

### BAB III

#### METODE PENELITIAN

Pada bab ini akan dijelaskan proses perancangan simulasi *Self Balancing Robot* menggunakan Matlab 2013a menggunakan metode *Linear Quadratic Regulator*

#### 3.1 ANALISA

Untuk merancang simulasi robot pendulum roda dua atau *Self Balancing Robot* diperlukan beberapa langkah, antara lain:

1. Mendapatkan nilai state pada *plant* robot pendulum.
2. Menentukan bobot matrix Q dan R, penentuan bobot matrix melalui coba coba atau *trial and error*, yang pemilihanya berdasarkan nilai sisa relatif paling sedikit
3. Menentukan indeks performansi kuadratik berikut

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + u^T \hat{\mathbf{R}} u) dt \dots\dots\dots(3.1)$$

4. Menentukan matriks yang merupakan solusi dari persamaan aljabar Riccati berikut

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \hat{\mathbf{R}}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0 \dots\dots\dots(3.2)$$

5. Menghitung nilai *gain feedback* untuk mendapatkan matriks kontrol atas sistem

$$\mathbf{u} = -\hat{\mathbf{R}}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{x} = -\mathbf{K} \mathbf{x} \dots\dots\dots(3.3)$$

#### 3.2 PERANCANGAN

##### 3.2.1 PERANCANGAN MENGGUNAKAN LQR

Suatu sistem dikatakan baik bila sistem tersebut mempunyai unjuk kerja terbaik (*best performance*) terhadap suatu acuan tertentu. Sistem kontrol mempunyai tolak ukur, artinya sistem tersebut harus sesuai terhadap suatu refrensi. Dinamika sistem yang akan dikontrol digambarkan oleh sekumpulan persamaan *state* yang dapat dituliskan dalam bentuk :

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} u(t) \\ y(t) &= \mathbf{C} \mathbf{x}(t) \dots\dots\dots(3.4) \end{aligned}$$

Permasalahan umum dalam kontrol adalah :

- Pada sistem (*plant*) yang akan dikendalikan, dapatkan fungsi kontrol  $u^*(t)$  yang akan bekerja pada *plant*, yang mana hal itu merupakan aksi kontrol terbaik. Terbaik dalam artian selain memenuhi criteria desain tertentu juga memenuhi kendala-kendala yang ada.



- Bagaimana merealisasikan fungsi kontrol yang didapat dari poin di atas ke dalam suatu kontroler.

Pengendalian deterministik membutuhkan variabel keadaan secara lengkap untuk membangkitkan sinyal kontrol dengan jalan meminimumkan suatu fungsi yang disebut *cost function*. *Cost function* ini mewakili indeks performansi yang harus dicari solusinya untuk mendapatkan hasil yang baik (Ogata, 1984:401). Untuk sistem dengan indeks performansi kuadratik dalam bentuk :

$$J = \int_0^{\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \dots\dots\dots(3.5)$$

Dengan tujuan untuk membawa state sistem dari state awal  $x(t) = x^0$  ke state yang diinginkan  $x^1$  dengan meminimumkan  $J$ , maka permasalahan sistem demikian disebut permasalahan *Linier Quadratic Regulator*. Secara umum permasalahan LQR adalah :

Untuk suatu sistem linier *,time invariant*,

$$\dot{x}(t) = Ax(t) + Bu(t) \dots\dots\dots(3.6)$$

Dimana A dan B adalah matriks bernilai konstan berdimensi  $n \times n$  dan  $n \times m$  berturut-turut, dengan indeks performansi :

$$J = \int_0^{\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \dots\dots\dots(3.7)$$

Dimana Q dan R adalah matriks semidefinit positif dan definit positif yang bernilai konstan berdimensi  $n \times n$  dan  $n \times m$ . Dapatkan sinyal kontrol optimal  $u$  berumpan balik :

$$u = -Kx(t) \dots\dots\dots(3.8)$$

Dengan K adalah :

$$K = R^{-1}B^T P \dots\dots\dots(3.9)$$

Dimana P adalah suatu matriks definit positif yang merupakan solusi dari persamaan :

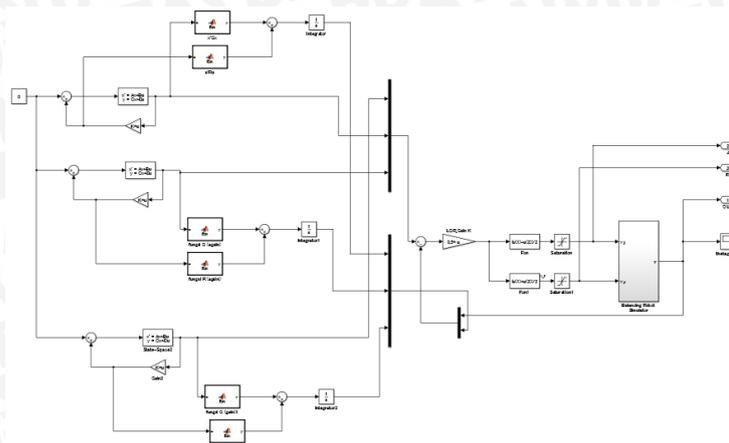
$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \dots\dots\dots(3.10)$$

Persamaan tersebut dikenal dengan sebagai persamaan *Algebraic Riccati Equation (ARE)*. Solusi persamaan di atas hanya dapat diselesaikan bila pasangan matriks (A,B) adalah terkontrol dan pasangan matriks (A,C) adalah teramati.

### 3.2.2 PERANCANGAN MODEL SIMULASI MATLAB

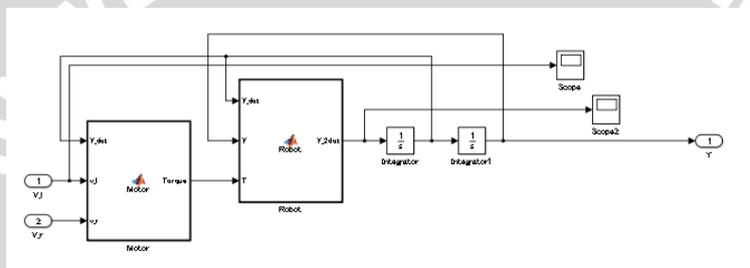
Sebagaimana telah dijelaskan pada bab sebelumnya tujuan dari desain kontrol LQR adalah menentukan matriks kontrol yang mampu menstabilkan robot self balancing. Berikut model perancangan matlab untuk kontrol LQR.





Gambar 3.1 Model Plant

Dari Gambar (3.1) terdapat blok plant balancing robot yang dapat dilihat pada gambar (3.2)



Gambar 3.2 Model balancing robot

Dapat dilihat dari Gambar (3.2) terdapat blok motor dan blok robot yang merupakan bagian dari perancang sistem *self balancing robot*

