Kendali Proposional-Integral Pada Pelacakan Formasi Berdasarkan Jarak

Anggoro Dwi Nur Rohman
anggoro_dwi@student.ub.ac.id

Universitas Brawijaya— September 30, 2020

Pendahuluan

Rangkuman ini ditujukan untuk memahami dan review jurnal. Judul dalam bahasa inggris *A Proportional-Integral Controller for Distance-Based Formation Tacking* yang diteliti oleh Oshri Rozenheck dari Israel Institute of Technology, Haifa, Israel. Peneliti menerangkan pada jurnal ini tentang permasalahan kendali formasi pada multi-agent apabila pada salah satu agent nya diberikan kecepatan tambahan sebagai refrensi.

1 Notasi-Notasi

Notasi	Keterangan
$n \triangleq \|V\|$	...
$m \triangleq \|\varepsilon\|$	..
$A_1, \ldots, A_n \in \mathbb{R}^{p\times q}$	...
$x = \begin{bmatrix} x_1^T \\ \ldots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{2n}$ $x_i \in \mathbb{R}^n$ $x_i \neq x_j; \forall i \neq j$	Konfigurasi titik
$e_k \triangleq x_j - x_i$	Relative position vector
$e = \begin{bmatrix} e_1^T \\ \vdots \\ e_m^T \end{bmatrix} \in \mathbb{R}^{2m}$ $E \in R^{n\times m}$	Edge Vector Incidence matrix dimana isinya adalah $\{0, \pm 1\}$ barisnya menandakan vertices dan kolom nya menandakan edge
$A_1, \ldots, A_i \in \mathbb{R}^{p\times q}$ $diag(A_i) \triangleq blkdiag\{A_1, \ldots, A_n\} \in \mathbb{R}^{np\times nq}$	 https://www.mathworks.com/help/matlab/ref/blkdiag.html

2 Edge Function

Diketahui framework $G(x)$ dengan vector edge $e_{k=1}^m$, $F : \mathbb{R}^{2n} \times G \to \mathbb{R}^m$, maka dapat didefinisi

$$F(x, G) \triangleq= \begin{bmatrix} |e_1||^2 \\ \vdots \\ ||e_m||^2 \end{bmatrix} \tag{1}$$

Peneliti mendefinisi *rigidity matrix R(x)* sebagai fungsi jacobian dari Edge function,

$$R(x) = \frac{\partial F(x, G)}{\partial x} \in \mathbb{R}^{m\times 2n}$$

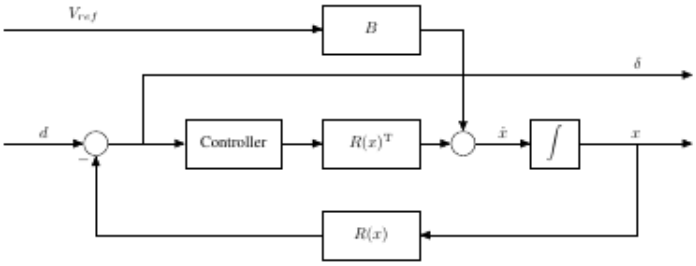


Fig. 2: The formation control is augmented with an additional controller to eliminate the steady-state error in the formation.

Gambar. 1: Skema General

dan menyingkat perhitungannya dan menghasilkan persamaan

$$R(x) = diag(e_i^T)(E^T \otimes I_2) \tag{2}$$

3 Kendali

3.1 Orde Satu

Error jarak dinotasikan dengan $\delta \in \mathbb{R}^m$ dimana

$$\delta_k = ||e_k||^2 - d_k^2, k \in \{1, \ldots, m\} \tag{3}$$

Peneliti menggunakan fungsi potensial

$$\Phi(e) = \frac{1}{2}\sum_{k-1}^{m}\Big(||e_k||^2 - d_k^2\Big)^2 = \frac{1}{2}\sum_{k=1}^{m}\delta_k^2 \tag{4}$$

digunakan untuk kendali pada setiap robot dengan cara negative gradient dari fungsi potensial.

$$u_i(t) = -\Big(\frac{\partial\Phi(e)}{\partial x_i}\Big) = -\sum_{j\ o}\Big(||e_k||^2 - d_k^2\Big)e_k \tag{5}$$

Apabila ditulis ulang dalam bentuk state-space

$$\dot{x}(t) = -R(x)^T R(x)x(t) + R(x)^T d \tag{6}$$

Menggunakan skema seperti pada gambar.1 maka dapat diperoleh

$$\dot{x} = u(t) + B.v_{ref} \tag{7}$$

$$u(t) = -R(x)^T.C.\Big(R(x).x(t) - d\Big) \tag{8}$$

dan cikal bakal C sebagai controller.

dengan menerapkan Proportional-integrator pada persamaan (8) maka diperoleh

$$u(t) = -R(x)^T k_p(R(x).x(T) - d) - R(x)^T.K_i\int_0^T (R(x).x(\tau) - d)d\tau \tag{9}$$

dimana kp_p dan k_i adalah konstanta.

