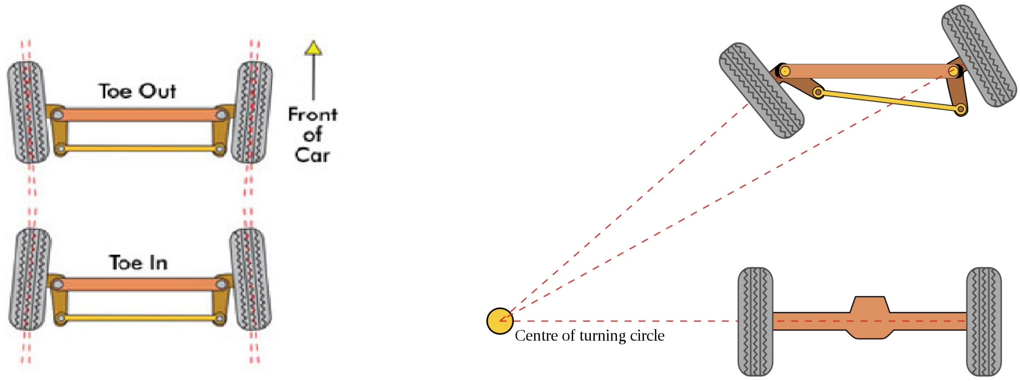
فصل ۳

مدل‌سازی سیستم فرمان و طراحی کنترل‌کننده

۱.۳ مقدمه

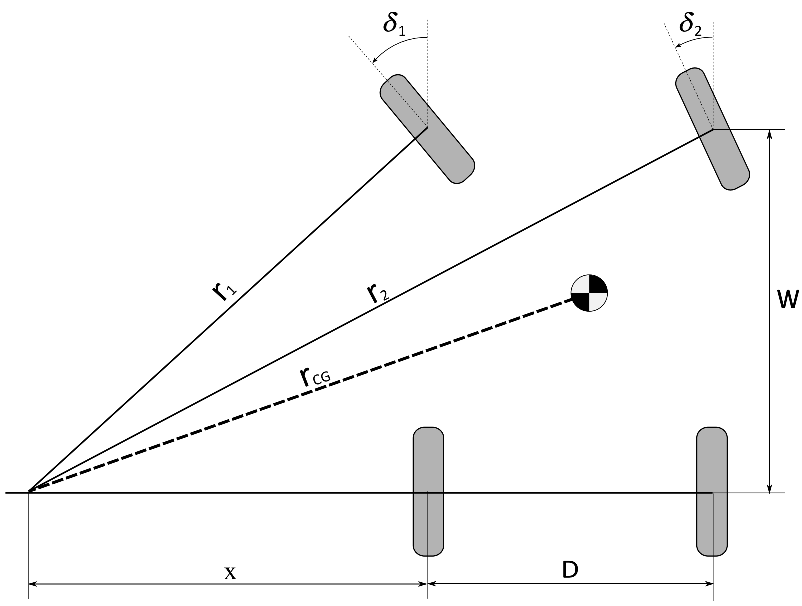
این فصل به شرح تحقیق انجام شده می‌پردازد و از دو بخش تشکیل شده است. بخش اول به نحوه طراحی سیستم فرمان و استخراج معادلات دینامیکی حاکم بر آن می‌پردازد و در بخش دوم پس از ساده سازی معادلات دینامیکی و روش طراحی کنترل‌کننده مدلغزشی تحمل‌پذیر عیب برای سیستم هدایت-با-سیم دارای تأخیر بیان می‌شود. کنترل‌کننده ارائه شده شامل یک مشاهده‌کننده اغتشاش و یک کنترل‌کننده زمان گسسته است. به منظور افزایش سرعت و کارایی، مشاهده‌کننده اغتشاش بخشی از مجموعه عملگر جعبه فرمان قرار داده شده است که بصورت بی‌درنگ فعالیت می‌کند. از آنجایی که مشاهده‌کننده برای تخمین اغتشاش فقط به نیروی ورودی و متغیرهای حالت سیستم نیاز دارد، قراردادن آن نزدیک عملگر و حسگر جعبه فرمان عملی و منطقی است. در این صورت مشاهده‌کننده اغتشاش فارغ از محدودیت‌های شبکه خواهدبود، و همین امر باعث می‌شود بیدرنگ بودن فعالیت آن امکان پذیر شود. از طرفی ارتباط کنترل‌کننده با جعبه فرمان به ناچار فقط از طریق شبکه امکان پذیر است به همین خاطر در این فصل برای عمل ردیابی مرجع کنترل کننده ای طراحی می‌شود که مقاوم به محدودیت‌های شبکه باشد.

۲.۳ طراحی مکانیزم سیستم فرمان



شکل ۱.۳: مرکز دوران در سیستم فرمان اکرمن، تعریف زاویه toe

بر اساس اصل اکرمن [1] در حالت ایده‌آل لازم است امتداد خط عمود بر بردار سرعت خطی هر چهار چرخ، مشابه شکل ۱.۳ در یک نقطه تلاقی کند. به این ترتیب نقطه به دست آمده مرکز دوران همه چرخ‌ها و درنتیجه مرکز دوران کل خودرو خواهد بود. [۴۵]



شکل ۲.۳: روابط هندسی حاکم بر دوران خودرو

[1] Ackermann

با توجه به شکل ۲.۳ برای $\delta > 0$ داریم:

$$\tan\left(\frac{\pi}{2} - \delta_1\right) = \frac{x}{W} \tag{۱.۳آ}$$

$$\tan\left(\frac{\pi}{2} - \delta_2\right) = \frac{x + D}{W} \tag{۱.۳ب}$$

درنتیجه می‌توان رابطه زیر را بین زاویه فرمان چرخ‌ها بدست آورد:

$$\delta_2 = \cot^{-1}\left(\frac{D}{W} + \cot(\delta_1)\right) \tag{۲.۳}$$

معادله (۲.۳) رابطه زاویه فرمان چرخ داخل خارج پیچ را نسبت به چرخ درون پیچ در حالت ایده‌آل توصیف می‌کند. هدف طراحی سیستم فرمانی است که بیشترین تطابق را با این رابطه داشته باشد.

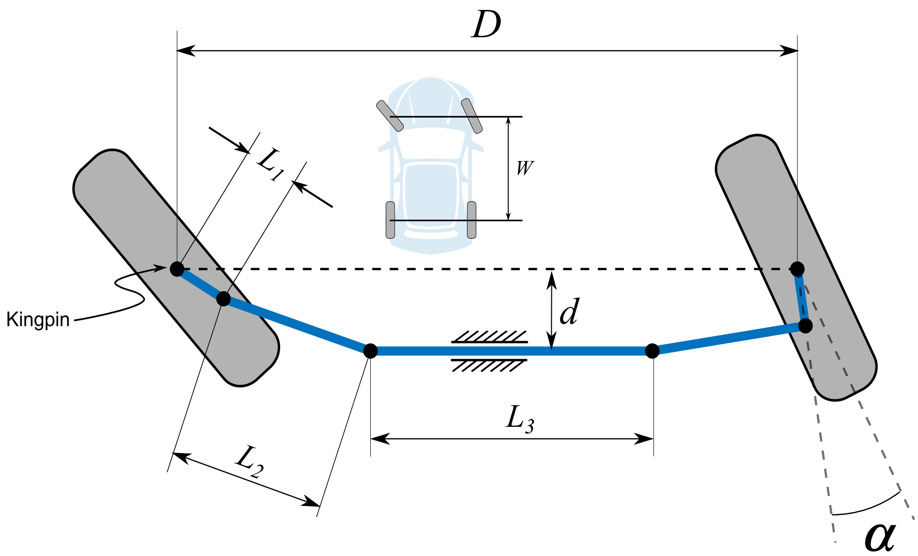
یکی از روش‌های رایج برای طراحی سیستم فرمان اکرمن استفاده از رک و پینیون، و اتصال رک به بازوی فرمان [1] توسط تای‌راد [2] است. برای ارضای رابطه (۲.۳) توسط مکانیزم رک و پینیون بایستی ابعاد بازوها و زاویه α که در شکل ۳.۳ مشخص شده‌اند را بطور بهینه طراحی کرد. به این منظور با استفاده از برنامه نوشته شده به زبان پایتون (پیوست آ.۱) پارامترهای سیستم فرمان شامل L_2 ، L_3 ، d و α متناسب با L_1 ، D و W بطور بهینه بدست می‌آید. در برنامه ی مذکور پس از وارد کردن مقادیر L_1 ، D و W نموداری مشابه شکل ۴.۳ نمایش داده می‌شود؛ سپس، در محیط گرافیکی موجود می‌توان پارامترهای L_2 ، d و α را با سعی و خطا طوری تنظیم کرد که نمودار مربوط به رک و پینیون به نمودار ایده‌آل منطبق شود. همچنین، با داشتن سایر پارامترها و صفر در نظر گرفتن زاویه toe که در شکل ۱.۳ معرفی شد ، L_3 از رابطه زیر بدست می‌آید:

$$L_3 = D - 2\left(L_1 \sin(\alpha) + L_2 \cos(\phi_0)\right) \tag{۳.۳}$$

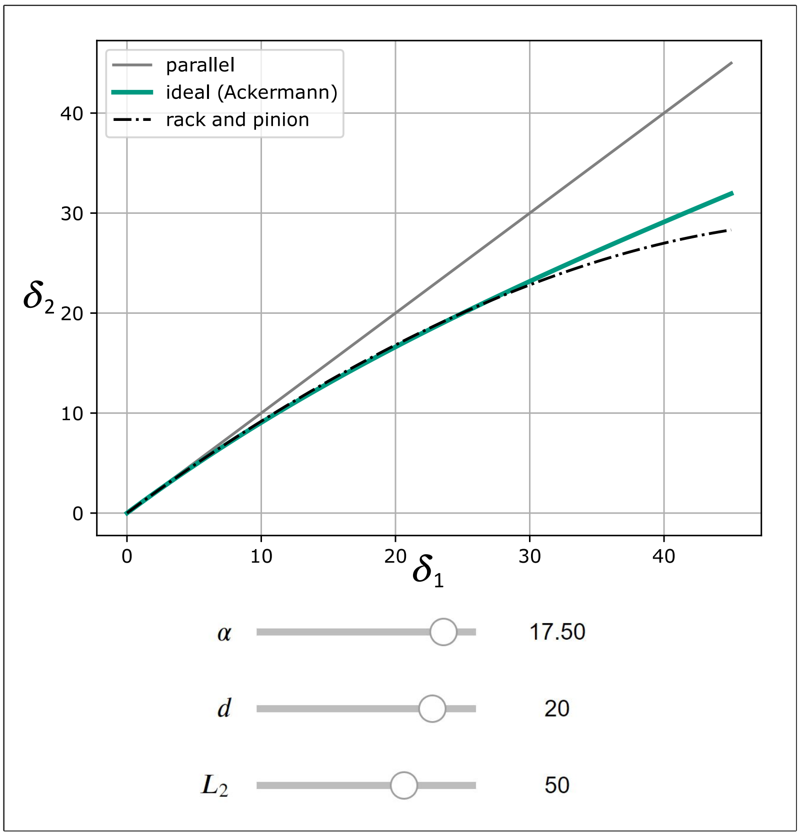
که ϕ_0 برابر است با:

