = k = [0 \ 0 \ 1]^T$ dimana k adalah matriks *skew symmetric* untuk rotasi terhadap sumbu z dan $p_0^0 = [0 \ 0 \ 0]^T$.

B.2. Komponen Matriks M dan N PUMA 560

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 & 0 & 0 \\ m_{12} & m_{22} & m_{23} & 0 & 0 & 0 \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} & 0 \\ 0 & 0 & m_{34} & m_{44} & 0 & 0 \\ 0 & 0 & m_{35} & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} \quad N = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ 0 \end{bmatrix}$$

Keterangan Notasi :

$$S_i = \sin \theta_i; C_i = \cos \theta_i; S_{ij} = \sin(\theta_i + \theta_j); C_{ij} = \cos(\theta_i + \theta_j);$$

$$S_{ijk} = \sin(\theta_i + \theta_j + \theta_k); C_{ijk} = \cos(\theta_i + \theta_j + \theta_k);$$

$$SS_i = \sin^2 \theta_i ; CC_i = \cos^2 \theta_i ; CS_i = \cos \theta_i \sin \theta_i ;$$

$$SS_{ij} = \sin^2 (\theta_i + \theta_j)$$

$$m_{11} = 2,57 + 1,38CC_2 + 0,3SS_{23} + 0,744C_2S_{23}$$

$$m_{12} = 0,69S_2 - 0,134C_{23} + 0,0238C_2$$

$$m_{13} = -0,134C_{23} - 0,00397S_{23}$$

$$m_{22} = 6,79 + 0,744S_3$$

$$m_{23} = 0,333 + 0,372S_3 - 0,011C_3$$

$$m_{33} = 1,16$$

$$m_{34} = -0,00125S_4S_5$$

$$m_{35} = -0,00125C_4C_5$$

$$m_{44} = 0,2$$

$$m_{55} = 0,18$$

$$m_{66} = 0,19$$

$$\begin{aligned} n_1 = & [0,69C_2 + 0,134S_{23} - 0,0238S_2] \dot{\theta}_2^2 \\ & + [0,1335S_{23} - 0,00379C_{23}] \dot{\theta}_3^2 \\ & + [-2,76SC_2 + 0,744C_{223} + 0,6SC_{23} - 0,021(1 - 2SS_{23})] \dot{\theta}_1 \dot{\theta}_2 \\ & + [0,74C_2C_{23} + 0,6SC_{23} + 0,02C_2S_{23} + 0,021(1 - 2SS_{23})] \dot{\theta}_1 \dot{\theta}_3 \\ & + [-0,0025SC_{23}S_4S_5 + 0,0008C_4S_5 - 0,0024C_2C_{23}S_4S_5] \dot{\theta}_1 \dot{\theta}_4 \\ & + [0,0025(SS_{23}S_5 - SC_{23}C_4C_5) - 0,00248C_2(S_{23}S_5C_{23}C_4C_5) \\ & + 0,000864S_4C_5] \dot{\theta}_1 \dot{\theta}_5 + [0,267S_{23} - 0,00758C_{23}] \dot{\theta}_2 \dot{\theta}_3 \end{aligned}$$

$$\begin{aligned}
n_2 = & -\frac{1}{2}[-2,76SC_2 + 0,74C_{223} + 0,6SC_{23} - 0,02(1-2SS_{23})]\dot{\theta}_1^2 \\
& + \frac{1}{2}[0,022S_3 + 0,744C_3]\dot{\theta}_3^2 + [0,0016S_{23} - 0,0025C_{23}C_4S_5 \\
& + 0,00248S_2C_4S_5 + 0,00003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_4 \\
& + [-0,0021C_{23}S_4C_5 + 0,0024S_2S_4C_5 - 0,00064C_{23}S_4]\dot{\theta}_1\dot{\theta}_5 \\
& + [0,022S_3 + 0,744C_3]\dot{\theta}_2\dot{\theta}_3 - [0,00248C_3S_4S_5]\dot{\theta}_2\dot{\theta}_4 \\
& + [-0,0025S_5 + 0,00248(C_3C_4C_5 - S_3S_5)]\dot{\theta}_2\dot{\theta}_5 \\
& - [0,00248C_3S_4S_5]\dot{\theta}_3\dot{\theta}_4 \\
& + [-0,0025S_5 + 0,00248(C_3C_4C_5 - S_3S_5)]\dot{\theta}_3\dot{\theta}_5 \\
& - 37,2C_2 - 8,4S_{23} + 1,02S_2
\end{aligned}$$

$$\begin{aligned}
n_3 = & -\frac{1}{2}[-2,76SC_2 + 0,74C_{223} + 0,6SC_{23} - 0,02(1-2SS_{23})]\dot{\theta}_1^2 \\
& - \frac{1}{2}[0,02S_3 + 0,74C_3]\dot{\theta}_2^2 - [0,001C_4S_5]\dot{\theta}_5^2 - [0,001C_4S_5]\dot{\theta}_4^2 \\
& + [-0,002C_{23}C_4S_5 + 0,0016S_{23} + 0,0003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_4 \\
& - [0,0025C_{23}S_4C_5 + 0,00064C_{23}S_4]\dot{\theta}_1\dot{\theta}_5 - [0,0025S_5]\dot{\theta}_2\dot{\theta}_5 \\
& - [0,0025S_5]\dot{\theta}_3\dot{\theta}_5 - [0,0025S_4C_5]\dot{\theta}_4\dot{\theta}_5 - 8,4S_{23} + 0,25C_{23}
\end{aligned}$$

$$\begin{aligned}
n_4 = & \frac{1}{2}[0,0025SC_{23}S_4S_5 - 0,00086C_4S_5 + 0,00248C_2C_{23}S_4S_5]\dot{\theta}_1^2 \\
& - \frac{1}{2}[0,00248C_3S_4S_5]\dot{\theta}_2^2 + [0,00164S_{23} - 0,0025C_{23}C_4S_5 \\
& + 0,00248S_2C_4C_5 + 0,0003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_2 \\
& + [0,0025C_{23}C_4S_5 - 0,00164S_{23} - 0,003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_3 \\
& - [0,00064S_{23}C_4]\dot{\theta}_1\dot{\theta}_5 + [0,00064S_4]\dot{\theta}_2\dot{\theta}_5 + [0,00064S_4]\dot{\theta}_3\dot{\theta}_5 \\
& + 0,028S_{23}S_4S_5
\end{aligned}$$

$$\begin{aligned}
n_5 = & \frac{1}{2}[0,0025(SS_{23}S_5 - SC_{23}C_4C_5) + 0,002C_2(S_{23}S_5 - C_{23}C_4C_5) \\
& - 0,0008S_4C_5]\dot{\theta}_1^2 - \frac{1}{2}[0,0025S_5 - 0,002(C_3C_4C_5 - S_3S_5)]\dot{\theta}_2^2 \\
& + [0,0026C_{23}S_4C_5 - 0,00248S_2S_4C_5 + 0,000642C_{23}S_4]\dot{\theta}_1\dot{\theta}_2 \\
& + [0,0025C_{23}S_4C_5 + 0,0006C_{23}S_4]\dot{\theta}_1\dot{\theta}_3 + [0,0006S_{23}C_4]\dot{\theta}_1\dot{\theta}_4 \\
& - [0,0006S_4]\dot{\theta}_2\dot{\theta}_4 - [0,0006S_4]\dot{\theta}_3\dot{\theta}_4 - 0,028(C_{23}S_5 + S_{23}C_4S_5)
\end{aligned}$$

LAMPIRAN C

C.1. Listing Program Forward Kinematic Pada Matlab

```
function T = forward(q)
al = [pi/2 0 -pi/2 pi/2 -pi/2 0];
d = [0 0 0.1254 0.4318 0 0];
a = [0 0.4318 0.0191 0 0 0];

for i=1:6
A = [cos(q(i)) -sin(q(i))*cos(al(i)) sin(q(i))*sin(al(i)) a(i)*cos(q(i));
sin(q(i)) cos(q(i))*cos(al(i)) -cos(q(i))*sin(al(i)) a(i)*sin(q(i));
0 sin(al(i)) cos(al(i)) d(i) ;
0 0 0 1 ];
if i==1
T1 = A;
elseif i==2
T2 = A;
elseif i==3
T3 = A;
elseif i==4
T4 = A;
elseif i==5
T5 = A;
else
T6 = A;
end
end

T = T1*T2*T3*T4*T5*T6;
%%posisi_ee = [T(1,4);T(2,4);T(3,4)];
```

C.2. Listing Program Jacobian Manipulator

```
function J_inv = jac(q)
al = [pi/2 0 -pi/2 pi/2 -pi/2 0];
d = [0 0 0.1254 0.4318 0 0];
a = [0 0.4318 0.0191 0 0 0];
```

$k = [0;0;1]$; $z0 = k$; $o0 = [0;0;0]$;

for i=1:6

$A = \begin{bmatrix} \cos(q(i)) & -\sin(q(i))*\cos(\alpha(i)) & \sin(q(i))*\sin(\alpha(i)) & a(i)*\cos(q(i)); \\ \sin(q(i)) & \cos(q(i))*\cos(\alpha(i)) & -\cos(q(i))*\sin(\alpha(i)) & a(i)*\sin(q(i)); \\ 0 & \sin(\alpha(i)) & \cos(\alpha(i)) & d(i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$;

if i==1

T1 = A;

R1 = T1(1:3,1:3);

o1 = T1(1:3,4);

z1 = R1*k;

elseif i==2

T2 = T1*A;

R2 = T2(1:3,1:3);

o2 = T2(1:3,4);

z2 = R2*k;

elseif i==3

T3 = T2*A;

R3 = T3(1:3,1:3);

o3 = T3(1:3,4);

z3 = R3*k;

elseif i==4

T4 = T3*A;

R4 = T4(1:3,1:3);

o4 = T4(1:3,4);

z4 = R4*k;

elseif i==5

T5 = T4*A;

R5 = T5(1:3,1:3);

o5 = T5(1:3,4);

z5 = R5*k;

else

T6 = T5*A;

%%R6 = T6(1:3,1:3);

o6 = T6(1:3,4);

end

end

```

j1 = [cross(z0,(o6-o0));z0]; j2 = [cross(z1,(o6-o1));z1]; j3 =
[cross(z2,(o6-o2));z2];
j4 = [cross(z3,(o6-o3));z3]; j5 = [cross(z4,(o6-o4));z4]; j6 =
[cross(z5,(o6-o5));z5];
J = [j1 j2 j3 j4 j5 j6];
J_inv = inv(J);

```

C.3.Listing Program Computed Torque Control

```

function T = inv_dynamic(in)
t = in(1); qa = in(2:7); q_dota = in(8:13); q_ddotd = in(14:19); u =
in(20:25);
%%Initiation%%
%%if t == 0
%%  qa = [0;0;0;0;0;0];
%%  q_dota = [0;0;0;0;0;0];
%%else
%%  qa = qa;
%%  q_dota = q_dota;
%%end
%%Angular Position%%
c2 = cos(qa(2)); s2 = sin(qa(2)); c3 = cos(qa(3)); s3 = sin(qa(3));
c4 = cos(qa(4)); s4 = sin(qa(4)); c5 = cos(qa(5)); s5 = sin(qa(5));
c23 = cos(qa(2)+qa(3)); s23 = sin(qa(2)+qa(3));
c223 = cos(qa(2)*2+qa(3));

%%Angular Velocity%%
q1 = q_dota(1); q2 = q_dota(2); q3 = q_dota(3); q4 = q_dota(4);
q5 = q_dota(5);

%%Puma Inertia Matrix%%
m11 = 2.57+1.38*c2^2+0.3*s23^2+0.744*c2*s23;
m12 = 0.69*s2-0.134*c23+0.0238*c2;
m13 = -0.134*c23-0.00397*s23;
m14 = 0; m15 = 0; m16 = 0;
m22 = 6.79+0.744*s3;
m23 = 0.333+0.372*s3-0.011*c3;

```

$$m24 = 0; m25 = 0; m26 = 0;$$

$$m33 = 1.16; m34 = -0.00125*s4*s5; m35 = -0.00125*c4*c5; m36 = 0;$$

$$m44 = 0.2; m45 = 0; m46 = 0;$$

$$m55 = 0.18; m56 = 0;$$

$$m66 = 0.19;$$

$$m21 = m12; m31 = m13; m32 = m23; m41 = 0; m42 = 0;$$

$$m51 = 0; m52 = 0; m54 = 0; m61 = 0; m62 = 0; m63 = 0;$$

$$m64 = 0; m65 = 0;$$

$$M = [m11 \ m12 \ m13 \ m14 \ m15 \ m16;$$

$$m21 \ m22 \ m23 \ m24 \ m25 \ m26;$$

$$m31 \ m32 \ m33 \ m34 \ m35 \ m36;$$

$$m41 \ m42 \ m34 \ m44 \ m45 \ m46;$$

$$m51 \ m52 \ m35 \ m54 \ m55 \ m56;$$

$$m61 \ m62 \ m63 \ m64 \ m65 \ m66];$$

%%Puma Coriolis, Centrifugal, and Gravitational Matrix%%

$$n11 = (0.69*c2+0.134*s23-0.0238*s2)*q2^2;$$

$$n12 = (0.1335*s23-0.00379*c23)*q3^2;$$

$$n13 = (-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1-2*s23^2))*q1*q2;$$

$$n14 = (0.744*c2*c23+0.6*s23*c23+0.022*c2*s23-0.0213*(1-2*s23^2))*q1*q3;$$

$$n15 = (-0.0025*s23*c23*s4*s5+0.00086*c4*c5-0.00248*c2*(s23*s5-c23*c4*c5))*q1*q4;$$

$$n16 = (((-0.0025*(s23^2*s5-s23*c23*c4*c5)-0.00248*c2*(s23*s5-c23*c4*c5))+0.000864*s4*c5)*q1*q5;$$

$$n17 = (0.267*s23-0.00758*c23)*q2*q3;$$

$$n1 = (n11+n12+n13+n14+n15+n16+n17);$$

$$n21 = -0.5*(-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1-2*s23^2))*q1^2;$$

$$n22 = 0.5*(0.022*s3+0.744*c3)*q3^2;$$

$$\begin{aligned}
n23 &= (0.00164*s23- \\
&0.0025*c23*c4*s5+0.00248*s2*c4*s5+0.00003*s23*(1- \\
&2*s4^2))*q1*q4; \\
n24 &= (-0.00215*c23*s4*c5+0.00248*s2*s4*c5- \\
&0.000642*c23*s4)*q1*q5; \\
n25 &= (0.022*s3+0.744*c3)*q2*q3-(0.00248*c3*s4*s5)*q2*q4; \\
n26 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q2*q5- \\
&(0.00248*c3*s4*s5)*q3*q4; \\
n27 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q3*q5-37.2*c2- \\
&8.4*s23+1.02*s2; \\
n2 &= (n21+n22+n23+n24+n25+n26+n27);
\end{aligned}$$

$$\begin{aligned}
n31 &= -0.5*(-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1- \\
&2*s23^2))*q1^2; \\
n32 &= -0.5*(0.022*s3+0.744*c3)*q2^2-(0.00123*c4*s5)*q5^2- \\
&(0.00125*c4*s5)*q4^2; \\
n33 &= (-0.0025*c23*s4*c5+0.00164*s23+0.0003*s23*(1- \\
&2*s4^2))*q1*q4; \\
n34 &= -(0.0025*c23*s4*c5+0.000642*c23*s4)*q1*q5- \\
&(0.0025*s5)*q2*q5; \\
n35 &= -(0.0025*s5)*q3*q5-(0.0025*s4*c5)*q4*q5-8.4*s23+0.25*c23; \\
n3 &= n31+n32+n33+n34+n35; \\
n41 &= 0.5*(0.0025*s23*c23*s4*s5- \\
&0.00086*c4*s5+0.00248*c2*c23*s4*s5)*q1^2- \\
&0.5*(0.00248*c3*s4*s5)*q2^2; \\
n42 &= (0.00164*s23- \\
&0.0025*c23*c4*s5+0.00248*s2*c4*s5+0.0003*s23*(1- \\
&2*s4^2))*q1*q2; \\
n43 &= (0.0025*c23*c4*s5-0.00164*s23-0.003*s23*(1-2*s4^2))*q1*q3; \\
n44 &= -(0.000642*s23*c4)*q1*q5+(0.000642*s4)*q2*q5; \\
n45 &= (0.000642*s4)*q3*q5+0.028*s23*s4*s5; \\
n4 &= n41+n42+n43+n44+n45; \\
n51 &= 0.5*(0.0025*(s23^2*s5-s23*c23*c4*c5)+0.00248*c2*(s23*s5- \\
&c23*c4*c5)-0.000864*s4*c5)*q1^2; \\
n52 &= -0.5*(0.0025*s5-0.00248*(c3*c4*c5-s3*s5))*q2^2; \\
n53 &= (0.0026*c23*s4*c5-0.00258*s2*s4*c5+0.000642*c23*s4)*q2^2; \\
n54 &= -(0.000642*s4)*q2*q4-(0.000642*s4)*q3*q4- \\
&0.028*(c23*s5+s23*c4*c5); \\
n5 &= n51+n52+n53+n54;
\end{aligned}$$

n6 = 0;

N = [n1;n2;n3;n4;n5;n6];

T = M*(q_ddotd+u)+N;

C.4.Listing Program Dinamika PUMA 560

```
function q_ddota = dynamic(in)
t = in(1); T = in(2:7); qa = in(8:13); q_dota = in(14:19);
%%Initiation%%
%%if t == 0
%%  qa = [0;0;0;0;0;0];
%%  q_dota = [0;0;0;0;0;0];
%%else
%%  qa = qaa;
%%  q_dota = q_dota;
%%end
%%Angular Position%%
c2 = cos(qa(2)); s2 = sin(qa(2)); c3 = cos(qa(3)); s3 = sin(qa(3));
c4 = cos(qa(4)); s4 = sin(qa(4)); c5 = cos(qa(5)); s5 = sin(qa(5));
c23 = cos(qa(2)+qa(3)); s23 = sin(qa(2)+qa(3));
c223 = cos(qa(2)*2+qa(3));

%%Angular Velocity%%
q1 = q_dota(1); q2 = q_dota(2); q3 = q_dota(3); q4 = q_dota(4);
q5 = q_dota(5);

%%Puma Inertia Matrix%%
m11 = 2.57+1.38*c2^2+0.3*s23^2+0.744*c2*s23;
m12 = 0.69*s2-0.134*c23+0.0238*c2;
m13 = -0.134*c23-0.00397*s23;

m22 = 6.79+0.744*s3;
m23 = 0.333+0.372*s3-0.011*c3;

m33 = 1.16; m34 = -0.00125*s4*s5; m35 = -0.00125*c4*c5;
```

$$m44 = 0.2;$$

$$m55 = 0.18;$$

$$m66 = 0.19;$$

$$M = \begin{bmatrix} m11 & m12 & m13 & 0 & 0 & 0; \\ m12 & m22 & m23 & 0 & 0 & 0; \\ m13 & m23 & m33 & m34 & m35 & 0; \\ 0 & 0 & m34 & m44 & 0 & 0; \\ 0 & 0 & m35 & 0 & m55 & 0; \\ 0 & 0 & 0 & 0 & 0 & m66 \end{bmatrix};$$

%Puma Coriolis, Centrifugal, and Gravitational Matrix%%

$$n11 = (0.69*s^2 + 0.134*s^23 - 0.0238*s^2)*q^2^2;$$

$$n12 = (0.1335*s^23 - 0.00379*c^23)*q^3^2;$$

$$n13 = (-2.76*s^2*c^2 + 0.744*c^223 + 0.6*s^23*c^23 - 0.0213*(1 - 2*s^23^2))*q^1*q^2;$$

$$n14 = (0.744*c^2*c^23 + 0.6*s^23*c^23 + 0.022*c^2*s^23 - 0.0213*(1 - 2*s^23^2))*q^1*q^3;$$

$$n15 = (-0.0025*s^23*c^23*s^4*s^5 + 0.00086*c^4*c^5 - 0.00248*c^2*(s^23*s^5 - c^23*c^4*c^5))*q^1*q^4;$$

$$n16 = ((-0.0025*(s^23^2*s^5 - s^23*c^23*c^4*c^5) - 0.00248*c^2*(s^23*s^5 - c^23*c^4*c^5)) + 0.000864*s^4*c^5)*q^1*q^5;$$

$$n17 = (0.267*s^23 - 0.00758*c^23)*q^2*q^3;$$

$$n1 = (n11 + n12 + n13 + n14 + n15 + n16 + n17);$$

$$n21 = -0.5*(-2.76*s^2*c^2 + 0.744*c^223 + 0.6*s^23*c^23 - 0.0213*(1 - 2*s^23^2))*q^1^2;$$

$$n22 = 0.5*(0.022*s^3 + 0.744*c^3)*q^3^2;$$

$$n23 = (0.00164*s^23 -$$

$$0.0025*c^23*c^4*s^5 + 0.00248*s^2*c^4*s^5 + 0.00003*s^23*(1 - 2*s^4^2))*q^1*q^4;$$

$$n24 = (-0.00215*c^23*s^4*c^5 + 0.00248*s^2*s^4*c^5 - 0.000642*c^23*s^4)*q^1*q^5;$$

$$n25 = (0.022*s^3 + 0.744*c^3)*q^2*q^3 - (0.00248*c^3*s^4*s^5)*q^2*q^4;$$

$$\begin{aligned}n26 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q2*q5- \\ &(0.00248*c3*s4*s5)*q3*q4; \\ n27 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q3*q5-37.2*c2- \\ &8.4*s23+1.02*s2; \\ n2 &= (n21+n22+n23+n24+n25+n26+n27);\end{aligned}$$

$$\begin{aligned}n31 &= -0.5*(-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1- \\ &2*s23^2))*q1^2; \\ n32 &= -0.5*(0.022*s3+0.744*c3)*q2^2-(0.00123*c4*s5)*q5^2- \\ &(0.00125*c4*s5)*q4^2; \\ n33 &= (-0.0025*c23*s4*c5+0.00164*s23+0.0003*s23*(1- \\ &2*s4^2))*q1*q4; \\ n34 &= -(0.0025*c23*s4*c5+0.000642*c23*s4)*q1*q5- \\ &(0.0025*s5)*q2*q5; \\ n35 &= -(0.0025*s5)*q3*q5-(0.0025*s4*c5)*q4*q5-8.4*s23+0.25*c23; \\ n3 &= n31+n32+n33+n34+n35;\end{aligned}$$

$$\begin{aligned}n41 &= 0.5*(0.0025*s23*c23*s4*s5- \\ &0.00086*c4*s5+0.00248*c2*c23*s4*s5)*q1^2- \\ &0.5*(0.00248*c3*s4*s5)*q2^2; \\ n42 &= (0.00164*s23- \\ &0.0025*c23*c4*s5+0.00248*s2*c4*s5+0.0003*s23*(1- \\ &2*s4^2))*q1*q2; \\ n43 &= (0.0025*c23*c4*s5-0.00164*s23-0.003*s23*(1-2*s4^2))*q1*q3; \\ n44 &= -(0.000642*s23*c4)*q1*q5+(0.000642*s4)*q2*q5; \\ n45 &= (0.000642*s4)*q3*q5+0.028*s23*s4*s5; \\ n4 &= n41+n42+n43+n44+n45;\end{aligned}$$

$$\begin{aligned}n51 &= 0.5*(0.0025*(s23^2*s5-s23*c23*c4*c5)+0.00248*c2*(s23*s5- \\ &c23*c4*c5)-0.000864*s4*c5)*q1^2; \\ n52 &= -0.5*(0.0025*s5-0.00248*(c3*c4*c5-s3*s5))*q2^2; \\ n53 &= (0.0026*c23*s4*c5-0.00258*s2*s4*c5+0.000642*c23*s4)*q2^2; \\ n54 &= -(0.000642*s4)*q2*q4-(0.000642*s4)*q3*q4- \\ &0.028*(c23*s5+s23*c4*c5); \\ n5 &= n51+n52+n53+n54;\end{aligned}$$

$$N = [n1;n2;n3;n4;n5;0];$$

$q_ddota = M \backslash (T - N);$

D.3. Listing Program Trayektori Lingkaran

```
function out=lingkaran1(t)
tmax=10;
k=.5;
if t<=3
    p0=[.4509 -.1254 .4318];
    p1=[.5 0 k];
    x=((p1(1)-p0(1))*t/3)+p0(1);
    y=((p1(2)-p0(2))*t/3)+p0(2);
    z=((p1(3)-p0(3))*t/3)+p0(3);
else
    q=(2*pi)*(t-3)/(tmax-3);
    gain=k;
    x=.50;
    y=sin(q)*gain;
    z=cos(q)*gain;
end
out=[x,y,z];
```





PUMA 560 MANIPULATOR MOTION CONTROL USING OPTIMAL COMPUTED TORQUE CONTROL (OPTIMAL CTC)

Name : Aditya Rahmat Abdillah
Supervisor : Dr. Trihastuti Agustinah, S.T., M.T.

