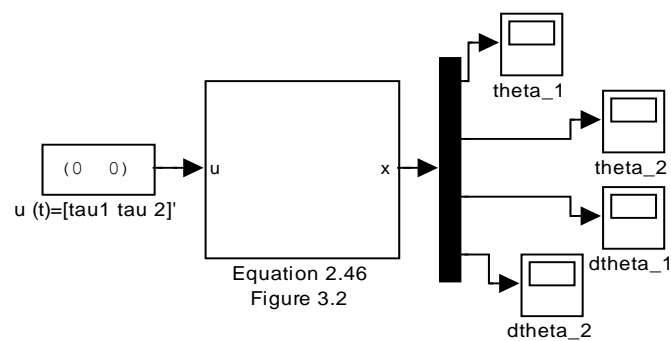


BAB III

PEMODELAN DINAMIKA *SWING LEG* DAN DESAIN KONTROL

3.1 Pemodelan Sistem

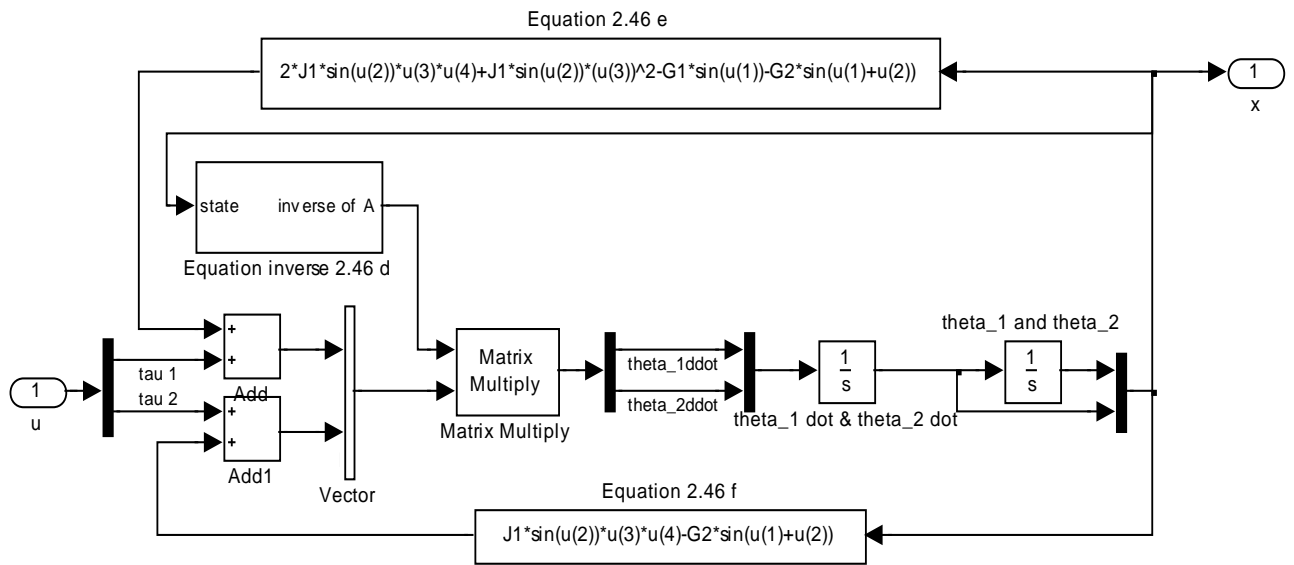
Dalam subbab pemodelan sistem ini seluruh model dari *swing leg* dan *adaptive control* dimodelkan menggunakan *software* MATLAB/Simulink. Seluruh pemodelan dibangun dari interkoneksi blok yang mewakili persamaan dinamik dari sistem *swing leg*. Gambar 3.1 merupakan model dinamika *swing leg* secara keseluruhan. Dinamika *swing leg* merepresentasikan *swing leg* yang menghasilkan *state* berupa posisi dan kecepatan.



Gambar 3.1 Model dinamika *swing leg*.

Dari gambar 3.1 *state* yang dihasilkan yaitu berupa posisi dan kecepatan. Dimana blok *Scope* *theta_1* adalah posisi pada θ_1 , blok *Scope* *theta_2* adalah posisi pada θ_2 , blok *Scope* *dtheta_1* adalah kecepatan pada posisi θ_1 dan blok *Scope* *dtheta_2* adalah kecepatan pada posisi θ_2 .

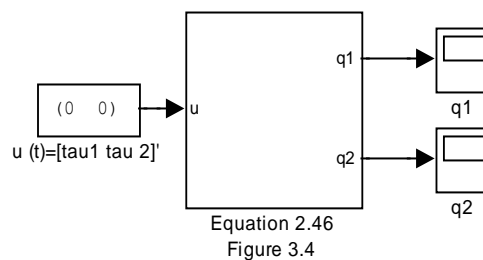
Plant swing leg pada gambar 3.1 merupakan *subsystem* yang merepresentasikan persamaan dinamik *swing leg* pada persamaan (2.46). Gambar 3.2 berikut adalah gambar blok-blok dalam *subsystem swing leg* yang merepresentasikan persamaan (2.46) pada gambar 3.1.



Gambar 3.2 Subsystem swing leg.

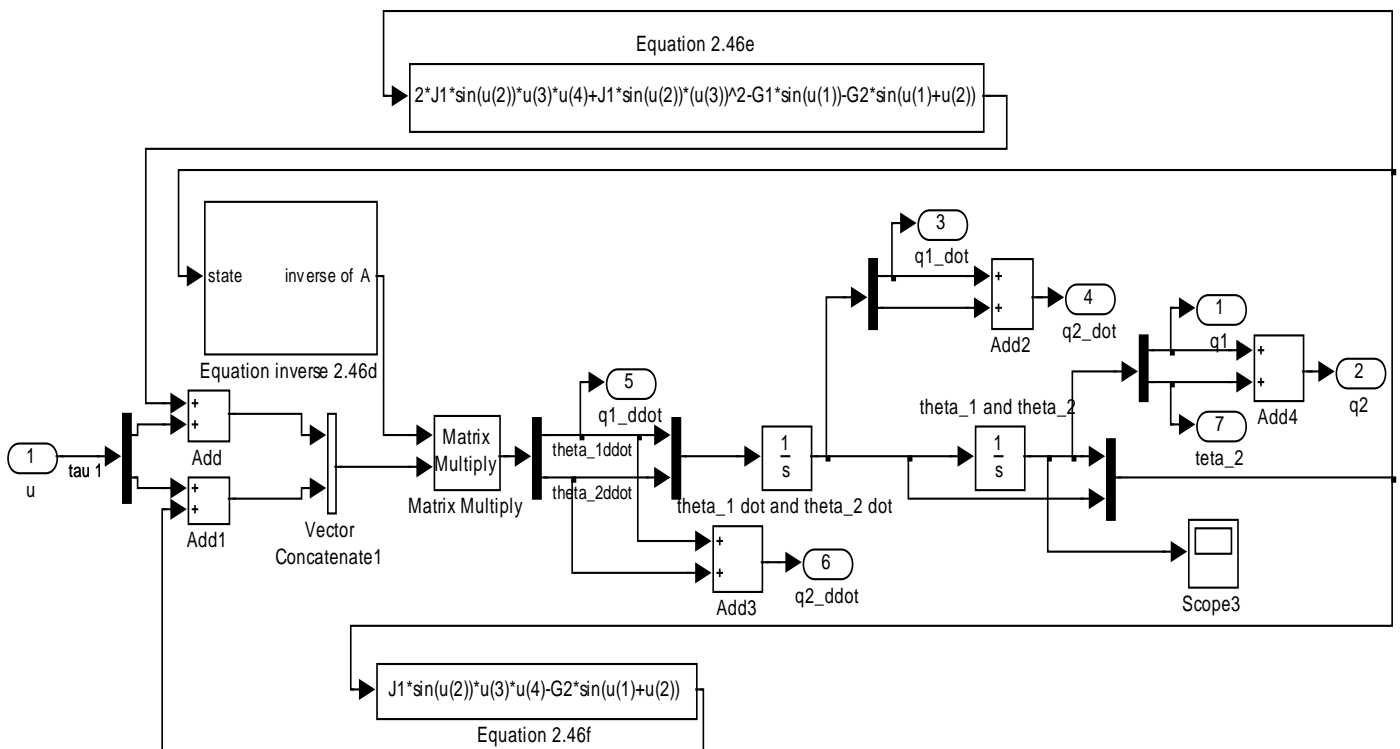
Dari gambar 3.2 blok *Function equation inverse 2.46 d*, *equation 2.46 e* dan *equation 2.46 f* merepresentasikan persamaan dinamik pada *swing leg*. Blok *Function equation inverse 2.46 d* merepresentasikan matriks massa yaitu matriks A pada persamaan (2.46 d). Blok *Function equation 2.46 e* merepresentasikan matriks B pada persamaan (2.46 e). Blok *Function equation 2.46 f* merepresentasikan matriks D pada persamaan (2.46 f). Sesuai dengan persamaan (2.47) dan (2.48) pada bab II sudut pada *upper leg* dan *lower leg* merupakan sudut q_1 dan q_2 , dimana sudut q_1 sama dengan sudut θ_1 dan sudut q_2 merupakan sudut θ_1 ditambah θ_2 .

Pada simulasi *plant swing leg* dengan *software Simulink state* yang dihasilkan berupa sudut q_1 dan q_2 . Sehingga pada simulasi ditambahkan persamaan (2.47) dan (2.48). Gambar 3.3 berikut merupakan gambar model simulasi *swing leg* dengan *state* berupa sudut q_1 dan q_2 :



Gambar 3.3 Model dinamika *swing leg* dengan *state* q_1 dan q_2 .

Dari gambar 3.3 dapat dilihat bahwa output yang dihasilkan berupa sudut q_1 dan q_2 sesuai dengan persamaan (2.47) dan (2.48). Gambar 3.4 berikut adalah gambar blok-blok pada *subsystem swing leg* yang merepresentasikan persamaan (2.46 d), (2.46 e), (2.46 f), (2.47) dan (2.48).



Gambar 3.4 *Subsystem swing leg* dengan state q_1 , q_2 , \dot{q}_1 , \dot{q}_2 , \ddot{q}_1 dan \ddot{q}_2 .

Pada gambar 3.4 *Output* q_1 merepresentasikan posisi sudut q_1 yaitu sudut pada *thigh* (paha) sedangkan pada blok *Output* q_2 merepresentasikan posisi sudut q_2 yaitu sudut *shin* (betis).

Output q_1_dot merepresentasikan kecepatan sudut pada *thigh* (paha). *Output* q_2_dot merepresentasikan kecepatan sudut pada *shin* (betis). *Output* q_1_ddot merepresentasikan percepatan sudut pada *thigh* (paha) sedangkan *output* q_2_ddot merepresentasikan percepatan sudut pada *shin* (betis). Data yang dipakai pada simulasi *swing leg* dapat dilihat pada tabel 3.1 berikut ini:

Tabel 3.1 Data Simulasi *Swing Leg* [2]

Parameter	Keterangan	Satuan	Nilai
m_1	Massa <i>upper leg</i>	kg	0.46
m_2	Massa <i>lower leg</i>	kg	0.31
l_1	Panjang <i>upper leg</i>	m	0.42
l_2	Panjang <i>lower leg</i>	m	0.42
b_1	Jarak pusat massa pada <i>upper leg</i>	m	0.21
b_2	Jarak pusat massa pada <i>lower leg</i>	m	0.21
I_1	Momen inersia <i>upper leg</i>	kg.m ²	0.13
I_2	Momen inersia <i>lower leg</i>	kg.m ²	0.0095

Data pada tabel 3.1 berdasarkan data *real* robot Spring Flamingo buatan Jerry E. Pratt dari MIT [2]. Pada simulasi *swing leg* dengan Simulink parameter-parameter pada tabel 3.1 di atas disederhanakan dalam bentuk persamaan momen inersia yang dapat dilihat pada persamaan (2.25), (2.26), (2.27), (2.28), (2.29), dan (2.30). Berikut perhitungan persamaan momen inersia tersebut.

$$\begin{aligned}
 J_0 &= m_1 b_1^2 + I_1 \\
 &= (0.46^2 \times 0.21^2) + 0.13 = 0.1503 \text{ kg.m}^2
 \end{aligned}$$

$$\begin{aligned}
 J_1 &= m_2 L_1 b_2 \\
 &= 0.31 \times 0.42 \times 0.21 = 0.0273 \text{ kg.m}^2
 \end{aligned}$$

$$\begin{aligned}
 J_2 &= m_2 L_1^2 + m_2 b_2^2 + I_2 \\
 &= (0.31 \times 0.42^2) + (0.31 \times 0.21^2) + 0.0095 = 0.1534 \text{ kg.m}^2
 \end{aligned}$$

$$\begin{aligned}
 J_3 &= m_2 b_2^2 + I_2 \\
 &= (0.31^2 \times 0.21^2) + 0.0095 = 0.0232 \text{ kg.m}^2
 \end{aligned}$$

$$G_1 = (m_1 b_1 + m_2 L_1)g$$

$$= ((0.46 \times 0.21) + (0.31 \times 0.42)) \times 9.81 = 2.2249 \text{ kg} \cdot \text{m}^3 / \text{s}^2$$

$$G_2 = m_2 b_2 g$$

$$= 0.31 \times 0.21 \times 9.81 = 0.6386 \text{ kg} \cdot \text{m}^3 / \text{s}^2$$

Data-data hasil perhitungan di atas kemudian dimasukkan ke model *swing leg* yang telah dibuat.

3.2 Verifikasi *Double Pendulum* Menggunakan SimMechanics

Pada bab II telah dijelaskan bahwa *swing leg* pada *bipedal walking robot* dimodelkan sebagai *double pendulum*. *Double pendulum* adalah dua buah massa yang melekat pada batang yang *rigid* dan tak bermassa [14]. Sumbu pada *double pendulum* diasumsikan tidak mempunyai gaya gesek [14]. Perbedaannya dengan *double pendulum* pada model *swing leg* ini yaitu pusat massa pada *double pendulum* ini terletak di tengah-tengah masing-masing batang. Simulasi *swing leg* yang dilakukan dengan Simulink, dimodelkan berdasarkan persamaan matematik (2.48). Untuk itu diperlukan adanya verifikasi menggunakan SimMechanics agar dapat dibandingkan antara hasil simulasi dengan menggunakan Simulink dan dengan menggunakan SimMechanics.