$$\phi_0 = \sin^{-1}\left(\frac{d - L_1 \cos(\alpha)}{L_2}\right) \tag{۴.۳}$$

[1] steering arm
[2] Tie-Rod



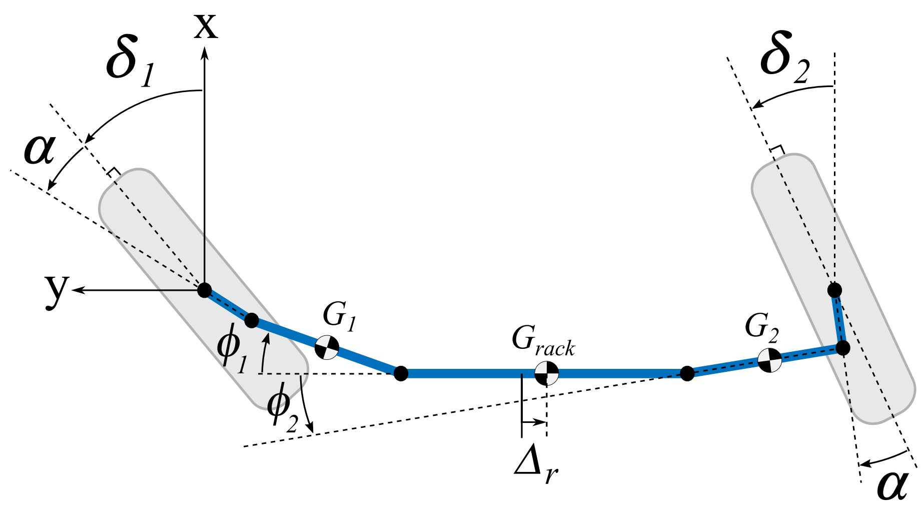
شکل ۳.۳: پارامترهای مربوط به طراحی هندسه سیستم فرمان با رک و پینیون



شکل ۴.۳: محیط گرافیکی تنظیم پارامترهای هندسی سیستم فرمان

۳.۳ مدل سازی سیستم فرمان

در این بخش با استفاده از روش لاگرانژ معادلات سیستم فرمان در حالت کلی محاسبه می‌شود. پارامترهای مربوط به جرم، ابعاد و زوایا استفاده شده در این بخش در شکل‌های ۳.۳ و ۵.۳ و در جدول ۱.۳ معرفی شده‌اند.



شکل ۵.۳: معرفی زوایا و محور مختصات

با انتخاب $q_1 = \delta_1$، $q_2 = \phi_1$، $q_3 = \delta_2$ و $q_4 = \phi_2$ به عنوان مختصات تعمیم یافته، درحالتی که انرژی پتانسیل سیستم ثابت است از معادله لاگرانژ داریم: [۲۳]

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} = R_j + Q_j, \quad j = 1, 2, 3, 4 \tag{۵.۳}$$

که T انرژی جنبشی کل سیستم، R_j مربوط به قیود سیستم و ضرایب لاگرانژ، و Q_j اثر کار نیروهای خارجی بر مختصات j ام است.

۱.۳.۳ معادلات قید و ضرایب لاگرانژ

این بخش به محاسبه R_j از معادله (۵.۳) می‌پردازد. از آنجایی که سیستم یک درجه آزادی دارد و ما چهار مختصات تعمیم یافته انتخاب کردیم، باید سه معادله قید و سه ضریب لاگرانژ λ_1 ، λ_2 و λ_3 موجود باشد. با

توجه به شکل ۵.۳ و ۳.۳ معادلات قید برابر است با:

$$\lambda_1\left(L_1\cos(\alpha+\delta_1)+L_2\sin(\phi_1)-d\right)=0 \tag{۶.۳آ}$$

$$\lambda_2\left(L_1\cos(\alpha-\delta_2)+L_2\sin(\phi_2)-d\right)=0 \tag{۶.۳ب}$$

$$\lambda_3\left(L_1\sin(\alpha+\delta_1)+L_2\cos(\phi_1)+L_1\sin(\alpha-\delta_2)+L_2\cos(\phi_2)+L_3-D\right)=0 \tag{۶.۳ج}$$

اگر معادلات (۶.۳) را باهم جمع کنیم و از مجموع آن‌ها نسبت به زمان مشتق بگیریم خواهیم داشت:

$$\begin{aligned}&\lambda_1\left(-L_1\dot{\delta}_1\sin\left(\alpha+\delta_1\right)+L_2\dot{\phi}_1\cos\left(\phi_1\right)\right)+\lambda_2\left(L_1\dot{\delta}_2\sin\left(\alpha-\delta_2\right)+L_2\dot{\phi}_2\cos\left(\phi_2\right)\right)\\&+\lambda_3\left(L_1\dot{\delta}_1\cos\left(\alpha+\delta_1\right)-L_1\dot{\delta}_2\cos\left(\alpha-\delta_2\right)-L_2\dot{\phi}_1\sin\left(\phi_1\right)-L_2\dot{\phi}_2\sin\left(\phi_2\right)\right)=0\end{aligned} \tag{۷.۳}$$

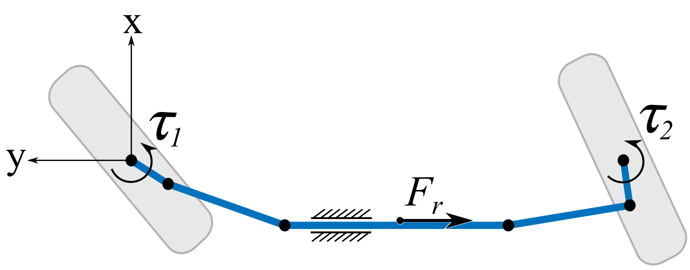
در معادله (۷.۳) ضریب $\dot{q}_j$ مقدار R_j را بدست می‌دهد. بنابراین:

$$\begin{bmatrix}R_1\\R_2\\R_3\\R_4\end{bmatrix}=\begin{bmatrix}-L_1\sin\left(\alpha+\delta_1\right) & 0 & L_1\cos\left(\alpha+\delta_1\right)\\ L_2\cos\left(\phi_1\right) & 0 & -L_2\sin\left(\phi_1\right)\\ 0 & L_1\sin\left(\alpha-\delta_2\right) & -L_1\cos\left(\alpha-\delta_2\right)\\ 0 & L_2\cos\left(\phi_2\right) & -L_2\sin\left(\phi_2\right)\end{bmatrix}\begin{bmatrix}\lambda_1\\\lambda_2\\\lambda_3\end{bmatrix} \tag{۸.۳}$$

۲.۳.۳ محاسبه اثر نیروهای خارجی از روش کار مجازی

این بخش به محاسبه Q_j از معادله (۵.۳) می‌پردازد. برای اعمال نیروی F_r و گشتاورهای τ_1 و τ_2 از روش کار مجازی داریم:

$$\delta W = F \cdot \delta r \qquad \text{یا} \qquad \delta W = M \cdot \delta\theta \tag{۹.۳}$$



شکل ۶.۳: معرفی نیروهای خارجی

باتوجه به شکل ۶.۳ و روابط کار مجازی (۹.۳) می‌توان نوشت:

$$\delta W_1 = \tau_1 \delta\delta_1 \tag{۱۰.۳آ}$$

$$\delta W_2 = \tau_2 \delta\delta_2 \tag{۱۰.۳ب}$$

$$\delta W_3 = \vec{F_r} \cdot \delta r_{rack} = F_r \Big(L_1 \delta\delta_1 \cos\left(\alpha + \delta_1\right) - L_2 \delta\phi_1 \sin\left(\phi_1\right) \Big) \tag{۱۰.۳ج}$$

از مقایسه معادلات (۱۰.۳) با $\delta W = Q_j \delta q_j$ مقادیر Q_j بدست می‌آید:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L_1 \cos(\alpha + \delta_1) \\ 0 & 0 & -L_2 \sin(\phi_1) \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_r \end{bmatrix} \tag{۱۱.۳}$$

۳.۳.۳ محاسبه سمت چپ معادله لاگرانژ

این بخش پس از بدست آوردن انرژی جنبشی سیستم، به محاسبه جمله‌های سمت چپ تساوی در معادله (۵.۳) می‌پردازد.

جدول ۱.۳: پارامترهای مربوط به جرم و نیروهای خارجی سیستم فرمان

پارامتر	توضیح
m_2	جرم بازوی تای‌راد
m_r	جرم رک
I_1	لختی دورانی مجموعه‌ی چرخ، هاب و بازوی فرمان حول کینگ‌پین
I_2	لختی دورانی بازوی اتصال حول مرکزجرم
F_r	نیروی وارد شده از طرف پینیون به رک (درجهت $-\vec{j}$)
τ_1، τ_2	گشتاور اغتشاش وارد شده به چرخ‌ها حول کینگ‌پین

انرژی جنبشی سیستم برحسب مختصات تعمیم یافته $q_1 = \delta_1$، $q_2 = \phi_1$، $q_3 = \delta_2$ و $q_4 = \phi_2$ برابر است با:

$$T = \frac{1}{2}\Big(I_1\dot{\delta}_1^2 + I_2\dot{\phi}_1^2 + m_2(V_{G_1}\cdot V_{G_1}) + m_r(V_{rack}\cdot V_{rack}) + I_1\dot{\delta}_2^2 + I_2\dot{\phi}_2^2 + m_2(V_{G_2}\cdot V_{G_2})\Big) \tag{۱۲.۳}$$

مطابق شکل ۵.۳ بردار موقعیت مرکز جرم G_1 ، G_2 و رک بصورت زیر است:

$$r_{G_1} = \begin{bmatrix} -L_1 \cos(\alpha+\delta_1) - \frac{1}{2}L_2 \sin(\phi_1) \\ -L_1 \sin(\alpha+\delta_1) - \frac{1}{2}L_2 \cos(\phi_1) \end{bmatrix} \tag{۱۳.۳}$$

$$r_{G_2} = \begin{bmatrix} -L_1 \cos(\alpha-\delta_2) - \frac{1}{2}L_2 \sin(\phi_2) \\ -D - L_1 \sin(\alpha-\delta_2) + \frac{1}{2}L_2 \cos(\phi_2) \end{bmatrix} \tag{۱۴.۳}$$

$$r_{G_{rack}} = \begin{bmatrix} -d \\ -L_1 \sin(\alpha+\delta_1) - L_2 \cos(\phi_1) - \frac{1}{2}L_3 \end{bmatrix} \tag{۱۵.۳}$$