ABSTRACT

There is a rapid development of manipulator application in industrial environment. It was concerned with performance upgrading and cost reduction. Most of manipulators are used for solving tasks such material handling, assembling, welding, etc. The tasks given contain some trajectories with vary joint reference values. Nonlinearity of system and friction for each joint cause the accuracy of manipulator decrease. Optimal CTC is used to solve the problem with solutions in joint space. Inverse kinematic solutions is applied to the system for obtaining solutions in cartesian space, so the task can be applied to manipulator. A circular trajectory with radius 10 to 50 cm are given for the manipulator as a task. The simulation results show that RMSE values average of x-axis is 0.0065 cm, 0.0099 cm at y-axis, and 0.0082 cm at z-axis.

Keywords : Accuracy, Inverse Kinematic, Optimal CTC



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah mencurahkan segala rahmat dan hidayahnya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul *Pengaturan Gerak Manipulator PUMA 560 Menggunakan Optimal Computed Torque Control (Optimal CTC)* tepat waktu.

Tugas Akhir ini merupakan syarat untuk menyelesaikan studi dan memperoleh gelar Sarjana di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih banyak atas bantuan dan motivasi yang diberikan baik secara langsung ataupun tidak langsung dalam pelaksanaan Tugas Akhir ini kepada:

1. Ibu Dr. Trihastuti Agustinah, S.T., M.T., selaku Dosen Pembimbing dalam Tugas Akhir ini
2. Orang tua dan keluarga yang telah mencurahkan doa, kasih sayang dan semangat,
3. Serta semua pihak yang tidak bisa disebutkan satu-persatu yang turut membantu dan memperlancar jalankan Tugas Akhir ini.

Penulis menyadari bahwa penyusunan buku ini masih belum sempurna. Untuk itu, penulis mengharapkan adanya kritik dan saran demi kebaikan di kemudian hari. Akhir kata, penulis mohon maaf jika ada kekurangan selama pengerjaan Tugas Akhir ini.

Surabaya, Juni 2015

Penulis

Aditya Rahmat A



DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Metodologi Penelitian.....	2
1.6. Sistematika Penulisan.....	3
1.7. Relevansi.....	4
BAB II TEORI PENUNJANG.....	5
2.1. <i>Manipulator</i>	5
2.1.1. Komponen dan Struktur dari <i>Manipulator</i>	5
2.1.2. Klasifikasi dari <i>Manipulator</i>	6
2.2. Kinematika <i>Manipulator</i>	8
2.2.1. <i>Forward Kinematic</i>	8
2.2.2. <i>Inverse Kinematic</i>	11
2.3. <i>Jacobian Manipulator</i>	12
2.4. Model Dinamika PUMA 560.....	13
2.5. <i>Computed Torque Control (CTC)</i>	16
2.6. <i>Linear Quadratic Regulator (LQR)</i>	17
2.7. Perencanaan Trayektori.....	18
2.8. Spesifikasi PUMA 560.....	19
BAB III PERANCANGAN SISTEM.....	21
3.1. Perancangan Sistem Keseluruhan.....	21
3.2. Perancangan <i>Forward Kinematic</i> PUMA 560.....	22
3.3. Perancangan <i>Inverse Kinematic</i> PUMA 560.....	23

3.4. Perancangan Model Dinamika PUMA 560.....	26
3.5. Perancangan Kontroler <i>Linear Quadratic Regulator</i>	28
3.6. Perancangan Kontroler <i>Computed Torque Control</i>	30
3.7. Perancangan Trayektori Kartesian.....	31
3.8. Perancangan Sistem Keseluruhan Menggunakan Matlab dengan <i>Toolbox Robot</i> Peter I Corke.....	32
BAB IV PENGUJIAN DAN ANALISIS SISTEM.....	33
4.1. Pengujian dan Analisis Respons <i>Joint</i>	33
4.1.1. Pengujian dan Analisis Menggunakan Sinyal <i>Step</i>	33
4.1.2. Pengujian dan Analisis Menggunakan Sinyal <i>Cubical Polynomial</i>	38
4.2. Pengujian dan Analisis <i>Inverse Kinematic</i> PUMA 560 Menggunakan <i>Jacobian Manipulator</i>	41
4.3. Pengujian dan Analisis Akurasi Posisi Aktual <i>End-Effector</i> Melalui Suatu Penugasan.....	42
BAB V PENUTUP.....	47
5.1. Kesimpulan.....	47
5.2. Saran.....	47
DAFTAR PUSTAKA.....	49
LAMPIRAN A.....	51
LAMPIRAN B.....	55
LAMPIRAN C.....	59
RIWAYAT PENULIS.....	69

DAFTAR TABEL

Tabel 2.1 Batas Gerak <i>Joint</i> PUMA 560.....	19
Tabel 2.2 Torsi Maksimum PUMA 560.....	19
Tabel 3.1 Parameter D-H Standar PUMA 560.....	22
Tabel 3.2 Konstanta Inersia PUMA 560.....	27
Tabel 3.3 Konstanta Gravitasi PUMA 560.....	27
Tabel 4.1 Rata-rata Nilai RMSE dan Waktu <i>Steady State</i> Respons Sistem.....	36
Tabel 4.2 Perbandingan Respons CTC Terhadap <i>Optimal</i> CTC....	38
Tabel 4.3 Nilai RMSE dari Respons Tiap <i>Joint</i> Terhadap Sinyal <i>Cubical Polynomial</i>	39
Tabel 4.4 Hasil Perhitungan <i>Inverse Kinematic</i>	41
Tabel 4.5 Pengujian Validitas Hasil <i>Inverse Kinematic</i>	42
Tabel 4.6 Nilai RMSE Posisi <i>End-Effector</i> 1.....	42
Tabel 4.7 Nilai RMSE Posisi <i>End-Effector</i> 2.....	43



DAFTAR GAMBAR

Gambar 2.1	Bagian-bagian <i>Manipulator</i>	6
Gambar 2.2	Jenis-jenis <i>Manipulator</i> Berdasarkan Bentuk Geometri.....	7
Gambar 2.3	Sumbu <i>Joint Manipulator</i>	9
Gambar 2.4	Acuan Pengukuran θ_i	9
Gambar 2.5	Acuan Pengukuran α_i	10
Gambar 2.6	Gaya Sentripetal.....	14
Gambar 2.7	Gaya Koriolis.....	14
Gambar 3.1	Diagram Blok Sistem Keseluruhan.....	21
Gambar 3.2	<i>Flowchart</i> Perancangan <i>Forward Kinematic</i> PUMA 560 Menggunakan Matlab.....	24
Gambar 3.3	Perancangan <i>Jacobian</i> PUMA 560.....	25
Gambar 3.4	Perancangan <i>Inverse Kinematic</i> PUMA 560.....	26
Gambar 3.5	<i>Flowchart</i> Perancangan Model Dinamik PUMA 560.....	28
Gambar 3.6	Diagram Blok Kontrol LQR.....	29
Gambar 3.7	<i>Flowchart</i> Perancangan Kontrol CTC.....	30
Gambar 3.8	Perancangan Trayektori Kartesian.....	31
Gambar 3.9	Animasi <i>Manipulator</i> Pada <i>Software</i> Matlab.....	32
Gambar 4.1	Respons <i>Joint</i> 1 CTC Terhadap Sinyal Uji <i>Step</i>	34
Gambar 4.2	Respons <i>Joint</i> 2 CTC Terhadap Sinyal Uji <i>Step</i>	34
Gambar 4.3	Respons <i>Joint</i> 3 CTC Terhadap Sinyal Uji <i>Step</i>	34
Gambar 4.4	Respons <i>Joint</i> 4 CTC Terhadap Sinyal Uji <i>Step</i>	35
Gambar 4.5	Respons <i>Joint</i> 5 CTC Terhadap Sinyal Uji <i>Step</i>	35
Gambar 4.6	Respons <i>Joint</i> 6 CTC Terhadap Sinyal Uji <i>Step</i>	35
Gambar 4.7	Respons <i>Joint</i> 1 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	36
Gambar 4.8	Respons <i>Joint</i> 2 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	36
Gambar 4.9	Respons <i>Joint</i> 3 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	37
Gambar 4.10	Respons <i>Joint</i> 4 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	37
Gambar 4.11	Respons <i>Joint</i> 5 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	37
Gambar 4.12	Respons <i>Joint</i> 6 <i>Optimal</i> CTC Terhadap Sinyal Uji <i>Step</i>	38

Gambar 4.13	Respons <i>Joint</i> 1 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	39
Gambar 4.14	Respons <i>Joint</i> 2 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	39
Gambar 4.15	Respons <i>Joint</i> 3 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	40
Gambar 4.16	Respons <i>Joint</i> 4 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	40
Gambar 4.17	Respons <i>Joint</i> 5 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	40
Gambar 4.18	Respons <i>Joint</i> 6 Terhadap Sinyal Uji <i>Cubical Polynomial</i>	41
Gambar 4.19	Hasil Penugasan Trayektori Lingkaran Untuk <i>Manipulator</i>	43
Gambar 4.20	Respons <i>Joint</i> 1 Untuk Penugasan Trayektori Lingkaran.....	44
Gambar 4.21	Respons <i>Joint</i> 2 Untuk Penugasan Trayektori Lingkaran.....	44
Gambar 4.22	Respons <i>Joint</i> 3 Untuk Penugasan Trayektori Lingkaran.....	44
Gambar 4.23	Respons <i>Joint</i> 4 Untuk Penugasan Trayektori Lingkaran.....	45
Gambar 4.24	Respons <i>Joint</i> 5 Untuk Penugasan Trayektori Lingkaran.....	45
Gambar 4.25	Respons <i>Joint</i> 6 Untuk Penugasan Trayektori Lingkaran.....	45

RIWAYAT PENULIS



Aditya Rahmat Abdillah, dilahirkan di Surabaya, 19 Maret 1993 sebagai putra sulung dari Bapak Panca Suluh DS dan Ibu Aisyah Nikma S. Penulis sempat merasakan beberapa tempat selama menempuh masa pendidikan. Sekolah dasar di SD Negeri Bangselok 1 Sumenep, sempat pindah ke SD Negeri Pucang Anom 1 Sidoarjo saat kelas 2 hingga kelas 3, melanjutkan sekolah menengah di SMP Negeri 1 Sumenep, dan SMA Negeri 1 Sumenep. Setelah lulus sekolah menengah dilanjutkan ke Diploma 3 Teknik Elektro ITS dan lulus pada Tahun 2013, kemudian melanjutkan ke S1 Teknik Elektro di bidang studi teknik sistem pengaturan. Pada bulan juni 2015 penulis melaksanakan seminar dan sidang Tugas Akhir untuk memperoleh gelar Sarjana S1 Teknik Elektro.

Email: aditya.rahmat10@mhs.ee.its.ac.id



BAB I

PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, perumusan masalah, batasan masalah dan tujuan dari tugas akhir yang dilakukan. Metodologi penelitian dan sistematika penulisan dalam penyelesaian tugas akhir juga disertakan. Relevansi diberikan sebagai langkah untuk penelitian selanjutnya.

1.1. Latar Belakang

Penggunaan robot di dunia industri berkembang pesat, khususnya *manipulator* [1]. Peningkatan performa kerja dari *manipulator* seperti efektivitas kerja, akurasi, dan presisi merupakan faktor utama yang memicu hal tersebut. Seiring dengan meningkatnya performa kerja tersebut, jumlah produksi menjadi lebih banyak, dan sebaliknya biaya produksi menjadi berkurang. *Manipulator* juga memiliki keunggulan berupa kemampuan menyelesaikan pekerjaan yang berbahaya untuk manusia seperti tempat radioaktif dan penjelajahan luar angkasa [1].

Pada umumnya *manipulator* melakukan tugas seperti *material handling*, *assembling*, *welding* dan sebagainya. Akurasi pada *manipulator* akan berkurang akibat adanya nonlinearitas dari sistem dan gaya friksi pada tiap *joint*, sehingga hasil dari tugas yang diberikan tidak sesuai. Oleh karena permasalahan tersebut, dilakukan pengaturan gerak untuk *manipulator* agar hasil dari penugasan sesuai. Pengaturan gerak yang digunakan menghasilkan penyelesaian dalam ruang *joint* sehingga digunakan *inverse kinematic* untuk memperoleh solusi pengaturan gerak *manipulator* dalam ruang kartesian.

Metode pengaturan gerak yang digunakan berupa *model-based control* yang membutuhkan model dinamika dari *manipulator* untuk proses pengaturannya. Metode *model-based control* yang umum digunakan yaitu *feedforward* dan *computed torque control*. Pada tugas akhir ini digunakan *computed torque control* sebagai metode untuk pengaturan gerak *manipulator*. Metode tersebut dikombinasikan dengan kontrol optimal untuk perhitungan *gain feedback outer loop*, sehingga akurasi dari *manipulator* dapat ditingkatkan. Perhitungan *inverse kinematic* dilakukan menggunakan *Jacobian manipulator*. *Jacobian manipulator* mendeskripsikan hubungan antara kecepatan *end-effector* dengan kecepatan *joint* dari *manipulator*.

1.2. Perumusan Masalah

Permasalahan yang muncul dalam penggunaan *manipulator* yaitu sifat nonlinear dan besaran seperti gaya friksi pada tiap *joint* yang menyebabkan akurasi *manipulator* berkurang, sehingga hasil dari penugasan tidak sesuai dengan referensi. Suatu kontroler nonlinear dirancang untuk mengatasi permasalahan akurasi tersebut. Penugasan untuk *manipulator* dilakukan dengan merencanakan suatu trayektori dalam ruang *joint* atau kartesian. Trayektori yang direncanakan dalam ruang *joint* lebih sulit dilakukan dibandingkan trayektori dalam ruang kartesian karena kemungkinan nilai-nilai sudut yang mewakili posisi dalam ruang kartesian sangat banyak. Penyelesaian *inverse kinematic* digunakan agar perencanaan trayektori kartesian dapat dilakukan. Trayektori yang digunakan sebagai penugasan berisi nilai referensi *joint* yang bervariasi dalam waktu yang singkat sehingga dapat meningkatkan *error tracking* jika respons dari tiap *joint* lambat. Oleh karena itu digunakan metode *optimal* untuk meningkatkan kecepatan respons tiap *joint*.

1.3. Batasan Masalah

Pada tugas akhir ini *manipulator* yang dikontrol adalah PUMA 560. Pengaturan gerak PUMA 560 meliputi posisi, kecepatan, dan percepatan dari sudut *joint*. Oleh karena itu pengaturan gerak *manipulator* pada tugas akhir ini dikhususkan pada penugasan *material handling*. Penyelesaian *inverse kinematic* yang digunakan menggunakan *jacobian manipulator*. Diasumsikan *workspace* bebas dari penghalang.

1.4. Tujuan Penelitian

Penelitian pada tugas akhir ini memiliki tujuan yaitu merancang kontroler *optimal* CTC untuk mengatur gerak *manipulator* PUMA 560 dalam ruang *joint* dan merancang *inverse kinematic* menggunakan *jacobian manipulator* untuk merubah nilai-nilai koordinat dari trayektori kartesian menjadi nilai-nilai sudut tiap *joint* sehingga perencanaan trayektori dalam ruang kartesian dapat dilakukan untuk penugasan *manipulator* PUMA 560.

1.5. Metodologi Penelitian

Metodologi yang dilakukan pada Tugas Akhir ini yaitu :

a. Studi Literatur

Pada tahap ini dilakukan kajian teori yang berkaitan dengan pemodelan kinematika dan dinamika dari *manipulator* khususnya PUMA 560. Sumber yang digunakan berupa buku referensi, *paper* penelitian terkait, dan jurnal ilmiah.

b. Pemodelan Sistem

Pemodelan sistem terdiri dari *kinematic* dan dinamika dari PUMA 560 yang diperoleh dari jurnal ilmiah terkait.

c. Perancangan Kontroler

Pada tahap ini dilakukan perancangan kontroler *computed torque control* yang dikombinasikan dengan *linear quadratic regulator*.

d. Perancangan Simulasi

Hasil perancangan sistem pengaturan gerak *manipulator* disimulasikan menggunakan *software* Matlab 2010a dengan *toolbox* robot Peter I Corke versi 8.0.

e. Penulisan Buku

Hasil dari tahap studi literatur hingga perancangan simulasi dituliskan pada buku laporan tugas akhir.

1.6. Sistematika Penulisan

Penulisan tugas akhir ini disusun dalam lima bab yang masing-masing membahas permasalahan yang berhubungan dengan tugas akhir yang telah dibuat dengan sistematika penulisan sebagai berikut:

a. BAB I PENDAHULUAN

Berisi tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan, metodologi, sistematika penulisan dan relevansi pembahasan tugas akhir ini.

b. BAB II TEORI PENUNJANG

Memberi gambaran secara umum mengenai konsep teori yang mendasari perancangan tugas akhir ini, meliputi teori *manipulator*, model kinematika *manipulator*, model dinamika *manipulator*, *jacobian manipulator*, *computed torque control*,

linear quadratic regulator, dan perencanaan trayektori untuk *manipulator*.

c. BAB III PERANCANGAN SISTEM

Dijelaskan secara menyeluruh tentang perancangan simulasi sistem yang terdiri dari perancangan *forward kinematic*, perancangan *inverse kinematic*, perancangan model dinamika, perancangan *computed torque control*, perancangan *linear quadratic regulator*, dan perancangan trayektori *manipulator*.

d. BAB IV PENGUJIAN DAN ANALISIS SISTEM

Berisi hasil pengujian dan analisis dari sistem yang telah dirancang. Pengujian yang dilakukan meliputi respons *joint* terhadap sinyal uji, *inverse kinematic*, dan penugasan trayektori lingkaran. Hasil dari tiap pengujian dianalisis dengan cara melihat *settling time* untuk respons sinyal tiap sinyal uji, validitas solusi nilai *joint* untuk *inverse kinematic*, dan akurasi dari hasil penugasan yang diberikan pada *manipulator*.

e. BAB V PENUTUP

Berisi tentang kesimpulan dan saran untuk pengembangan tugas akhir ini.

1.7. Relevansi

Tugas akhir ini diharapkan dapat digunakan sebagai referensi pengaturan gerak pada *manipulator* jenis lainnya, dilakukan implementasi pada *plant manipulator*, serta pengembangan untuk perancangan metode kontrol selanjutnya.

BAB II

TEORI PENUNJANG

Pada bab ini dibahas teori-teori yang berkaitan dengan topik penelitian yang dilakukan. Dasar teori pada bab ini meliputi pendahuluan mengenai *manipulator*, komponen dan struktur dari *manipulator*, kinematika dan dinamika *manipulator*, metode *linear quadratic regulator* (LQR), *computed torque control* (CTC), perencanaan trayektori, dan spesifikasi PUMA 560 yang akan diterapkan pada *manipulator*.

2.1. Manipulator

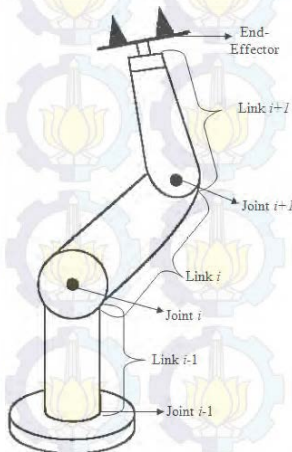
Robot Institute of America (RIA) mendefinisikan robot sebagai perangkat multifungsi terprogram yang dirancang untuk memindahkan material, atau alat khusus dengan perencanaan pergerakan untuk menyelesaikan tugas tertentu [2]. Secara struktur mekanik, robot diklasifikasikan menjadi *manipulator (fixed base frame)* dan *mobile robot (mobile base frame)*. Kedua jenis robot tersebut dibedakan berdasarkan *base frame*-nya. *Base frame* adalah sistem koordinat yang dijadikan sebagai acuan dari *manipulator* dan segala objek pada *workspace manipulator*. *Base frame* dapat bersifat tetap (*fixed*) atau bergerak (*mobile*). Penjelasan selanjutnya akan dikhususkan untuk *manipulator*.