Pada umumnya, selalu dibutuhkan persamaan gerak untuk mensimulasikan dinamika. Namun, pada kasus-kasus tertentu untuk menurunkan persamaan gerak sangatlah rumit. Seperti yang telah dibahas pada bab II, SimMechanics menyediakan alat bantu untuk mensimulasikan sistem dinamik tanpa menggunakan persamaan dinamik dari suatu sistem tersebut. Begitu pula ketika simulasi *swing leg* dengan menggunakan SimMechanics tidak diperlukan persamaan dinamik *swing leg*. Pada SimMechanics hanya membutuhkan parameter kondisi awal pada *swing leg*. Parameter kondisi awal tersebut antara lain parameter massa, momen inersia, serta jarak pusat massa.

Berikut langkah-langkah untuk membuat model *swing leg* dengan menggunakan SimMechanics:

1. Buka MATLAB dengan *double* klik MATLAB *icon* atau pilih dari *windows* menu.

2. Pada MATLAB *prompt* ketik kata berikut untuk membuka *Simulink library browser* : >> Simulink.
3. Dari *library browser*, pilih “File → New → Model” dan akan keluar *window* kosong yang nantinya akan menjadi lembar kerja atau *worksheet*.
4. Klik “SimMechanics” atau “Seascope → SimMechanics” pada *library browser*.
5. *Double* klik icon “Bodies”.
6. Klik dan tahan icon “Ground”.
7. *Drag* dan *drop* icon “Ground” di dalam lembar kerja yang telah di buka pada langkah ke tiga.
8. *Double* klik pada blok “Ground” pada lembar kerja, dan pastikan lokasi berada pada “[0 0 0]”. Ini merupakan koordinat referensi x,y dan z. Gambar 3.5 berikut merupakan kotak dialog blok *Ground* :



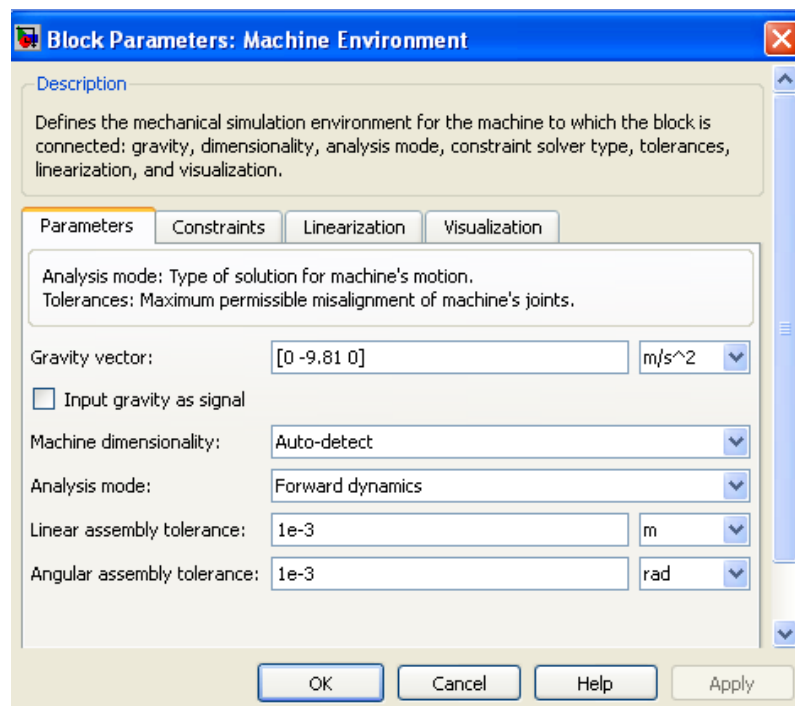
Gambar 3.5 Kotak dialog pada blok *Ground*.

9. Cek “*Show machine environment port*” dan akan muncul *port* bundar di bagian kiri blok “*Ground*”. Seperti terlihat pada gambar 3.6 berikut :



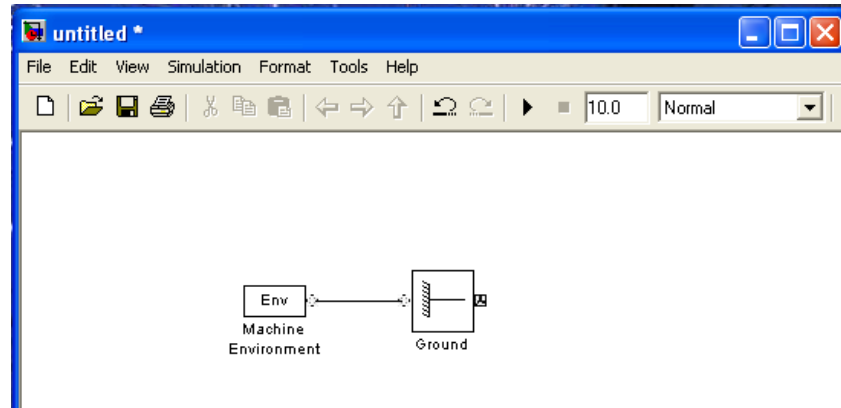
Gambar 3.6 Kotak dialog pada blok *Ground* setelah cek *Show Machine Environment port*.

10. Pada SimMechanics, sumbu x positif berada pada arah kanan, sumbu y positif berada pada arah atas dan sumbu z positif berada pada arah keluar dari layar komputer.
11. Seperti pada langkah ke delapan, *drag* dan *drop icon* “*Machine Environment*” pada lembar kerja.
12. *Double click* pada “*Machine Environment*” akan muncul kotak dialog seperti pada gambar 3.7 berikut. Kemudian masukkan nilai vektor gravitasi sebesar $[0 \ -9.81 \ 0]$. Hal ini memastikan bahwa gaya gravitasi berada pada sumbu y negatif. Blok *Machine Environment* ini juga menunjukkan mode analisis yang terdiri dari *Forwards Dynamics*, *Inverse Dynamics*, *Kinematics* dan *Trimming*. Untuk pemodelan *plant swing leg open loop* dengan SimMechanics ini pilih mode analisis *Forwards Dynamic*. Sedangkan pada kolom *Machine dimensionality* terdapat tiga pilihan yaitu *3-D Only*, *2-D Only* dan *Auto-detect*. Untuk pemodelan *plant swing leg open loop* ini pilih *Auto-detect*.



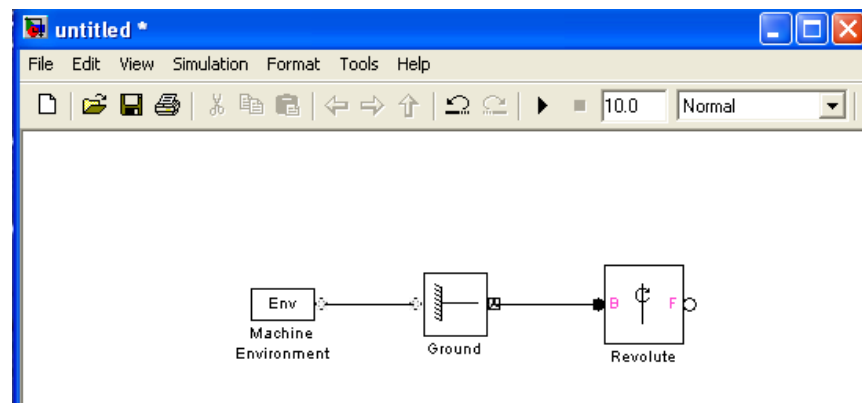
Gambar 3.7 Kotak dialog pada blok *Machine Environment*.

13. Kemudian hubungkan blok “*Machine Environment*” dengan blok “*Ground*” dengan klik dan *drag* antara *port* blok yang ingin dihubungkan. Kemudian akan muncul diagram seperti gambar 3.8 berikut :



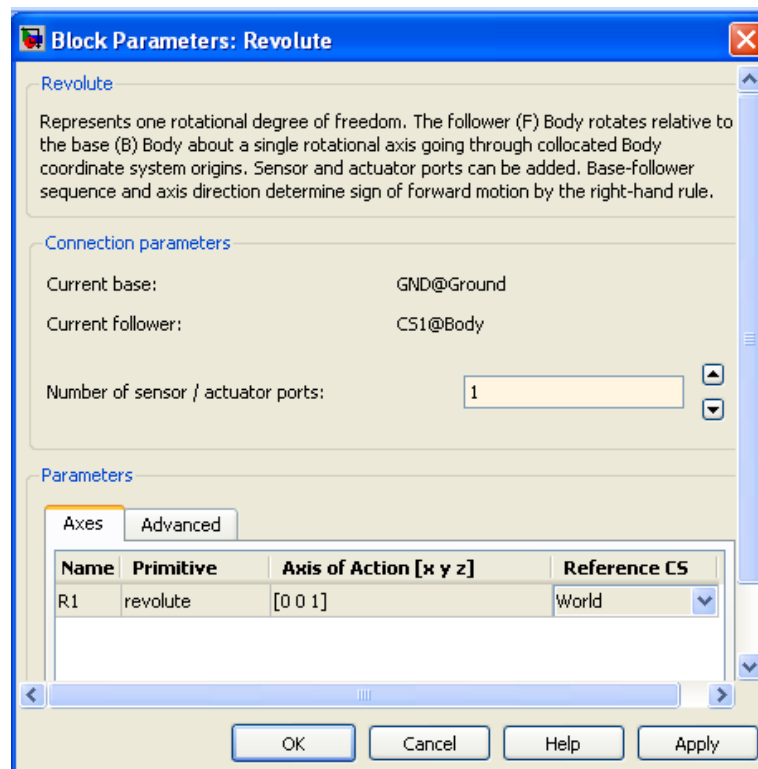
Gambar 3.8 Tampilan blok *Machine Environment* dan *Ground* pada lembar kerja.

14. Buka *Library Browser*.
15. Klik “*SimMechanics* → *Joints*” dan cari “*Revolute*”, yang merupakan representasi dari *rotational joint* pada tiap-tiap pendulum.
16. *Drag* dan *drop* blok “*Revolute*” pada lembar kerja dan hubungkan dengan blok “*Ground*”. Sehingga akan terlihat tampilan seperti pada gambar 3.9 berikut :



Gambar 3.9 Tampilan blok *Machine Environment*, *Ground*, dan *Revolute* pada lembar kerja.

17. Pada langkah ini, sebaiknya simpan dulu pekerjaan yang telah dilakukan. Klik “*File* → *Save*” dan simpan dengan nama “*double_pendulum.mdl*”.
18. *Double* klik “*Revolute*” maka akan muncul kotak dialog seperti pada gambar 3.10 berikut. Blok *Revolute* tersebut mewakili sebuah derajat kebebasan. Pada blok tersebut dapat ditambahkan *port* sensor dan aktuator.
- Dari gambar 3.10 kemudian pastikan “*Axis of Action [x y z]*” bernilai [0 0 1]. Karena pendulum berputar terhadap sumbu z, maka sumbu *revolute* harus terletak pada sumbu z yang diwakilkan dengan vektor [0 0 1]. Kemudian pada kolom *Reference CS* pilih *World*.



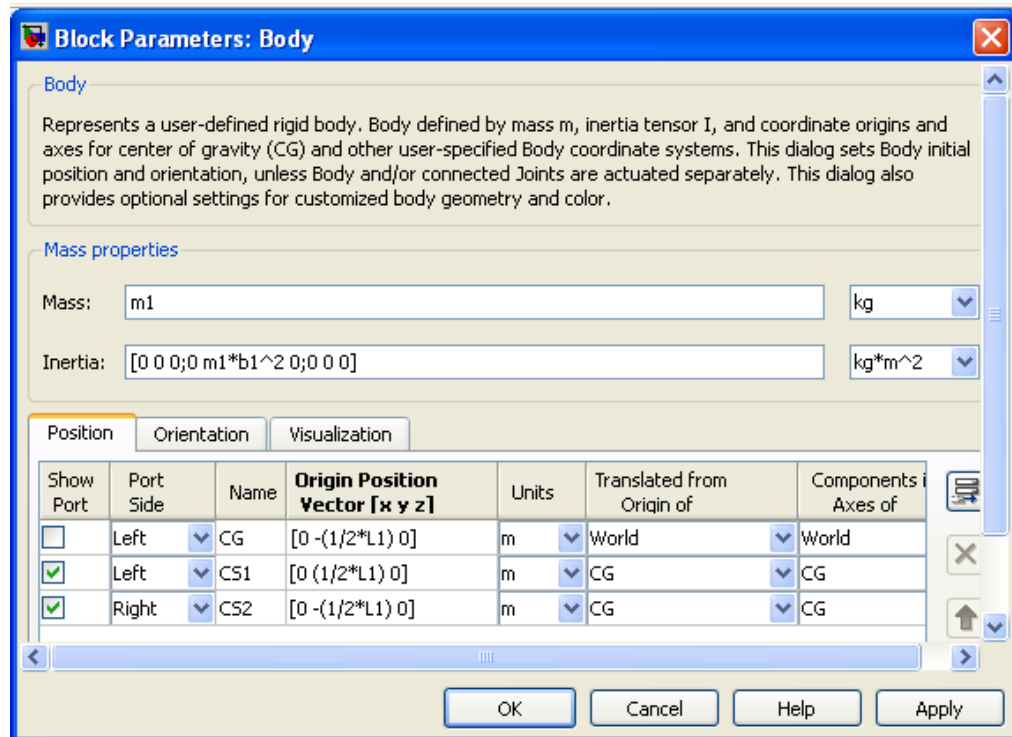
Gambar 3.10 Kotak dialog pada blok *Revolute*.

19. Pada *library browser*, buka “*SimMechanics* → *Bodies*” kemudian *drag* dan *drop* blok “*Body*” pada lembar kerja. Blok ini merepresentasikan pendulum pertama.
20. *Double* klik pada blok “*Body*” dan masukkan nilai parameter untuk pendulum pertama.