برای بدست آوردن سرعت از بردار موقعیت نسبت به زمان مشتق می‌گیریم:

$$V_{G_1} = \begin{bmatrix} L_1\dot{\delta}_1 \sin(\alpha+\delta_1) - \frac{1}{2}L_2\dot{\phi}_1 \cos(\phi_1) \\ -L_1\dot{\delta}_1 \cos(\alpha+\delta_1) + \frac{1}{2}L_2\dot{\phi}_1 \sin(\phi_1) \end{bmatrix} \tag{۱۶.۳}$$

$$V_{G_2} = \begin{bmatrix} -L_1\dot{\delta}_2 \sin(\alpha-\delta_2) - \frac{1}{2}L_2\dot{\phi}_2 \cos(\phi_2) \\ L_1\dot{\delta}_2 \cos(\alpha-\delta_2) - \frac{1}{2}L_2\dot{\phi}_2 \sin(\phi_2) \end{bmatrix} \tag{۱۷.۳}$$

$$V_{rack} = \begin{bmatrix} 0 \\ -L_1\dot{\delta}_1 \cos(\alpha+\delta_1) + L_2\dot{\phi}_1 \sin(\phi_1) \end{bmatrix} \tag{۱۸.۳}$$

پس از جاگذاری روابط سرعت در معادله (۱۲.۳) و ساده‌سازی بدست می‌آید:

$$\begin{aligned} T =& \frac{1}{2}\left(I_1\dot{\delta}_1^2 + I_1\dot{\delta}_2^2 + I_2\dot{\phi}_1^2 + I_2\dot{\phi}_2^2\right) \\ &+ \frac{1}{8}m_2\left(4L_1^2\dot{\delta}_1^2 - 4L_1L_2\dot{\delta}_1\dot{\phi}_1 \sin(\alpha+\delta_1+\phi_1) + L_2^2\dot{\phi}_1^2\right) \\ &+ \frac{1}{8}m_2\left(4L_1^2\dot{\delta}_2^2 - 4L_1L_2\dot{\delta}_2\dot{\phi}_2 \sin(-\alpha+\delta_2+\phi_2) + L_2^2\dot{\phi}_2^2\right) \\ &+ \frac{1}{2}m_r\left(L_1\dot{\delta}_1 \cos(\alpha+\delta_1) - L_2\dot{\phi}_1 \sin(\phi_1)\right)^2 \end{aligned} \tag{۱۹.۳}$$

سمت چپ رابطه (۵.۳) را می‌توان به فرم ماتریسی زیر نوشت:

$$\begin{bmatrix} \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_1}\right)-\frac{\partial T}{\partial q_1} \\ \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_2}\right)-\frac{\partial T}{\partial q_2} \\ \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_3}\right)-\frac{\partial T}{\partial q_3} \\ \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_4}\right)-\frac{\partial T}{\partial q_4} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \\ a_{15} & a_{25} & a_{35} & a_{45} \\ a_{16} & a_{26} & a_{36} & a_{46} \\ a_{17} & a_{27} & a_{37} & a_{47} \\ a_{18} & a_{28} & a_{38} & a_{48} \end{bmatrix}^T \begin{bmatrix} \ddot{\delta}_1 \\ \ddot{\phi}_1 \\ \dot{\delta}_1^2 \\ \dot{\phi}_1^2 \\ \ddot{\delta}_2 \\ \ddot{\phi}_2 \\ \dot{\delta}_2^2 \\ \dot{\phi}_2^2 \end{bmatrix} \tag{۲۰.۳}$$

که درایه‌های ماتریس برابر است با:

$$\begin{aligned}
&a_{11} = I_1 + L_1^2 m_2 + L_1^2 m_r \cos^2(\alpha+\delta_1) \quad , \quad a_{22} = I_2 + \tfrac{1}{4}L_2^2 m_2 + L_2^2 m_r \sin^2(\phi_1) \\
&a_{35} = I_1 + L_1^2 m_2 \quad , \quad a_{46} = I_2 + \tfrac{1}{4}L_2^2 m_2 \\
&a_{13} = -L_1^2 m_r \sin(\alpha+\delta_1)\cos(\alpha+\delta_1) \quad , \quad a_{24} = L_2^2 m_r \sin(\phi_1)\cos(\phi_1) \\
&a_{12} = -\tfrac{1}{2}L_1L_2m_2\sin(\alpha+\delta_1+\phi_1) - L_1L_2m_r\sin(\phi_1)\cos(\alpha+\delta_1) \\
&a_{14} = -\tfrac{1}{2}L_1L_2m_2\cos(\alpha+\delta_1+\phi_1) - L_1L_2m_r\cos(\alpha+\delta_1)\cos(\phi_1) \\
&a_{21} = -\tfrac{1}{2}L_1L_2m_2\sin(\alpha+\delta_1+\phi_1) - L_1L_2m_r\sin(\phi_1)\cos(\alpha+\delta_1) \\
&a_{23} = -\tfrac{1}{2}L_1L_2m_2\cos(\alpha+\delta_1+\phi_1) + L_1L_2m_r\sin(\alpha+\delta_1)\sin(\phi_1) \\
&a_{36} = a_{45} = -\tfrac{1}{2}L_1L_2m_2\sin(-\alpha+\delta_2+\phi_2) \\
&a_{38} = a_{47} = -\tfrac{1}{2}L_1L_2m_2\cos(-\alpha+\delta_2+\phi_2) \\
&a_{15} = a_{16} = a_{17} = a_{18} = a_{25} = a_{26} = a_{27} = a_{28} = 0 \\
&a_{31} = a_{32} = a_{33} = a_{34} = a_{37} = a_{41} = a_{42} = a_{43} = a_{44} = a_{48} = 0
\end{aligned}$$

در نتیجه روابط (۸.۳) ، (۱۱.۳) و (۲۰.۳) معادله حاکم بر سیستم را کامل می‌کنند:

$$\begin{bmatrix} a_{ji} \end{bmatrix} \begin{bmatrix} \ddot{\delta}_1 & \ddot{\phi}_1 & \dot{\delta}_1^2 & \dot{\phi}_1^2 & \ddot{\delta}_2 & \ddot{\phi}_2 & \dot{\delta}_2^2 & \dot{\phi}_2^2 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & L_1\cos(\alpha+\delta_1) & -L_1\sin(\alpha+\delta_1) & 0 & L_1\cos(\alpha+\delta_1) \\ 0 & 0 & -L_2\sin(\phi_1) & L_2\cos(\phi_1) & 0 & -L_2\sin(\phi_1) \\ 0 & 1 & 0 & 0 & L_1\sin(\alpha-\delta_2) & -L_1\cos(\alpha-\delta_2) \\ 0 & 0 & 0 & 0 & L_2\cos(\phi_2) & -L_2\sin(\phi_2) \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_r \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \tag{۲۱.۳}$$

۴.۳.۳ حل معادلات حاکم برای F_r

برای بدست آوردن نیروی رک رابطه (۲۱.۳) بصورت زیر باز نویسی شده است:

$$\overbrace{\begin{bmatrix} a_{ji} \end{bmatrix} \begin{bmatrix} \ddot{\delta}_1 & \ddot{\phi}_1 & \dot{\delta}_1^2 & \dot{\phi}_1^2 & \ddot{\delta}_2 & \ddot{\phi}_2 & \dot{\delta}_2^2 & \dot{\phi}_2^2 \end{bmatrix}^T - \begin{bmatrix} \tau_1 & 0 & \tau_2 & 0 \end{bmatrix}^T}^{LS} = \underbrace{\begin{bmatrix} L_1\cos(\alpha+\delta_1) & -L_1\sin(\alpha+\delta_1) & 0 & L_1\cos(\alpha+\delta_1) \\ -L_2\sin(\phi_1) & L_2\cos(\phi_1) & 0 & -L_2\sin(\phi_1) \\ 0 & 0 & L_1\sin(\alpha-\delta_2) & -L_1\cos(\alpha-\delta_2) \\ 0 & 0 & L_2\cos(\phi_2) & -L_2\sin(\phi_2) \end{bmatrix}}_{\Gamma} \begin{bmatrix} F_r \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \tag{۲۲.۳}$$

بنابراین، می‌توان نوشت:

$$\begin{bmatrix} F_r & \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix}^T = \Gamma^{-1} LS \tag{۲۳.۳}$$

۵.۳.۳ حل معادلات حاکم برای δ_1

با فرض معلوم بودن ϕ_1 ، ϕ_2 ، δ_2 و مشتق اول و دوم آن‌ها ، برای بدست آوردن $\ddot{\delta}_1$ رابطه (۲۱.۳) بصورت زیر مرتب می‌شود:

$$\overbrace{\begin{bmatrix} a_{[1][2:8]} & -1 & 0 & -L_1\cos(\alpha+\delta_1) \\ a_{[2][2:8]} & 0 & 0 & L_2\sin(\phi_1) \\ a_{[3][2:8]} & 0 & -1 & 0 \\ a_{[4][2:8]} & 0 & 0 & 0 \end{bmatrix}}^{\Omega} \begin{bmatrix} \ddot{\phi}_1 & \dot{\delta}_1^2 & \dot{\phi}_1^2 & \ddot{\delta}_2 & \ddot{\phi}_2 & \dot{\delta}_2^2 & \dot{\phi}_2^2 & \tau_1 & \tau_2 & F_r \end{bmatrix}^T$$
$$= \underbrace{\begin{bmatrix} -a_{11} & -L_1\sin(\alpha+\delta_1) & 0 & L_1\cos(\alpha+\delta_1) \\ -a_{21} & L_2\cos(\phi_1) & 0 & -L_2\sin(\phi_1) \\ 0 & 0 & L_1\sin(\alpha-\delta_2) & -L_1\cos(\alpha-\delta_2) \\ 0 & 0 & L_2\cos(\phi_2) & -L_2\sin(\phi_2) \end{bmatrix}}_{\Psi} \begin{bmatrix} \ddot{\delta}_1 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (۲۴.۳)$$

با استفاده از نام گذاری‌های مشخص شده در رابطه (۲۴.۳) می‌توان نوشت:

$$\begin{bmatrix} \ddot{\delta}_1 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \Psi^{-1}\Omega \begin{bmatrix} \ddot{\phi}_1 & \dot{\delta}_1^2 & \dot{\phi}_1^2 & \ddot{\delta}_2 & \ddot{\phi}_2 & \dot{\delta}_2^2 & \dot{\phi}_2^2 & \tau_1 & \tau_2 & F_r \end{bmatrix}^T \quad (۲۵.۳)$$

و با تعریف مقادیر زیر رابطه (۲۶.۳) وارون ماتریس Ψ را بدست می‌دهد.