Manipulator merupakan jenis robot yang memiliki *base frame* tetap. Tujuan dari penggunaan *manipulator* untuk membantu pekerjaan yang dilakukan oleh manusia khususnya untuk pekerjaan yang tidak memungkinkan seperti eksplorasi bawah laut dan luar angkasa. Penggunaan *manipulator* di dunia industri juga mampu mengurangi biaya produksi, serta meningkatkan kualitas dan kapasitas produksi.

2.1.1. Komponen dan Struktur dari Manipulator

Manipulator terdiri dari beberapa komponen antara lain *joint*, *link*, dan *end-effector*. *Joint* adalah komponen yang menggerakkan *link* dan *end-effector*. Struktur mekanik *manipulator* terdiri dari rangkaian *link* yang terhubung melalui *joint*. Rangkaian dari *joint*, *link*, dan *end-effector* disebut *kinematic chain*. *Kinematic chain* dapat berupa *serial/open* atau *parallel/close* [3]. Struktur dasar dari *manipulator* sebagian besar berupa *serial/open kinematic chain*. Pergerakan *manipulator* ditentukan oleh adanya *joint* yang dapat berupa *prismatic* atau *revolute joint*. Untuk menggunakan *manipulator* dalam berbagai tugas, diperlukan spesifikasi

workspace. *Workspace* merupakan area yang dapat dijangkau oleh *manipulator* berdasarkan nilai variabel *joint* yang memungkinkan. *Workspace* dibagi menjadi *reachable* dan *dextrous* [2]. Bagian-bagian dari *manipulator* ditunjukkan pada Gambar 2.1.



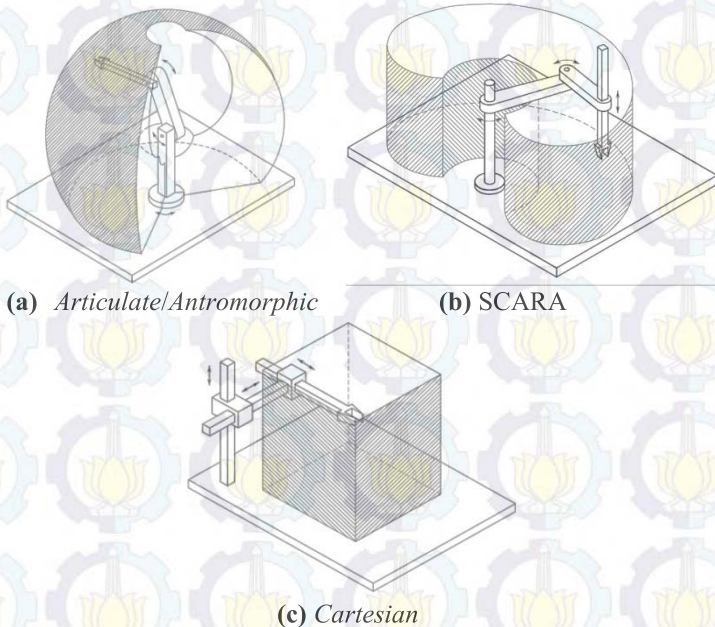
Gambar 2.1 Bagian-bagian *Manipulator*

Dalam suatu *open kinematic chain*, tiap *joint* memberikan pergerakan sebanyak satu derajat kebebasan (DOF). Derajat kebebasan merupakan suatu ukuran yang menunjukkan kemampuan *manipulator* untuk menuju ke titik-titik kerja dalam *workspace*. Pada umumnya *manipulator* memiliki enam derajat kebebasan yang berfungsi untuk pengerjaan tugas dengan posisi dan orientasi dalam ruang tiga dimensi. Sistem robot selain dari komponen mekanik yang telah dijelaskan antara lain *power supply*, lengan mekanik, sensor, *teach pendant*, kontroler, penyimpanan program, dan *tool end-effector* [3].

2.1.2. Klasifikasi dari *Manipulator*

Manipulator diklasifikasikan berdasarkan sumber daya, area aplikasi, dan geometrinya [2]. Sumber daya *manipulator* dapat berupa hidrolik, pneumatik, atau motor servo. Berdasarkan area aplikasi *manipulator* dibagi menjadi *assembly* dan *non-assembly*. Ukuran *manipulator* untuk area *assembly* umumnya berbentuk kecil. Area *non-*

assembly meliputi pengecatan, pengelasan, dan *material handling*. Berdasarkan geometri *manipulator* terbagi menjadi lima antara lain *articulate/antromorphic*, *spherical*, SCARA, *cylindrical*, dan *cartesian*. Beberapa jenis *manipulator* berdasarkan bentuk geometrinya antara lain, *Articulate/Antromorphic*, *Cartesian*, dan SCARA [2]. Beberapa Bentuk fisik dari *manipulator* tersebut ditunjukkan pada Gambar 2.2.



Gambar 2.2 Jenis-jenis *Manipulator* Berdasarkan Bentuk Geometri [2]

Dari ketiga *manipulator* tersebut memiliki perbedaan dari konfigurasi *joint*. *Articulate/antromorphic* terdiri atas enam *revolute joint*, SCARA terdiri atas dua atau tiga *revolute joint* dan satu *prismatic joint* yang terletak pada *end-effector*. *Cartesian* memiliki tiga sampai lima *prismatic joint*. Dalam melakukan tugas yang diprogramkan, *manipulator* harus memenuhi tiga syarat yaitu:

- *Manipulator* tidak boleh membahayakan manusia.
- *Manipulator* harus mematuhi perintah yang diberikan oleh manusia, kecuali perintah yang melanggar syarat pertama.

- *Manipulator* harus menjaga kondisinya sendiri selama proteksi tersebut tidak bertentangan dengan syarat pertama dan kedua.

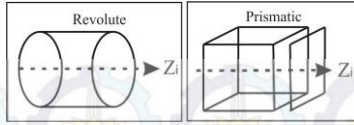
Penyelesaian tugas dengan persyaratan di atas dapat dipenuhi dengan perancangan dan perencanaan terkait kontrol dan *trajectory planning* dari *manipulator* tersebut. Kontrol *manipulator* berhubungan dengan model dinamika dari *manipulator*. Hasil dari kontrol *manipulator* berhubungan dengan akurasi dan presisi. Sebagian besar *manipulator* saat ini memiliki presisi yang tinggi, namun tidak terlalu akurat. Pengukuran nilai variabel *joint* umumnya menggunakan *encoder*. *Encoder* diletakkan pada *joint* atau lebih tepatnya pada *shaft* dari motor servo. Posisi dan orientasi dari *end-effector* umumnya tidak diukur secara langsung. Oleh karena itu pengukuran dari posisi dan orientasi *end-effector* menggunakan kinematika berdasarkan nilai variabel *joint*.

2.2. Kinematika *Manipulator* [2]

Kinematika merupakan studi tentang gerak dari suatu benda tanpa menghiraukan gaya yang menyebabkannya. Dalam penerapannya untuk *manipulator*, kinematika digunakan untuk mengetahui posisi dan orientasi dari *end-effector* terhadap *base frame manipulator* serta benda lain yang ada dalam lingkup *workspace manipulator*. Posisi dan orientasi ini dapat direpresentasikan dalam ruang kartesian maupun ruang *joint*. Untuk memperoleh koordinat kartesian berdasarkan nilai-nilai sudut tiap *joint* digunakan *forward kinematic* sedangkan sebaliknya digunakan *inverse kinematic*.

2.2.1. *Forward Kinematic* [2]

Perhitungan *forward kinematic* menggunakan metode yang biasa disebut dengan kaidah Denavit-Hartenberg. Kaidah ini merepresentasikan tiap transformasi homogen ' A_i ' sebagai perkalian dari empat transformasi dasar yang ditunjukkan pada Persamaan 2.1. Matriks transformasi homogen A_i memiliki empat parameter yaitu θ_i , a_i , d_i , dan α_i . Untuk *prismatic joint* nilai d_i berupa variabel, sedangkan untuk *revolute joint* nilai θ_i berupa variabel. Perhitungan kuantitas tersebut dilakukan dengan cara menentukan *frame* untuk tiap *joint manipulator* yang terbagi menjadi sembilan langkah. Langkah pertama dalam pembentukan *frame* yaitu menentukan sumbu *joint* $z_0, \dots, z_{n-1}$ dimana n adalah jumlah *joint manipulator*. Sumbu *joint manipulator* ditunjukkan pada Gambar 2.3.



Gambar 2.3 Sumbu *Joint Manipulator*

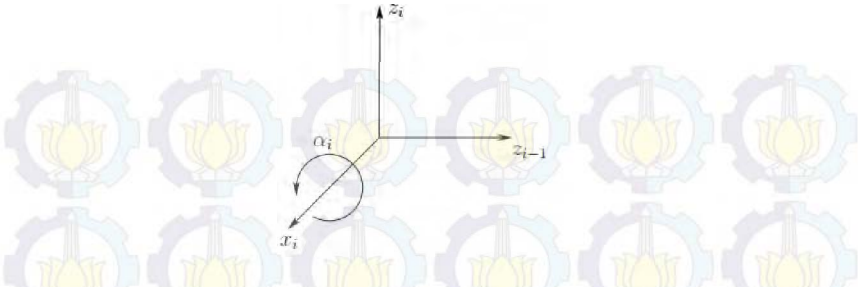
Langkah kedua yaitu menentukan *base frame*. Titik *origin* o_0 diletakkan bebas di sepanjang sumbu z_0 . Sumbu x_0 dan y_0 dibentuk berdasarkan kaidah *right-hand frame*. Langkah ketiga yaitu dibentuk *origin* o_i dimana $i = 1, \dots, n-1$ tegak lurus terhadap z_i dan z_{i-1} memotong z_i . Untuk z_i berpotongan dengan z_{i-1} , o_i diletakkan pada titik perpotongan tersebut, sedangkan jika z_i paralel dengan z_{i-1} , o_i diletakkan pada posisi yang sesuai di sepanjang z_i . Langkah keempat yaitu dibentuk sumbu x_i sepanjang tegak lurus antara z_{i-1} dan z_i melalui o_i atau searah tegak lurus terhadap bidang z_{i-1} - z_i jika z_{i-1} dan z_i berpotongan. Langkah kelima yaitu dibentuk sumbu y_i sesuai dengan kaidah *right-hand frame*. Langkah keenam yaitu dibentuk *end-effector frame* $o_n x_n y_n z_n$. Diasumsikan *joint* ke- n adalah *revolute*, bentuk z_n sepanjang arah z_{n-1} . Dibentuk titik *origin* o_n sepanjang z_n , sebaiknya ditengah *gripper* atau *tool end-effector*. Dibentuk y_n searah penutupan *gripper* dan $x_n = y_n \times z_n$. Langkah ketujuh yaitu membuat tabel parameter θ_i , a_i , d_i , dan α_i dengan ketentuan sebagai berikut:

- θ_i : Sudut antara x_{i-1} dan x_i diukur terhadap z_{i-1} .



Gambar 2.4 Acuan Pengukuran θ_i

- α_i : Sudut antara z_{i-1} dan z_i diukur terhadap x_i .



Gambar 2.5 Acuan Pengukuran α_i

- a_i : Jarak sepanjang x_i dari o_i hingga perpotongan sumbu x_i dan z_{i-1} .
- d_i : Jarak sepanjang z_{i-1} dari o_{i-1} hingga perpotongan sumbu x_i dan z_{i-1} .

Langkah kedelapan yaitu dibentuk matriks transformasi homogen A_i dengan mensubstitusi parameter pada langkah ketujuh ke Persamaan 2.1.

$$\begin{aligned}
 A_i &= \text{Rotasi}_{z, \theta_i} \text{Translasi}_{z, d_i} \text{Translasi}_{x, a_i} \text{Rotasi}_{x, \alpha_i} \\
 &= \begin{bmatrix} R_i^{i-1} & p_i^{i-1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)
 \end{aligned}$$

dengan matriks rotasi $R_3 = [n \ s \ a]$, dan matriks posisi $p_{3 \times 1} = [p_x \ p_y \ p_z]^T$. Langkah kesembilan yaitu membentuk matriks T_n^{i-1} untuk *manipulator* dengan n -DOF yang ditunjukkan pada Persamaan 2.2.

$$T_n^{i-1} = A_i A_{i+1} \dots A_n, \quad i = 1 \quad (2.2)$$

2.2.2. Inverse Kinematic [2]

Inverse kinematic merupakan suatu perhitungan nilai-nilai variabel tiap *joint manipulator* berdasarkan posisi dan orientasi yang telah

diketahui sebelumnya. Pada umumnya metode aljabar digunakan untuk menyelesaikan perhitungan ini. Perhitungan tersebut dilakukan dengan cara menyelesaikan persamaan-persamaan yang berisi variabel *joint* pada elemen dari matriks transformasi homogen berdasarkan matriks rotasi dan posisi yang telah diketahui. Perhitungan metode aljabar menjadi sulit dilakukan untuk *manipulator* yang memiliki *joint* banyak. Contoh persoalan *inverse kinematic* untuk *Stanford Manipulator*[3] ditunjukkan pada Persamaan 2.3.

$$n_x = C_1 [C_2 (C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] - S_1 (S_4 C_5 C_6 + C_4 S_6) \quad (2.3.a)$$

$$n_y = S_1 [C_2 (C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] + C_1 (S_4 C_5 C_6 + C_4 S_6) \quad (2.3.b)$$

$$n_z = -S_2 (C_4 C_5 C_6 - S_4 S_6) - C_2 S_5 S_6 \quad (2.3.c)$$

$$s_x = C_1 [-C_2 (C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] - S_1 (-S_4 C_5 S_6 + C_4 C_6) \quad (2.3.d)$$

$$s_y = S_1 [-C_2 (C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] + C_1 (-S_4 C_5 S_6 + C_4 C_6) \quad (2.3.e)$$

$$s_z = S_2 (C_4 C_5 S_6 + S_4 C_6) + C_2 S_5 S_6 \quad (2.3.f)$$

$$a_x = C_1 (C_2 C_4 S_5 + S_2 C_5) - S_1 S_4 S_5 \quad (2.3.g)$$

$$a_y = S_1 (C_2 C_4 S_5 + S_2 C_5) + C_1 S_4 S_5 \quad (2.3.h)$$

$$a_z = -S_2 C_4 S_5 + C_2 C_5 \quad (2.3.i)$$

$$p_x = C_1 S_2 d_3 - S_1 d_2 + d_6 (C_1 C_2 C_4 S_5 + C_1 C_5 S_2 - S_1 S_4 S_5) \quad (2.3.j)$$

$$p_y = S_1 S_2 d_3 + C_1 d_2 + d_6 (C_1 S_4 S_5 + C_2 C_4 S_1 S_5 - C_5 S_1 S_2) \quad (2.3.k)$$

$$p_z = C_2 d_3 + d_6 (C_2 C_5 - C_2 S_4 S_5) \quad (2.3.l)$$

Keterangan:

$C_i = \cos \theta_i$, $S_i = \sin \theta_i$, d_i = Parameter D-H untuk panjang *link* ke- i

Pada Persamaan 2.3 tampak bahwa penyelesaian tiap variabel *joint* harus memenuhi semua persamaan. Metode lain yang dapat digunakan untuk persoalan *inverse kinematic* antara lain pendekatan geometri, dan melalui *Jacobian manipulator*.

2.3. *Jacobian Manipulator* [2]

Pada penjelasan *forward kinematic* diketahui bahwa dari nilai-nilai variabel *joint* yang diketahui, dapat peroleh nilai posisi dan orientasi dari

end-effector manipulator tersebut. Untuk menentukan hubungan antara kecepatan linear dengan kecepatan sudut dari *end-effector*, digunakan suatu fungsi yang disebut *Jacobian*. Melalui *Jacobian* dapat pula diperoleh nilai-nilai variabel *joint* berdasarkan posisi dan orientasi dari *end-effector* yang telah diketahui.

Hubungan antara kecepatan linear dengan kecepatan sudut dari *end-effector* ditunjukkan pada Persamaan 2.4.

$$v = J_{n \times n} \dot{\theta} \quad (2.4)$$

dengan:

v : Vektor kecepatan *end-effector*, $v \in R^n$

$J_{n \times n}$: *Jacobian manipulator*, n adalah jumlah DOF manipulator

$\dot{\theta}$: Kecepatan sudut *joint*, $\dot{\theta} \in R^n$

Untuk Manipulator n -DOF variabel *joint* $\theta_1, \theta_2, \dots, \theta_n$ memiliki matriks transformasi homogen seperti yang ditunjukkan pada Persamaan 2.5.

$$T_n^0(\theta) = \begin{bmatrix} R_{n \times 3}^0(\theta) & p_{n \times 1}^0(\theta) \\ 0 & 1 \end{bmatrix} \quad (2.5)$$

Dari matriks transformasi homogen pada Persamaan 2.5 diperoleh kecepatan linear $\dot{v}_n^0$ dari *end-effector* melalui perpindahan posisi *end-effector* seperti yang ditunjukkan pada Persamaan 2.6.

$$\dot{v}_{n \times 1}^0 = \dot{p}_{n \times 1}^0 \quad (2.6)$$

Untuk kecepatan sudut dari *end-effector* $\omega_{n \times 1}^0$ diperoleh melalui representasi sudut euler pada Persamaan 2.6.

$$\omega_{n \times 1}^0 = \begin{bmatrix} 1 & 0 & s\theta_p \\ 0 & c\theta_r & -c\theta_p s\theta_r \\ 0 & s\theta_r & c\theta_p s\theta_r \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_p \\ \dot{\theta}_y \end{bmatrix} \quad (2.7)$$

Berdasarkan Persamaan 2.4, diperoleh hubungan antara vektor kecepatan *end-effector* dengan kecepatan sudut *joint* melalui Persamaan 2.8 berikut

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{\theta} = \begin{bmatrix} J_v \dot{\theta} \\ J_\omega \dot{\theta} \end{bmatrix} \quad (2.8)$$

Dengan J_v dan J_ω berupa matriks $3 \times n$ merupakan *jacobian* untuk kecepatan linear dan kecepatan sudut dari *end-effector*. Oleh karena semua *joint* berupa *revolute*, maka *jacobian* untuk kecepatan sudut *end-effector* diperoleh melalui Persamaan 2.9

$$J_\omega = [z_0 \ z_1 \ \dots \ z_5] \quad (2.9)$$

dengan $z_{i-1}^0 = R_{i-1}^0 k$ untuk $i = 1, \dots, n$. $k = [0 \ 0 \ 1]^T$ adalah vektor koordinat satuan sumbu z dan $z_0^0 = k$. Untuk *jacobian* kecepatan linear *end-effector* diperoleh melalui Persamaan 2.10

$$J_{vi} = z_{i-1}^0 \times (p_n^0 - p_{i-1}^0) \quad (2.10)$$

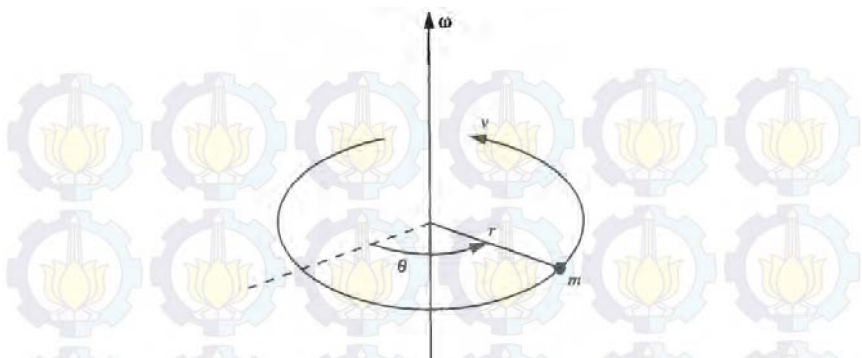
dengan $\times$ adalah notasi *cross product* dan $p_0^0 = [0 \ 0 \ 0]^T$.

2.4. Model Dinamika PUMA 560 [4]

Secara umum pemodelan *manipulator* diturunkan menggunakan dinamika euler-lagrange. Dinamika tersebut berisi perhitungan energi kinetik dan potensial dari tiap *link manipulator*. Perhitungan energi kinetik dan potensial berhubungan dengan gaya, momen inersia dan energi yang mempengaruhi tiap *link manipulator*.

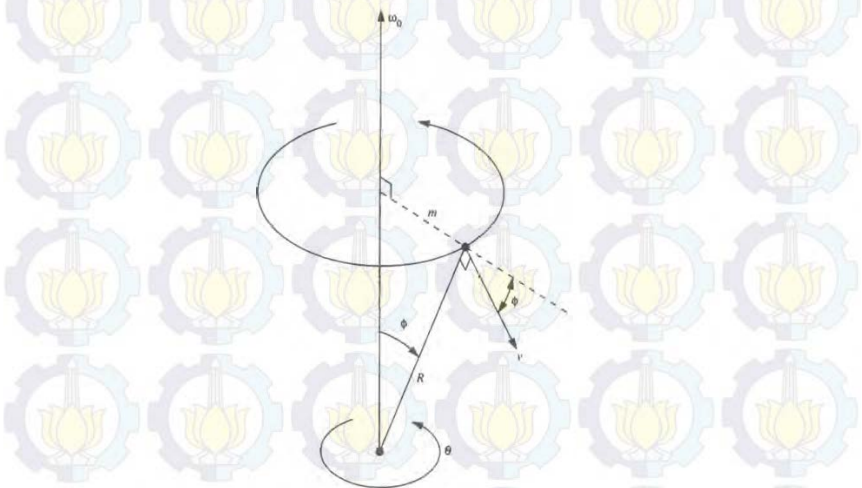
Gaya sentripetal dari suatu massa m yang mengitari titik dengan radius r dan kecepatan sudut ω yang ditunjukkan pada Gambar 2.6 diperoleh berdasarkan Persamaan 2.11.

$$F_{sen} = \frac{mv^2}{r} = m\omega^2 r = m\dot{\theta}^2 r \quad (2.11)$$



Gambar 2.6 Gaya Sentripetal

Diasumsikan suatu lingkaran berputar terhadap pusatnya dengan kecepatan sudut ω_0 seperti yang ditunjukkan pada Gambar 2.7.