Contoh $Mass = \text{“m1” [kg]}$

$Inertia = \text{“[0 0 0; 0 m1*b1^2 0; 0 0 0]” [kg*m^2]}$

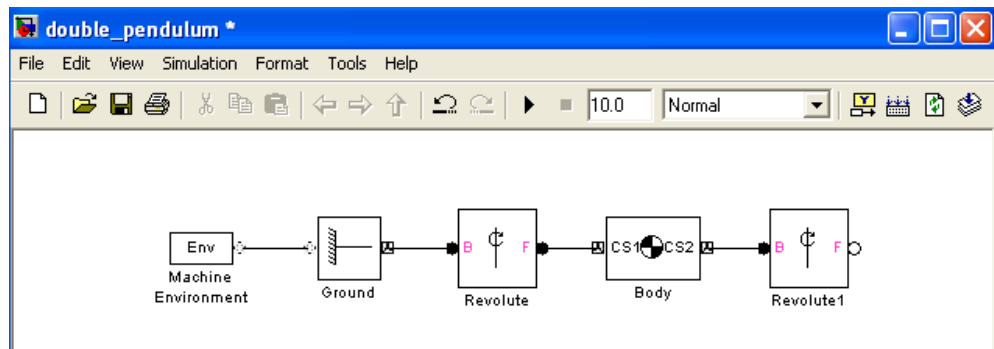
Dimana momen inersia adalah matriks rotasi inersia yang merupakan matriks dengan dimensi 3x3. Karena yang diperlukan hanya rotasi terhadap sumbu y maka inersia pada sumbu y berdasarkan data parameter pada tabel 3.1 pada pendulum pertama. Untuk lebih jelas dapat dilihat pada gambar 3.11 berikut:



Gambar 3.11 Kotak dialog pada blok *Body*.

Gambar 3.11 menunjukkan variasi sistem koordinat yaitu koordinat CG, CS1 dan CS2. CG adalah posisi *center of gravity* (pusat massa), yaitu posisi pusat massa pada *upper leg* atau posisi b_1 yang sesuai dengan acuan *world coordinat* sesuai dengan data pada tabel 3.1. Nilai minus menunjukkan bahwa pusat massa berada pada sumbu y negatif. Sedangkan CS1 adalah koordinat sistem yang berhubungan dengan *joint* pertama dan CS2 adalah koordinat sistem yang berhubungan dengan bagian bawah pendulum dimana akan menjadi posisi untuk *joint* yang selanjutnya.

21. Tambahkan “*Revolute*” kedua dan hubungkan dengan pendulum pertama, seperti pada gambar 3.12 berikut. Blok *Revolute* tersebut merepresentasikan *joint* kedua pada simulasi *plant swing leg open loop* ini.

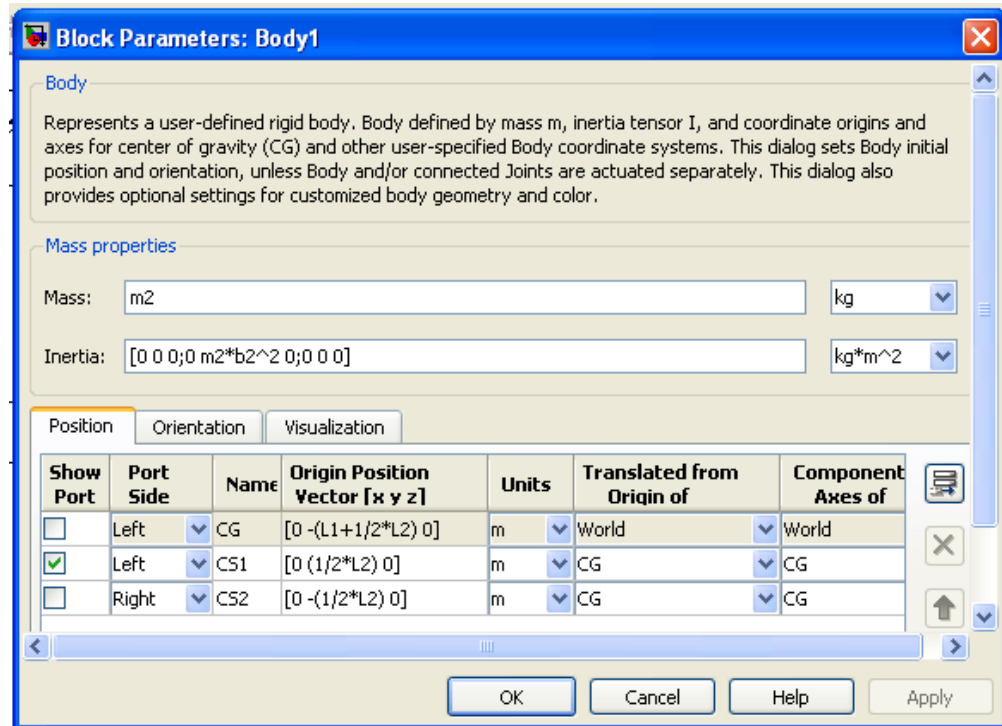


Gambar 3.12 Tampilan blok *Machine Environment*, *Ground*, *Revolute*, *Body* dan *Revolute 1* pada lembar kerja *double pendulum*.

22. Tambahkan “*Body*” yang merepresentasikan pendulum kedua, dan hubungkan dengan blok “*Revolute*” yang kedua. *Double* klik blok *Body* kemudian akan muncul kotak dialog seperti pada gambar 3.11. Kemudian masukkan nilai parameter massa m_2 pada kolom *Mass* dan nilai parameter momen inersia $[0 \ 0 \ 0; 0 \ m_2 * b_2^2 \ 0; 0 \ 0 \ 0]$ pada kolom *Inertia*. Data parameter tersebut berdasarkan tabel 3.1 dan dicantumkan pada *m file* lampiran.

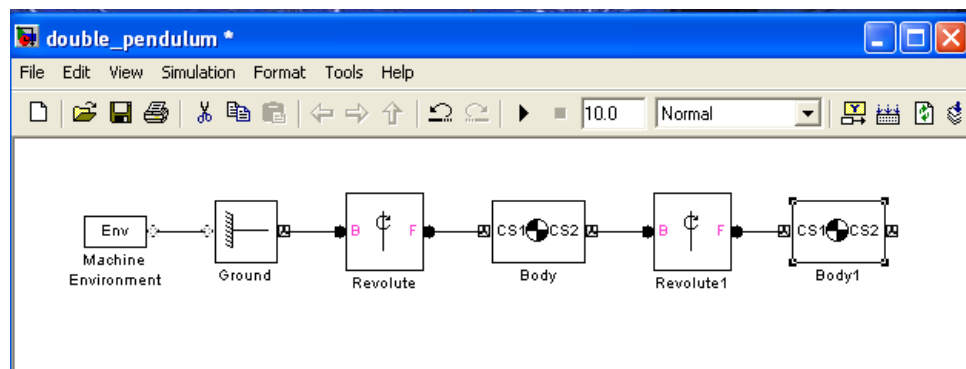
Pada gambar 3.13 kotak dialog *Body* di bawah ini terdapat variasi sistem koordinat yang sama seperti pada langkah nomor 20. Variasi koordinat tersebut antara lain CG, CS1 dan CS2. CG adalah posisi *center of gravity* (pusat massa), yaitu posisi pusat massa pada *lower leg* atau posisi b_2 yang sesuai dengan acuan *world coordinate* yang bernilai $-(L_1 + 1/2 * L_2)$. Sedangkan CS1 dan CS2 menunjukkan koordinat posisi awal dan posisi akhir pada pendulum kedua atau pada kaki bawah *plant swing leg*.

Dimana *port* ketiga yang tidak dibutuhkan bisa dihapus dengan cara, pilih *port* untuk CS2 dan klik tombol *delete*. Tombol *delete* ada pada sisi kanan blok parameter *window* berupa tanda X warna merah. Untuk lebih jelas dapat dilihat pada gambar 3.13 berikut:



Gambar 3.13 Kotak dialog blok *Body*.

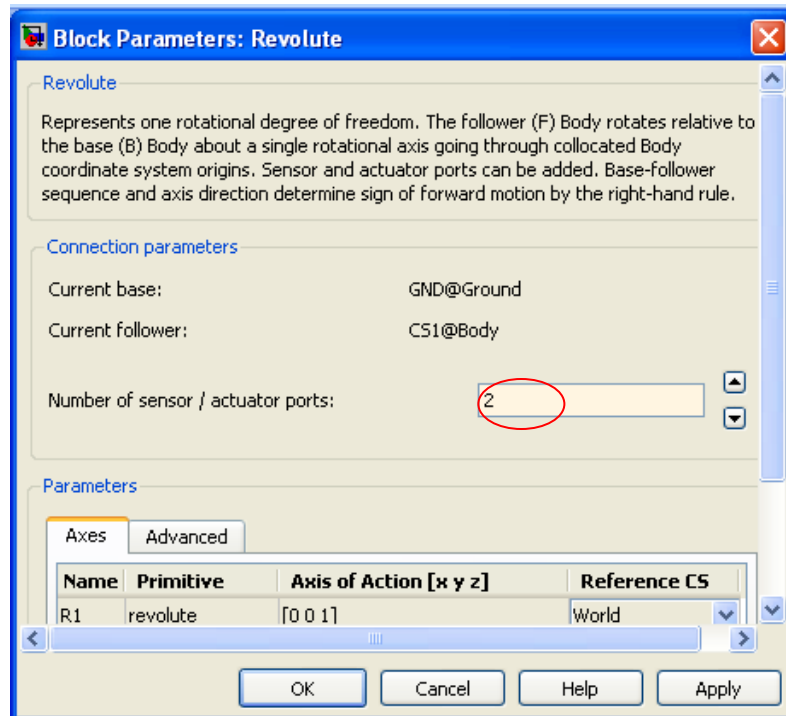
23. Setelah dilakukan langkah nomor 22 maka lembar kerja akan terlihat seperti pada gambar 3.14 berikut :



Gambar 3.14 Tampilan blok *Machine Environment*, *Ground*, *Body*, *Revolute 1*, dan *Body 1* pada lembar kerja *double pendulum*.

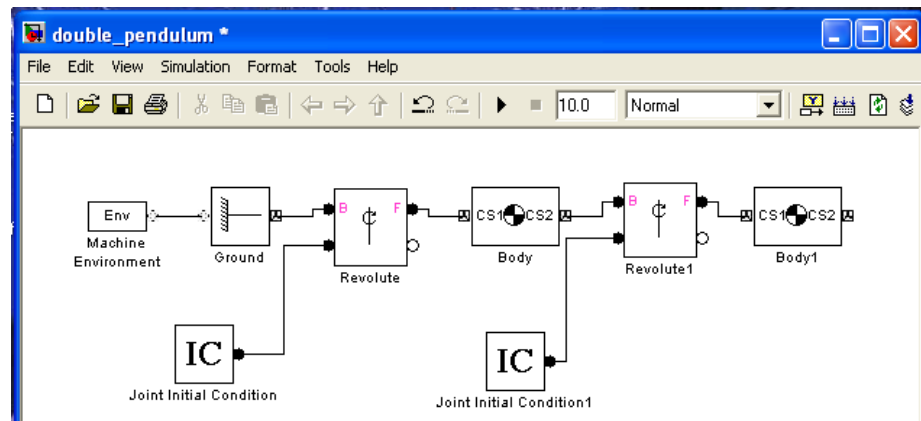
24. Untuk memasukkan parameter sudut untuk tiap blok *Revolute*, *double* klik tiap blok *Revolute* untuk membuka menu *window* dan masukkan nilai “*Number of*

sensors/actuator ports” dengan angka 2. Seperti pada gambar kotak dialog 3.15 berikut:



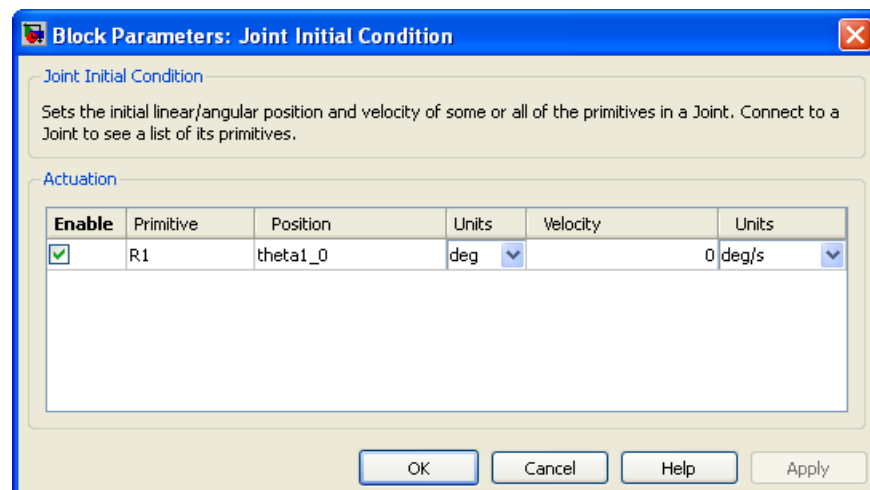
Gambar 3.15 Tampilan kotak dialog *Revolute*.

25. Dari “SimMechanics → *Sensors and Actuators*”, drag dan drop blok “*Joint Initial Condition*” dua kali dan hubungkan pada *port* baru yang telah di buat pada langkah nomor 25. Blok *Joint Initial Condition* merupakan blok untuk menentukan kondisi awal sistem yang akan disimulasikan. Kondisi awal tersebut dapat berupa posisi awal ataupun kecepatan awal pada sistem yang akan disimulasikan. Gambar 3.16 berikut adalah gambar tampilan lembar kerja setelah penambahan blok *Joint Initial Condition* :



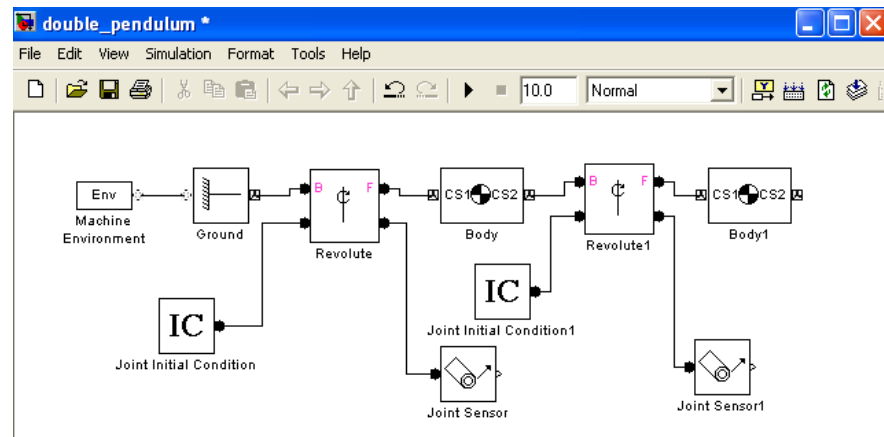
Gambar 3.16 Tampilan lembar kerja *double pendulum* setelah penambahan blok *Joint Initial Condition* (IC).