Pada bagian integrator menghasilkan state baru

$$\dot{\xi} = k_i(R(x).x(t) - d) \tag{10}$$

lalu kombinasi persamaan tersebut diperoleh state- space close loop

$$\begin{bmatrix}\dot{x}(t)\\ \dot{\xi}(t)\end{bmatrix} = \begin{bmatrix}-k_p.R(x)^T.R(x) & -R(x)^T\\ k_i.R(x) & 0\end{bmatrix}\begin{bmatrix}x(t)\\ \xi(t)\end{bmatrix} + \begin{bmatrix}k_p.R(x)^T\\ -k_i.I\end{bmatrix}.d + \begin{bmatrix}B\\ 0\end{bmatrix}.v_{ref} \tag{11}$$

3.2 Orde Dua

Definisi persamaan orde dua

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases} \tag{12}$$

Dengan fungsi potensial

$$\Phi(e) = \frac{1}{2}\sum_{i \in V} ||x_{2i}||^2 + \frac{1}{2}\sum_{k=1}^{m} \delta_k^2 \tag{13}$$

Maka persamaan state space nya

$$\begin{cases} \dot{x}_1 & = \nabla_{x_2}\Phi = x_2 \\ \dot{x}_2 & = c(-\nabla_{x_2}\phi - \nabla_{x_1}\phi) \\ & = c(-x_2(t) - R(x_1)^T(R(x_1)x_1(t) - d)) \end{cases} \tag{14}$$

Dengan menerapkan kendali PI

$$\dot{x}_2 = C(-x_2(t) - R(x_1)^T(R(x_1)x_1(t) - d)) \tag{15}$$

$$= -K_{p1}x_2(t) - R(x_1)^T K_{p2}(R(x_1)x_1(t) - d)) \tag{16}$$

$$- K_{i1}\int_0^\tau x_2(\tau)d\tau - R(x_1)^T K_{i2}\int_0^\tau (R(x_1)x_1(\tau) - d))d\tau \tag{17}$$

$$\tag{18}$$

Menghasilkan state baru dan Ditulis ulang

$$\dot{x}_1 = x_2(t) \tag{19}$$

$$\dot{x}_2 = -k_{p1}x_2(t) - R(x_1)^T k_{p2}(R(x_1)x_1(t) - d)) - \xi_1 - R(x_1)^T\xi_2 \tag{20}$$

$$\dot{\xi}_1 = k_{i1}x_2(t) \tag{21}$$

$$\dot{\xi}_2 = k_{i2}(R(x_1)x_1(t) - d) \tag{22}$$

Dalam bentuk matrix

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_{p2}R(x_1)^T R(x_1) & -k_{p1} & -1 & -R(x_1)^T \\ 0 & k_{i1} & 0 & 0 \\ k_{i2}R(x_1) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ k_{p2}R(x_1)^T \\ 0 \\ -k_{i2} \end{bmatrix} d \tag{23}$$

3.3 Implementasi

Pada seksi implementasi ini akan dibahas mengenai code yang digunakan untuk menyimulasi persamaan yang diciptakan oleh peneliti. Dapat diperhatikan hasil simulasi yang dibuat ulang oleh penulis pada gambar(2). Pada gambar tersebut $vref$ diberikan pada robot 1 dengan kecepatan vector $[-5;\ 5]$. Akan tetapi ada sebuah error apabila $vref$ dengan kecepatan vector $[5;\ 5]$. Ini diakibatkan ada masalah pada rigiditas graph.

3.3.1 time varian bentuk diskrit

Sederhana dari persamaan-()23)

$$\dot{x} = A(x)x + B(x)d \tag{24}$$

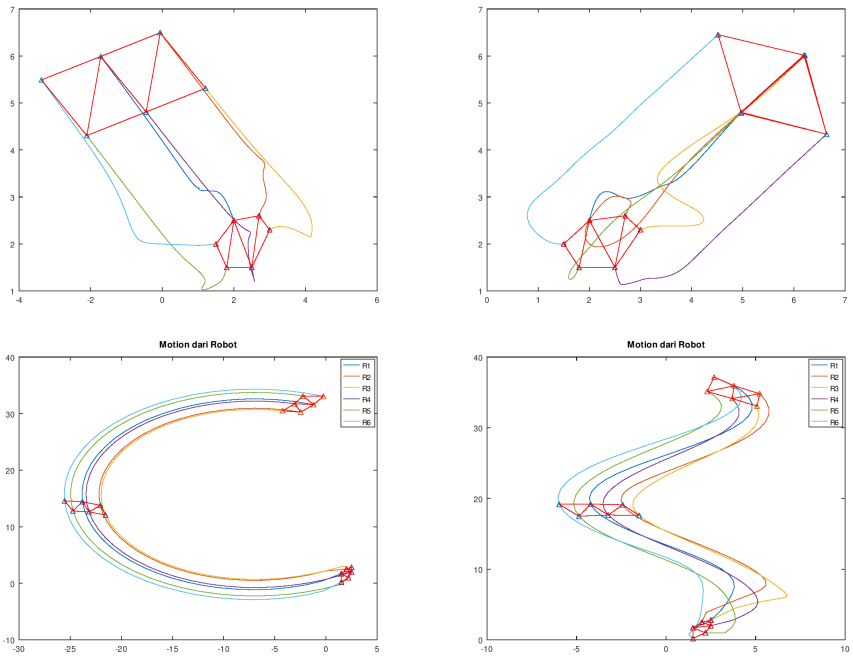
$$\frac{x[k+1] - x[k]}{h} = A(x[k])x[k] + B(x[k])d \tag{25}$$

$$x[k+1] = I + hA(x[k])x[k] + hB(x[k])d \tag{26}$$

$$x]k+1] = A_d[k]x[k] + B_d[k]d \tag{27}$$

$$A_d[k] = I + hA(x[k]) \tag{28}$$

$$B_d[k] = hB(x[k]) \tag{29}$$



Gambar. 2: Hasil Membuat Ulang Simulasi

4 Kesimpulan

Implementasi berhasil. Penelitian mengalami kekurangan apabila vref diberikan pada robot yang berpotensi menghasilkan lipatan pada graph.