Gambar 2.7 Gaya Koriolis

Gaya koriolis pada suatu benda dengan massa m yang bergerak dengan kecepatan v pada permukaan lingkaran tersebut ditunjukkan pada Persamaan 2.12 dengan menggunakan kaidah putaran tangan kanan.

$$F_{kor} = -2m\omega_0 \times v \quad (2.12)$$

Energi kinetik suatu massa m yang bergerak dengan kecepatan linear v ditunjukkan pada Persamaan 2.13.

$$K = \frac{1}{2}mv^2 \quad (2.13)$$

Energi kinetik rotasi $K_{rot} = \frac{1}{2}I\omega^2$ dengan $I = \int_{vol} \rho(r)r^2 dr$, dimana $\rho(r)$ adalah distribusi massa pada radius r dalam suatu volume. Untuk m yang diasumsikan sebagai suatu titik massa maka $I = mr^2$. Oleh karena itu energi kinetik rotasi ditunjukkan pada Persamaan 2.14.

$$K_{rot} = \frac{1}{2}mr^2\dot{\theta}^2 \quad (2.14)$$

Energi potensial dari suatu massa m pada ketinggian h pada suatu medan gravitasi g yang konstan diperoleh melalui Persamaan 2.15.

$$P = mgh \quad (2.15)$$

Berdasarkan gaya dan energi tersebut, persamaan gerak lagrange diperoleh dari selisih antara energi kinetik terhadap energi potensial. Secara umum persamaan lagrange tersebut ditunjukkan pada Persamaan 2.16.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \quad (2.16)$$

Pada Persamaan 2.16, q adalah variabel *joint manipulator* dalam satuan derajat untuk *revolute* atau meter untuk *prismatic* dan τ adalah torsi dengan satuan newton-meter.

Berdasarkan penjabaran energi kinetik dan potensial sebelumnya, diperoleh suatu model dinamika *manipulator* secara umum yang dinotasikan pada Persamaan 2.17.

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) = \tau \cong M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (2.17)$$

dengan:

$$M(q) = [m_{ij}(q)]_{n \times n}, i, j = 1, \dots, n$$

$$m_{ij}(q) = m_{ji}(q)$$

$$V(q, \dot{q}) + F(\dot{q}) + G(q) \cong N(q, \dot{q}) = [n_i(q, \dot{q})]_{n \times 1}, i = 1, \dots, n$$

Model dinamika PUMA 560 direpresentasikan melalui matriks M dan vektor N yang ditunjukkan pada Lampiran B.

2.5. Computed Torque Control (CTC) [4]

Metode kontrol ini merupakan salah satu cara untuk mengatur gerak *manipulator* dalam ruang *joint*. Pada metode ini digunakan *feedback linearization* untuk menghilangkan nonlinearitas dari model *error* variabel *joint*. Torsi yang dibutuhkan agar *manipulator* bergerak menuju nilai *joint* yang diinginkan dihitung menggunakan *inverse dynamic* yang diperoleh dari model dinamika *manipulator*.

Nilai variabel *joint* yang diinginkan $q_d(t)$ ditentukan sebagai referensi *manipulator*. *Error* dari variabel *joint* didefinisikan sebagai selisih nilai variabel *joint* yang diinginkan terhadap nilai variabel *joint* aktual $q(t)$ yang ditunjukkan pada Persamaan 2.18. Untuk memperoleh relasi antara torsi dengan *error* dilakukan turunan sebanyak dua kali dari *error* terhadap waktu t .

$$e_i(t) = q_{di}(t) - q_i(t), i = 1, \dots, n \quad (2.18)$$

$$\ddot{e}_i(t) = \ddot{q}_{di}(t) - \ddot{q}_i(t), i = 1, \dots, n \quad (2.19)$$

Pada Persamaan 2.19, $\ddot{q}_i(t)$ disubstitusikan berdasarkan model dinamika pada Persamaan 2.17, sehingga diperoleh hubungan antara torsi dan *error* yang ditunjukkan pada Persamaan 2.20.

$$\ddot{e}_i(t) = \ddot{q}_{di}(t) + M^{-1}(N - \tau_i), i = 1, \dots, n \quad (2.20)$$

Didefinisikan suatu fungsi masukan kontrol yaitu $u_i = \ddot{q}_{di}(t) + M^{-1}(N - \tau_i)$ dan *state variable* $x = [e_i \dot{e}_i]^T$, diperoleh suatu model linear dari *error* variabel *joint* yang ditunjukkan pada Persamaan 2.21.

$$\frac{d}{dt} \begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (2.21)$$

Sistem linear dari Persamaan 2.21 disebut bentuk *Brunovsky canonical* dengan nilai *error* dan *delta error* sebagai *state variable* untuk tiap *joint* dari *manipulator*. Metode *feedback linearization* dilakukan pada pendefinisian fungsi masukan kontrol u . Torsi masukan untuk *manipulator* ditunjukkan pada Persamaan 2.22.

$$\tau_i = M(\ddot{q}_{di} - u_i) + N \quad (2.22)$$

Masukan kontrol dapat dilakukan dengan metode kontrol yang dapat membuat *error* dan *delta error* menjadi nol.

2.6. Linear Quadratic Regulator (LQR) [4]

Metode *linear quadratic regulator* merupakan salah satu metode kontrol optimal yang digunakan untuk membuat *state variable* yang telah didefinisikan pada Subbab 2.5 menuju ke nol.

Suatu sistem LTI dalam bentuk *state space* ditunjukkan pada Persamaan 2.23.

$$\dot{x} = Ax + Bu \quad (2.23)$$

dengan *state feedback gain* K dipilih untuk memperoleh *closed-loop* stabil yang ditunjukkan pada Persamaan 2.24.

$$\dot{x} = (A - BK)x \quad (2.24)$$

Didefinisikan suatu indeks performansi PI yang ditunjukkan pada Persamaan 2.25.

$$PI = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.25)$$

Q adalah matriks semidefinit positif pembobot *state* dan R adalah matriks pembobot kontrol. Matriks Q dipilih dengan cara membentuk matriks diagonal dengan nilai elemen diagonal lebih dari nol. Nilai R akan mempengaruhi besarnya sinyal kontrol yang dikeluarkan. Untuk kebutuhan sistem dengan energi minimal, maka nilai R yang dipilih dibuat besar.

Nilai K diperoleh dengan menyelesaikan aljabar Riccati ($A^T P + PA + Q - PBR^{-1}B^T P = 0$) dengan $K = R^{-1}B^T P$.

2.7. Perencanaan Trayektori

Perencanaan trayektori dilakukan dengan menentukan titik-titik pada koordinat kartesian. Pergerakan dari titik ke titik selanjutnya dibuat dengan menentukan posisi awal, posisi tujuan, kecepatan awal, kecepatan tujuan, percepatan awal, percepatan tujuan, waktu awal, dan waktu tujuan. Dari beberapa ketentuan tersebut dihitung persamaan polinomial yang disebut *quintic cubical polynomial* yang ditunjukkan pada Persamaan 2.26 [2].

$$p_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \quad (2.26.a)$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \quad (2.26.b)$$

$$ac_0 = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \quad (2.26.c)$$

$$p_d = a_0 + a_1 t_d + a_2 t_d^2 + a_3 t_d^3 + a_4 t_d^4 + a_5 t_d^5 \quad (2.26.d)$$

$$v_d = a_1 + 2a_2 t_d + 3a_3 t_d^2 + 4a_4 t_d^3 + 5a_5 t_d^4 \quad (2.26.e)$$

$$ac_d = 2a_2 + 6a_3 t_d + 12a_4 t_d^2 + 20a_5 t_d^3 \quad (2.26.f)$$

Dari Persamaan 2.26 diubah ke bentuk matriks menjadi Persamaan 2.27.

$$Da = b$$

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_d & t_d^2 & t_d^3 & t_d^4 & t_d^5 \\ 0 & 1 & 2t_d & 3t_d^2 & 4t_d^3 & 5t_d^4 \\ 0 & 0 & 2 & 6t_d & 12t_d^2 & 20t_d^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} p_0 \\ v_0 \\ ac_0 \\ p_d \\ v_d \\ ac_d \end{bmatrix} \quad (2.27)$$

Koefisien a_0 a_1 a_2 a_3 a_4 a_5 pada matriks a dihitung dengan cara $a = D^{-1}b$.

2.8. Spesifikasi PUMA 560

Spesifikasi PUMA 560 digunakan untuk perancangan simulasi sistem. Spesifikasi terdiri dari batas gerak *joint*, dan torsi maksimum yang mampu diterapkan pada PUMA 560. Batas gerak *joint* PUMA 560 ditunjukkan pada Tabel 2.1 dan torsi maksimum PUMA 560 ditunjukkan pada Tabel 2.2. Bentuk fisik dari PUMA 560 ditunjukkan pada Gambar 2.8.

Tabel 2.1 Batas Gerak *Joint* PUMA 560 [5]

<i>Joint</i>	<i>Range</i> (°)
1	-160 – +160
2	-245 – +45
3	-45 – +225
4	-110 – +170
5	-100 – +100
6	-266 – +266

Tabel 2.2 Torsi Maksimum PUMA 560 [6]

<i>Joint</i>	Torsi Maksimum (N-m)
1	97,6
2	186,4
3	89,4
4	24,2
5	20,1
6	21,3



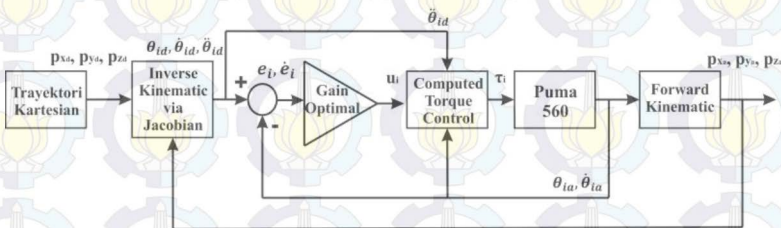
Halaman ini sengaja dikosongkan

BAB III PERANCANGAN SISTEM

Pada bab ini dibahas perancangan sistem secara lengkap meliputi model kinematika dan dinamika dari *manipulator* PUMA 560, *Jacobian manipulator*, trayektori koordinat kartesian, perancangan metode *optimal computed torque control*, dan perancangan *software* yang akan digunakan untuk simulasi.

3.1. Perancangan Sistem Keseluruhan

Perancangan sistem pengaturan gerak *manipulator* secara keseluruhan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem Keseluruhan

Secara umum pergerakan *manipulator* dikontrol menggunakan *optimal computed torque control* untuk mengikuti referensi yang diberikan yaitu posisi dari *end-effector* (p_{xd}, p_{yd}, p_{zd}). Posisi yang diberikan diubah menjadi sudut *joint* menggunakan *inverse kinematic*. Nilai sudut *joint* diturunkan terhadap waktu sebanyak dua kali sehingga diperoleh percepatan sudut *joint* yang digunakan untuk perhitungan *computed torque control*. Keluaran dari perhitungan tersebut berupa torsi untuk tiap *joint* dimasukkan ke *manipulator* PUMA 560 sehingga terjadi perubahan nilai *joint* sampai nilai *joint* aktual sama dengan nilai *joint* hasil perhitungan *inverse kinematic*. Posisi aktual dari *manipulator* diperoleh melalui perhitungan *forward kinematic* berdasarkan nilai *joint* aktual. Hasil perhitungan berupa transformasi homogen yang berisi matriks rotasi dan posisi aktual dari *end-effector* digunakan untuk perhitungan *Jacobian manipulator*.

3.2. Perancangan *Forward Kinematic* PUMA 560

Perhitungan *forward kinematic* menggunakan parameter D-H standar hasil penelitian R.P Paul yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Parameter D-H Standar PUMA 560 [6]

Joint ke- <i>i</i>	θ_i ($^{\circ}$)	α_i ($^{\circ}$)	a_i (mm)	d_i (mm)
1	*	90	0	0
2	*	0	431,8	0
3	*	-90	19,1	125,4
4	*	90	0	431,8
5	*	-90	0	0
6	*	0	0	0

Keterangan: * Menandakan Nilai Parameter Berupa Variabel Sesuai Spesifikasi PUMA 560

Penurunan matriks transformasi yang digunakan untuk menghitung posisi dan orientasi dari *end-effector* yaitu $T_6^0 = A_1 A_2 A_3 A_4 A_5 A_6$ dengan nilai-nilai A_i untuk $i = 1, 2, \dots, 6$ ditunjukkan pada Persamaan 3.1 hingga Persamaan 3.6.

$$A_1 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$A_2 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 431,8c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & 431,8s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$A_3 = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & 19,1c\theta_3 \\ s\theta_3 & 0 & c\theta_3 & 19,1s\theta_3 \\ 0 & -1 & 0 & 125,4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$A_4 = \begin{bmatrix} c\theta_4 & 0 & s\theta_4 & 0 \\ s\theta_4 & 0 & -c\theta_4 & 0 \\ 0 & 1 & 0 & 431,8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$A_5 = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$A_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Perancangan *forward kinematic* pada *software* Matlab 2010a ditunjukkan oleh *flowchart* pada Gambar 3.2. *Listing program* ditunjukkan pada Lampiran C.

3.3. Perancangan *Inverse Kinematic* PUMA 560

Perhitungan *inverse kinematic* dilakukan menghitung matriks *jacobian* manipulator dengan masukan berupa posisi *end-effector* dari manipulator. Orientasi dari *end-effector* dibuat tetap dengan matriks rotasi $R_3 = I_3$. Perubahan sudut *joint* diperoleh dengan melakukan perkalian matriks *jacobian* “ J ” dengan vektor $v_{6 \times 1} = [v_n \ \omega_n]^T$ dimana:

$$v_{n3 \times 1} = [v_x \ v_y \ v_z]^T \quad (3.7)$$

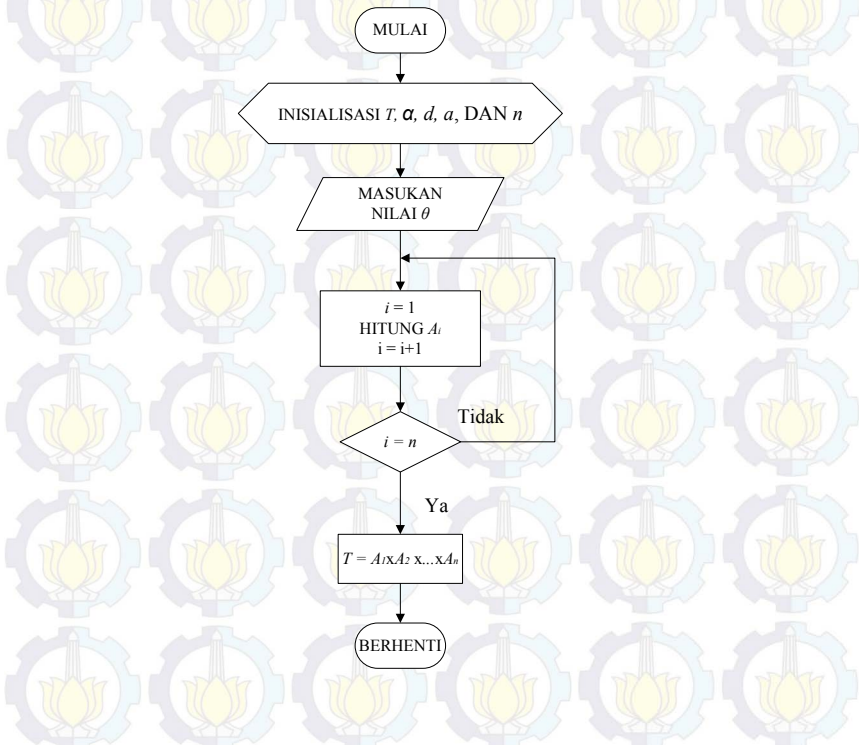
$$\omega_{n3 \times 1} = \begin{bmatrix} 1 & 0 & s\theta_p \\ 0 & c\theta_r & -c\theta_p s\theta_r \\ 0 & s\theta_r & c\theta_p s\theta_r \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_p \\ \dot{\theta}_y \end{bmatrix} \quad (3.8)$$

Dengan matriks *jacobian* manipulator yang ditunjukkan pada Persamaan 3.9 dapat dihitung nilai perubahan sudut *joint* berdasarkan perpindahan titik pada koordinat kartesian dengan menggunakan Persamaan 3.10.

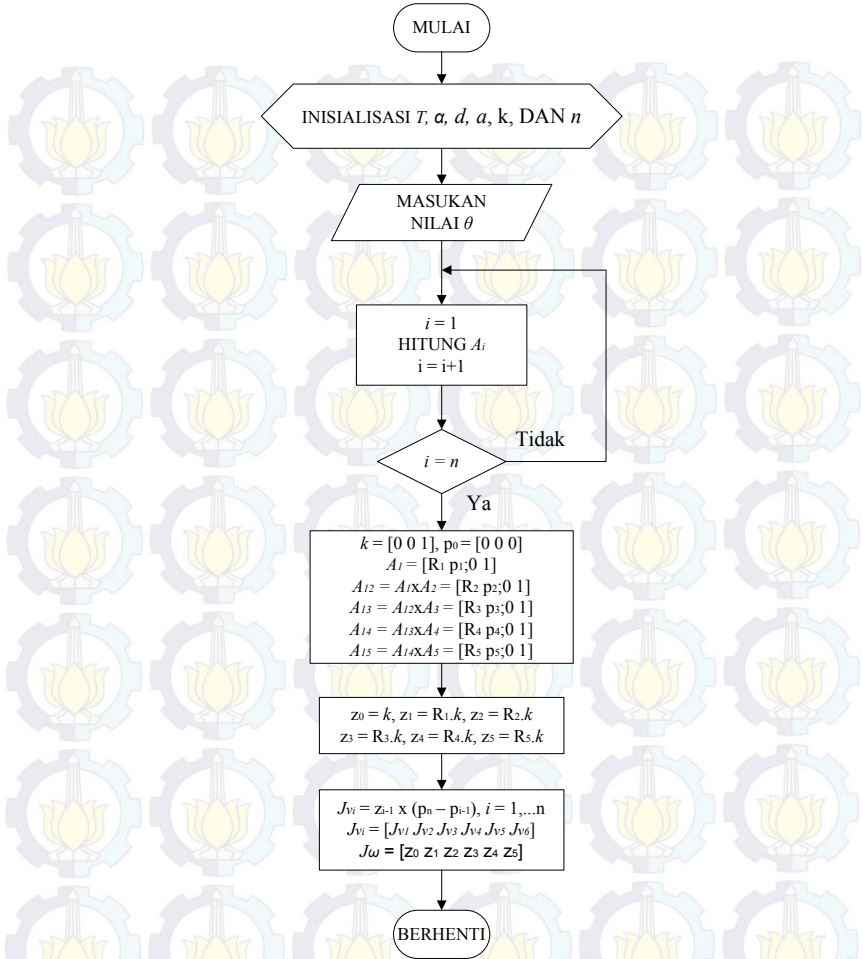
$$J = \begin{bmatrix} J_{v0} & J_{v1} & J_{v2} & J_{v3} & J_{v4} & J_{v5} \\ J_{\omega1} & J_{\omega2} & J_{\omega3} & J_{\omega4} & J_{\omega4} & J_{\omega5} \end{bmatrix} \quad (3.9)$$

$$\dot{\theta} = J^{-1}v = J^{-1} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_{n \times 1} \end{bmatrix} \quad (3.10)$$

Perancangan *jacobian* pada Matlab ditunjukkan melalui *flowchart* pada Gambar 3.3. *Listing program* ditunjukkan pada Lampiran C.

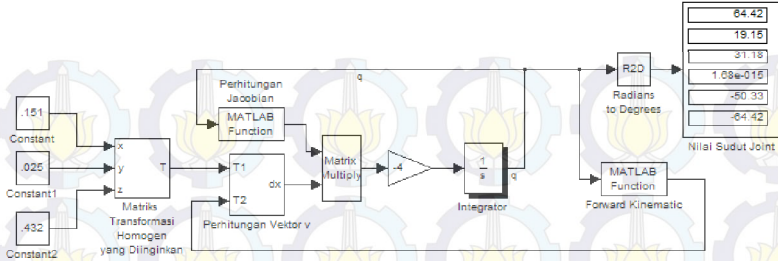


Gambar 3.2 *Flowchart* Perancangan *Forward Kinematic* PUMA 560 Menggunakan Matlab



Gambar 3.3 Perancangan *Jacobian* PUMA 560

Perubahan sudut *joint* tersebut diintegrasikan untuk memperoleh nilai sudut *joint* yang akan dibentuk untuk mencapai posisi *end-effector* yang diinginkan. Perancangan keseluruhan *inverse kinematic* menggunakan *jacobian* pada Matlab ditunjukkan pada Gambar 3.4.



Gambar 3.4 Perancangan *Inverse Kinematic* PUMA 560

3.4. Perancangan Model Dinamika PUMA 560

Pemodelan dinamika dari PUMA 560 terdiri dari matriks $M_{6 \times 6}$ yang berisi konstanta inersia dan matriks $N_{6 \times 1} \equiv V + F + G$ dengan V berupa vektor Koriolis, F berupa vektor friksi viskos dan dinamik, serta G berupa matriks gravitasi. Berdasarkan Persamaan 2.16, diperoleh elemen-elemen matriks M dan N ditunjukkan pada Persamaan 3.9 dan 3.10 [4].