Double klik tiap IC blok dan atur posisi awal sudut untuk *joint* pertama pada sebesar θ_{1_0} dan posisi awal sudut untuk *joint* kedua sebesar θ_{2_0} . Nilai kondisi awal tersebut ada pada *m file* yang akan dicantumkan pada lampiran. Dan kondisi awal untuk kecepatan sebesar nol pada kedua blok. Untuk lebih jelas dapat dilihat pada gambar 3.17 berikut :



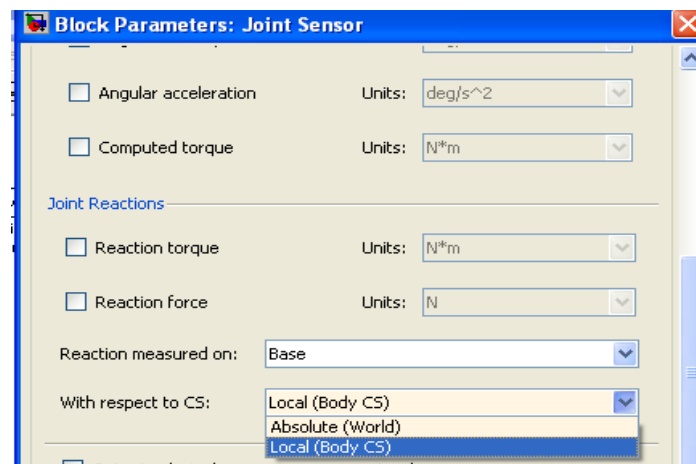
Gambar 3.17 Kotak dialog pada blok IC (*Joint Initial Condition*).

26. Masuk ke “SimMechanics → *Sensor and Actuators*” kemudian *drag* dan *drop* blok “*Joint Sensor*” dua kali dan hubungkan kemudian hubungkan dengan *port* pada “*Revolute*”. Seperti pada gambar 3.18 berikut :



Gambar 3.18 Tampilan lembar kerja *double pendulum* setelah penambahan blok *Joint Sensor*.

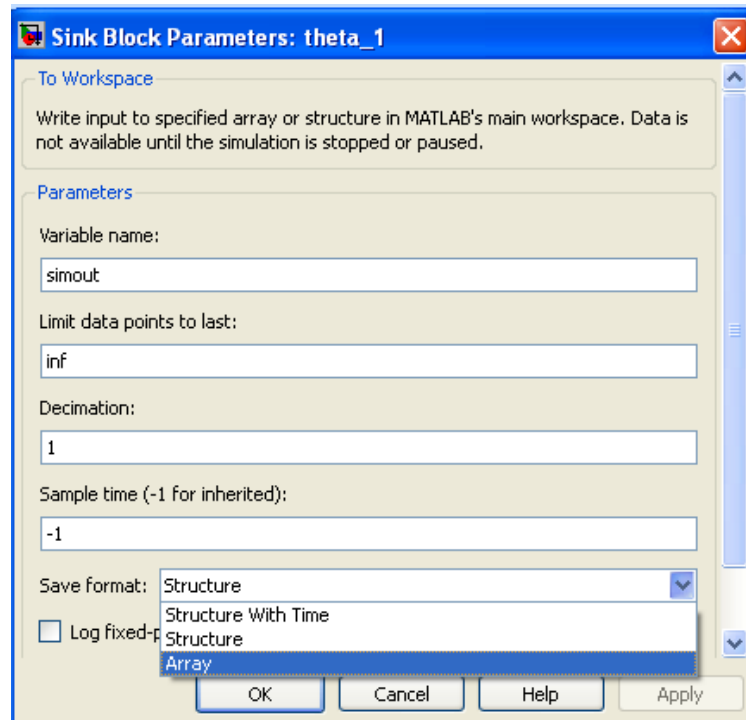
27. Double klik pada blok “*Joint Sensor*” dan ubah “*Absolute (World)*” menjadi “*Local (Body CS)*” pada “*Joint Reactions/With respect to CS*”. Seperti pada gambar 3.19 berikut :



Gambar 3.19 Kotak dialog pada blok *Joint Sensor*.

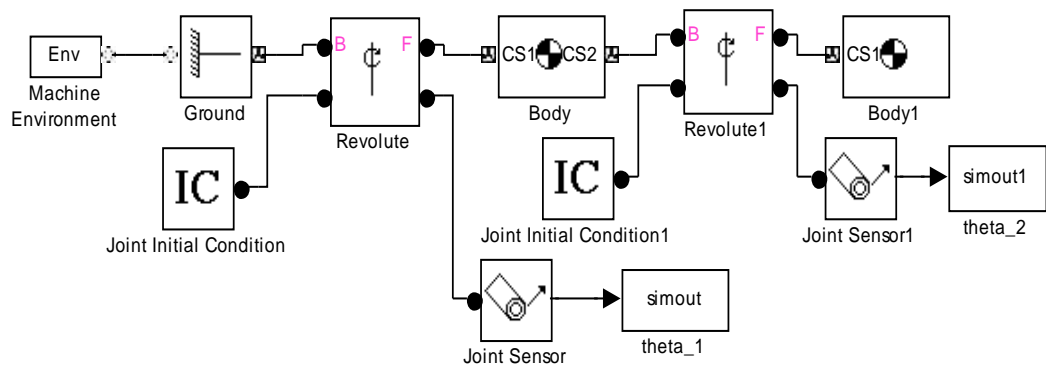
28. Pada Simulink *library browser*, buka “Simulink → sink” kemudian *drag* dan *drop* blok “*To Workspace*” ke lembar kerja dan hubungkan pada “*Joint Sensor*” dan lakukan hal yang sama pada “*Joint Sensor*” yang kedua.
29. Double klik tiap-tiap blok “*To Workspace*” dan ganti namanya menjadi “*theta_1*” dan “*theta_2*” untuk tiap-tiap *joint*. Ganti “*save format*” menjadi “*Array*” untuk

kedua blok dengan cara *double* klik pada blok *To Workspace* kemudian akan muncul kotak dialog seperti pada gambar 3.20 berikut :



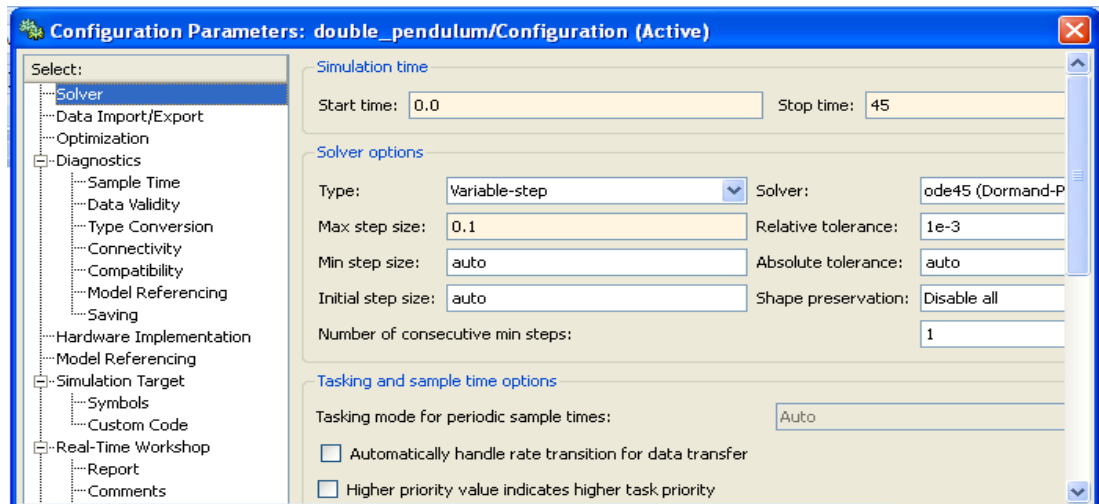
Gambar 3.20 Kotak dialog pada blok *To Workspace*.

Kemudian simpan file. Hasil simulasi akan terlihat seperti pada gambar 3.21. Gambar 3.21 berikut merupakan gambar model simulasi *double pendulum* secara keseluruhan. Model pada gambar 3.21 tersebut tidak ditambahkan sistem kontrol. Hal ini dikarenakan model tersebut digunakan untuk verifikasi gerakan *swing leg*.



Gambar 3.21 Simulasi *double pendulum* secara keseluruhan.

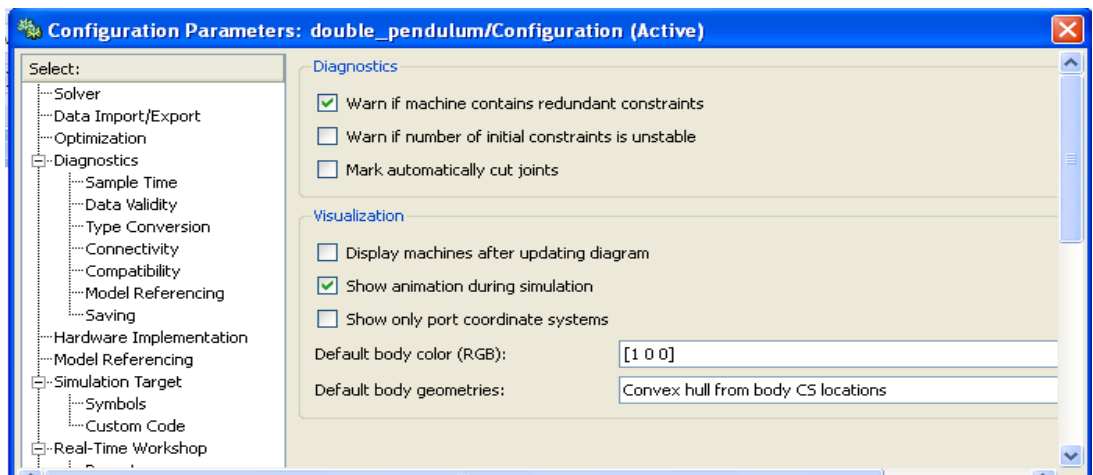
30. Pada pemodelan fisik *double pendulum* ini sudut pada kedua pendulum mengacu pada sumbu θ_1 sehingga sudut *thigh* (paha) yaitu $q_1 = \theta_1$, sedangkan sudut *shin* (betis) yaitu $q_2 = \theta_1 + \theta_2$. Pada gambar 3.21 di atas hasil simulasi berupa nilai θ_1 dan θ_2 sehingga perlu dijumlahkan untuk mendapatkan nilai q_1 dan q_2 . Dengan cara klik “*Simulink library browser* → *math operation*” kemudian *drag* dan *drop* blok *Add* pada lembar kerja. Untuk melihat hasil grafik yang diperoleh dari simulasi *double pendulum*, tambahkan blok *scope* pada lembar kerja dengan cara klik “*Simulink library browser* → *commonly used blocks*”. Kemudian *drag* dan *drop* blok *Scope* pada lembar kerja. Kemudian beri nama masing-masing blok *Scope* dengan nama q_1 dan q_2 . Setelah itu simpan kembali hasil pekerjaan yang telah dilakukan.
31. Pada lembar kerja *window*, klik “*Simulation* → *Configuration parameters*”. Pada kolom *Select* pilih *Solver*. *Solver* merupakan penyelesaian untuk model dinamik dari *plant swing leg open loop* dengan menggunakan *SimMechanics*. Kemudian atur *stop time* sebesar 45. *Stop time* tersebut merupakan waktu berhenti simulasi. Kemudian atur *max step size* sebesar 0,1. Pada kolom *Solver* pilih *ode45* (*Dormand-prince*). Untuk lebih jelas lihat pada gambar 3.22 berikut :



Gambar 3.22 Kotak dialog *Configuration Parameter-Solver option*.

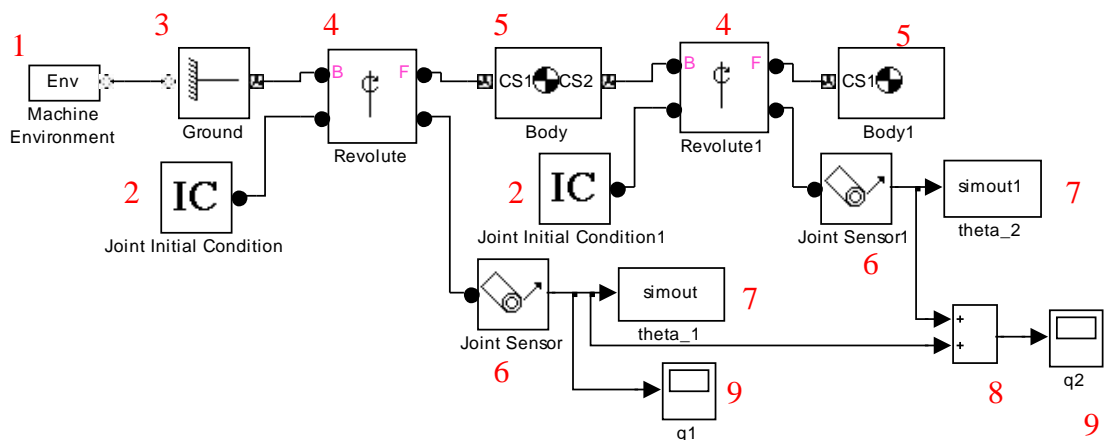
32. Pada *configuration window*, klik “*Data Import/Export*” dan klik “*Limit Data points to be*” pada *save option*.

33. Pada *configuration window*, klik “SimMechanics” *option* pada sisi kiri *window*. Pilih “*Show animation during simulation*”. Pilihan *Show animation during simulation* akan menampilkan animasi *double pendulum* selama simulasi berjalan sehingga dapat dilihat bagaimana gerakan pendulum beresilasi. Klik OK untuk menutup *configuration window*. Gambar 3.23 berikut adalah gambar kotak dialog pada *Configuration Parameter-SimMechanics option* :



Gambar 3.23 Kotak dialog *Configuration Parameter-SimMechanics option*.

34. Buka lembar kerja dan klik “*Simulation → Start*”. Akan terlihat gerakan pendulum beresilasi. Untuk melihat hasil grafik setelah pendulum beresilasi *double* klik pada blok *scope*, maka akan muncul grafik untuk q_1 dan q_2 . Gambar 3.24 berikut merupakan gambar simulasi *double pendulum* dengan asumsi pusat massa berada pada tengah-tengah batang pendulum.