$$\begin{aligned} r_{11} &= -L_1\sin(\alpha+\delta_1) & r_{13} &= L_1\cos(\alpha+\delta_1) \\ r_{21} &= L_2\cos(\phi_1) & r_{23} &= -L_2\sin(\phi_1) \\ r_{32} &= L_1\sin(\alpha-\delta_2) & r_{33} &= -L_1\cos(\alpha-\delta_2) \\ r_{42} &= L_2\cos(\phi_2) & r_{43} &= -L_2\sin(\phi_2) \end{aligned}$$

$$\Psi^{-1}=\begin{bmatrix}-a_{11}&r_{11}&0&r_{13}\\-a_{21}&r_{21}&0&r_{23}\\0&0&r_{32}&r_{33}\\0&0&r_{42}&r_{43}\end{bmatrix}^{-1}=\frac{1}{\begin{pmatrix}a_{11}r_{21}r_{32}r_{43}-a_{11}r_{21}r_{33}r_{42}\\-a_{21}r_{11}r_{32}r_{43}+a_{21}r_{11}r_{33}r_{42}\end{pmatrix}}\times$$
$$\begin{bmatrix}\begin{pmatrix}r_{21}r_{33}r_{42}\\-r_{21}r_{32}r_{43}\end{pmatrix}&\begin{pmatrix}r_{11}r_{32}r_{43}\\-r_{11}r_{33}r_{42}\end{pmatrix}&\begin{pmatrix}r_{11}r_{23}r_{42}\\-r_{13}r_{21}r_{42}\end{pmatrix}&\begin{pmatrix}r_{13}r_{21}r_{32}\\-r_{11}r_{23}r_{32}\end{pmatrix}\\\begin{pmatrix}a_{21}r_{33}r_{42}\\-a_{21}r_{32}r_{43}\end{pmatrix}&\begin{pmatrix}a_{11}r_{32}r_{43}\\-a_{11}r_{33}r_{42}\end{pmatrix}&\begin{pmatrix}a_{11}r_{23}r_{42}\\-a_{21}r_{13}r_{42}\end{pmatrix}&\begin{pmatrix}a_{21}r_{13}r_{32}\\-a_{11}r_{23}r_{32}\end{pmatrix}\\0&0&\begin{pmatrix}a_{11}r_{21}r_{43}\\-a_{21}r_{11}r_{43}\end{pmatrix}&\begin{pmatrix}a_{21}r_{11}r_{33}\\-a_{11}r_{21}r_{33}\end{pmatrix}\\0&0&\begin{pmatrix}a_{21}r_{11}r_{42}\\-a_{11}r_{21}r_{42}\end{pmatrix}&\begin{pmatrix}a_{11}r_{21}r_{32}\\-a_{21}r_{11}r_{32}\end{pmatrix}\end{bmatrix} \tag{۲۶.۳}$$

پس از جاگذاری رابطه (۲۶.۳) در معادله (۲۵.۳) و حذف ردیف‌های ۲ تا ۴ (که مربوط به ضرایب لاگرانژ هستند)، $\ddot{\delta}_1$ محاسبه می‌شود:

$$\ddot{\delta}_1=\frac{1}{a_{11}r_{21}r_{32}r_{43}-a_{11}r_{21}r_{33}r_{42}-a_{21}r_{11}r_{32}r_{43}+a_{21}r_{11}r_{33}r_{42}}\times$$
$$\begin{bmatrix}r_{21}r_{33}r_{42}-r_{21}r_{32}r_{43}\\r_{11}r_{32}r_{43}-r_{11}r_{33}r_{42}\\r_{11}r_{23}r_{42}-r_{13}r_{21}r_{42}\\r_{13}r_{21}r_{32}-r_{11}r_{23}r_{32}\end{bmatrix}^T\begin{bmatrix}a_{12}&a_{22}&0&0\\a_{13}&a_{23}&0&0\\a_{14}&a_{24}&0&0\\0&0&a_{35}&a_{45}\\0&0&a_{36}&a_{46}\\0&0&0&a_{47}\\0&0&a_{38}&0\\-1&0&0&0\\0&0&-1&0\\-L_1\cos(\alpha+\delta_1)&L_2\sin(\phi_1)&0&0\end{bmatrix}^T\begin{bmatrix}\ddot{\phi}_1\\\dot{\delta}_1^2\\\dot{\phi}_1^2\\\ddot{\delta}_2\\\ddot{\phi}_2\\\dot{\delta}_2^2\\\dot{\phi}_2^2\\\tau_1\\\tau_2\\F_r\end{bmatrix} \tag{۲۷.۳}$$

۶.۳.۳ محاسبه ϕ_1 ، ϕ_2 ، δ_2 و مشتق اول و دوم آن‌ها برحسب δ_1

قدم آخر محاسبه ϕ_1 ، ϕ_2 ، δ_2 و مشتق اول و دوم آن‌ها برحسب δ_1 است. با استفاده از (۶.۳) می‌توان نوشت:

$$\phi_1 = -\sin^{-1}\left(\frac{L_1\cos(\alpha+\delta_1)-d}{L_2}\right) \tag{۲۸.۳آ}$$

$$\phi_2 = 2\tan^{-1}\left(\frac{2L_2 d \pm \sqrt{\begin{matrix}-L_1^4+2L_1^2L_2^2+2L_1^2d^2-L_2^4+2L_2^2d^2\\ -d^4+2L_1^2\xi^2+2L_2^2\xi^2-2d^2\xi^2-\xi^4\end{matrix}}}{-L_1^2+L_2^2+2L_2\xi+d^2+\xi^2}\right) \tag{۲۸.۳ب}$$

$$\delta_2 = \alpha+\sin^{-1}\left(\frac{L_2\cos(\phi_2)-\xi}{L_1}\right) \tag{۲۸.۳ج}$$

$$\xi = D - L_1\sin(\alpha+\delta_1) - L_2\cos(\phi_1) - L_3$$

همچنین پس از مشتق گرفتن از روابط (۶.۳) نسبت به زمان، خواهیم داشت:

$$\dot{\phi}_1 = \frac{L_1\dot{\delta}_1\sin(\alpha+\delta_1)}{L_2\cos(\phi_1)} \tag{۲۹.۳آ}$$

$$\dot{\phi}_2 = -\frac{\left(L_1\dot{\delta}_1\cos(\alpha+\delta_1)-L_2\dot{\phi}_1\sin(\phi_1)\right)\sin(\alpha-\delta_2)}{L_2\cos(\alpha-\delta_2+\phi_2)} \tag{۲۹.۳ب}$$

$$\dot{\delta}_2 = \frac{\left(L_1\dot{\delta}_1\cos(\alpha+\delta_1)-L_2\dot{\phi}_1\sin(\phi_1)\right)\cos(\phi_2)}{L_1\cos(\alpha-\delta_2+\phi_2)} \tag{۲۹.۳ج}$$

به طور مشابه مشتق مرتبه دوم روابط (۶.۳) نسبت به زمان، معادلات زیر را بدست می‌دهد:

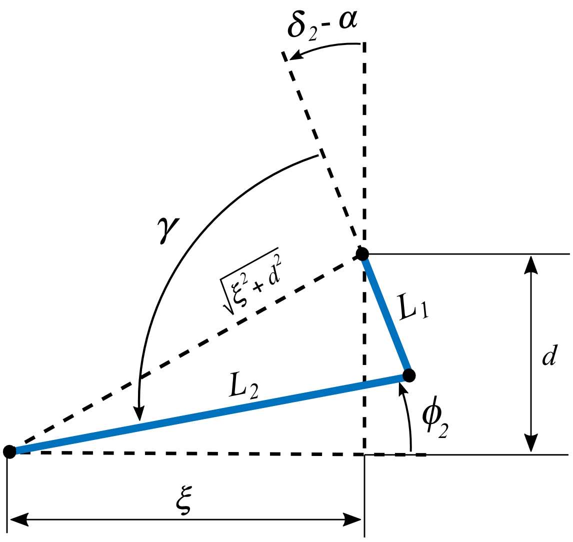
$$\ddot{\phi}_1 = \frac{L_1\left(\ddot{\delta}_1\sin(\alpha+\delta_1)+\dot{\delta}_1^2\cos(\alpha+\delta_1)\right)+L_2\dot{\phi}_1^2\sin(\phi_1)}{L_2\cos(\phi_1)} \tag{۳۰.۳آ}$$

$$
\begin{aligned}
\ddot{\phi}_2 = \frac{1}{L_2 \cos(\alpha - \delta_2 + \phi_2)} \Big(& - L_1 \ddot{\delta}_1 \sin(\alpha - \delta_2) \cos(\alpha + \delta_1) + L_1 \dot{\delta}_2^2 \\
& + L_1 \dot{\delta}_1^2 \sin(\alpha + \delta_1) \sin(\alpha - \delta_2) + L_2 \ddot{\phi}_1 \sin(\alpha - \delta_2) \sin(\phi_1) \\
& + L_2 \dot{\phi}_1^2 \sin(\alpha - \delta_2) \cos(\phi_1) + L_2 \dot{\phi}_2^2 \sin(\alpha - \delta_2 + \phi_2) \Big)
\end{aligned}
\quad (۳۰.۳ب)
$$

$$
\begin{aligned}
\ddot{\delta}_2 = \frac{1}{L_1 \cos(\alpha - \delta_2 + \phi_2)} \Big(& L_1 \ddot{\delta}_1 \cos(\alpha + \delta_1) \cos(\phi_2) - L_1 \dot{\delta}_1^2 \sin(\alpha + \delta_1) \cos(\phi_2) \\
- L_1 \dot{\delta}_2^2 \sin(\alpha - \delta_2 + \phi_2) - L_2 \ddot{\phi}_1 \sin(\phi_1) \cos(\phi_2) & - L_2 \dot{\phi}_1^2 \cos(\phi_1) \cos(\phi_2) - L_2 \dot{\phi}_2^2 \Big)
\end{aligned}
\quad (۳۰.۳ج)
$$

۷.۳.۳ حل عددی سینماتیک معکوس

هدف این بخش معرفی روش عددی برای جایگزین کردن معادله (۲۸.۳ب) به منظور جلوگیری از مشکلات عددی ناشی از جذر و وارون تانژانت است.



شکل ۷.۳: معرفی زاویه گاما

با توجه به شکل ۷.۳ از قانون کسینوس داریم:

$$
\begin{aligned}
&d^2+\xi^2=L_1^2+L_2^2-2L_1L_2\cos(\gamma)\\
&\Rightarrow \gamma=\cos^{-1}\left(\frac{L_1^2+L_2^2-d^2-\xi^2}{2L_1L_2}\right)\\
&\xi=D-L_1\sin\left(\alpha+\delta_1\right)-L_2\cos\left(\phi_1\right)-L_3
\end{aligned}
\tag{۳۱.۳}
$$

از طرفی با توجه به شکل می‌توان نوشت:

$$
\frac{\pi}{2}=\delta_2-\alpha+\gamma-\phi_2 \quad\Rightarrow\quad \delta_2=\frac{\pi}{2}-\gamma+\phi_2+\alpha \tag{۳۲.۳}
$$

با جاگذاری (۳۲.۳) در معادله قید (۶.۳ب) داریم:

$$
\begin{aligned}
&L_1\cos\left(\alpha-(\frac{\pi}{2}-\gamma+\phi_2+\alpha)\right)+L_2\sin(\phi_2)-d=0\\
&\rightarrow L_1\sin(\gamma-\phi_2)+L_2\sin(\phi_2)-d=0\\
&\Rightarrow \phi_2=\sin^{-1}\left(\frac{d-L_1\sin(\gamma-\phi_2)}{L_2}\right) \quad\equiv\quad x=\mathrm{g}(x)
\end{aligned}
\tag{۳۳.۳}
$$

با استفاده از (۳۳.۳) به کمک روش عددی نقطه ثابت مقدار ϕ_2 بصورت زیر محاسبه می‌شود:

$$
\begin{aligned}
&n=0\\
&x_{n+1}=x_n+2\epsilon
\end{aligned}
$$

$$
\begin{aligned}
&\text{While}\quad |x_{n+1}-x_n|>\epsilon:\\
&\qquad x_{n+1}=\mathrm{g}(x_n)\\
&\qquad n=n+1
\end{aligned}
$$