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 & 0 & 0 \\ m_{12} & m_{22} & m_{23} & 0 & 0 & 0 \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} & 0 \\ 0 & 0 & m_{34} & m_{44} & 0 & 0 \\ 0 & 0 & m_{35} & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} \quad (3.9)$$

$$N = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ 0 \end{bmatrix} \quad (3.10)$$

Beberapa konstanta yang digunakan sebagai penyusun model dinamik PUMA 560 ditunjukkan pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.2 Konstanta Inersia PUMA 560 [7]

$I_1 = 1,43 \pm 0,05$	$I_{13} = -3,79 \times 10^{-3} \pm 0,9 \times 10^{-3}$
$I_2 = 1,75 \pm 0,05$	$I_{14} = 1,64 \times 10^{-3} \pm 0,07 \times 10^{-3}$
$I_3 = 1,38 \pm 0,05$	$I_{15} = 1,25 \times 10^{-3} \pm 0,3 \times 10^{-3}$
$I_4 = 6,9 \times 10^{-1} \pm 0,2 \times 10^{-1}$	$I_{16} = 1,24 \times 10^{-3} \pm 0,3 \times 10^{-3}$
$I_5 = 3,72 \times 10^{-1} \pm 0,31 \times 10^{-1}$	$I_{17} = 6,42 \times 10^{-4} \pm 3 \times 10^{-4}$
$I_6 = 3,33 \times 10^{-1} \pm 0,16 \times 10^{-1}$	$I_{18} = 4,31 \times 10^{-4} \pm 1,3 \times 10^{-4}$
$I_7 = 2,98 \times 10^{-1} \pm 0,29 \times 10^{-1}$	$I_{19} = 3 \times 10^{-4} \pm 14 \times 10^{-4}$
$I_8 = -1,34 \times 10^{-1} \pm 0,14 \times 10^{-1}$	$I_{20} = -2,02 \times 10^{-4} \pm 8 \times 10^{-4}$
$I_9 = 2,38 \times 10^{-2} \pm 1,2 \times 10^{-2}$	$I_{21} = -1 \times 10^{-4} \pm 6 \times 10^{-4}$
$I_{10} = -2,13 \times 10^{-2} \pm 0,22 \times 10^{-2}$	$I_{22} = -5,8 \times 10^{-5} \pm 1,5 \times 10^{-5}$
$I_{11} = -1,42 \times 10^{-2} \pm 0,7 \times 10^{-2}$	$I_{23} = 4 \times 10^{-5} \pm 2 \times 10^{-5}$
$I_{12} = -1,1 \times 10^{-2} \pm 0,11 \times 10^{-2}$	
$I_{m1} = 1,14 \pm 0,27$	$I_{m4} = 2 \times 10^{-1} \pm 0,16 \times 10^{-1}$
$I_{m2} = 4,71 \pm 0,54$	$I_{m5} = 1,79 \times 10^{-1} \pm 0,14 \times 10^{-1}$
$I_{m3} = 8,27 \times 10^{-1} \pm 0,93 \times 10^{-1}$	$I_{m6} = 1,93 \times 10^{-1} \pm 0,16 \times 10^{-1}$

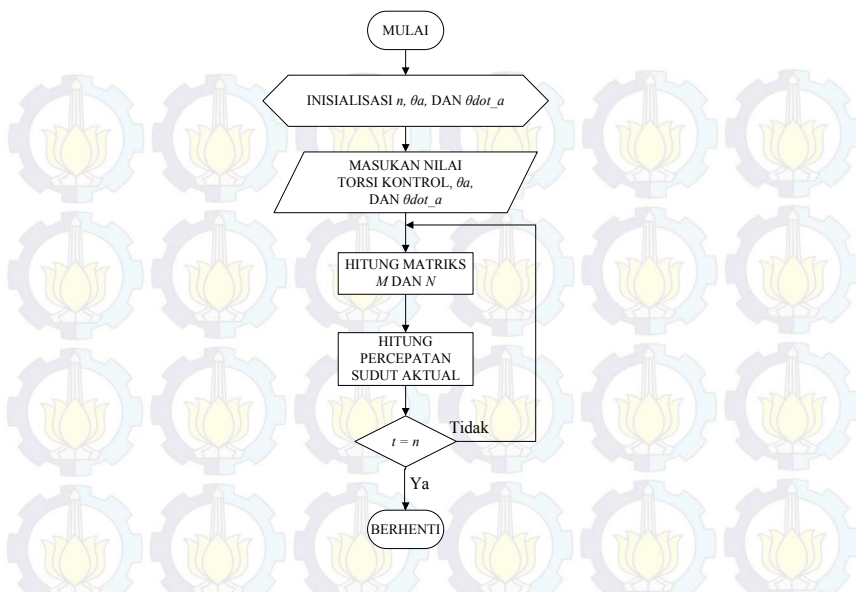
Keterangan: Satuan dari konstanta inersia adalah kg/m^2 .

Tabel 3.3 Konstanta Gravitasi PUMA 560 [7]

$g_1 = -37,2 \pm 0,05$	$g_4 = 2,49 \times 10^{-1} \pm 0,25 \times 10^{-1}$
$g_2 = -8,44 \pm 0,2$	$g_5 = -2,82 \times 10^{-2} \pm 0,56 \times 10^{-2}$
$g_3 = 1,02 \pm 0,5$	

Keterangan: Satuan dari konstanta gravitasi adalah newton-meter.

Berdasarkan konstanta tersebut, dibentuk nilai-nilai untuk elemen matriks pada model dinamika *manipulator*. Nilai dari tiap elemen ditunjukkan pada Lampiran B. Pada simulasi semua elemen didefinisikan dengan masukan antara lain torsi kontrol, posisi sudut *joint* aktual, dan kecepatan sudut *joint* aktual. Perancangan model dinamika PUMA 560 melalui *flowchart* pada Gambar 3.5. *Listing program* ditunjukkan pada Lampiran C.



Gambar 3.5 Flowchart Perancangan Model Dinamika PUMA 560

Keluaran dari model dinamika berupa percepatan aktual yang kemudian diintegrasikan dua kali untuk memperoleh posisi dan kecepatan *joint*. Nilai posisi *joint* dibatasi sesuai dengan spesifikasi sudut maksimum pergerakan tiap *joint*.

3.5. Perancangan Kontroler *Linear Quadratic Regulator* (LQR)

Pada perancangan ini ditentukan matriks Q sebagai pembobot *state variable* dari sistem yang akan dikontrol dan konstanta R untuk pembobot sinyal kontrol. Matriks A dan B diperoleh dari model manipulator setelah dilakukan linearisasi pada Persamaan 2.21. Matriks A dan B dibuat untuk masing-masing *joint*. Matriks P , solusi dari aljabar Riccati dihitung menggunakan fungsi matlab “[K P e] = lqr (A, B, Q, R, N)” dimana $K = [K_1 K_2]$ digunakan sebagai *gain* untuk *error* dan *delta error* pada tiap *joint* dan nilai $N = 0$. Untuk perancangan didefinisikan A , B , Q , dan R sebagai berikut:

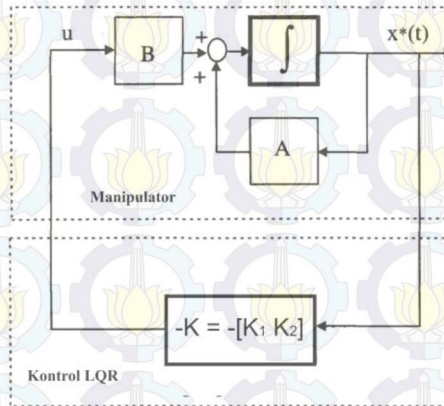
$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (3.11)$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.12)$$

$$Q = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix}, \text{ dengan } Q_{11}, Q_{22} > 0 \quad (3.13)$$

$$R = c, \text{ dengan } c > 0 \quad (3.14)$$

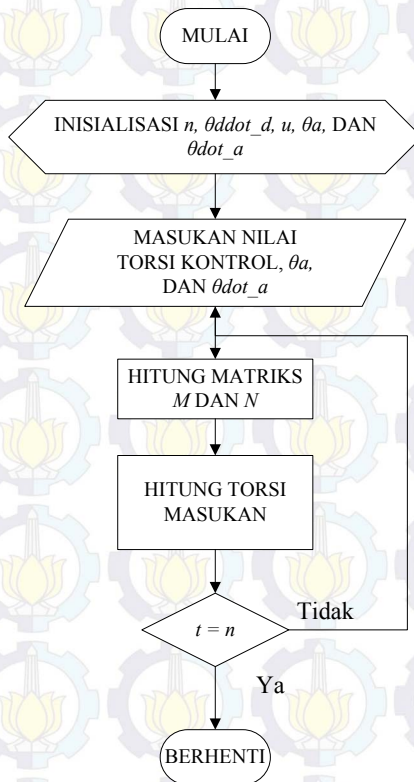
Dari Persamaan 3.11 hingga Persamaan 3.14, dihitung matriks solusi aljabar Ricatti dengan keluaran yaitu *gain* K , solusi Riccati P , dan nilai eigen e dari sistem baru. Nilai dari *state variable* $x = [e_i \dot{e}_i]$ akan semakin cepat menuju ke nol jika elemen matriks pembobot Q_{11} dan Q_{22} semakin besar. Pada simulasi ini nilai R dibuat tetap dan pembobot Q diubah-ubah untuk memperoleh respons yang diinginkan. Diagram blok dari LQR ditunjukkan pada Gambar 3.6. Hasil perkalian dari *state variable* dan gain K didefinisikan sebagai u dan digunakan sebagai masukan pada perhitungan *computed torque control*.



Gambar 3.6 Diagram Blok Kontrol LQR

3.6. Perancangan Kontroler *Computed Torque Control* (CTC)

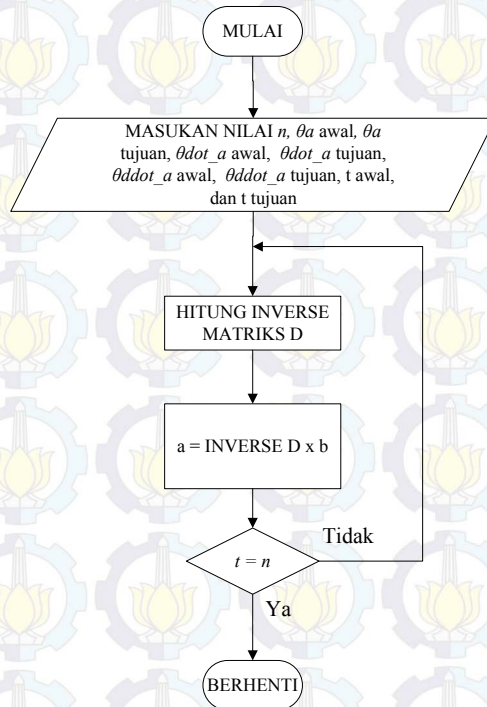
Pada bab 2 Subbab 2.5 telah dijelaskan perhitungan kontrol ini menggunakan model dinamika dari PUMA 560 untuk menghitung torsi yang dibutuhkan untuk membentuk nilai sudut *joint* yang diinginkan. Perancangan kontroler *computed torque control* melalui *flowchart* pada Gambar 3.7. *Listing program* ditunjukkan pada Lampiran C.



Gambar 3.7 *Flowchart* Perancangan Kontroler CTC

3.7. Perancangan Trayektori Kartesian

Pada perancangan ini dilakukan penentuan jalur berdasarkan titik-titik yang berada dalam ruang kartesian. Jalur yang dibentuk berbentuk lingkaran dengan radius 10 hingga 50 cm. Perancangan trayektori lingkaran pada Matlab ditunjukkan melalui *flowchart* pada Gambar 3.8. *Listing program* ditunjukkan pada Lampiran C.



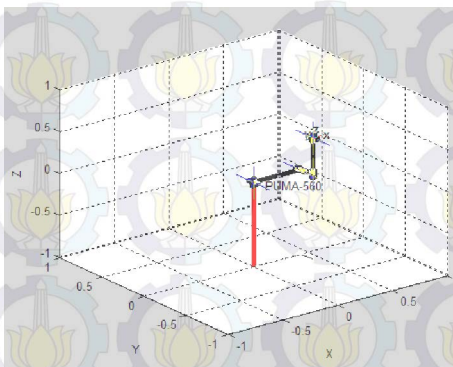
Gambar 3.8 Perancangan Trayektori Kartesian

Tujuan dari perancangan untuk penugasan *manipulator* dengan lintasan berbentuk lingkaran yaitu untuk mengetahui kemampuan PUMA 560 dalam menyelesaikan tugas dengan trayektori yang relatif sulit. Koordinat dari trayektori tersebut diubah ke bentuk sudut *joint* melalui perhitungan *inverse kinematic* yang telah dirancang pada Subbab 3.3.

3.8. Perancangan Sistem Keseluruhan Menggunakan Matlab dengan *Toolbox Robot Peter I Corke* [6]

Sistem secara keseluruhan disusun berdasarkan beberapa subsistem yang telah dijelaskan sebelumnya hingga sesuai dengan diagram blok secara keseluruhan yang telah dijelaskan pada Subbab 3.1. Diagram blok simulasi sistem secara keseluruhan ditunjukkan pada Lampiran A.

Dalam simulasi tersebut hasil dari sistem ditunjukkan dalam suatu animasi gerak manipulator. Animasi tersebut dibentuk menggunakan perintah *plot* dari Matlab berdasarkan parameter D-H PUMA 560. Bentuk animasi manipulator tersebut ditunjukkan pada Gambar 3.9.



Gambar 3.9 Animasi Manipulator pada *Software* Matlab

BAB IV

PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini dibahas hasil pengujian simulasi sistem secara lengkap meliputi respons *joint* terhadap dua sinyal uji, *inverse kinematic* dari PUMA 560, dan pengujian akurasi dari posisi *end-effector* melalui penugasan tertentu.

4.1. Pengujian dan Analisis Respons *Joint*

Pada pengujian ini dipilih beberapa sinyal uji antara lain *step* dan *cubical polynomial*. Untuk sinyal uji *step*, diberikan pada *manipulator* yang telah dikontrol menggunakan CTC dan *Optimal* CTC. Untuk metode *Optimal* CTC digunakan nilai pembobot Q dan R yang sesuai untuk mendapatkan respons dengan *settling time* paling kecil.

4.1.1. Pengujian dan Analisis Menggunakan Sinyal *Step*

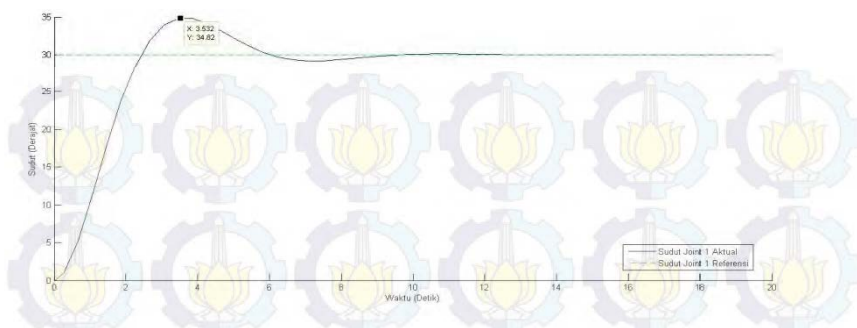
Sinyal *step* yang diberikan berupa enam nilai sudut referensi untuk masing-masing *joint* yang ditunjukkan pada Persamaan 4.1. Untuk pengujian sinyal *step* dipilih kondisi awal (IC) dari sudut *joint* yang ditunjukkan pada Persamaan 4.2. Pengujian dilakukan dengan membandingkan sistem yang dikontrol menggunakan CTC tanpa sinyal kontrol optimal terhadap CTC dengan sinyal kontrol *optimal*. Kontroler CTC tanpa sinyal kontrol *optimal* ditunjukkan pada Persamaan 4.3, sedangkan CTC dengan sinyal kontrol *optimal* ditunjukkan pada Persamaan 4.4. Nilai K_1 dan K_2 pada Persamaan 4.4 dihitung menggunakan metode LQR. Pemilihan matriks pembobot Q dan R untuk perhitungan metode LQR dilakukan berdasarkan *settling* yang paling kecil dari respons sistem. Rata-rata RMSE dan *settling time* dari respons tersebut ditunjukkan pada Tabel 4.1. Respons dari tiap *joint* yang dikontrol menggunakan CTC ditunjukkan pada Gambar 4.1 hingga Gambar 4.6.

$$\theta_d = [30^\circ \ 30^\circ \ 30^\circ \ 45^\circ \ 100^\circ \ 166^\circ]^T \quad (4.1)$$

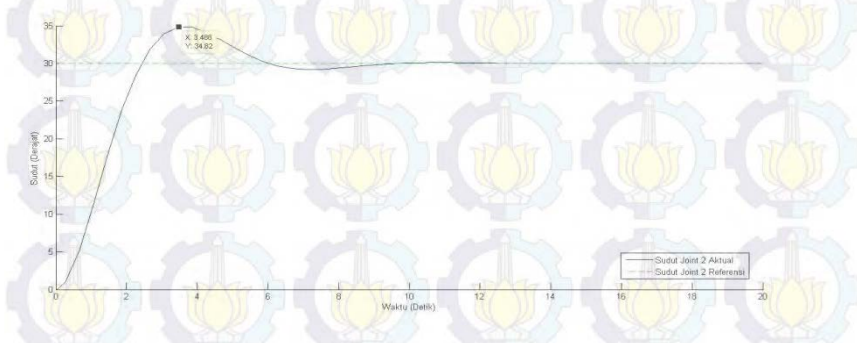
$$IC = [0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ]^T \quad (4.2)$$

$$u = -(e_i + \dot{e}_i) \quad (4.3)$$

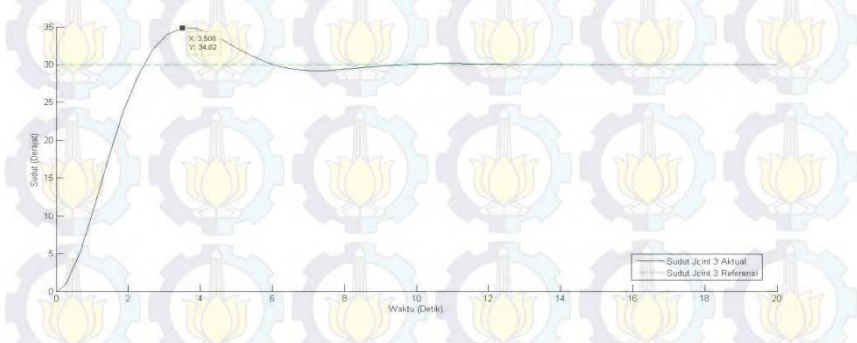
$$u = -(K_1 e_i + K_2 \dot{e}_i) \quad (4.3)$$



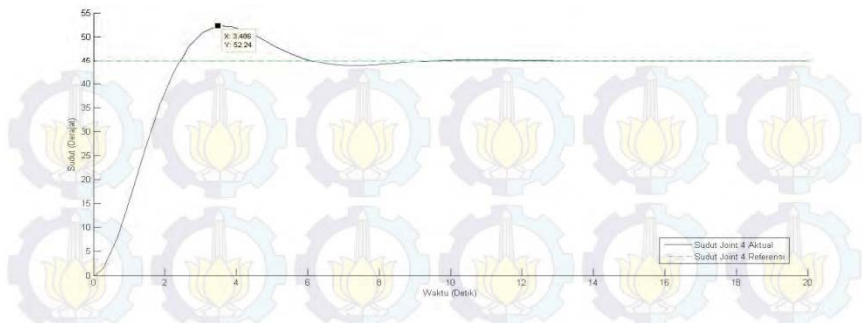
Gambar 4.1 Respons *Joint 1* CTC Terhadap Sinyal Uji Step



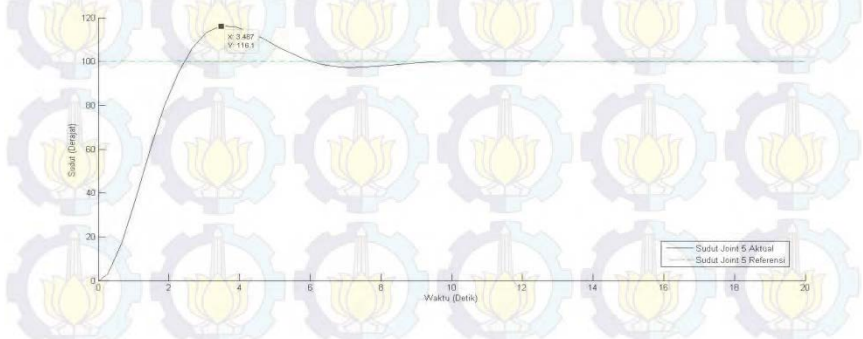
Gambar 4.2 Respons *Joint 2* CTC Terhadap Sinyal Uji Step



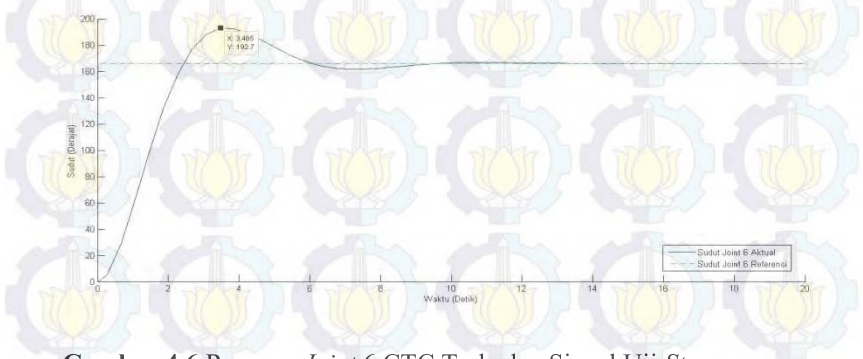
Gambar 4.3 Respons *Joint 3* CTC Terhadap Sinyal Uji Step



Gambar 4.4 Respons *Joint 4* CTC Terhadap Sinyal Uji Step



Gambar 4.5 Respons *Joint 5* CTC Terhadap Sinyal Uji Step



Gambar 4.6 Respons *Joint 6* CTC Terhadap Sinyal Uji Step

Tabel 4.1 Rata-rata Nilai RMSE dan *Settling Time* Respons Sistem

Sistem	1	2	3	4
Rata-rata RMSE ($^{\circ}$)	6,001	5,884	6,101	7,567
Rata-rata T_s (Detik)	0,336	0,375	0,472	0,564

dengan:

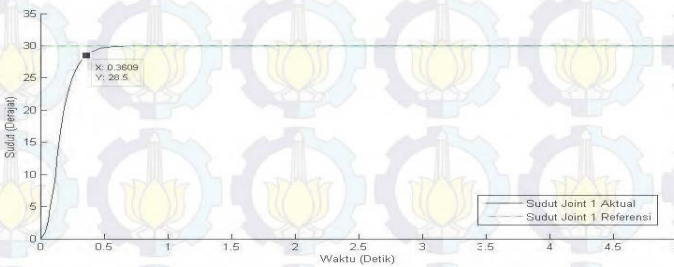
Sistem 1: $Q = [1200000 \ 0;0 \ 10000]$; $R = 2$; $K = [774,59 \ 80,9271]$

Sistem 2: $Q = [900000 \ 0;0 \ 10000]$; $R = 2$; $K = [670,82 \ 79,6344]$

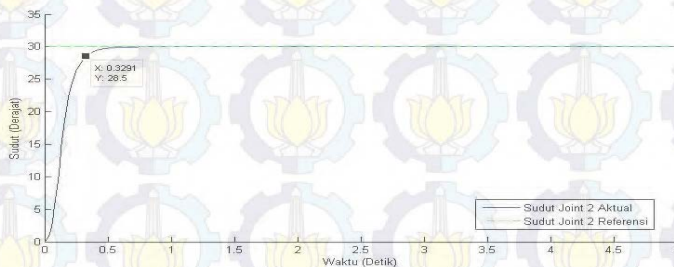
Sistem 3: $Q = [300000 \ 0;0 \ 9000]$; $R = 2$; $K = [387,298 \ 72,6264]$

Sistem 4: $Q = [100000 \ 0;0 \ 3000]$; $R = 2$; $K = [223,6068 \ 44,1272]$

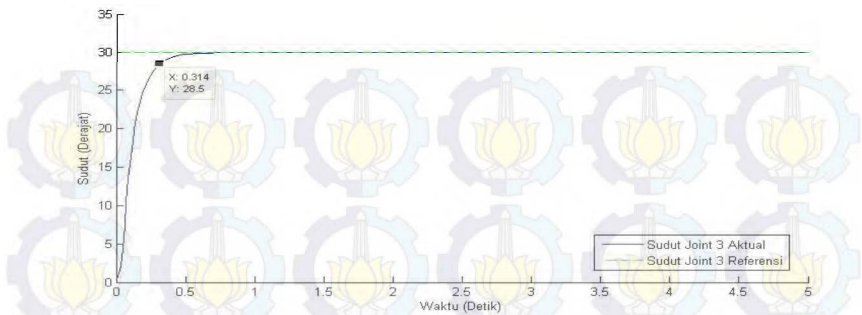
Berdasarkan kondisi tersebut, dipilih sistem 1 dengan *settling time* yang relatif kecil dibandingkan sistem lainnya. Dari Sistem 1, diperoleh nilai *gain* $K = [774,59 \ 80,9271]$. Respons dari tiap *joint* yang dikontrol menggunakan CTC dengan sinyal kontrol *optimal* ditunjukkan pada Gambar 4.7 hingga Gambar 4.12.