Gambar 3.24 Simulasi *double pendulum*.

Gambar 3.24 merupakan model simulasi *double pendulum* secara keseluruhan dimana tiap-tiap blok pada SimMechanics mewakili dan merepresentasikan parameter-parameter kondisi yang berpengaruh pada simulasi *double pendulum*. Berikut ini penjelasan dari masing-masing blok pada gambar 3.24:

1. Blok nomor 1 merupakan blok *Machine Environment* yang berfungsi untuk menyesuaikan *mechanical environment* pada mesin yang akan disimulasikan. *Mechanical environment* yang akan disimulasikan antara lain berupa bagaimana mensimulasikan mesin, bagaimana menginterpretasikan *mechanical constraint*, bagaimana melinearisasikan simulasi yang dilakukan, dan bagaimana menampilkan mesin dalam bentuk visual.
2. Blok nomor 2 merupakan blok *Joint Initial Condition* yang berfungsi untuk menentukan kondisi awal sesuai dengan data simulasi yang dibutuhkan. Kondisi awal tersebut antara lain kondisi awal untuk posisi dan kondisi awal untuk kecepatan. Kondisi awal pada simulasi *double pendulum* ini yaitu posisi sebesar 45° sedangkan kondisi awal untuk kecepatan yaitu sebesar 0 m/s.
3. Blok nomor 3 merupakan blok *Ground* yang merepresentasikan sebuah *ground point* yang tidak bisa bergerak. *Ground point* adalah titik tetap yang acuannya terletak pada *absolute world coordinate*.
4. Blok nomor 4 merupakan blok *Revolute* yang merepresentasikan sebuah *joint* dengan satu rotasional *degree of freedom*. Blok *Revolute* tersebut merepresentasikan *joint* pertama pada simulasi *double pendulum*. Sedangkan blok *Revolute 1* merepresentasikan *joint* kedua pada simulasi *double pendulum*.
5. Blok nomor 5 merupakan blok *Body* yang merepresentasikan sebuah *rigid body* dengan parameter yang bisa disesuaikan menurut kebutuhan. Parameter-parameter yang dapat disesuaikan antara lain massa *body* dan momen inersia, koordinat dari *center of gravity* pada *body* dan sistem koordinat *body*. Blok *Body* ini merepresentasikan bagian *thigh* (paha). Sedangkan blok *Body 1* merepresentasikan bagian *shin* atau (betis).
6. Blok nomor 6 merupakan blok *Joint Sensor* yang berfungsi untuk mengukur posisi, kecepatan dan percepatan pada suatu *joint*.

7. Blok nomor 7 merupakan blok *To Workspace* yang berfungsi untuk memasukkan data sinyal hasil simulasi ke lembar kerja MATLAB.
8. Blok nomor 8 merupakan blok *Add* yang berfungsi untuk menambahkan suatu input. Blok ini dapat menambahkan besaran skalar, vektor maupun matriks.
9. Blok nomor 9 merupakan blok *Scope* yang berfungsi untuk memvisualisasikan sinyal berdasarkan waktu simulasi. Hasil visualisasi berupa grafik. Pada blok *Scope* grafik yang ada merupakan grafik posisi θ_1 atau q_1 . Sedangkan pada blok *Scope 1* grafik yang ada merupakan grafik posisi $\theta_1 + \theta_2$ atau q_2 .

3.3 Desain *Adaptive Control*

Sistem kontrol yang digunakan pada *plant swing leg* yaitu sistem *adaptive control*. *Adaptive control* merupakan sistem yang umumnya digunakan pada dunia robotika. Dengan asumsi bahwa kurva posisi yang diinginkan ($q_d(t)$) diketahui dan beberapa dinamik parameter pada sistem tidak diketahui. Permasalahan pada desain *adaptive control* adalah menentukan hukum kontrol untuk torsi pada *revolute joint* dan menentukan estimasi untuk parameter-parameter sistem yang tidak diketahui sehingga output pada grafik pada *plant swing leg* (q) dapat mengikuti kurva yang diinginkan ($q_d(t)$) dengan presisi.

Pada laporan penelitian ini kurva posisi yang diinginkan dinyatakan dalam bentuk fungsi $q_d(t)$ [2]. Fungsi tersebut dinyatakan dalam bentuk matriks yaitu fungsi $q_{1d}(t)$ dan $q_{2d}(t)$. Berikut ini fungsi $q_{1d}(t)$ dan $q_{2d}(t)$ berdasarkan referensi [2] :

$$q_{1d}(t) = 0.8 \sin(2\pi(0.05)t) + 0.25 \sin(2\pi(1.0)t) \quad (3.1)$$

$$q_{2d}(t) = -0.4 + q_{1d}(t) + 0.4 \sin(2\pi(0.03)t) + 0.15 \sin(2\pi(1.5)t) \quad (3.2)$$

Pada fungsi (3.1) fungsi $q_{1d}(t)$ merupakan kurva posisi yang diinginkan (*desired position*) untuk posisi sudut q_1 atau posisi sudut thigh (paha). Sehingga output kurva q_1 menghasilkan kurva yang presisi dengan kurva yang diinginkan yaitu kurva $q_{1d}(t)$.

Sedangkan pada fungsi (3.2) fungsi $q_{2d}(t)$ merupakan kurva posisi yang diinginkan (*desired position*) untuk posisi sudut q_2 atau posisi sudut *shin* (betis).

Sehingga output kurva q_2 menghasilkan kurva yang presisi dengan kurva yang diinginkan yaitu kurva $q_{2d}(t)$. Pemodelan sistem *adaptive control* disimulasikan menggunakan *software* MATLAB/Simulink berdasarkan persamaan (2.60), (2.61), (2.62), (2.63), (2.64), (2.65) dan (2.66) dengan memasukkan data-data parameter yang sudah ada pada referensi [2]. Berikut ini merupakan parameter kontrol yang digunakan pada *adaptive control*:

Tabel 3.2 Data Parameter *Adaptive Control* pada *Swing Leg* [2]

Parameter	Keterangan	Nilai
K_{d1}	Konstanta kontroler <i>derivative</i>	1.0
K_{d2}	Konstanta kontroler <i>derivative</i>	0.7
λ_1	Konstanta <i>adaptive gain</i>	16.0
λ_2	Konstanta <i>adaptive gain</i>	10.0
Γ_{11}	Matriks adaptasi massa	0.005
Γ_{22}	Matriks adaptasi massa	0.001
Γ_{33}	Matriks adaptasi massa	0.001
Γ_{44}	Matriks adaptasi massa	0.30
Γ_{55}	Matriks adaptasi massa	0.08

Pada simulasi dibutuhkan nilai persamaan \dot{q}_{1d} , \dot{q}_{2d} , \ddot{q}_{1d} dan \ddot{q}_{2d} yang merupakan turunan pertama dan turunan kedua dari persamaan (3.1) dan (3.2). Persamaan berikut merupakan turunan dari persamaan (3.1) dan (3.2).

$$\dot{q}_{1d} = 0,25 \cos(2\pi(0.05)t) + 1,57 \cos(2\pi(1.0)t) \quad (3.3)$$

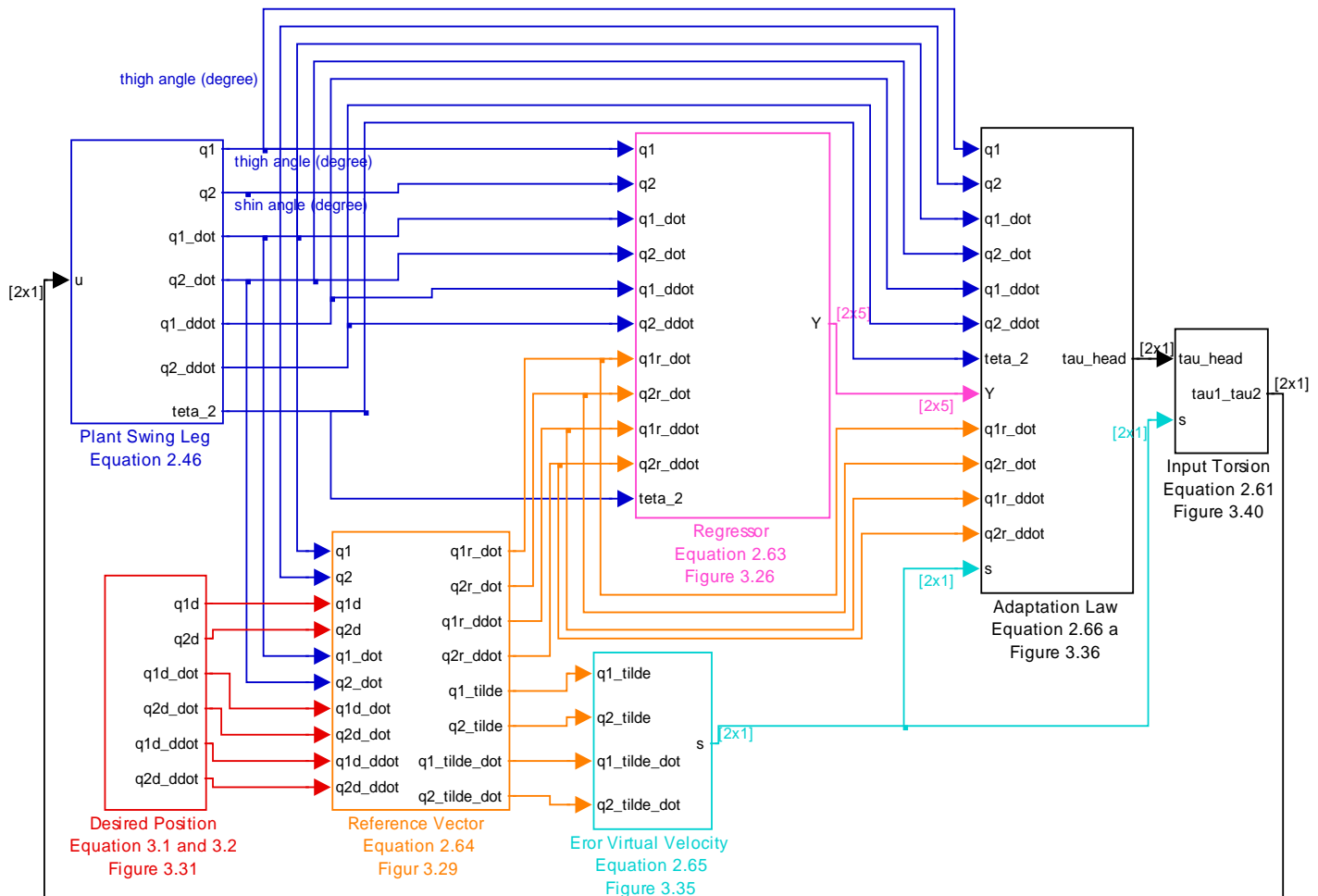
$$\dot{q}_{2d} = \dot{q}_{1d} + 0,075 \cos(2\pi(0.03)t) + 1,41 \cos(2\pi(1.5)t) \quad (3.4)$$

$$\ddot{q}_{1d} = -0,078 \sin(2\pi(0.05)t) - 9,86 \sin(2\pi(1.0)t) \quad (3.5)$$

$$\ddot{q}_{2d} = \ddot{q}_{1d} - 0,014 \sin(2\pi(0.03)t) - 13,28 \sin(2\pi(1.5)t) \quad (3.6)$$

Gambar 3.25 berikut adalah gambar simulasi *swing leg* dan desain kontrol secara keseluruhan yang dilakukan dengan alat bantu *software* MATLAB/Simulink. Masing-

masing *subsystem* mewakili persamaan-persamaan hukum *adaptive control* pada persamaan (2.61), (2.62), (2.63), (2.64), (2.65), (2.66) dan (2.67).



Gambar 3.25 Gambar simulasi *swing leg* beserta desain *adaptive control*.

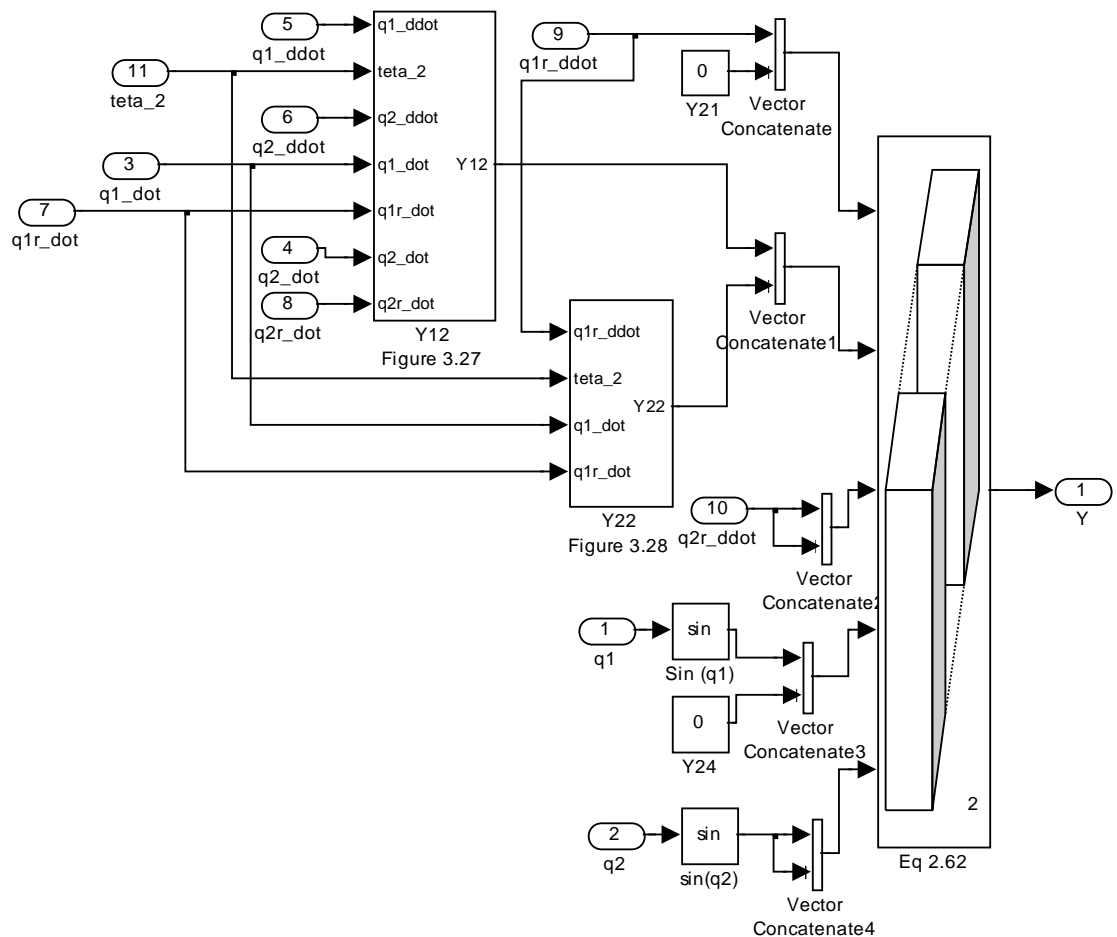
Dari gambar 3.25 dapat dilihat bahwa masing-masing *subsystem* mewakili persamaan-persamaan pada sistem *adaptive control*. Berikut ini penjelasan masing-masing *subsystem* untuk gambar 3.25 :

1. *Subsystem Plant Swing Leg*

Blok *Subsystem plant swing leg* mewakili persamaan (2.46), (2.46 d), (2.46 e) dan (2.46 f). Dimana sudut pada *thigh* (paha) dan *shin* (betis) dinyatakan dengan notasi q_1 dan q_2 sesuai dengan persamaan (2.47) dan (2.48).