۸.۳.۳ محاسبه رابطه بین زاویه فرمان و جابجایی رک

در این بخش معادلات لازم برای بدست آوردن Δ_r برحسب δ_1 و برعکس محاسبه می‌شود. از آنجایی که هدف کنترلی برای سیستم فرمان موقعیت رک است، این معادلات برای اعمال ورودی و گرفتن فیدبک از موقعیت سیستم فرمان کاربرد دارند. با توجه به شکل ۵.۳ و ۳.۳ می‌توان نوشت:

$$\Delta_r = L_1 \sin(\alpha + \delta_1) + L_2 \cos(\phi_1) - (D - L_3)/2 \tag{۳۴.۳}$$

اگر از (۳۴.۳) نسبت به زمان مشتق بگیریم خواهیم داشت:

$$\dot{\Delta}_r = L_1 \dot{\delta}_1 \cos(\alpha + \delta_1) - L_2 \dot{\phi}_1 \sin(\phi_1) \tag{۳۵.۳}$$

$$\ddot{\Delta}_r = L_1 \ddot{\delta}_1 \cos(\alpha + \delta_1) - L_1 \dot{\delta}_1^2 \sin(\alpha + \delta_1) - L_2 \ddot{\phi}_1 \sin(\phi_1) - L_2 \dot{\phi}_1^2 \cos(\phi_1) \tag{۳۶.۳}$$

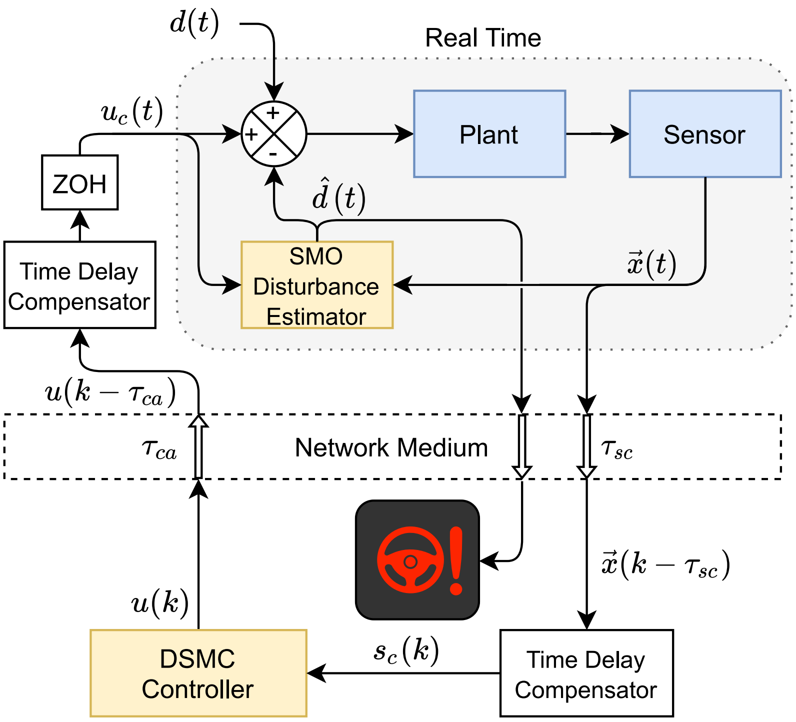
و معکوس روابط (۳۴.۳) ، (۳۵.۳) و (۳۶.۳) به ترتیب برابر است با:

$$\delta_1 = -\alpha + \operatorname{asin}\left(\frac{(D - L_3)/2 - L_2 \cos(\phi_1) + \Delta_r}{L_1}\right) \tag{۳۷.۳آ}$$

$$\dot{\delta}_1 = \frac{L_2 \dot{\phi}_1 \sin(\phi_1) + \dot{\Delta}_r}{L_1 \cos(\alpha + \delta_1)} \tag{۳۷.۳ب}$$

$$\ddot{\delta}_1 = \frac{L_1 \dot{\delta}_1^2 \sin(\alpha + \delta_1) + L_2 \ddot{\phi}_1 \sin(\phi_1) + L_2 \dot{\phi}_1^2 \cos(\phi_1) + \ddot{\Delta}_r}{L_1 \cos(\alpha + \delta_1)} \tag{۳۷.۳ج}$$

۴.۳ طراحی کنترل‌کننده تحمل‌پذیر عیب



شکل ۸.۳: دیاگرام سیستم حلقه بسته

۱.۴.۳ طراحی مشاهده کننده اغتشاش [۱۴]

در این بخش با کمک لم ۱.۳ یک مشاهده‌کننده زمان پیوسته اغتشاش برای سیستم غیرخطی (۳۸.۳)، که دارای عدم قطعیت و اغتشاش ورودی است، درحالت کلی طراحی می‌شود.

$$\begin{cases} \dot{x}_i = x_{i+1} & i = 1, 2, \dots, n-1 \\ \dot{x}_n = f(x) + \Delta f(x) + \big(g(x) + \Delta g(x)\big)\big(u + d_0(t)\big) \\ y = x_1 \end{cases} \tag{۳۸.۳}$$

که $\Delta f(x)$ و $\Delta g(x)$ عدم قطعیت، و $d_0(t)$ اغتشاش ورودی به سیستم است.

لم ۱.۳. تابع پیوسته و مثبت معین $V(t)$ را در نظر بگیرید. اگر نامساوی زیر برقرار باشد. [۷۳]

$$\dot{V}(t) + \xi V^{\chi}(t) + \vartheta V(t) \leq 0, \quad \forall t > t_0 \tag{۳۹.۳}$$

$V(t)$ در زمان محدود t_s به نقطه تعادل همگرا می‌شود، که برابر است با

$$t_s \leq t_0 + \frac{1}{\vartheta(1+\chi)} \ln\left(\frac{\vartheta V^{1-\chi}\left(t_0\right)}{\varepsilon} + 1\right) \tag{۴۰.۳}$$

که $0 < \chi < 1$ و $\vartheta > \xi > 0$.

درصورتی که مجموع عدم قطعیت و اغتشاش بوجود آمده محدود و کران بالای آن مشخص باشد $\beta > |d(t)|$ ، می‌توان معادلات سیستم (۳۸.۳) را با معادله خطی (۴۱.۳) تخمین زد و برای طراحی مشاهده‌کننده اغتشاش از آن استفاده کرد.

$$\begin{cases} \dot{x}_i = x_{i+1} & i = 1, 2, \ldots, n-1 \\ \dot{x}_n = \vec{a} \cdot x + b(u + d(t)) \\ y = x_1 \end{cases} \tag{۴۱.۳}$$

$$d(t) = \Delta' f(x) + \Delta' g(x) u + d_0(t).$$

فرض ۲.۳. تابع اغتشاش ورودی نسبت به زمان پیوسته و مشتق‌پذیر با کران بالای $\beta > |d(t)|$ فرض شده است.

برای طراحی مشاهده‌کننده، متغیر کمکی s_o را بصورت زیر تعریف می‌کنیم:

$$s_o = z - x_n \tag{۴۲.۳}$$

که z از رابطه زیر بدست می‌آید:

$$\dot{z} = -ks_o - \varepsilon s_o^{p_o/q_o} - \beta \operatorname{sign}(s_o) - |\vec{a}\cdot x| \operatorname{sign}(s_o) + bu \tag{۴۳.۳}$$

و، p_o و q_o اعداد فرد مثبت هستند که $p_o < q_o$ است. k و ε اعداد مثبت و $|d(t)| < \beta$ است. همچنین، تابع اغتشاش مطابق فرض ۲.۳ پیوسته و مشتق پذیر فرض شده است. در نتیجه تخمین اغتشاش از رابطه زیر بدست می‌آید:

$$\hat{d} = \frac{1}{b}\Big(-ks_o - \varepsilon s_o^{p_o/q_o} - \beta \operatorname{sign}(s_o) - |\vec{a}\cdot x| \operatorname{sign}(s_o) - \vec{a}\cdot x \Big) \tag{۴۴.۳}$$

با درنظر گرفتن (۴۱.۳)، (۴۲.۳) و (۴۳.۳) داریم:

$$\dot{s_o} = \dot{z} - \dot{x}_n = -ks_o - \varepsilon s_o^{p_o/q_o} - \beta \operatorname{sign}(s_o) - |\vec{a}\cdot x| \operatorname{sign}(s_o) - \vec{a}\cdot x - bd \tag{۴۵.۳}$$

و با توجه به (۴۴.۳) و (۴۲.۳) خطای تخمین اغتشاش برابر است با:

$$\begin{aligned}
\tilde{d} =& \hat{d} - d = \frac{1}{b}(b\hat{d} - bd) \\
=& \frac{1}{b}(-ks_o - \varepsilon s_o^{p_o/q_o} - \beta \operatorname{sign}(s_o) - |\vec{a}\cdot x| \operatorname{sign}(s_o) - \vec{a}\cdot x - bd) \\
=& \frac{1}{b}(-ks_o - \varepsilon s_o^{p_o/q_o} - \beta \operatorname{sign}(s_o) - |\vec{a}\cdot x| \operatorname{sign}(s_o) - \vec{a}\cdot x - \dot{x}_n + \vec{a}\cdot x + bu) \\
=& \frac{1}{b}(-ks_o - \varepsilon s_o^{p_o/q_o} - \beta \operatorname{sign}(s_o) - |\vec{a}\cdot x| \operatorname{sign}(s_o) + bu - \dot{x}_n) \\
=& \frac{1}{b}(\dot{z} - \dot{x}_n) = \frac{1}{b}\dot{s_o}
\end{aligned} \tag{۴۶.۳}$$

قضیه ۳.۳. برای سیستم (۳۸.۳) با اعمال مشاهده‌کننده اغتشاش (۴۴.۳) خطای تخمین اغتشاش (۴۶.۳) در زمان محدود به صفر همگرا می‌شود.

اثبات. تابع لیاپانوف منتخب زیر را درنظر بگیرید:

$$V_o = \frac{1}{2}s_o^2 \tag{۴۷.۳}$$

مشتق تابع V_o نسبت به زمان برابر است با:

$$\begin{aligned}\dot{V}_o =& s_o\dot{s}_o = s_o\Big(- ks_o - \varepsilon s_o^{\frac{p_o}{q_o}} - \beta\,\mathrm{sign}\,(s_o) - |\vec{a}\cdot x|\,\mathrm{sign}\,(s_o) - \vec{a}\cdot x - bd\Big)\\ \leq& - ks_o^2 - \varepsilon s_o^{(p_o+q_o)/q_o} - \beta s_o\,\mathrm{sign}\,(s_o) - |\vec{a}\cdot x| s_o\,\mathrm{sign}\,(s_o) - s_o\vec{a}\cdot x - s_o d\\ \leq& - ks_o^2 - \varepsilon s_o^{(p_o+q_o)/q_o} - \beta\,|s_o| - |\vec{a}\cdot x|\,|s_o| - s_o\vec{a}\cdot x + |s_o|\,|d|\\ \leq& - ks_o^2 - \varepsilon s_o^{(p_o+q_o)/q_o}\\ \leq& - 2kV_o - 2^{(p_o+q_o)/2q_o}\varepsilon V_o^{(p_o+q_o)/2q_o}.\end{aligned} \tag{۴۸.۳}$$

بنابراین، مطابق لم ۱.۳ متغیر کمکی s_o در زمان محدود به صفر همگرا می‌شود، در نتیجه خطای تخمین اغتشاش نیز در زمان محدود به صفر میل می‌کند. □

ملاحظه ۴.۳. بر اساس لم ۱.۳ و معادله (۴۸.۳) زمان همگرایی تخمین اغتشاش برابر است با:

$$t_s \leq t_0 + \frac{1}{k(q_o - p_o)}\ln\left(\frac{k(s_o\,(t_0))^{(q_o-p_o)/q_o}}{\varepsilon} + 1\right) \tag{۴۹.۳}$$

۱.۱.۴.۳ تشخیص عیب

مشابه دیاگرام شکل ۸.۳ تخمین اغتشاش وارد شده $\hat{d}(t)$ که از رابطه (۴۴.۳) بصورت بی‌درنگ بدست می‌آید، علاوه بر خنثی کردن اغتشاش خارجی و جبران عدم قطعیت سیستم، برای تشخیص و اعلام وجود عیب در سیستم حلقه بسته استفاده می‌شود.