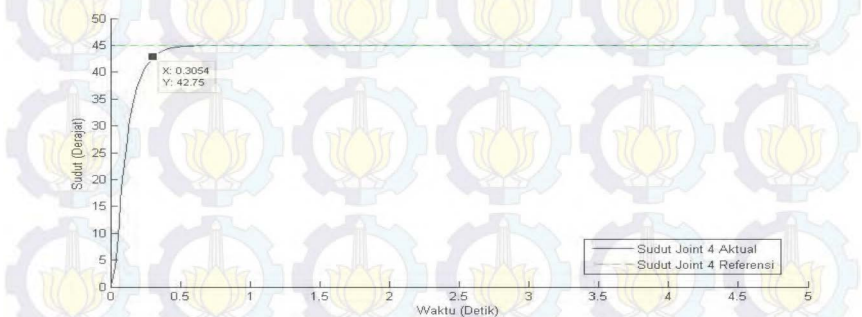
Gambar 4.7 Respons *Joint 1* *Optimal* CTC Terhadap Sinyal Uji *Step*



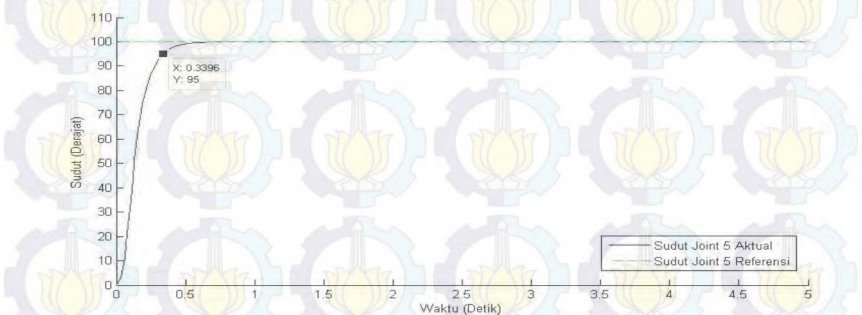
Gambar 4.8 Respons *Joint 2* *Optimal* CTC Terhadap Sinyal Uji *Step*



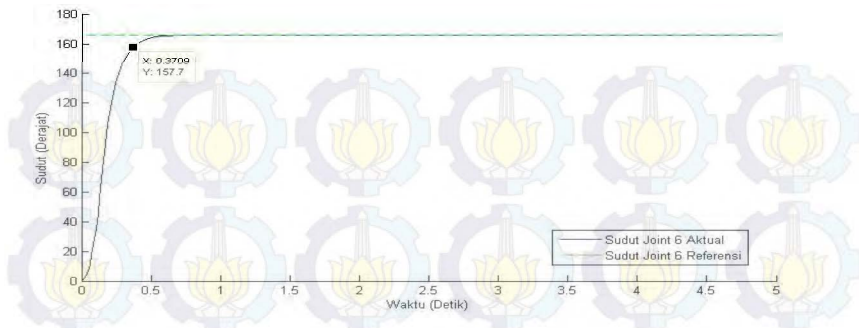
Gambar 4.9 Respons *Joint 3 Optimal* CTC Terhadap Sinyal Uji Step



Gambar 4.10 Respons *Joint 4 Optimal* CTC Terhadap Sinyal Uji Step



Gambar 4.11 Respons *Joint 5 Optimal* CTC Terhadap Sinyal Uji Step



Gambar 4.12 Respons *Joint 6 Optimal CTC* Terhadap Sinyal Uji *Step*

Berdasarkan pengujian menggunakan sinyal *step*, diperoleh *settling time* di sekitar $< 0,4$ detik dan tanpa *overshoot* untuk tiap *joint*. Pengujian dengan nilai *joint* statis menunjukkan bahwa *manipulator* mampu mengikuti referensi *joint* yang diberikan dengan toleransi perubahan nilai sudut referensi kurang maksimal 0,4 detik. Perbandingan respons dari CTC terhadap *Optimal CTC* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Perbandingan Respons CTC Terhadap *Optimal CTC*

CTC	Joint	RMSE ($^{\circ}$)	Optimal CTC	Joint	RMSE ($^{\circ}$)
	1	3,9392		1	2,6941
	2	3,9392		2	2,6941
	3	3,9392		3	2,6941
	4	5,9088		4	4,0411
	5	13,1308		5	8,9804
	6	21,7969		6	14,9072

Berdasarkan Tabel 4.2, diketahui *optimal CTC* mampu memperbaiki respons sudut dari tiap *joint* maksimal sebesar $6,8897^{\circ}$.

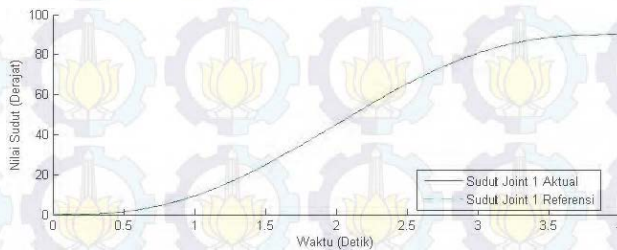
4.1.2. Pengujian dan Analisis Menggunakan Sinyal *Cubical Polynomial*

Sinyal *cubical polynomial* digunakan untuk mengetahui respons *joint* terhadap perubahan referensi dalam rentang waktu tertentu. Sinyal *cubical polynomial* dibuat dengan ketentuan yang telah dijelaskan pada Subbab 2.7. Nilai *joint* referensi $\theta_d = [90^{\circ}40^{\circ} - 25^{\circ} - 100^{\circ}35^{\circ}150^{\circ}]^T$ dan

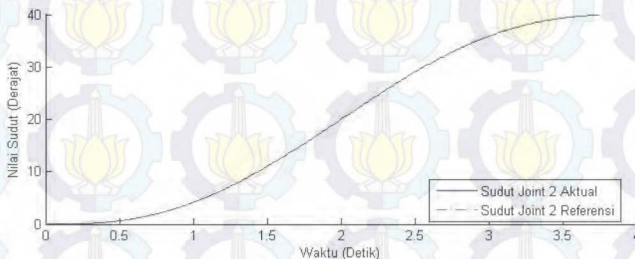
kondisi awal dari sistem dibuat sama dengan pengujian menggunakan sinyal *step* seperti yang ditunjukkan pada Persamaan 4.1 dan Persamaan 4.2. Nilai matriks pembobot *state* $Q = \text{diagonal}(1200000, 10000)$ dan pembobot sinyal kontrol $R = 2$. Respons dari tiap *joint* terhadap kondisi yang telah ditentukan sebelumnya ditunjukkan pada Gambar 4.13 hingga Gambar 4.18. Berdasarkan pengujian yang dilakukan menggunakan sinyal uji *cubical polynomial*, disimpulkan bahwa respons sistem dengan kontroler *optimal* CTC mampu mengikuti perubahan referensi dengan nilai RMSE dari tiap respons *joint* yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Nilai RMSE dari Respons Tiap *Joint* Terhadap Sinyal *Cubical Polynomial*

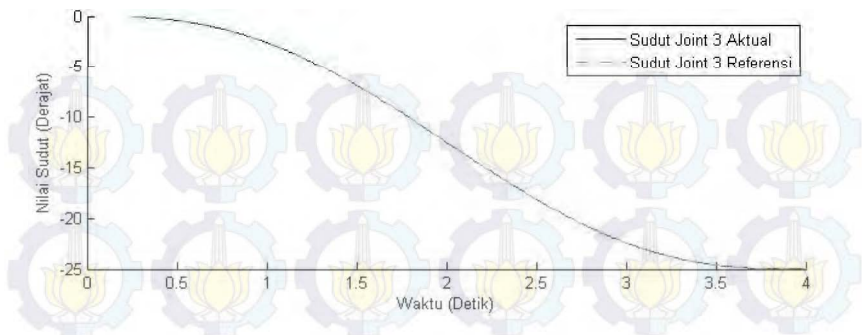
<i>Joint</i>	1	2	3	4	5	6
RMSE ($^{\circ}$)	0,0108	0,0048	0,003	0,012	0,0042	0,0192



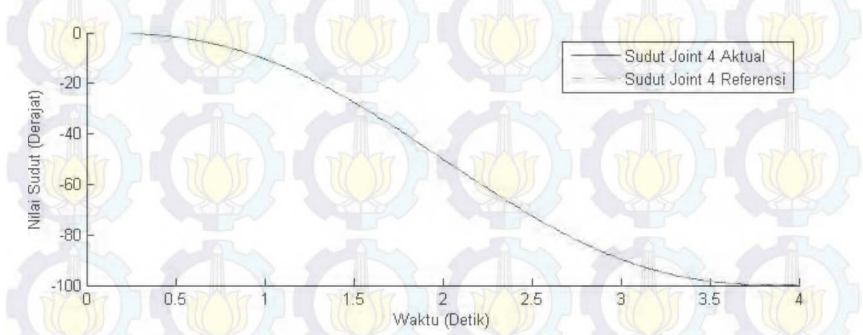
Gambar 4.13 Respons *Joint* 1 Terhadap Sinyal Uji *Cubical Polynomial*



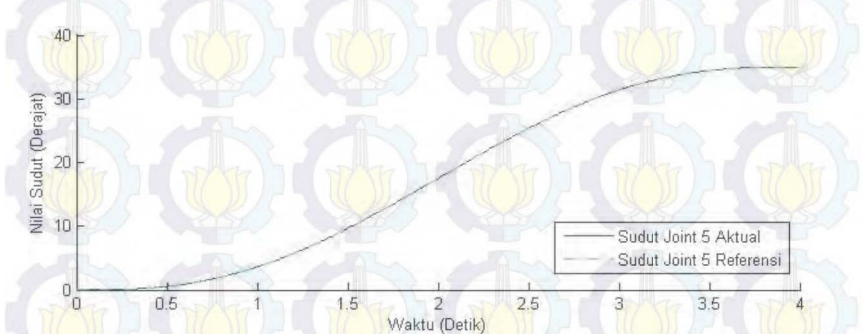
Gambar 4.14 Respons *Joint* 2 Terhadap Sinyal Uji *Cubical Polynomial*



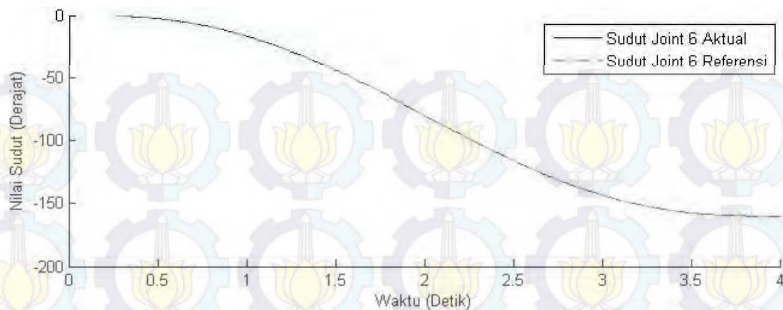
Gambar 4.15 Respons *Joint 3* Terhadap Sinyal Uji *Cubical Polynomial*



Gambar 4.16 Respons *Joint 4* Terhadap Sinyal Uji *Cubical Polynomial*



Gambar 4.17 Respons *Joint 5* Terhadap Sinyal Uji *Cubical Polynomial*



Gambar 4.18 Respons *Joint 6* Terhadap Sinyal Uji *Cubical Polynomial*

4.2. Pengujian dan Analisis *Inverse Kinematic* PUMA 560 Menggunakan *Jacobian Manipulator*

Pengujian *inverse kinematic* dilakukan dengan diberikan masukan posisi *end-effector* dalam ruang kartesian dan menghasilkan keluaran berupa nilai sudut *joint*. Hasil dari pengujian tersebut ditunjukkan pada Tabel 4.4. Untuk validitas hasil dari nilai sudut *joint*, dilakukan perhitungan *forward kinematic* berdasarkan keluaran nilai sudut *joint* sehingga diperoleh posisi dari *end-effector*. Hasil dari pengujian tersebut ditunjukkan pada Tabel 4.5.

Tabel 4.4 Hasil Perhitungan *Inverse Kinematic*

No.	Masukan (cm)			Keluaran ($^{\circ}$)					
	p_x	p_y	p_z	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	0	-50	1	-75,48	-54,78	24,38	-180	-30,4	255,5
2	12,36	-40	1	-55,4	-61,07	37,44	0	23,63	55,4
3	32,36	-20	1	-12,47	-63,89	43,39	0	20,51	12,47
4	40	0	1	18,27	-62,46	40,35	0	22,11	-18,27
5	32,36	20	1	50,96	-63,89	43,39	0	20,51	-50,96
6	12,36	40	1	90,26	-61,07	37,44	0	23,63	-90,26
7	0	50	1	104,5	-54,78	24,38	0	30,4	-104,5

Tabel 4.5 Pengujian Validitas Hasil *Inverse Kinematic*

No.	Masukan (derajat)						Keluaran (cm)		
	1	2	3	4	5	6	px	py	pz
1	-75,48	-54,78	24,38	-180	-30,4	255,5	0	-50	1
2	-55,4	-61,07	37,44	0	23,63	55,4	12,36	-40	1
3	-12,47	-63,89	43,39	0	20,51	12,47	32,36	-20	1
4	18,27	-62,46	40,35	0	22,11	-18,27	40	0	1
5	50,96	-63,89	43,39	0	20,51	-50,96	32,36	20	1
6	90,26	-61,07	37,44	0	23,63	-90,26	12,36	40	1
7	104,5	-54,78	24,38	0	30,4	-104,5	0	50	1

Dari Tabel 4.2 dan Tabel 4.3, diketahui hasil perhitungan *inverse kinematic* menghasilkan posisi *end-effector* yang sama dengan masukan posisi yang diinginkan, sehingga perhitungan *inverse kinematic* sesuai.

4.3. Pengujian dan Analisis Akurasi Posisi Aktual *End-Effector* Melalui Suatu Penugasan

Penugasan untuk *manipulator* berupa trayektori kartesian dengan lintasan berbentuk lingkaran dengan radius 10 hingga 50 cm. *Error* dari posisi *end-effector* dihitung untuk menentukan akurasi dari *manipulator*. Hasil dari penugasan ditunjukkan pada Gambar 4.19. Nilai RMSE dari posisi *end-effector* ditunjukkan pada Tabel 4.6.

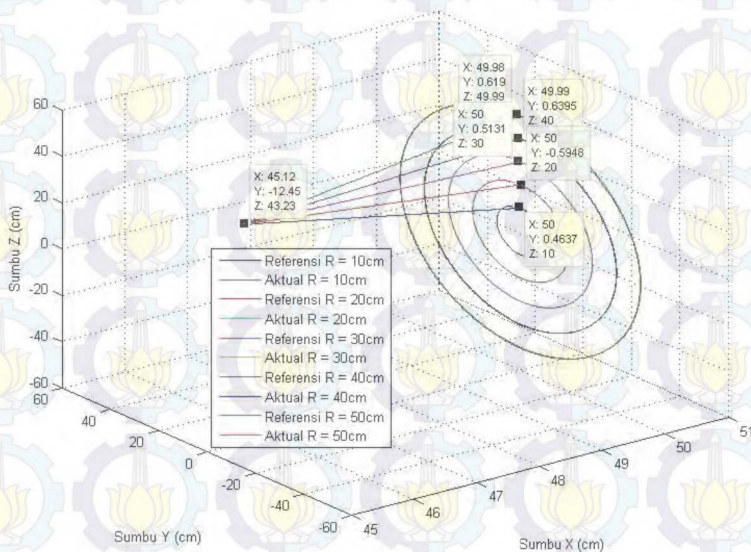
Tabel 4.6 Nilai RMSE Posisi *End-Effector* 1

Metode	Penugasan	RMSE Posisi End-Effector (cm)		
		X	Y	Z
Optimal CTC	Lingkaran Radius 10 cm	0,0016	0,0036	0,0075
	Lingkaran Radius 20 cm	0,0028	0,0061	0,0065
	Lingkaran Radius 30 cm	0,0053	0,0116	0,0071
	Lingkaran Radius 40 cm	0,0088	0,0117	0,0085
	Lingkaran Radius 50 cm	0,0142	0,0165	0,0112

Tabel 4.7 Nilai RMSE Posisi *End-Effector 2*

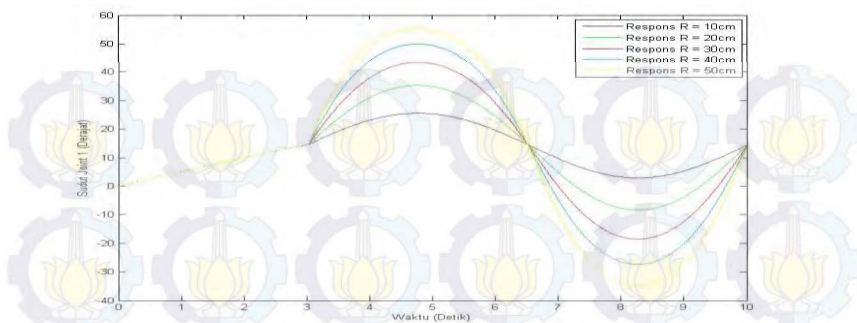
Rata-rata RMSE		0,0065	0,0099	0,0082
CTC	Lingkaran Radius 10 cm	0,5593	2,1514	2,4294
	Lingkaran Radius 20 cm	0,5778	3,7329	3,6488
	Lingkaran Radius 30 cm	1,2075	5,7475	5,7471
	Lingkaran Radius 40 cm	1,785	8,4642	9,0944
	Lingkaran Radius 50 cm	8,852	12,336	11,068
Rata-rata RMSE		2,5963	6,4864	6,3975

Berdasarkan Tabel 4.6, diketahui metode *Optimal* CTC menghasilkan akurasi posisi yang lebih tinggi.

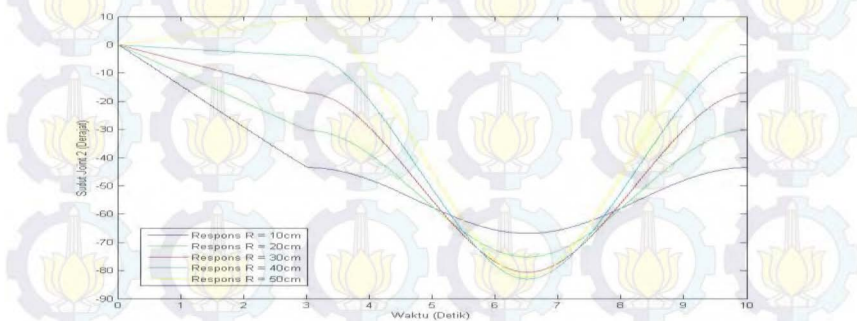


Gambar 4.19 Hasil Penugasan Trayektori Lingkaran Untuk *Manipulator*

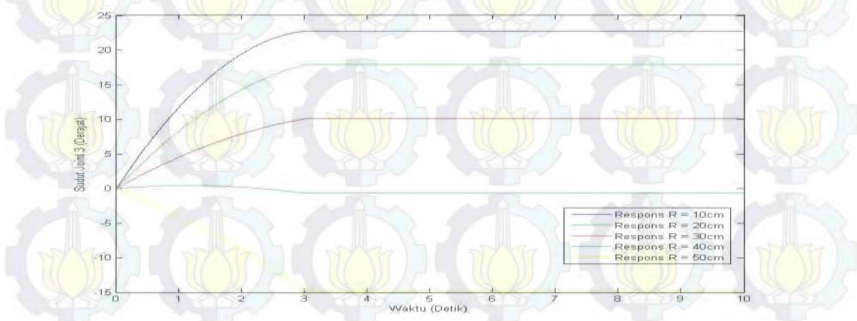
Untuk respons tiap *joint* dari penugasan trayektori lingkaran ditunjukkan pada Gambar 4.20 sampai Gambar 4.25.