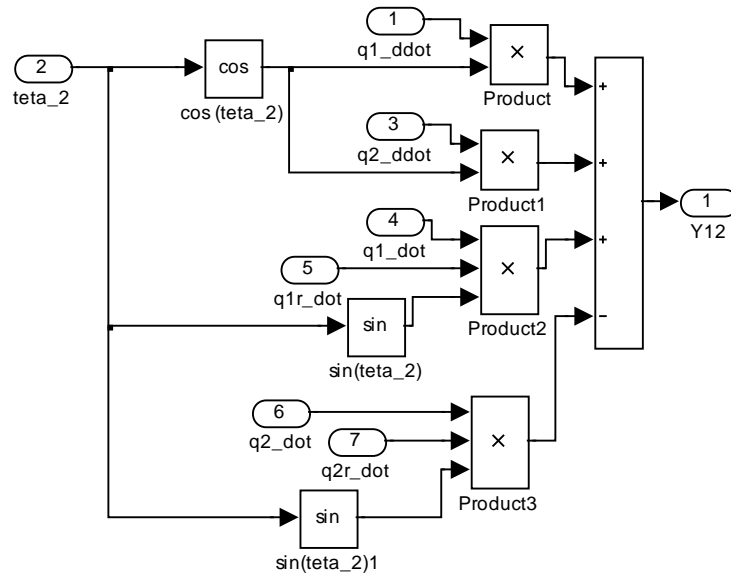
2. Subsystem Regressor

Blok *subsystem regressor* pada gambar 3.25 merupakan sebuah *subsystem* yang mewakili matriks Y pada persamaan (2.63), gambar 3.26 berikut merupakan gambar *subsystem* untuk matriks Y tersebut :



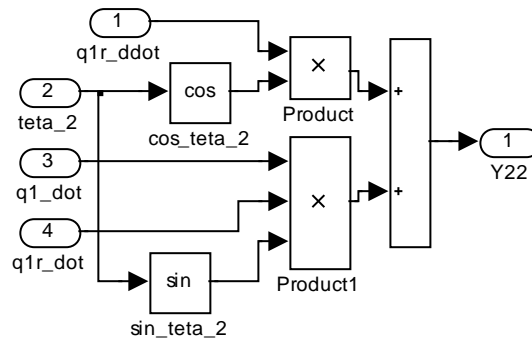
Gambar 3.26 Gambar *subsystem* untuk matriks Y pada persamaan (2.62).

Dari gambar 3.26 blok *subsystem* Y12 merupakan *subsystem* yang mewakili nilai matriks Y yang terletak pada baris pertama dan kolom kedua yaitu $(\ddot{q}_1 \cos(\theta_2) + \ddot{q}_2 \cos(\theta_2) + \dot{q}_1 \dot{q}_{1r} \sin(\theta_2) - \dot{q}_2 \dot{q}_{2r} \sin(\theta_2))$. Gambar 3.27 berikut adalah gambar blok *subsystem* Y12.



Gambar 3.27 Gambar blok *subsystem* Y12 yang mewakili matriks Y pada baris pertama dan kolom kedua.

Sedangkan blok *subsystem* Y22 merupakan *subsystem* yang mewakili nilai matriks Y yang terletak pada baris kedua dan kolom kedua yaitu $(\ddot{q}_{1r} \cos(\theta_2) + \dot{q}_1 \dot{q}_{1r} \sin(\theta_2))$. Gambar 3.28 berikut merupakan gambar blok *subsystem* Y22.

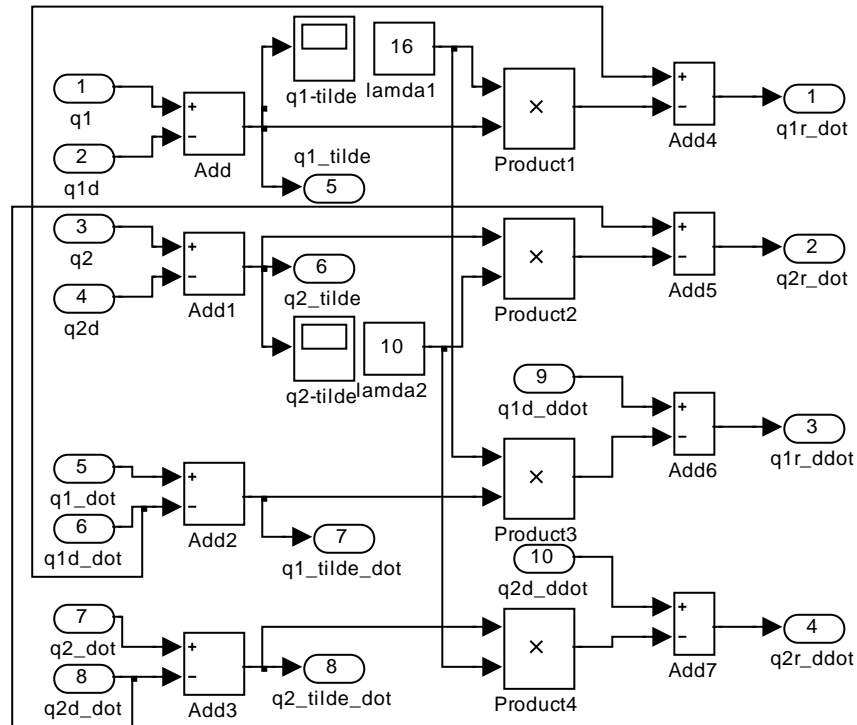


Gambar 3.28 Gambar blok *subsystem* Y22 yang mewakili matriks Y pada baris kedua dan kolom kedua.

3. Subsystem Reference Vector

Blok *subsystem reference vector* pada gambar 3.25 yang merupakan gambar simulasi *swing leg* beserta desain *adaptive control* merepresentasikan persamaan (2.64)

dan (2.66). Dimana dalam Simulink persamaan (2.64) dan (2.66) disimulasikan dalam bentuk matriks seperti persamaan (2.64 a) dan (2.66 a).

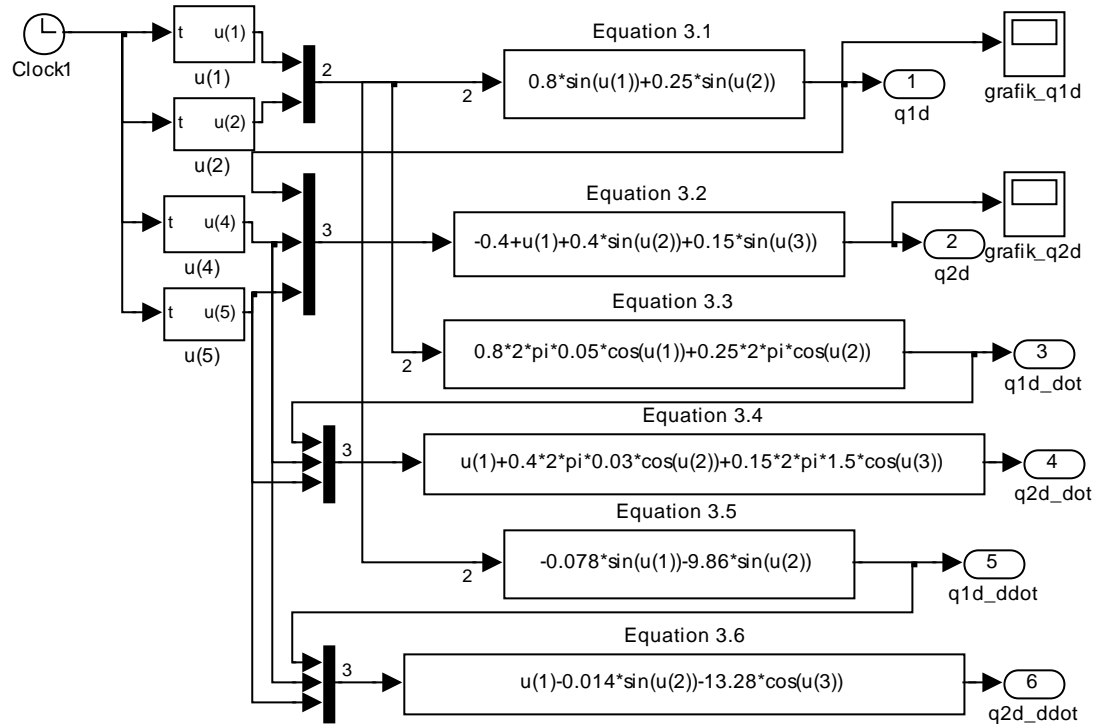


Gambar 3.29 Gambar blok *subsystem* yang mewakili persamaan (2.64 a) dan (2.66 a).

Gambar 3.29 merupakan gambar *subsystem equation* (2.64 a) dan (2.66 a) dimana terdapat delapan buah blok input yang masing-masing adalah input dari q_1 , q_{1d} , q_2 , dan q_{2d} , \dot{q}_1 , \dot{q}_2 , \dot{q}_{1d} dan \dot{q}_{2d} . Terdapat delapan blok output yang masing-masing adalah \tilde{q}_1 , \tilde{q}_2 , \ddot{q}_1 , \ddot{q}_2 , \dot{q}_{1r} , \dot{q}_{2r} , \ddot{q}_{1r} dan \ddot{q}_{2r} . Input dan output tersebut mewakili persamaan (2.64 a) dan (2.66 a). Sedangkan blok *Constant lamda1* dan *lamda2* mewakili nilai matriks Λ . Dimana nilai matriks Λ adalah matriks diagonal dari λ_1 dan λ_2 .

4. *Subsystem Desired Position*

Blok *subsystem desired position* pada gambar 3.25 yang merupakan gambar simulasi *swing leg* beserta desain *adaptive control* secara keseluruhan mewakili persamaan (3.1), (3.2), (3.3), (3.4), (3.5) dan (3.6). Gambar 3.30 berikut merupakan gambar *subsystem desired position*.

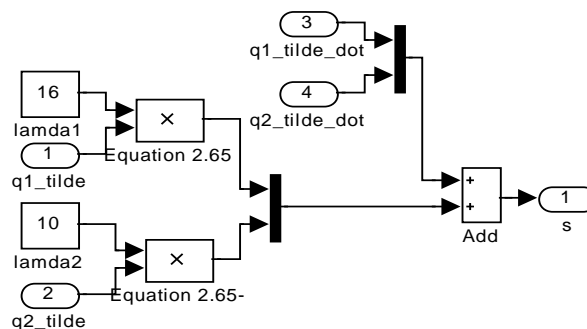


Gambar 3.30 *Subsystem desired position*.

Masing-masing blok *Function* pada gambar 3.30 mewakili persamaan posisi yang diinginkan yaitu persamaan (3.1) dan (3.2) dan turunan dari persamaan tersebut.

5. *Subsystem Error Virtual Velocity*

Blok *subsystem error virtual velocity* pada gambar 3.25 yang merupakan gambar simulasi *swing leg* beserta desain *adaptive control* secara keseluruhan mewakili persamaan (2.65 a). Gambar 3.31 berikut merupakan gambar *subsystem error virtual velocity*.



Gambar 3.31 *Subsystem error virtual velocity* yang mewakili persamaan (2.65 a).

Gambar 3.31 tersebut mewakili persamaan vektor eror kecepatan virtual (s) pada persamaan (2.65 a). Dimana persamaan vector eror kecepatan virtual (s) untuk *swing leg* merupakan vektor eror kecepatan ditambah dengan vektor eror posisi.