با این روش می‌توان وجود عیب در عملگر و یا مکانیسم فرمان را شناسایی کرد.

۲.۴.۳ طراحی کنترل‌کننده [۶۱]

سیستم خطی زمان پیوسته و نامتغیر با زمان زیر را که شامل عدم قطعیت و تأخیر ورودی است، در نظر بگیرید:

$$\begin{cases} \dot{x}(t) = Ax(t) + B\big(u(t-\tau_r) + d(t)\big), \\ y(t) = Cx(t) \end{cases} \tag{۵۰.۳}$$

$x \in R^n$ بردار متغیرهای حالت سیستم، $u \in R^m$ ورودی کنترل، و $y \in R^p$ خروجی سیستم است. در معادله فوق $A \in R^{n\times n}$، $B \in R^{n\times m}$ و $C \in R^{p\times n}$ ماتریس‌هایی با ابعاد مناسب هستند، و $d(t)$ اغتشاش و عدم قطعیت سیستم که $|d(t)| \leq \beta'$ و τ_r کل تأخیر به وجود آمده در شبکه در حوزه زمان پیوسته است.

سیستم زمان گسسته معادل سیستم (۵۰.۳) با نرخ نمونه برداری h برابر است با:

$$\begin{cases} x(k+1) = Fx(k) + G\big(u(k-\hat{\tau}) + d(k)\big), \\ y(k) = Cx(k) \end{cases} \tag{۵۱.۳}$$

که $F = e^{Ah}$ و $G = \int_0^h e^{At}B\,dt$ برای ساده سازی فرض می‌کنیم $d(k)$ به آرامی تغییر می‌کند و طی بازه زمانی $kh \leq t \leq (k+1)h$ ثابت می‌ماند.

اگر τ_{rsc} تأخیر شبکه بین حسگر و کنترل‌کننده و τ_{rcr} تأخیر شبکه بین کنترل‌کننده و عملگر در حوزه زمان پیوسته باشد، کل تأخیر به وجود آمده در شبکه در حوزه زمان گسسته برابر است با:

$$\hat{\tau} = \hat{\tau}_{sc} + \hat{\tau}_{ca} \tag{۵۲.۳}$$

که $\hat{\tau}_{sc} = \frac{\tau_{rsc}}{h}$ و $\hat{\tau}_{ca} = \frac{\tau_{rca}}{h}$

فرض ۵.۳. در معادله ۵۰.۳ مقدار $d(t)$ که مجموع خطای تخمین اغتشاش ($\tilde{d}(t)$) و عدم قطعیت معادلات سیستم است محدود و کمتر از β' فرض می‌شود.

فرض ۶.۳. در این روش $0 \leq \hat{\tau} < h$ مفروض است و تأخیرهای بزرگتر از نرخ نمونه‌برداری اتلاف بسته تلقی می‌شود.

۱.۲.۴.۳ تعریف سطح لغزش

لم ۷.۳. متغیر لغزش برای سیستم (۵۱.۳) با تأخیر شبکه در مسیر حسگر به کنترل‌کننده $\hat{\tau}_{sc}$ و احتمال اتلاف بسته سیگنال $\bar{\alpha}$ که فرضیات ۵.۳ و ۶.۳ برای آن صادق است از رابطه زیر بدست می‌آید: [۶۹]

$$s_c(k) = (1-\bar{\alpha})x'_c(k) - \bar{\alpha}x'_c(k-1) \tag{۵۳.۳}$$

که $x'_c(k) = C_s x(k) - \varsigma C_s x(k-1)$ ، $\varsigma = \frac{\hat{\tau}_{sc}}{1+\hat{\tau}_{sc}}$ و C_s ضرایب سطح لغزش هستند.

اثبات. بردار متغیرهای حالت با وجود تأخیر حسگر به کنترل‌کننده برابر است با:

$$x_c(k) \triangleq x(k-\hat{\tau}_{sc}) \tag{۵۴.۳}$$

با اعمال تبدیل z به (۵۴.۳) داریم:

$$x_c(z) = x(z)z^{-\hat{\tau}_{sc}} \tag{۵۵.۳}$$

با استفاده از روش تخمین تیران [۸] ، تأخیر در حوزه زمان گسسته برابر است با:

$$z^{-\hat{\tau}_{sc}} = \Sigma_{k=0}^{n}(-1)^k \begin{pmatrix} n \\ k \end{pmatrix} \Pi_{i=0}^{n}\frac{2\hat{\tau}_{sc}+i}{2\hat{\tau}_{sc}+k+i}z^{-k} \tag{۵۶.۳}$$

که n مرتبه تخمین را مشخص می‌کند.

در این صورت تخمین مرتبه اول تأخیر برابر است با:

$$z^{-\hat{\tau}_{sc}} = \left[(-1)^0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \left\{ \frac{2\hat{\tau}_{sc}}{2\hat{\tau}_{sc}} \times \frac{2\hat{\tau}_{sc}+1}{2\hat{\tau}_{sc}+1} \right\} z^0 + (-1)^1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \left\{ \frac{2\hat{\tau}_{sc}}{2\hat{\tau}_{sc}+1} \times \frac{2\hat{\tau}_{sc}+1}{2\hat{\tau}_{sc}+2} \right\} z^{-1} \right] \quad (۵۷.۳)$$

پس از ساده سازی خواهیم داشت:

$$z^{-\hat{\tau}_{sc}} = 1 - \varsigma z^{-1} \quad (۵۸.۳)$$

که $\varsigma = \frac{\hat{\tau}_{sc}}{1+\hat{\tau}_{sc}}$ با جاگذاری (۵۸.۳) در (۵۵.۳) داریم:

$$x_c(z) = x(z) - \varsigma x(z) z^{-1} \quad (۵۹.۳)$$

و با اعمال تبدیل معکوس z بردار متغیرهای حالت جبران شده برابر است با:

$$x_c(k) = x(k) - \varsigma x(k-1) \quad (۶۰.۳)$$

از طرف دیگر رابطه زیر برای بیان ریاضی اتلاف تصادفی بسته در شبکه برقرار است:

$$x_p(k) = (1-\alpha(k))x_c(k) - \alpha(k)x_c(k-1) \quad (۶۱.۳)$$

که $\alpha(k) \in \{0, 1\}$ متغیر تصادفی است و داریم:

$$P\{\alpha(k)=1\} = E\{\alpha(k)\} = \bar{\alpha} \quad (۶۲.۳آ)$$

$$P\{\alpha(k)=0\} = 1 - E\{\alpha(k)\} = 1 - \bar{\alpha} \quad (۶۲.۳ب)$$

که $0 \leq \bar{\alpha} < 1$ احتمال اتلاف بسته شبکه را نشان می‌دهد و $E\{\alpha(k)\}$ امید ریاضی متغیر تصادفی $\alpha(k)$ است.

در نتیجه می‌توان (۶۱.۳) را بصورت زیر تقریب زد:

$$x_p(k) = (1-\bar{\alpha})x_c(k) - \bar{\alpha}x_c(k-1) \tag{۶۳.۳}$$

متغیر لغزش که تأثیر تأخیر کسری و اتلاف بسته شبکه را جبران می‌کند برابر است با:

$$s_c(k) = C_s x_p(k) \tag{۶۴.۳}$$

که C_s بردار ضرایب تابع سطح لغزش است و پارامتر طراحی کنترل‌کننده محسوب می‌شود. با جاگذاری (۶۳.۳) در معادله سطح لغزش خواهیم داشت:

$$s_c(k) = (1-\bar{\alpha})x'_c(k) + \bar{\alpha}x'_c(k-1) \tag{۶۵.۳}$$

که $x'_c(k) = C_s x(k) - \varsigma C_s x(k-1)$، $\varsigma = \frac{\hat{\tau}_{sc}}{1+\hat{\tau}_{sc}}$ و این اثبات را کامل می‌کند. □

انتخاب ضرایب C_s باید به گونه‌ای صورت گیرد که معادله سطح لغزش $S_c(k)$ پایدار باشد؛ بنابراین اگر حالت‌های سیستم روی سطح لغزش نگه داشته شوند روی آن به سمت مبدأ سر می‌خورند، و در صورتی که متغیرهای حالت برحسب خطای ردیابی تعریف شوند مبدأ بیانگر صفر بودن خطای ردیابی خواهد بود.

۲.۲.۴.۳ طراحی کنترل‌کننده

این بخش با استفاده از سطح لغزش (۶۵.۳) کنترل‌کننده‌ی مد لغزشی زمان گسسته برای سیستم تحت شبکه (۵۱.۳) در قالب لم زیر ارائه می‌کند.

لم ۸.۳. کنترل‌کننده مد لغزشی غیرسوییچینگ زمان گسسته برای سیستم (۵۱.۳) در حضور تأخیر کسری

تصادفی با فرض ۶.۳ و اتلاف بسته شبکه با وجود عدم قطعیت $d(t)$ با فرض ۵.۳ برابر است با:

$$u(k) = -(C_sG)^{-1}\frac{1}{1-\bar{\alpha}}\left[Hx(k) - Ix(k) + Kx(k) - Lx(k-1) - J\right] - \hat{d}(kh) \tag{۶۶.۳}$$

که در آن:

$$\begin{aligned} &H = (1-\bar{\alpha})(C_sF) \quad , \quad I = \varsigma(1-\bar{\alpha})C_s \quad , \quad K = \bar{\alpha}C_s \\ &L = \varsigma\bar{\alpha}C_s \quad , \quad J = [1 - q(s_c(k))] \quad , \quad q[s(k)] = \beta'/(\beta' + |s_c(k)|) \end{aligned}$$

اثبات. برای نوشتن قانون کنترلی، از قانون همگرایی [1] غیر سوییچینگ [2] که باعث همگرایی سریعتر و بدون نوسان [3] در فضای زمان گسسته می‌شود، استفاده شده است [۷۱]. قانون همگرایی زیر را در نظر بگیرید:

$$\begin{aligned} &s(k+1) = \{1 - q[s(k)]\} \\ &q[s(k)] = \frac{\beta'}{\beta' + |s_c(k)|} \end{aligned} \tag{۶۷.۳}$$

که $\beta' > |d(t)|$ است. پس از جاگذاری مقدار $s(k+1)$ از (۶۵.۳) خواهیم داشت:

$$(1-\bar{\alpha})x'_c(k+1) + \bar{\alpha}x'_c(k) = \{1 - q[s(k)]\} \tag{۶۸.۳}$$

با جاگذاری x'_c می‌توان نوشت:

$$(1-\bar{\alpha})[C_sx(k+1) - \varsigma C_sx(k)] + \bar{\alpha}[C_sx(k) - \varsigma C_sx(k-1)] = \{1 - q[s(k)]\} \tag{۶۹.۳}$$

ملاحظه ۹.۳. در معادله (۶۵.۳) اثر تأخیر شبکه بین حسگر و کنترل‌کننده در رابطه سطح لغزش جبران شده است. بطور مشابه باید اثر تأخیر بین کنترل‌کننده و عملگر در سمت عملگر جبران شود، در نتیجه از دید کنترل‌کننده

[1] reaching law
[2] non-switching
[3] Chattering

$u(k-\tau_{ca}) = u(k)$ است.