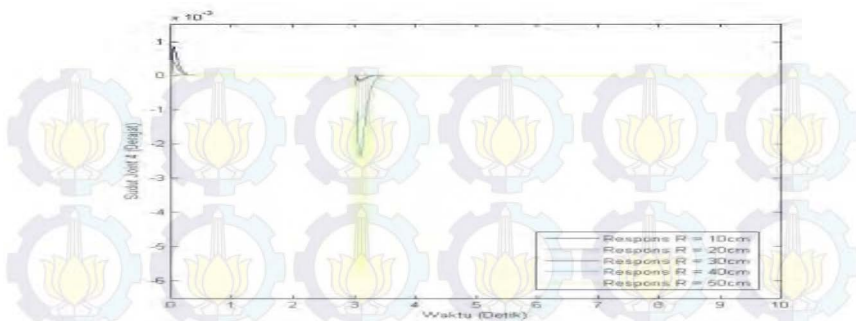
Gambar 4.20 Respons *Joint 1* Untuk Penugasan Trayektori Lingkaran



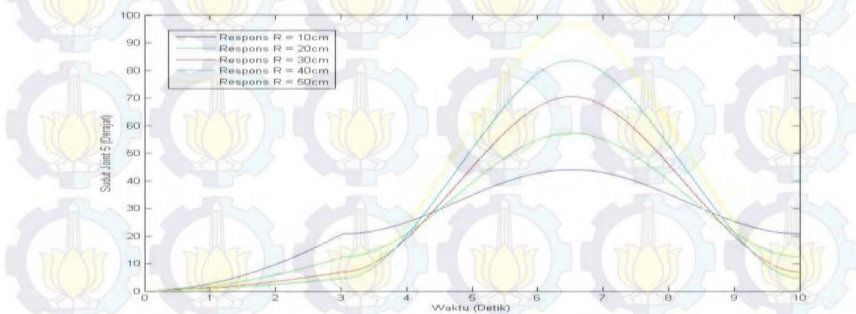
Gambar 4.21 Respons *Joint 2* Untuk Penugasan Trayektori Lingkaran



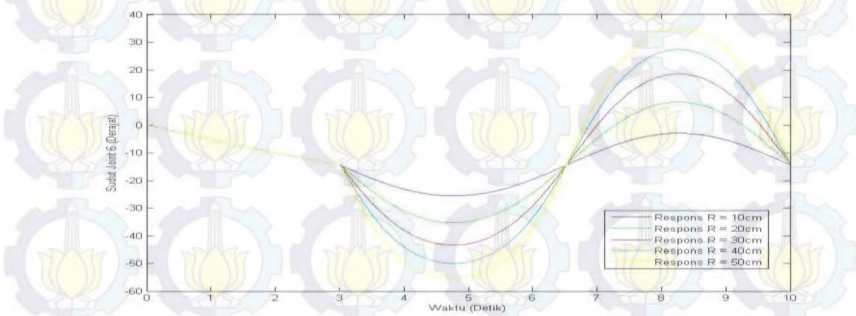
Gambar 4.22 Respons *Joint 3* Untuk Penugasan Trayektori Lingkaran



Gambar 4.23 Respons *Joint 4* Untuk Penugasan Trayektori Lingkaran

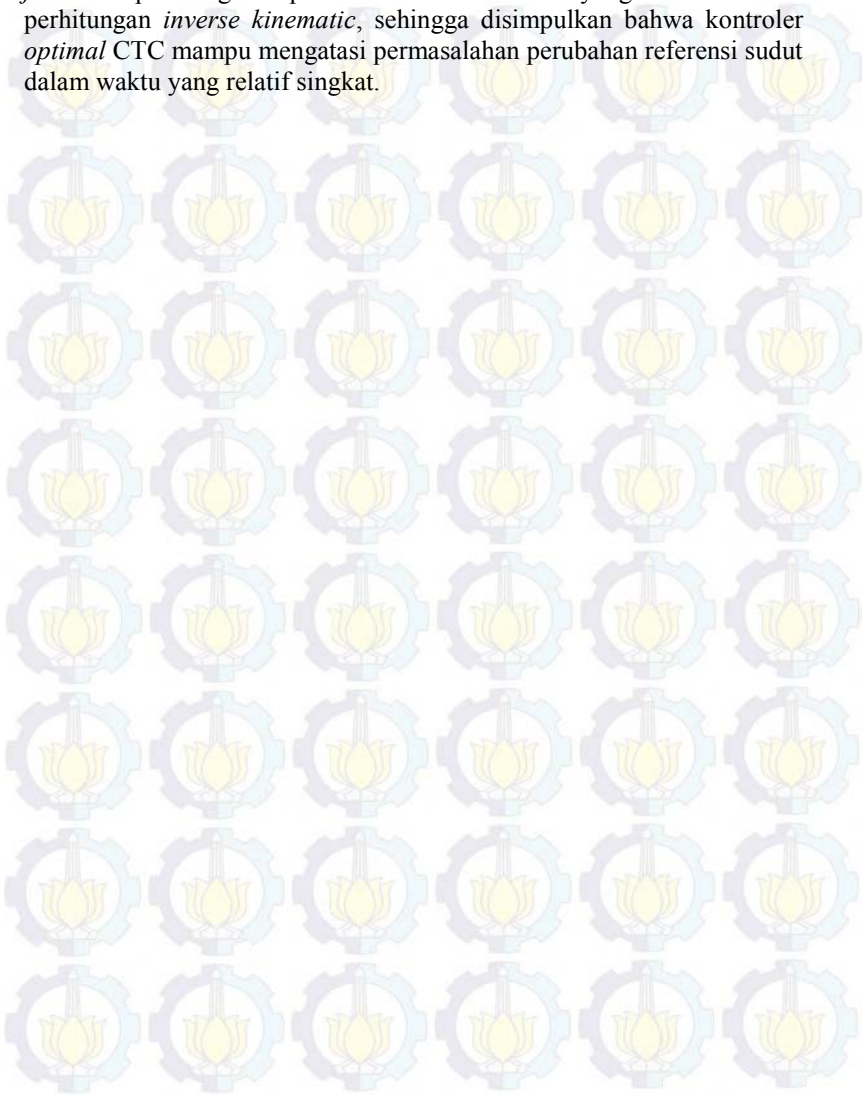


Gambar 4.24 Respons *Joint 5* Untuk Penugasan Trayektori Lingkaran



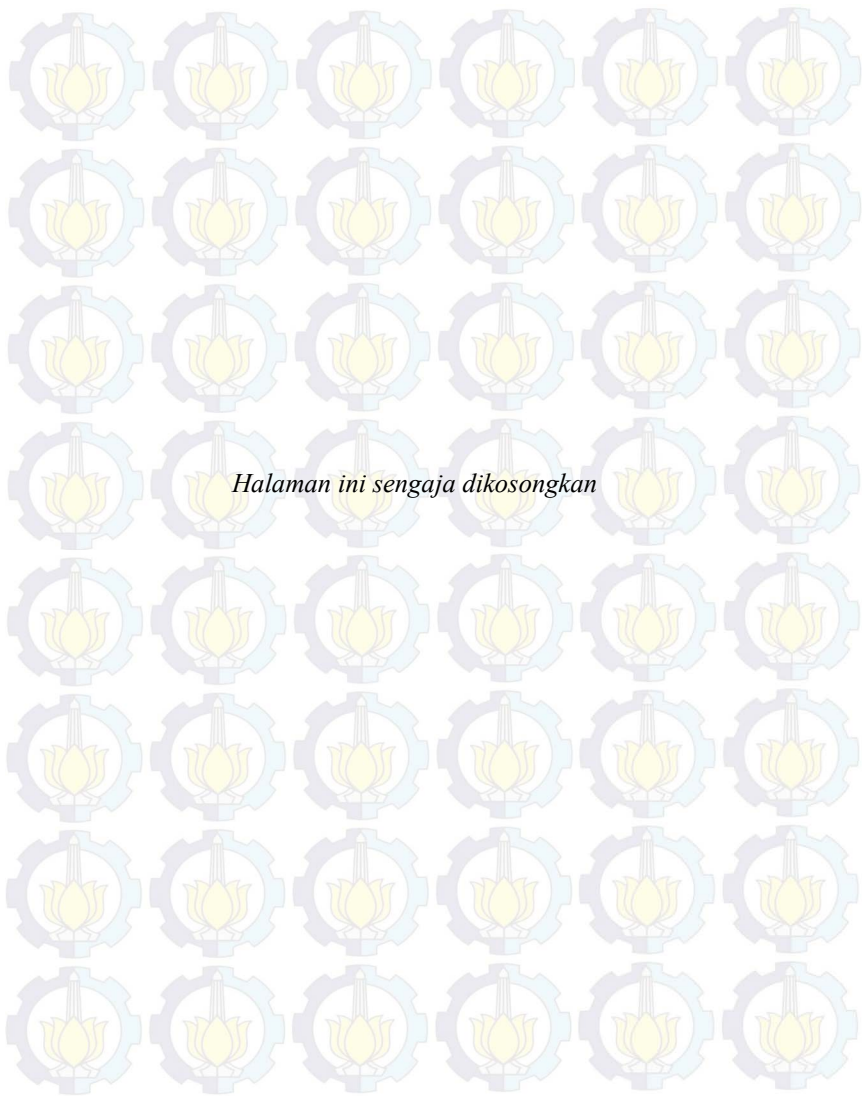
Gambar 4.25 Respons *Joint 6* Untuk Penugasan Trayektori Lingkaran

Berdasarkan Gambar 4.19 hingga Gambar 4.25, diketahui bahwa respons *joint* mampu mengikuti perubahan referensi sudut yang dihasilkan dari perhitungan *inverse kinematic*, sehingga disimpulkan bahwa kontroler *optimal* CTC mampu mengatasi permasalahan perubahan referensi sudut dalam waktu yang relatif singkat.



DAFTAR PUSTAKA

- [1] Craig, John J., "*Introduction to Robotics Mechanics and Control – Third Edition*", Prentice Hall, USA, 2005.
- [2] Spong, Mark W., Hutchinson, S. dan Vidyasagar, M., "*Robot Dynamics and Control – Third Edition*", John Wiley & Sons, 2004.
- [3] Siciliano, Bruce, "*Robotics Modelling Planning and Control*", Springer, UK, 2009.
- [4] Lewis, Frank L., Dawson, Darren M., Abdallah, Chaouki T., "*Robot Manipulator Control Theory and Practice - Second Edition*", Marcel Dekker, USA, 2004.
- [5] Benitez, Antonio., "PUMA 560: Robot Prototype with Graphic Simulation Environment", *Advanced In Mechanical Engineering* ISSN: 2160-0619 vol. 2 no. 1, Mexico, 2012.
- [6] Corke, Peter I., Armstrong-Helouvy, Brian., "A Search for Consensus Among Model Parameters Reported for the PUMA 560 Robot", *Proc. Int. Conf. Robotics and Automation.*, pp1608-1613, San Diego, 1994.
- [7] Armstrong-Helouvy, Brian., Khatib, Oussama., Burdick, Joe., "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", *Proc. 1986 IEEE Conf. Robot Autom.*, pp 510-518, San Fransisco, April 1986.
- [8] Corke, Peter I., "*Robotics, Vision, and Control Fundamental Algorithms in Matlab*", Springer, Australia, 2011.
- [9] Naidu, Desineni S., "*Optimal Control Systems*", CRC Press, Idaho, 2002.



Halaman ini sengaja dikosongkan

BAB V

PENUTUP

Pada bagian ini akan diberikan kesimpulan dan saran berdasarkan hasil penelitian yang telah dilakukan.

5.1. Kesimpulan

Berdasarkan analisis dan hasil percobaan dari penelitian yang dilakukan, dapat disimpulkan bahwa :

1. Penyelesaian *inverse kinematic* menghasilkan nilai sudut *joint* yang akurat terbukti melalui uji validitas dari hasil penyelesaian tersebut.
2. Kontroler *optimal* CTC mampu mengurangi nilai RMSE dari respons tiap sudut *joint* dibandingkan dengan CTC sebesar $6,8897^0$.
3. Kontroler *optimal* CTC mampu mengatur gerak PUMA 560 lebih akurat dari CTC dengan rata-rata nilai RMSE untuk pergerakan sumbu $x = 0,0065$ cm; sumbu $y = 0,0099$ cm; sumbu $z = 0,0082$ cm untuk pengugasan trayektori lingkaran dengan radius 10 hingga 50 cm.

5.2. Saran

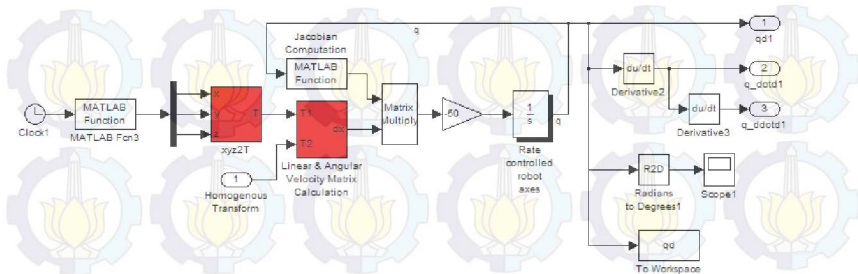
Berdasarkan hasil dari penelitian yang dilakukan, disarankan agar penelitian selanjutnya ditambahkan pengaturan gaya *end-effector* agar *manipulator* dapat melakukan tugas seperti *welding*, *assembling*, dan *milling*.



A.1.Diagram Blok Simulasi Sistem Keseluruhan Pada *Software* Matlab 2010a

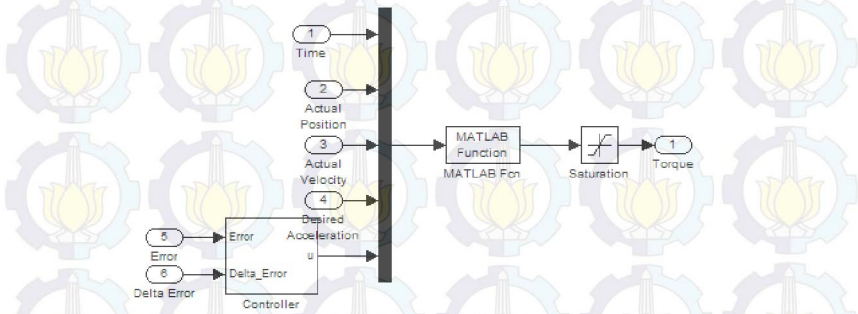


A.2. Diagram Blok Sub *Cartesian Trajectory Planning*



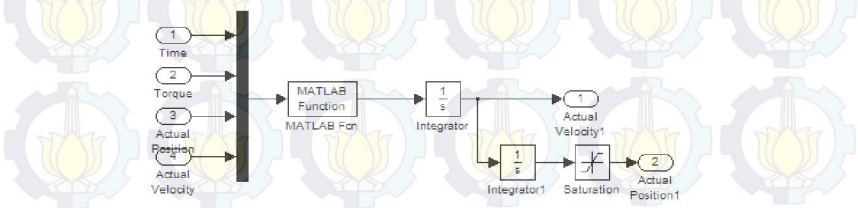
Gambar A.2 Diagram Blok Sub *Cartesian Trajectory Planning*

A.3. Diagram Blok Sub *Manipulator Controller*



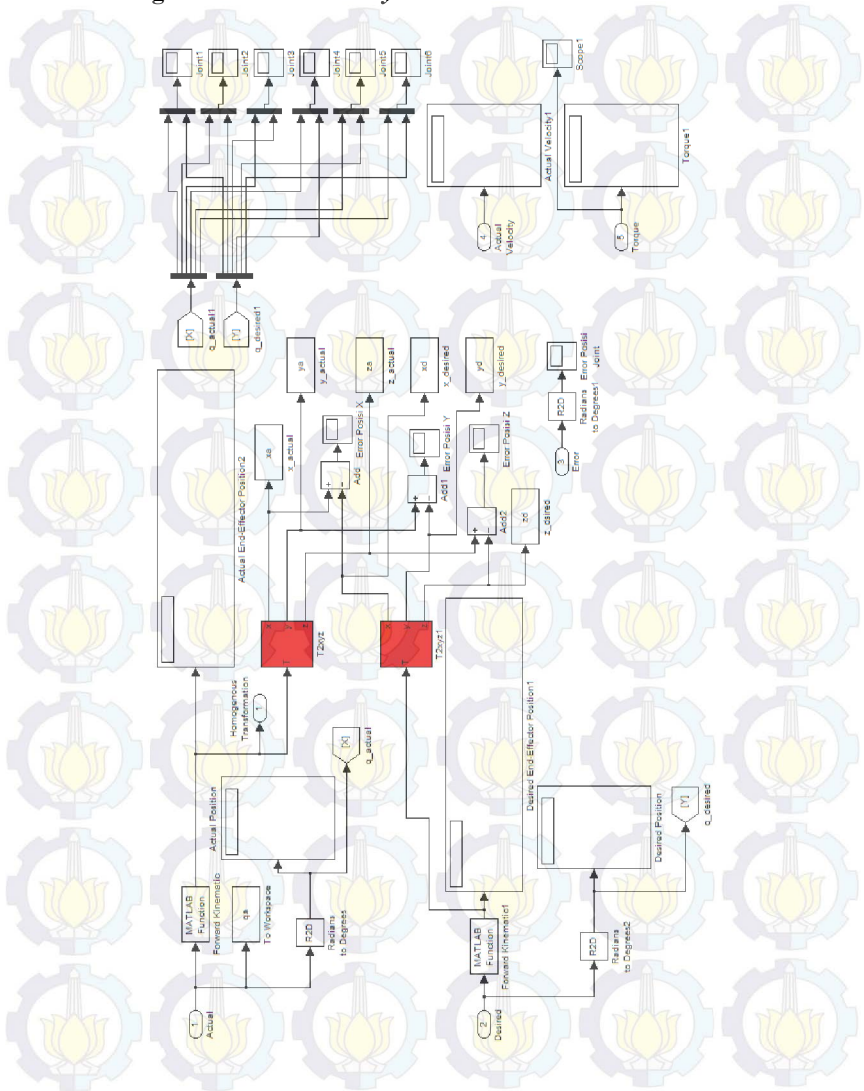
Gambar A.3 Diagram Blok Sub *Manipulator Controller*

A.4. Diagram Blok Sub PUMA 560



Gambar A.4 Diagram Blok Sub PUMA 560

A.5. Diagram Blok Sub Analysis



Gambar A.5 Diagram Blok Sub Analysis



LAMPIRAN B

B.1. Komponen Matriks *Jacobian* untuk *Manipulator 6-DOF*

$$J = \begin{bmatrix} J_{V0} & J_{V1} & J_{V2} & J_{V3} & J_{V4} & J_{V5} \\ J_{\omega0} & J_{\omega1} & J_{\omega2} & J_{\omega3} & J_{\omega4} & J_{\omega5} \end{bmatrix}$$

$$J_{V0} = z_0^0 x(p_6^0 - p_0^0); \quad J_{V1} = z_1^0 x(p_6^0 - p_1^0); \quad J_{V2} = z_2^0 x(p_6^0 - p_2^0)$$

$$J_{V3} = z_3^0 x(p_6^0 - p_3^0); \quad J_{V4} = z_4^0 x(p_6^0 - p_4^0); \quad J_{V5} = z_5^0 x(p_6^0 - p_5^0)$$

$$J_{\omega0} = z_0^0; \quad J_{\omega1} = z_1^0; \quad J_{\omega2} = z_2^0$$

$$J_{\omega3} = z_3^0; \quad J_{\omega4} = z_4^0; \quad J_{\omega5} = z_5^0$$

$z_0^0 = k = [0 \ 0 \ 1]^T$ dimana k adalah matriks *skew symmetric* untuk rotasi terhadap sumbu z dan $p_0^0 = [0 \ 0 \ 0]^T$.