6. *Subsystem Adaptation Law*

Blok *subsystem adaptation law* pada gambar 3.25 yang merupakan gambar simulasi *swing leg* beserta desain *adaptive control* secara keseluruhan mewakili persamaan (2.67 a). Persamaan (2.67) merupakan persamaan $\dot{\alpha}$ yang merupakan persamaan hukum adaptasi. Dalam simulasi diperlukan nilai dinamik parameter a_1 , a_2 , a_3 , a_4 dan a_5 yang didapat dari persamaan (2.56) sampai dengan persamaan (2.60). Berikut perhitungan nilai dinamik parameter tersebut.

$$a_1 = J_0 + J_2 - J_3 = 0,2805 \text{ kg.m}^2$$

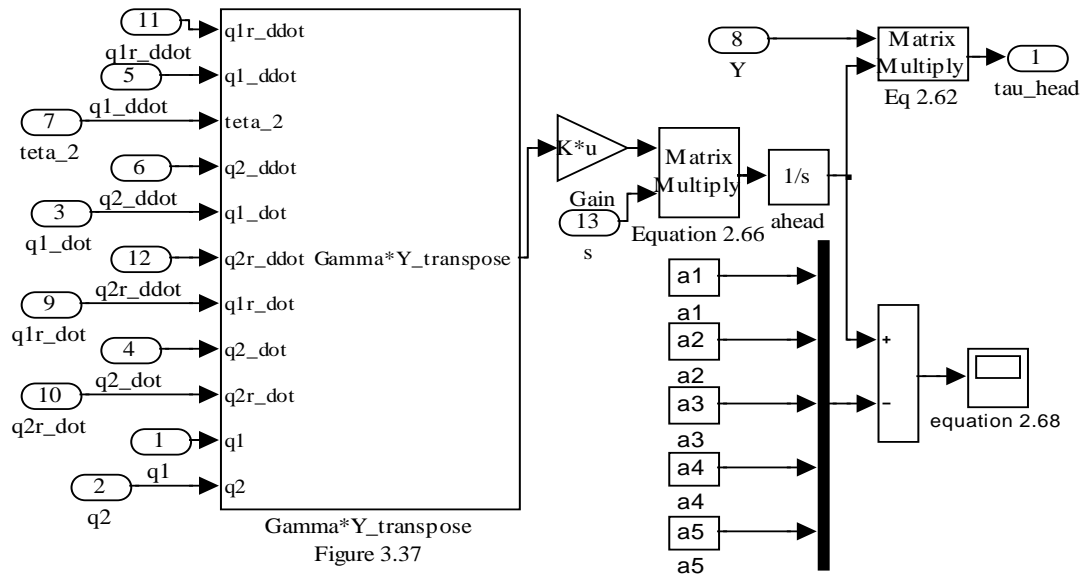
$$a_2 = J_1 = 0,0273 \text{ kg.m}^2$$

$$a_3 = J_3 = 0,0232 \text{ kg.m}^2$$

$$a_4 = G_1 = 2,2249 \text{ kg.m}^3/\text{s}^2$$

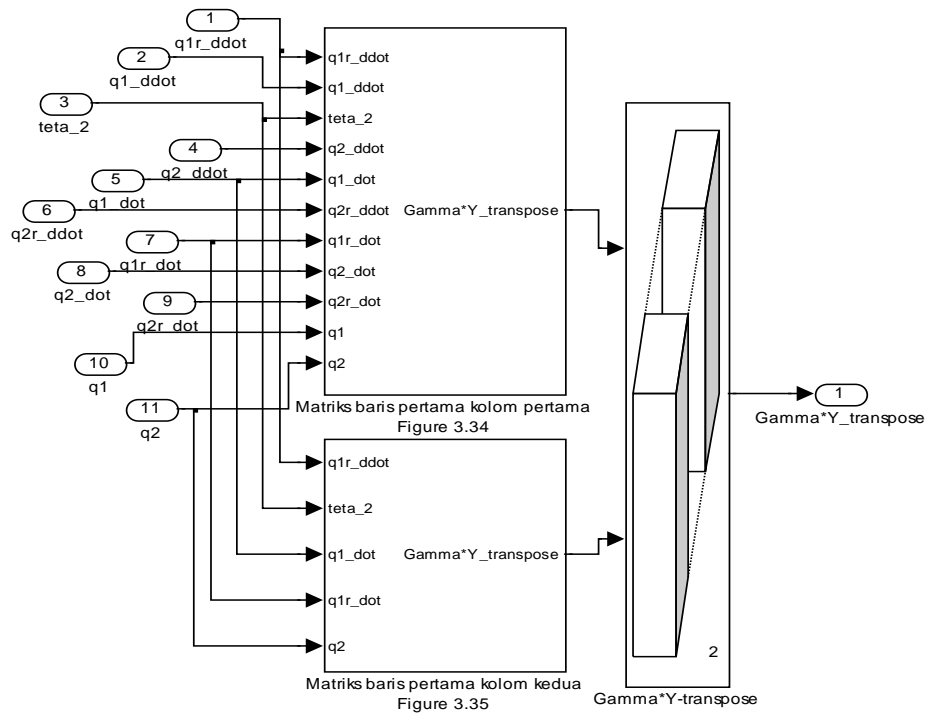
$$a_5 = G_2 = 0,6386 \text{ kg.m}^3/\text{s}^2$$

Gambar 3.32 berikut merupakan gambar *subsystem adaptation law* yang mewakili persamaan (2.67 a). Dimana blok *Constant* a1 sampai dengan a5 mewakili nilai dinamik parameter yang telah dihitung.



Gambar 3.32 *Subsystem adaptation law* yang mewakili persamaan (2.67 a).

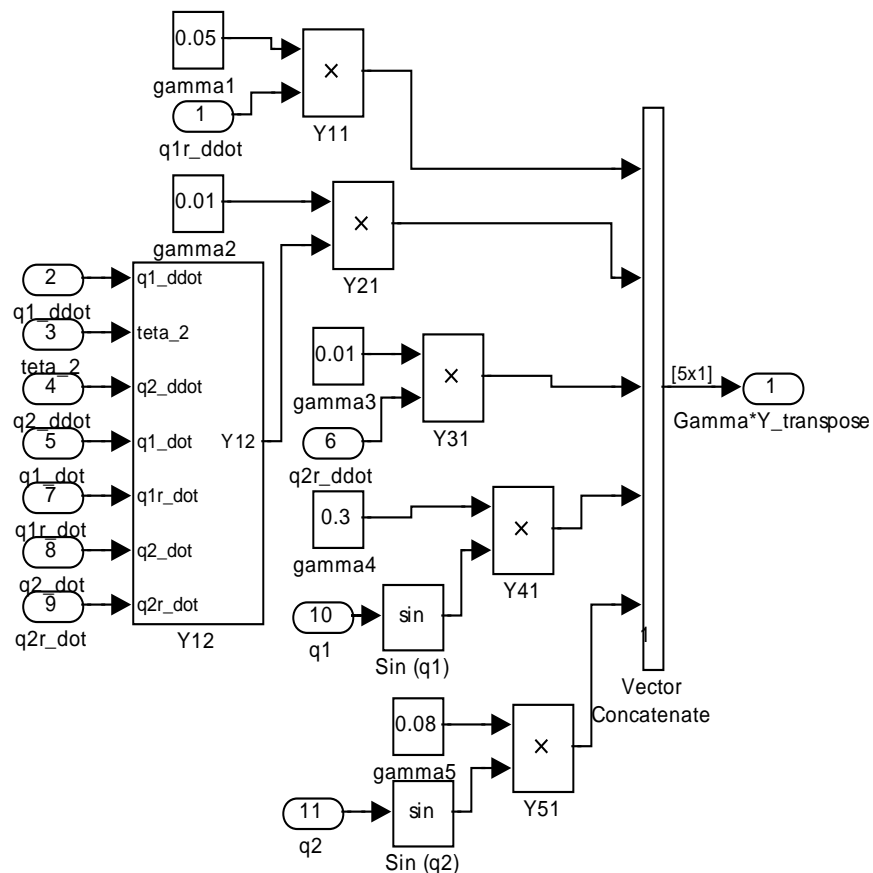
Subsystem $\gamma \cdot Y^T$ pada gambar 3.32 mewakili hasil perkalian dari matriks $\Gamma \cdot Y^T$ pada persamaan (2.67 a). Perkalian Γ dan Y transpose tersebut menghasilkan matriks dengan dimensi 1×2 . Gambar 3.33 berikut merupakan matriks penyusun hasil perkalian $\Gamma \cdot Y^T$.



Gambar 3.33 *Subsystem* $\Gamma \cdot Y^T$.

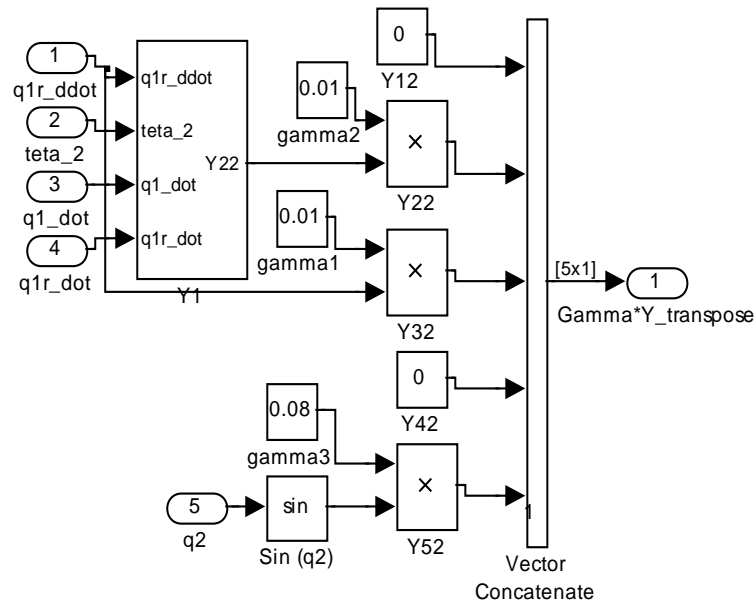
Gambar 3.33 mewakili matriks hasil perkalian $\Gamma.Y^T$ (γ dan Y transpose) dimana gambar 3.34 berikut menunjukkan matriks penyusun yang terletak pada baris pertama dan kolom pertama matriks hasil perkalian $\Gamma.Y^T$.

Nilai Γ didapat dari tabel parameter *adaptive control* untuk *swing leg* pada tabel tabel 3.2 sedangkan matriks Y didapat dari persamaan *regressor* (2.63) [2]. Dari gambar 3.33 dapat dilihat bahwa pada matriks tersebut terdapat sebelas input antara lain q_1 , q_2 , \dot{q}_1 , \dot{q}_2 , \ddot{q}_1 , \ddot{q}_2 , \dot{q}_{1r} , \dot{q}_{2r} , \ddot{q}_{1r} , \ddot{q}_{2r} dan θ_2 .



Gambar 3.34 Matriks penyusun hasil perkalian $\Gamma.Y^T$ yang terletak pada baris pertama dan kolom pertama.

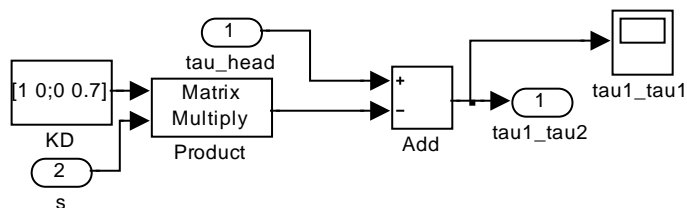
Gambar 3.34 menunjukkan blok *Constant gamma1*, *gamma2*, *gamma3*, *gamma4* dan *gamma5* merupakan konstanta yang mewakili nilai Γ . Nilai Γ pada blok *Constant* tersebut dikalikan dengan hasil *transpose* matriks Y atau yang biasa disebut *regressor* pada persamaan (2.63). Gambar 3.35 berikut merupakan gambar matriks penyusun yang terletak pada baris pertama dan kolom kedua matriks hasil perkalian $\Gamma.Y^T$.



Gambar 3.35 Matriks penyusun hasil perkalian $\Gamma \cdot Y^T$ yang terletak pada baris pertama dan kolom kedua.

7. Subsystem Input Torsion

Blok *subsystem input torsion* pada gambar 3.25 yang merupakan gambar simulasi *swing leg* beserta desain *adaptive control* secara keseluruhan mewakili persamaan (2.61 a). Gambar 3.36 berikut merupakan gambar *subsystem input torsion* :



Gambar 3.36 Subsystem yang mewakili persamaan (2.61 a).

3.4 Visualisasi Swing Leg dengan Virtual Reality

Gerakan *swing leg* dapat divisualisasikan dengan menggunakan *Virtual Reality* (VR). *Virtual Reality* merupakan fitur bawaan yang ada pada *software* MATLAB. *Virtual Reality* merupakan sebuah teknik simulasi yang divisualisasikan dengan grafis tiga dimensi (CAD) interatif. Teknik VR dapat digunakan untuk kepentingan pemodelan mekanika (rotasi dan translasi), kondisi (perubahan suhu, volume,

intensitas). Pemodelan mekanika dengan menggunakan teknik VR, secara khusus membahas respon sebuah konfigurasi mekanik yang diberi input berupa variabel rotasi dan translasi. Berikut ini adalah langkah-langkah untuk membuat visualisasi *swing leg* dengan menggunakan fitur *Virtual Reality*.

1. Pembuatan model tiga dimensi dengan CAD

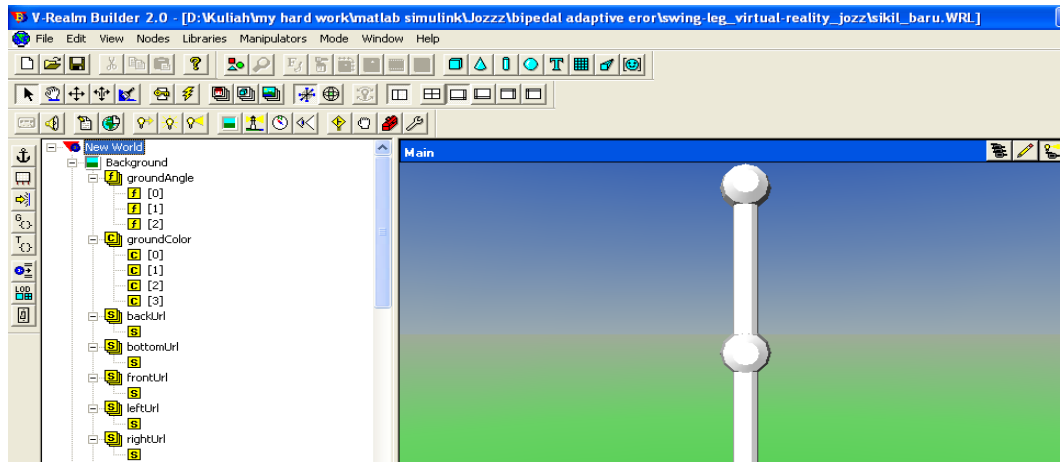
Model tiga dimensi yang digunakan pada VR sink, merupakan model gambar dari CAD tiga dimensi yang kemudian di-export dalam bentuk *file.wrl* (hanya *file.wrl* yang bisa dibaca oleh blok VR Sink) . Tahapan yang dilalui untuk membuat model 3D.wrl :

- a. Membuat model gambar dengan menggunakan program CAD

Model *swing leg* dapat digambar dengan menggunakan beberapa software CAD berikut AutoCAD 2004 (memiliki fitur *eksport to file.3ds*), SolidWorks dan CATIA. Pada model CAD setting titik absolute pada koordinat (0,0,0). Pada penelitian tugas akhir ini, model tiga dimensi dari *swing leg* dibuat dengan menggunakan *software* AutoCAD 2004.