با توجه به ملاحظه ۹.۳ می‌توان $x(k+1)$ را از رابطه (۵۱.۳) جایگزین کرد و نوشت:

$$(1-\bar{\alpha})\Big[C_s[Fx(k)+G\big(u(k)+d(k)\big)] - \varsigma C_s x(k)\Big] + \bar{\alpha}[C_s x(k) - \varsigma C_s x(k-1)] = \{1-q[s(k)]\} \tag{۷۰.۳}$$

همچنین، با توجه به قضیه ۳.۳ می‌توان $d(k)$ را با $\hat{d}(kh)$ از رابطه (۴۴.۳) جایگزین کرد. پس از ساده‌سازی داریم:

$$(1-\bar{\alpha})C_sFx(k) + (1-\bar{\alpha})C_sG\big(u(k)+\hat{d}(kh)\big) - \varsigma(1-\bar{\alpha})C_s x(k) + \bar{\alpha}C_s x(k) - \varsigma\bar{\alpha}C_s x(k-1) = \{1-q[s(k)]\} \tag{۷۱.۳}$$

قانون کنترلی از حل معادله (۷۱.۳) برحسب $u(k)$ بدست می‌آید، که برابر است با:

$$u(k) = -(C_sG)^{-1}\frac{1}{1-\bar{\alpha}}\Big[(1-\bar{\alpha})C_sFx(k) - \varsigma(1-\bar{\alpha})C_s x(k) + \bar{\alpha}C_s x(k) - \varsigma\bar{\alpha}C_s x(k-1) - \{1-q[s(k)]\}\Big] - \hat{d}(kh) \tag{۷۲.۳}$$

و می‌توان آن را به فرم زیر بازنویسی کرد:

$$u(k) = -(C_sG)^{-1}\frac{1}{1-\bar{\alpha}}\big[Hx(k) - Ix(k) + Kx(k) - Lx(k-1) - J\big] - \hat{d}(kh) \tag{۷۳.۳}$$

$$\begin{array}{llll} H = (1-\bar{\alpha})(C_sF) & , \quad I = \varsigma(1-\bar{\alpha})C_s & , \quad K = \bar{\alpha}C_s \\ L = \varsigma\bar{\alpha}C_s & , \quad J = [1-q(s_c(k))] & , \quad q[s(k)] = \beta'/(\beta'+|s_c(k)|) \end{array}$$

□

۳.۲.۴.۳ تحلیل پایداری

قضیه زیر شرط پایداری الگوریتم کنترلی ارائه شده در دیاگرام شکل ۸.۳ را بیان می‌کند.

قضیه ۱۰.۳. مسیرهای حالت سیستم حلقه بسته (۵۱.۳) در حضور تأخیر شبکه و اتلاف بسته، توسط کنترل‌کننده (۷۳.۳) به سمت سطح لغزش (۶۵.۳) حرکت می‌کنند و روی آن باقی می‌مانند اگر پارامترهای کنترل‌کننده به نحوی انتخاب شوند که:

$$
\begin{aligned}
&0 \le \Gamma^T \Gamma < s^T(k)s(k) \\
&\Gamma = \frac{1}{1-\bar{\alpha}}[1-q(s_c(k))]s_c(k)
\end{aligned}
\tag{۷۴.۳}
$$

اثبات. تابع کاندیدای لیاپانوف زیر را در نظر بگیرید:

$$
V_s(k) = s^T(k)s(k) \tag{۷۵.۳}
$$

با مشتق (دیفرنس) پیشرو گرفتن از آن داریم:

$$
\Delta V_s(k) = s^T(k+1)s(k+1) - s^T(k)s(k) \tag{۷۶.۳}
$$

و اگر مقدار $s_c(k+1)$ را از معادله سطح لغزش (۶۵.۳) جاگذاری کنیم خواهیم داشت:

$$
\Delta V_s(k) = [(1-\bar{\alpha})x'_c(k+1) + \bar{\alpha}x'_c(k)]^T[(1-\bar{\alpha})x'_c(k+1) + \bar{\alpha}x'_c(k)] - s^T(k)s(k) \tag{۷۷.۳}
$$

پس از جاگذاری $x'_c(k+1)$ داریم:

$$
\begin{aligned}
\Delta V_s(k) =&[(1-\bar{\alpha})[C_s x(k+1) - \varsigma C_s x(k)] + \bar{\alpha}x'_c(k)]^T \\
&[(1-\bar{\alpha})[C_s x(k+1) - \varsigma C_s x(k)] + \bar{\alpha}x'_c(k)] - s^T(k)s(k)
\end{aligned}
\tag{۷۸.۳}
$$

$x(k+1)$ را با معادله سیستم (۵۱.۳) جایگزین می‌کنیم:

$$\Delta V_s(k) = [(1-\bar{\alpha})[C_s[Fx(k)+G\big(u(k)+d(k)\big)] - \varsigma C_s x(k)] + \bar{\alpha} x_c'(k)]^T \\ [(1-\bar{\alpha})[C_s[Fx(k)+G\big(u(k)+d(k)\big)] - \varsigma C_s x(k)] + \bar{\alpha} x_c'(k)] - s^T(k)s(k) \tag{۷۹.۳}$$

درصورت قرار دادن قانون کنترلی (۷۳.۳) در (۷۹.۳) پس از ساده سازی خواهیم داشت:

$$\Delta V_s(k) = \Gamma^T \Gamma - s^T(k)s(k) \\ \Gamma = \frac{1}{1-\bar{\alpha}}[1-q(s_c(k))]s_c(k) \tag{۸۰.۳}$$

معادله (۸۰.۳) نشان می‌دهد درصورت انتخاب کردن پارامترهای کنترل‌کننده به نحوی که باعث شود $\Gamma^T \Gamma$ کوچکتر از $s^T(k)s(k)$ باشد، دیفرنس تابع لیاپانوف منفی خواهد بود:

$$\Gamma^T \Gamma < s^T(k)s(k) \quad \Rightarrow \quad \Delta V_s(k) < 0 \tag{۸۱.۳}$$

در نتیجه متغیرهای حالت سیستم به سمت سطح لغزش حرکت می‌کنند و روی آن باقی می‌مانند و نهایتاً به مبدأ می‌رسند. □

بنابراین، در صورت اقناع شدن شرط قضیه ۱۰.۳ با بکارگیری کنترل‌کننده $u(k)$ از رابطه (۷۳.۳)، متغیرهای حالت سیستم (۵۱.۳) به سمت سطح لغزش $S_c(k)$ حرکت می‌کنند، و پس از قرار گرفتن روی سطح لغزش روی آن نگه داشته می‌شوند و به سمت مبدأ سر می‌خورند.

۳.۴.۳ جمع بندی

در این بخش به منظور ساده سازی روابط لازم برای اجرای کنترل‌کننده جمع آوری شده‌اند. به منظور بررسی عملکرد کنترل‌کننده و مشاهده‌کننده در حضور عدم قطعیت مدل، و ساده‌تر شدن معادلات آن‌ها به هدف کاهش هزینه محاسبات، مدل خطی به فرم کلی (۸۲.۳) برای طراحی کنترل‌کننده و مشاهده‌کننده استفاده می‌شود.

بنابراین هدف طراحی کنترل‌کننده برای سیستم خطی زیر است:

$$\begin{cases} \dot{x}_n = Ax + B(u + d(t)) \\ y = Cx \end{cases} \tag{۸۲.۳}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \dots & a_n \end{bmatrix} \quad , \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ b \end{bmatrix} \quad , \quad C = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \tag{۸۳.۳}$$

که می‌توان به فرم رابطه (۵۰.۳) بازنویسی کرد:

$$\begin{cases} \dot{x}_i = x_{i+1} \qquad\qquad i = 1, 2, \dots, n-1 \\ \dot{x}_n = \vec{a} \cdot x + b(u + d(t)) \\ y = x_1 \end{cases} \tag{۸۴.۳}$$

$$\vec{a} = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_n \end{bmatrix}$$

و رابطه (۸۶.۳) مدل (۸۲.۳) را در حوزه زمان گسسته با نرخ نمونه برداری $T_s = h$ مطابق رابطه (۵۱.۳) با استفاده از معادلات (۸۵.۳) نشان می‌دهد.

$$F = e^{Ah} \tag{آ۸۵.۳}$$

$$G = \int_0^h e^{At} B \, dt \tag{ب۸۵.۳}$$

$$\begin{cases} x(k+1) = Fx(k) + G\big(u(k - \hat{\tau}) + d(k)\big), \\ y(k) = Cx(k) \end{cases} \tag{۸۶.۳}$$

به این ترتیب تخمین اغتشاش با استفاده از روابط (۴۲.۳) ، (۴۳.۳) و (۴۴.۳) بدست می‌آید:

$$
\begin{aligned}
&s_o = z - x_n = \int_0^t \dot{z}\,dt - x_n \\
&\dot{z} = -ks_o - \varepsilon s_o^{p_o/q_o} - \beta\,\mathrm{sign}\,(s_o) - |\vec{a}\cdot x|\,\mathrm{sign}\,(s_o) + bu \\
&\hat{d}(t) = \frac{1}{b}\Big(- ks_o - \varepsilon s_o^{p_o/q_o} - \beta\,\mathrm{sign}\,(s_o) - |\vec{a}\cdot x|\,\mathrm{sign}\,(s_o) - \vec{a}\cdot x \Big)
\end{aligned} \tag{۸۷.۳}
$$

و با استفاده از (۶۵.۳) و (۷۳.۳) خروجی کنترل‌کننده محاسبه می‌شود.

$$
\begin{aligned}
&\varsigma = \frac{\hat{\tau}_{sc}}{1+\hat{\tau}_{sc}} \\
&x_c'(k) = C_s x(k) - \varsigma C_s x(k-1) \\
&s_c(k) = (1-\bar{\alpha})x_c'(k) + \bar{\alpha}x_c'(k-1) \\
&\quad H = (1-\bar{\alpha})(C_sF) \quad , \quad I = \varsigma(1-\bar{\alpha})C_s \quad , \quad K = \bar{\alpha}C_s \\
&\quad L = \varsigma\bar{\alpha}C_s \quad , \quad J = [1 - q(s_c(k))] \quad , \quad q[s(k)] = \beta'/(\beta' + |s_c(k)|) \\
&u(k) = -(C_sG)^{-1}\frac{1}{1-\bar{\alpha}}\big[Hx(k) - Ix(k) + Kx(k) - Lx(k-1) - J\big] - \hat{d}(kh)
\end{aligned} \tag{۸۸.۳}
$$

کنترل‌کننده هر h ثانیه اعمال می‌شود درحالی که تخمین اغتشاش بی‌درنگ محاسبه و اعمال می‌شود.