B.2. Komponen Matriks *M* dan *N* PUMA 560

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 & 0 & 0 \\ m_{12} & m_{22} & m_{23} & 0 & 0 & 0 \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} & 0 \\ 0 & 0 & m_{34} & m_{44} & 0 & 0 \\ 0 & 0 & m_{35} & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} \quad N = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ 0 \end{bmatrix}$$

Keterangan Notasi :

$$S_i = \sin \theta_i; C_i = \cos \theta_i; S_{ij} = \sin(\theta_i + \theta_j); C_{ij} = \cos(\theta_i + \theta_j);$$

$$S_{ijk} = \sin(\theta_i + \theta_j + \theta_k); C_{ijk} = \cos(\theta_i + \theta_j + \theta_k);$$

$$SS_i = \sin^2 \theta_i ; CC_i = \cos^2 \theta_i ; CS_i = \cos \theta_i \sin \theta_i ;$$

$$SS_{ij} = \sin^2 (\theta_i + \theta_j)$$

$$m_{11} = 2,57 + 1,38CC_2 + 0,3SS_{23} + 0,744C_2S_{23}$$

$$m_{12} = 0,69S_2 - 0,134C_{23} + 0,0238C_2$$

$$m_{13} = -0,134C_{23} - 0,00397S_{23}$$

$$m_{22} = 6,79 + 0,744S_3$$

$$m_{23} = 0,333 + 0,372S_3 - 0,011C_3$$

$$m_{33} = 1,16$$

$$m_{34} = -0,00125S_4S_5$$

$$m_{35} = -0,00125C_4C_5$$

$$m_{44} = 0,2$$

$$m_{55} = 0,18$$

$$m_{66} = 0,19$$

$$\begin{aligned} n_1 = & [0,69C_2 + 0,134S_{23} - 0,0238S_2] \dot{\theta}_2^2 \\ & + [0,1335S_{23} - 0,00379C_{23}] \dot{\theta}_3^2 \\ & + [-2,76SC_2 + 0,744C_{223} + 0,6SC_{23} - 0,021(1 - 2SS_{23})] \dot{\theta}_1 \dot{\theta}_2 \\ & + [0,74C_2C_{23} + 0,6SC_{23} + 0,02C_2S_{23} + 0,021(1 - 2SS_{23})] \dot{\theta}_1 \dot{\theta}_3 \\ & + [-0,0025SC_{23}S_4S_5 + 0,0008C_4S_5 - 0,0024C_2C_{23}S_4S_5] \dot{\theta}_1 \dot{\theta}_4 \\ & + [0,0025(SS_{23}S_5 - SC_{23}C_4C_5) - 0,00248C_2(S_{23}S_5C_{23}C_4C_5) \\ & + 0,000864S_4C_5] \dot{\theta}_1 \dot{\theta}_5 + [0,267S_{23} - 0,00758C_{23}] \dot{\theta}_2 \dot{\theta}_3 \end{aligned}$$

$$\begin{aligned}
n_2 = & -\frac{1}{2}[-2,76SC_2 + 0,74C_{223} + 0,6SC_{23} - 0,02(1-2SS_{23})]\dot{\theta}_1^2 \\
& + \frac{1}{2}[0,022S_3 + 0,744C_3]\dot{\theta}_3^2 + [0,0016S_{23} - 0,0025C_{23}C_4S_5 \\
& + 0,00248S_2C_4S_5 + 0,00003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_4 \\
& + [-0,0021C_{23}S_4C_5 + 0,0024S_2S_4C_5 - 0,00064C_{23}S_4]\dot{\theta}_1\dot{\theta}_5 \\
& + [0,022S_3 + 0,744C_3]\dot{\theta}_2\dot{\theta}_3 - [0,00248C_3S_4S_5]\dot{\theta}_2\dot{\theta}_4 \\
& + [-0,0025S_5 + 0,00248(C_3C_4C_5 - S_3S_5)]\dot{\theta}_2\dot{\theta}_5 \\
& - [0,00248C_3S_4S_5]\dot{\theta}_3\dot{\theta}_4 \\
& + [-0,0025S_5 + 0,00248(C_3C_4C_5 - S_3S_5)]\dot{\theta}_3\dot{\theta}_5 \\
& - 37,2C_2 - 8,4S_{23} + 1,02S_2
\end{aligned}$$

$$\begin{aligned}
n_3 = & -\frac{1}{2}[-2,76SC_2 + 0,74C_{223} + 0,6SC_{23} - 0,02(1-2SS_{23})]\dot{\theta}_1^2 \\
& - \frac{1}{2}[0,02S_3 + 0,74C_3]\dot{\theta}_2^2 - [0,001C_4S_5]\dot{\theta}_5^2 - [0,001C_4S_5]\dot{\theta}_4^2 \\
& + [-0,002C_{23}C_4S_5 + 0,0016S_{23} + 0,0003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_4 \\
& - [0,0025C_{23}S_4C_5 + 0,00064C_{23}S_4]\dot{\theta}_1\dot{\theta}_5 - [0,0025S_5]\dot{\theta}_2\dot{\theta}_5 \\
& - [0,0025S_5]\dot{\theta}_3\dot{\theta}_5 - [0,0025S_4C_5]\dot{\theta}_4\dot{\theta}_5 - 8,4S_{23} + 0,25C_{23}
\end{aligned}$$

$$\begin{aligned}
n_4 = & \frac{1}{2}[0,0025SC_{23}S_4S_5 - 0,00086C_4S_5 + 0,00248C_2C_{23}S_4S_5]\dot{\theta}_1^2 \\
& - \frac{1}{2}[0,00248C_3S_4S_5]\dot{\theta}_2^2 + [0,00164S_{23} - 0,0025C_{23}C_4S_5 \\
& + 0,00248S_2C_4C_5 + 0,0003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_2 \\
& + [0,0025C_{23}C_4S_5 - 0,00164S_{23} - 0,003S_{23}(1-2SS_4)]\dot{\theta}_1\dot{\theta}_3 \\
& - [0,00064S_{23}C_4]\dot{\theta}_1\dot{\theta}_5 + [0,00064S_4]\dot{\theta}_2\dot{\theta}_5 + [0,00064S_4]\dot{\theta}_3\dot{\theta}_5 \\
& + 0,028S_{23}S_4S_5
\end{aligned}$$

$$\begin{aligned}
n_5 = & \frac{1}{2}[0,0025(SS_{23}S_5 - SC_{23}C_4C_5) + 0,002C_2(S_{23}S_5 - C_{23}C_4C_5) \\
& - 0,0008S_4C_5]\dot{\theta}_1^2 - \frac{1}{2}[0,0025S_5 - 0,002(C_3C_4C_5 - S_3S_5)]\dot{\theta}_2^2 \\
& + [0,0026C_{23}S_4C_5 - 0,00248S_2S_4C_5 + 0,000642C_{23}S_4]\dot{\theta}_1\dot{\theta}_2 \\
& + [0,0025C_{23}S_4C_5 + 0,0006C_{23}S_4]\dot{\theta}_1\dot{\theta}_3 + [0,0006S_{23}C_4]\dot{\theta}_1\dot{\theta}_4 \\
& - [0,0006S_4]\dot{\theta}_2\dot{\theta}_4 - [0,0006S_4]\dot{\theta}_3\dot{\theta}_4 - 0,028(C_{23}S_5 + S_{23}C_4S_5)
\end{aligned}$$

LAMPIRAN C

C.1. Listing Program Forward Kinematic Pada Matlab

```
function T = forward(q)
al = [pi/2 0 -pi/2 pi/2 -pi/2 0];
d = [0 0 0.1254 0.4318 0 0];
a = [0 0.4318 0.0191 0 0 0];

for i=1:6
A = [cos(q(i)) -sin(q(i))*cos(al(i)) sin(q(i))*sin(al(i)) a(i)*cos(q(i));
sin(q(i)) cos(q(i))*cos(al(i)) -cos(q(i))*sin(al(i)) a(i)*sin(q(i));
0 sin(al(i)) cos(al(i)) d(i) ;
0 0 0 1 ];
if i==1
T1 = A;
elseif i==2
T2 = A;
elseif i==3
T3 = A;
elseif i==4
T4 = A;
elseif i==5
T5 = A;
else
T6 = A;
end
end

T = T1*T2*T3*T4*T5*T6;
%%posisi_ee = [T(1,4);T(2,4);T(3,4)];
```

C.2. Listing Program Jacobian Manipulator

```
function J_inv = jac(q)
al = [pi/2 0 -pi/2 pi/2 -pi/2 0];
d = [0 0 0.1254 0.4318 0 0];
a = [0 0.4318 0.0191 0 0 0];
```

$k = [0;0;1]$; $z0 = k$; $o0 = [0;0;0]$;

for i=1:6

$A = \begin{bmatrix} \cos(q(i)) & -\sin(q(i))*\cos(al(i)) & \sin(q(i))*\sin(al(i)) & a(i)*\cos(q(i)); \\ \sin(q(i)) & \cos(q(i))*\cos(al(i)) & -\cos(q(i))*\sin(al(i)) & a(i)*\sin(q(i)); \\ 0 & \sin(al(i)) & \cos(al(i)) & d(i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$;

if i==1

T1 = A;

R1 = T1(1:3,1:3);

o1 = T1(1:3,4);

z1 = R1*k;

elseif i==2

T2 = T1*A;

R2 = T2(1:3,1:3);

o2 = T2(1:3,4);

z2 = R2*k;

elseif i==3

T3 = T2*A;

R3 = T3(1:3,1:3);

o3 = T3(1:3,4);

z3 = R3*k;

elseif i==4

T4 = T3*A;

R4 = T4(1:3,1:3);

o4 = T4(1:3,4);

z4 = R4*k;

elseif i==5

T5 = T4*A;

R5 = T5(1:3,1:3);

o5 = T5(1:3,4);

z5 = R5*k;

else

T6 = T5*A;

%%R6 = T6(1:3,1:3);

o6 = T6(1:3,4);

end

end

```

j1 = [cross(z0,(o6-o0));z0]; j2 = [cross(z1,(o6-o1));z1]; j3 =
[cross(z2,(o6-o2));z2];
j4 = [cross(z3,(o6-o3));z3]; j5 = [cross(z4,(o6-o4));z4]; j6 =
[cross(z5,(o6-o5));z5];

J = [j1 j2 j3 j4 j5 j6];
J_inv = inv(J);

```

C.3.Listing Program Computed Torque Control

```

function T = inv_dynamic(in)
t = in(1); qa = in(2:7); q_dota = in(8:13); q_ddotd = in(14:19); u =
in(20:25);
%%Initiation%%
%%if t == 0
%%  qa = [0;0;0;0;0;0];
%%  q_dota = [0;0;0;0;0;0];
%%else
%%  qa = qa;
%%  q_dota = q_dota;
%%end
%%Angular Position%%
c2 = cos(qa(2)); s2 = sin(qa(2)); c3 = cos(qa(3)); s3 = sin(qa(3));
c4 = cos(qa(4)); s4 = sin(qa(4)); c5 = cos(qa(5)); s5 = sin(qa(5));
c23 = cos(qa(2)+qa(3)); s23 = sin(qa(2)+qa(3));
c223 = cos(qa(2)*2+qa(3));

%%Angular Velocity%%
q1 = q_dota(1); q2 = q_dota(2); q3 = q_dota(3); q4 = q_dota(4);
q5 = q_dota(5);

%%Puma Inertia Matrix%%
m11 = 2.57+1.38*c2^2+0.3*s23^2+0.744*c2*s23;
m12 = 0.69*s2-0.134*c23+0.0238*c2;
m13 = -0.134*c23-0.00397*s23;
m14 = 0; m15 = 0; m16 = 0;
m22 = 6.79+0.744*s3;
m23 = 0.333+0.372*s3-0.011*c3;

```


$$m24 = 0; m25 = 0; m26 = 0;$$

$$m33 = 1.16; m34 = -0.00125*s4*s5; m35 = -0.00125*c4*c5; m36 = 0;$$

$$m44 = 0.2; m45 = 0; m46 = 0;$$

$$m55 = 0.18; m56 = 0;$$

$$m66 = 0.19;$$

$$m21 = m12; m31 = m13; m32 = m23; m41 = 0; m42 = 0;$$

$$m51 = 0; m52 = 0; m54 = 0; m61 = 0; m62 = 0; m63 = 0;$$

$$m64 = 0; m65 = 0;$$

$$M = [m11 \ m12 \ m13 \ m14 \ m15 \ m16;$$

$$m21 \ m22 \ m23 \ m24 \ m25 \ m26;$$

$$m31 \ m32 \ m33 \ m34 \ m35 \ m36;$$

$$m41 \ m42 \ m34 \ m44 \ m45 \ m46;$$

$$m51 \ m52 \ m35 \ m54 \ m55 \ m56;$$

$$m61 \ m62 \ m63 \ m64 \ m65 \ m66];$$

%%Puma Coriolis, Centrifugal, and Gravitational Matrix%%

$$n11 = (0.69*c2+0.134*s23-0.0238*s2)*q2^2;$$

$$n12 = (0.1335*s23-0.00379*c23)*q3^2;$$

$$n13 = (-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1-2*s23^2))*q1*q2;$$

$$n14 = (0.744*c2*c23+0.6*s23*c23+0.022*c2*s23-0.0213*(1-2*s23^2))*q1*q3;$$

$$n15 = (-0.0025*s23*c23*s4*s5+0.00086*c4*c5-0.00248*c2*(s23*s5-c23*c4*c5))*q1*q4;$$

$$n16 = (((-0.0025*(s23^2*s5-s23*c23*c4*c5)-0.00248*c2*(s23*s5-c23*c4*c5))+0.000864*s4*c5)*q1*q5;$$

$$n17 = (0.267*s23-0.00758*c23)*q2*q3;$$

$$n1 = (n11+n12+n13+n14+n15+n16+n17);$$

$$n21 = -0.5*(-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1-2*s23^2))*q1^2;$$

$$n22 = 0.5*(0.022*s3+0.744*c3)*q3^2;$$

$$\begin{aligned}
n23 &= (0.00164*s23- \\
&0.0025*c23*c4*s5+0.00248*s2*c4*s5+0.00003*s23*(1- \\
&2*s4^2))*q1*q4; \\
n24 &= (-0.00215*c23*s4*c5+0.00248*s2*s4*c5- \\
&0.000642*c23*s4)*q1*q5; \\
n25 &= (0.022*s3+0.744*c3)*q2*q3-(0.00248*c3*s4*s5)*q2*q4; \\
n26 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q2*q5- \\
&(0.00248*c3*s4*s5)*q3*q4; \\
n27 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q3*q5-37.2*c2- \\
&8.4*s23+1.02*s2; \\
n2 &= (n21+n22+n23+n24+n25+n26+n27);
\end{aligned}$$

$$\begin{aligned}
n31 &= -0.5*(-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1- \\
&2*s23^2))*q1^2; \\
n32 &= -0.5*(0.022*s3+0.744*c3)*q2^2-(0.00123*c4*s5)*q5^2- \\
&(0.00125*c4*s5)*q4^2; \\
n33 &= (-0.0025*c23*s4*c5+0.00164*s23+0.0003*s23*(1- \\
&2*s4^2))*q1*q4; \\
n34 &= -(0.0025*c23*s4*c5+0.000642*c23*s4)*q1*q5- \\
&(0.0025*s5)*q2*q5; \\
n35 &= -(0.0025*s5)*q3*q5-(0.0025*s4*c5)*q4*q5-8.4*s23+0.25*c23; \\
n3 &= n31+n32+n33+n34+n35; \\
n41 &= 0.5*(0.0025*s23*c23*s4*s5- \\
&0.00086*c4*s5+0.00248*c2*c23*s4*s5)*q1^2- \\
&0.5*(0.00248*c3*s4*s5)*q2^2; \\
n42 &= (0.00164*s23- \\
&0.0025*c23*c4*s5+0.00248*s2*c4*s5+0.0003*s23*(1- \\
&2*s4^2))*q1*q2; \\
n43 &= (0.0025*c23*c4*s5-0.00164*s23-0.003*s23*(1-2*s4^2))*q1*q3; \\
n44 &= -(0.000642*s23*c4)*q1*q5+(0.000642*s4)*q2*q5; \\
n45 &= (0.000642*s4)*q3*q5+0.028*s23*s4*s5; \\
n4 &= n41+n42+n43+n44+n45; \\
n51 &= 0.5*(0.0025*(s23^2*s5-s23*c23*c4*c5)+0.00248*c2*(s23*s5- \\
&c23*c4*c5)-0.000864*s4*c5)*q1^2; \\
n52 &= -0.5*(0.0025*s5-0.00248*(c3*c4*c5-s3*s5))*q2^2; \\
n53 &= (0.0026*c23*s4*c5-0.00258*s2*s4*c5+0.000642*c23*s4)*q2^2; \\
n54 &= -(0.000642*s4)*q2*q4-(0.000642*s4)*q3*q4- \\
&0.028*(c23*s5+s23*c4*c5); \\
n5 &= n51+n52+n53+n54;
\end{aligned}$$

n6 = 0;

N = [n1;n2;n3;n4;n5;n6];

T = M*(q_ddotd+u)+N;

C.4.Listing Program Dinamika PUMA 560

```
function q_ddota = dynamic(in)
t = in(1); T = in(2:7); qa = in(8:13); q_dota = in(14:19);
%%Initiation%%
%%if t == 0
%%  qa = [0;0;0;0;0;0];
%%  q_dota = [0;0;0;0;0;0];
%%else
%%  qa = qaa;
%%  q_dota = q_dotaa;
%%end
%%Angular Position%%
c2 = cos(qa(2)); s2 = sin(qa(2)); c3 = cos(qa(3)); s3 = sin(qa(3));
c4 = cos(qa(4)); s4 = sin(qa(4)); c5 = cos(qa(5)); s5 = sin(qa(5));
c23 = cos(qa(2)+qa(3)); s23 = sin(qa(2)+qa(3));
c223 = cos(qa(2)*2+qa(3));

%%Angular Velocity%%
q1 = q_dota(1); q2 = q_dota(2); q3 = q_dota(3); q4 = q_dota(4);
q5 = q_dota(5);

%%Puma Inertia Matrix%%
m11 = 2.57+1.38*c2^2+0.3*s23^2+0.744*c2*s23;
m12 = 0.69*s2-0.134*c23+0.0238*c2;
m13 = -0.134*c23-0.00397*s23;

m22 = 6.79+0.744*s3;
m23 = 0.333+0.372*s3-0.011*c3;

m33 = 1.16; m34 = -0.00125*s4*s5; m35 = -0.00125*c4*c5;
```

$$m44 = 0.2;$$

$$m55 = 0.18;$$

$$m66 = 0.19;$$

$$M = \begin{bmatrix} m11 & m12 & m13 & 0 & 0 & 0; \\ m12 & m22 & m23 & 0 & 0 & 0; \\ m13 & m23 & m33 & m34 & m35 & 0; \\ 0 & 0 & m34 & m44 & 0 & 0; \\ 0 & 0 & m35 & 0 & m55 & 0; \\ 0 & 0 & 0 & 0 & 0 & m66 \end{bmatrix};$$

%%Puma Coriolis, Centrifugal, and Gravitational Matrix%%

$$n11 = (0.69*c2 + 0.134*s23 - 0.0238*s2)*q2^2;$$

$$n12 = (0.1335*s23 - 0.00379*c23)*q3^2;$$

$$n13 = (-2.76*s2*c2 + 0.744*c223 + 0.6*s23*c23 - 0.0213*(1 - 2*s23^2))*q1*q2;$$

$$n14 = (0.744*c2*c23 + 0.6*s23*c23 + 0.022*c2*s23 - 0.0213*(1 - 2*s23^2))*q1*q3;$$

$$n15 = (-0.0025*s23*c23*s4*s5 + 0.00086*c4*c5 - 0.00248*c2*(s23*s5 - c23*c4*c5))*q1*q4;$$

$$n16 = ((-0.0025*(s23^2*s5 - s23*c23*c4*c5) - 0.00248*c2*(s23*s5 - c23*c4*c5)) + 0.000864*s4*c5)*q1*q5;$$

$$n17 = (0.267*s23 - 0.00758*c23)*q2*q3;$$

$$n1 = (n11 + n12 + n13 + n14 + n15 + n16 + n17);$$

$$n21 = -0.5*(-2.76*s2*c2 + 0.744*c223 + 0.6*s23*c23 - 0.0213*(1 - 2*s23^2))*q1^2;$$

$$n22 = 0.5*(0.022*s3 + 0.744*c3)*q3^2;$$

$$n23 = (0.00164*s23 - 0.0025*c23*c4*s5 + 0.00248*s2*c4*s5 + 0.00003*s23*(1 - 2*s4^2))*q1*q4;$$

$$n24 = (-0.00215*c23*s4*c5 + 0.00248*s2*s4*c5 - 0.000642*c23*s4)*q1*q5;$$

$$n25 = (0.022*s3 + 0.744*c3)*q2*q3 - (0.00248*c3*s4*s5)*q2*q4;$$

$$\begin{aligned} n26 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q2*q5- \\ &(0.00248*c3*s4*s5)*q3*q4; \\ n27 &= (-0.0025*s5+0.00248*(c3*c4*c5-s3*s5))*q3*q5-37.2*c2- \\ &8.4*s23+1.02*s2; \\ n2 &= (n21+n22+n23+n24+n25+n26+n27); \end{aligned}$$

$$\begin{aligned} n31 &= -0.5*(-2.76*s2*c2+0.744*c223+0.6*s23*c23-0.0213*(1- \\ &2*s23^2))*q1^2; \\ n32 &= -0.5*(0.022*s3+0.744*c3)*q2^2-(0.00123*c4*s5)*q5^2- \\ &(0.00125*c4*s5)*q4^2; \\ n33 &= (-0.0025*c23*s4*c5+0.00164*s23+0.0003*s23*(1- \\ &2*s4^2))*q1*q4; \\ n34 &= -(0.0025*c23*s4*c5+0.000642*c23*s4)*q1*q5- \\ &(0.0025*s5)*q2*q5; \\ n35 &= -(0.0025*s5)*q3*q5-(0.0025*s4*c5)*q4*q5-8.4*s23+0.25*c23; \\ n3 &= n31+n32+n33+n34+n35; \end{aligned}$$

$$\begin{aligned} n41 &= 0.5*(0.0025*s23*c23*s4*s5- \\ &0.00086*c4*s5+0.00248*c2*c23*s4*s5)*q1^2- \\ &0.5*(0.00248*c3*s4*s5)*q2^2; \\ n42 &= (0.00164*s23- \\ &0.0025*c23*c4*s5+0.00248*s2*c4*s5+0.0003*s23*(1- \\ &2*s4^2))*q1*q2; \\ n43 &= (0.0025*c23*c4*s5-0.00164*s23-0.003*s23*(1-2*s4^2))*q1*q3; \\ n44 &= -(0.000642*s23*c4)*q1*q5+(0.000642*s4)*q2*q5; \\ n45 &= (0.000642*s4)*q3*q5+0.028*s23*s4*s5; \\ n4 &= n41+n42+n43+n44+n45; \end{aligned}$$

$$\begin{aligned} n51 &= 0.5*(0.0025*(s23^2*s5-s23*c23*c4*c5)+0.00248*c2*(s23*s5- \\ &c23*c4*c5)-0.000864*s4*c5)*q1^2; \\ n52 &= -0.5*(0.0025*s5-0.00248*(c3*c4*c5-s3*s5))*q2^2; \\ n53 &= (0.0026*c23*s4*c5-0.00258*s2*s4*c5+0.000642*c23*s4)*q2^2; \\ n54 &= -(0.000642*s4)*q2*q4-(0.000642*s4)*q3*q4- \\ &0.028*(c23*s5+s23*c4*c5); \\ n5 &= n51+n52+n53+n54; \end{aligned}$$

$$N = [n1;n2;n3;n4;n5;0];$$

$q_ddota = M \backslash (T - N);$

D.3. Listing Program Trayektori Lingkaran

```
function out=lingkaran1(t)
tmax=10;
k=.5;
if t<=3
    p0=[.4509 -.1254 .4318];
    p1=[.5 0 k];
    x=((p1(1)-p0(1))*t/3)+p0(1);
    y=((p1(2)-p0(2))*t/3)+p0(2);
    z=((p1(3)-p0(3))*t/3)+p0(3);
else
    q=(2*pi)*(t-3)/(tmax-3);
    gain=k;
    x=.50;
    y=sin(q)*gain;
    z=cos(q)*gain;
end
out=[x,y,z];
```