- b. Eksport atau *save as* kedalam bentuk *file.wrl*

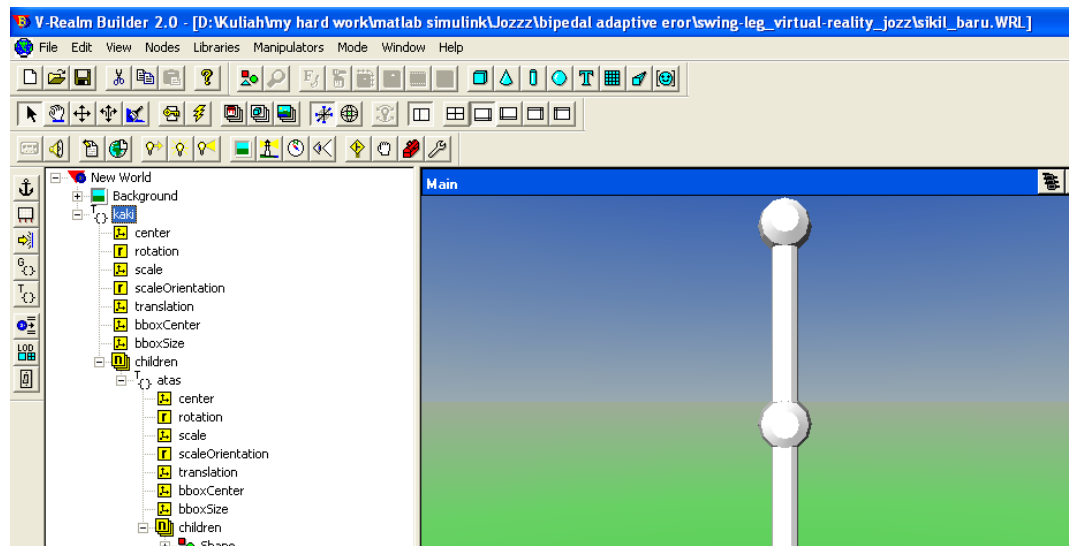
Pada model tiga dimensi yang dibuat dengan menggunakan SolidWorks dan CATIA untuk mengubah file dalam bentuk *file.wrl* klik *save as* kemudian pilih *file.wrl* sebagai *extension* file. Bila model tiga dimensi dibuat dengan menggunakan AutoCAD 2004 untuk mengubah file ke dalam bentuk *file.wrl* klik *File* kemudian klik *Export* lalu pilih *Export data* kemudian pilih *file.3ds* langkah terakhir klik *save*. Untuk mengubah file ke dalam bentuk *file.wrl* buka fitur *VRML Builder* kemudian klik *open* dan buka *file.3ds* yang sudah dibuat sebelumnya. Sehingga tampilan *VRML Builder* terlihat seperti pada gambar 3.37 berikut ini.



Gambar 3.37 Tampilan *file.wrl* dari model *swing leg* pada fitur *VRML Builder*.

c. Konfigurasi *file.wrl*

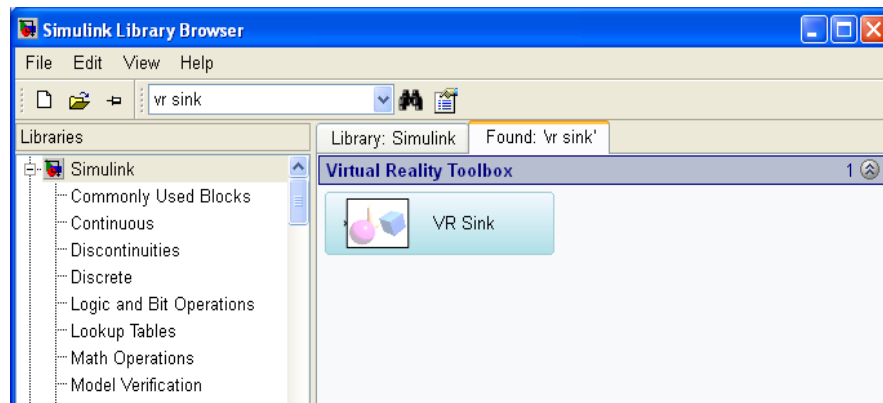
Pada konfigurasi *file.wrl* ini dapat dilakukan penyesuaian kondisi batas yang diinginkan antara lain *rotasi*, *translasi*, skala dan lain-lain. Sehingga tampilan pada fitur *VRML Builder* akan terlihat seperti pada gambar 3.38 berikut.



Gambar 3.38 Tampilan *file.wrl* dari model *swing leg* setelah dilakukan konfigurasi.

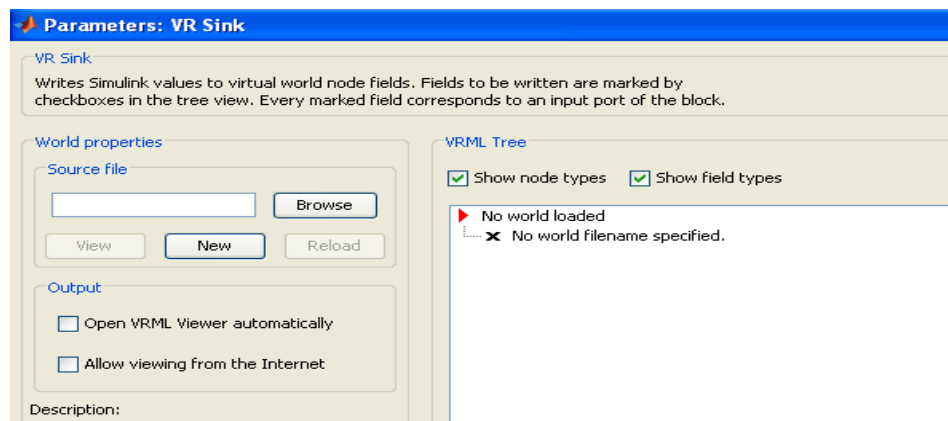
2. Menggabungkan model *file.wrl* dengan model *swing leg* pada Simulink

Langkah pertama untuk menggabungkan model *swing leg* dengan membuka model *plant swing leg* beserta *adaptive control*. Kemudian buka *Library Browser* pada Simulink lalu ketik *VR Sink* pada kolom pencarian. Untuk lebih jelas dapat dilihat pada gambar 3.39 berikut.



Gambar 3.39 Tampilan *Library Browser* pada *Simulink*.

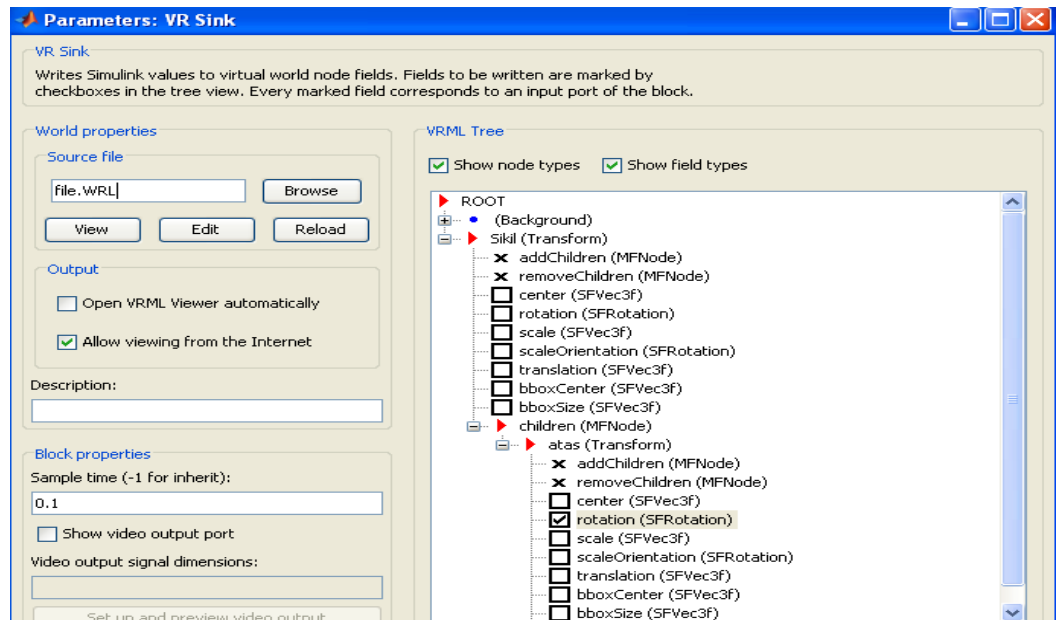
Langkah selanjutnya yaitu *drag* dan *drop* blok *VR Sink* tersebut ke lembar kerja model *plant swing leg* dan *adaptive control* yang telah dibuat sebelumnya. Kemudian *double klik* blok *VR Sink* tersebut. Sehingga akan muncul tampilan seperti pada gambar 3.40 berikut.



Gambar 3.40 Tampilan *VR Sink*.

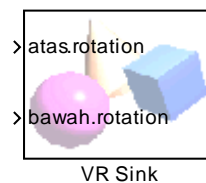
Dari gambar 3.40 pada kolom *Source File* klik *browse* kemudian pilih *file.wrl* yang telah dibuat pada langkah sebelumnya. Kemudian cek *rotation* pada *port rotation*. Langkah ini menunjukkan bahwa kaki mengalami

gerakan rotasi dengan sudut sesuai dengan hasil perhitungan pada model *plant swing leg* dan *adaptive control*. Untuk lebih jelas dapat dilihat pada gambar 3.41 berikut.



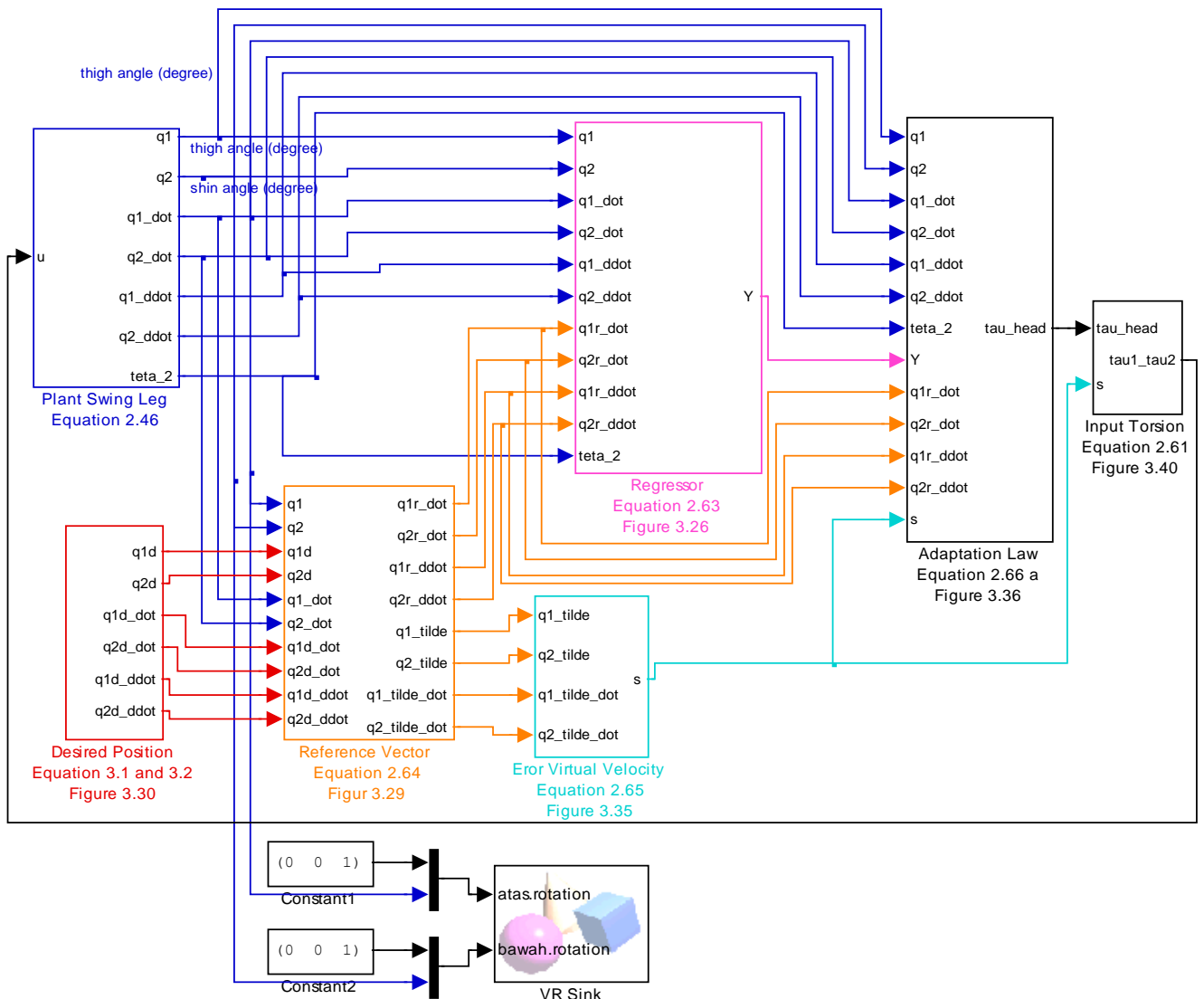
Gambar 3.41 Tampilan *VR Sink* dan *VRML Builder*.

Setelah langkah di atas, maka tampilan *VR Sink* pada model *plant swing leg* akan seperti pada gambar 3.42.



Gambar 3.42 Tampilan blok *VR Sink* pada model Simulink.

Kemudian hubungkan *port atas.rotation* dengan *port q_1* dan *port bawah.rotation* dengan *port q_2* . Tambahkan blok *Constant* kemudian masukkan nilai pada kolom *Constant Value* dengan nilai matriks $[0 \ 0 \ 1]$ yang berarti bahwa kaki bagian atas dan kaki bagian bawah berputar berdasarkan sumbu z. Sehingga tampilan pada model Simulink akan terlihat seperti pada gambar 3.43 berikut.



Gambar 3.43 Model *plant swing leg* dan *adaptive control* beserta simulasi

VR.

Langkah terakhir untuk melihat visualisasi pada VR klik *Start Simulation* dan *double* klik pada blok *VR Sink* maka akan tampak gambar tiga dimensi dari CAD yang telah dibuat sebelumnya berorientasi sesuai dengan sudut q_1 dan q_2 .