فصل ۴

شبیه‌سازی

۱.۴ مقدمه

در این فصل شرایط، جزییات و نتایج شبیه‌سازی‌های اجرا شده برای بررسی صحت مدل و عملکرد الگوریتم کنترلی ارائه شده در فصل ۳ بیان خواهد شد. به این منظور ابتدا مقادیر عددی پارامترهای هندسی سیستم فرمان مطابق بخش ۲.۳ انتخاب می‌شوند، سپس، با استفاده از آن‌ها یک مدل نرم‌افزاری به عنوان مدل مرجع ساخته می‌شود و برای اعتبار سنجی مدلِ بدست آمده در بخش ۳.۳ ، شبیه‌سازی عددی اجرا می‌شود. پس از اطمینان از صحت مدل، برای بررسی عملکرد کنترل‌کننده سیستم حلقه بسته در برنامه کامپیوتری پایتون به نحوی نوشته می‌شود که بتوان تأخیر شبکه و اتلاف بسته را به صورت تصادفی شبیه‌سازی کرد. همچنین، با اعمال اغتشاش به سیستم و شبیه‌سازی عیب در عملگر قابلیت تحمل‌پذیری کنترل‌کننده در برابر عیب را بررسی می‌کنیم. در آخر با استفاده از نرم افزار کارسیم [1] و سیمولینک دینامیک خوردو و نیروهای بین تایرهای خودرو و زمین شبیه‌سازی می‌شود. به این ترتیب اغتشاش وارد شده به سیستم فرمان مشابه مقادیر واقعی خواهد بود و عملکرد کنترل‌کننده در شرایط واقعی تر بررسی می‌شود.

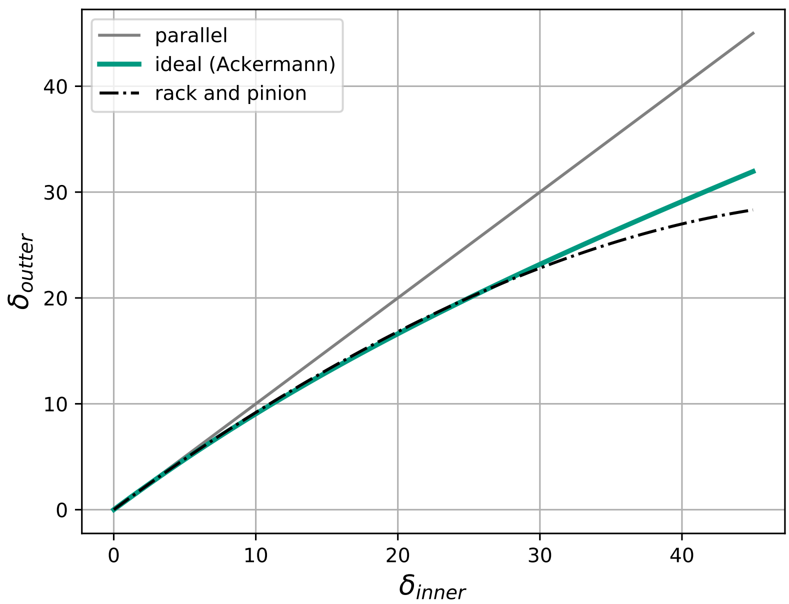
[1] CarSim

۲.۴ پارامترهای هندسی سیستم فرمان

به کمک برنامه کامپیوتری که در بخش ۲.۳ معرفی شد پارامترهای زیر برای هندسه سیستم فرمان طراحی شدند:

جدول ۱.۴: مقادیر مربوط به هندسه سیستم فرمان

پارامتر	مقدار	واحد	پارامتر	مقدار	واحد
W	2.65	m	L_1	14	cm
D	1.6	m	L_2	50	cm
d	20	cm	L_3	52.47	cm
α	17.5	$degree$			



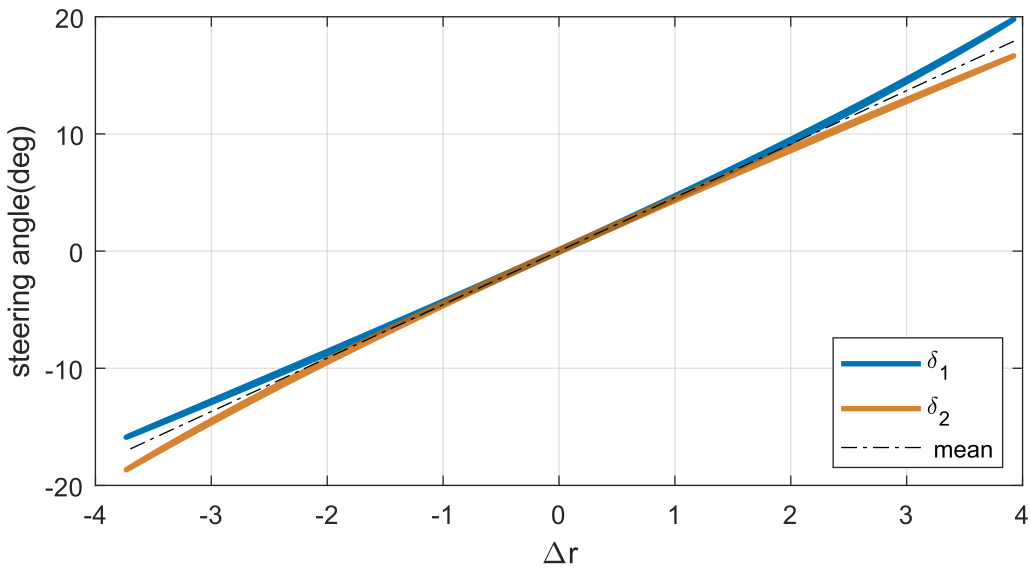
شکل ۱.۴: مقایسه مکانیزم رک و پینیون طراحی شده با حالت ایده آل

نمودار ۱.۴ رفتار سیستم فرمان رک و پینیون طراحی شده با مقادیر جدول ۱.۴ را نسبت به حالت ایده‌آل نشان می‌دهد. با توجه به نمودار، تا ۳۰ درجه زاویه فرمان برای چرخ داخل پیچ مکانیسم فرمان طراحی شده رفتار

مطلوبی دارد. همچنین، از آنجایی که در واقعیت زوایای بالاتر از ۳۰ درجه در سرعت خیلی کم قابل دستیابی هستند سایش محدودی در تایرها ایجاد می‌کنند. درنتیجه عدم تطابق نمودار با حالت ایده‌آل در زوایای بالاتر ۳۰ درجه مشکلی ایجاد نخواهد کرد.

نمودار شکل ۲.۴ زاویه فرمان هرچرخ (δ_1, δ_2) را برحسب جابجایی رک (Δr) نشان می‌دهد. با توجه به نمودار، شیب خطچین تقریب نسبت زاویه فرمان به جابه‌جایی رک را نشان می‌دهد که برابر است با:

$$\frac{\delta_{mean}}{\Delta r} = 4.56°/cm = 7.97 rad/m \tag{۱.۴}$$



شکل ۲.۴: زاویه فرمان برحسب جابجایی رک

با توجه به نمودار فوق و رابطه (۱.۴) به ازای هر سانتیمتر جابجا شدن رک، زاویه فرمان چرخ‌ها بطور میانگین 4.56 درجه تغییر می‌کند، و این میزان برای هر دو چرخ چپ و راست حول مبدأ تقریباً یکسان است.

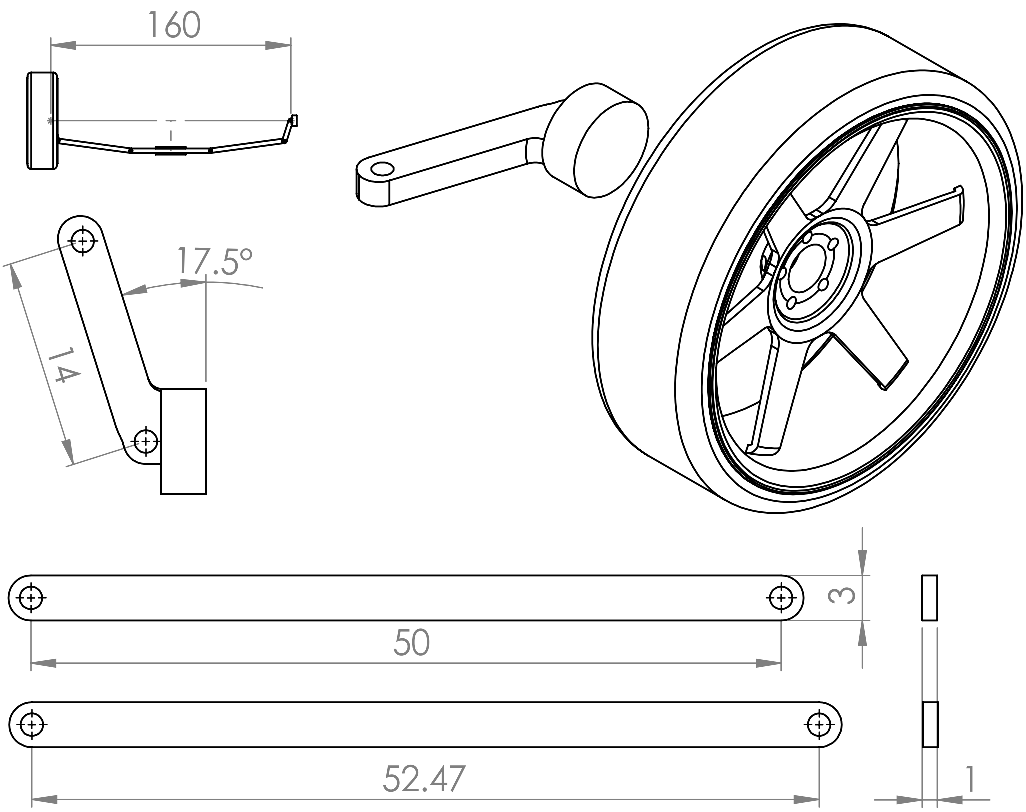
۳.۴ اعتبار سنجی مدل سیستم فرمان

هدف این بخش مقایسه مدل ریاضی سیستم فرمان که در بخش ۳.۳ ارائه شد، با مدل مرجع به منظور اطمینان از صحت آن است. در ادامه به ساخت مدل نرم‌افزاری پرداخته و سپس خواص جرمی قطعات مکانیسم ساخته

شده را جهت وارد کردن در مدل ریاضی ،استخراج می‌کنیم. در آخر این بخش نتایج مقایسه دو مدل ارائه شده است.

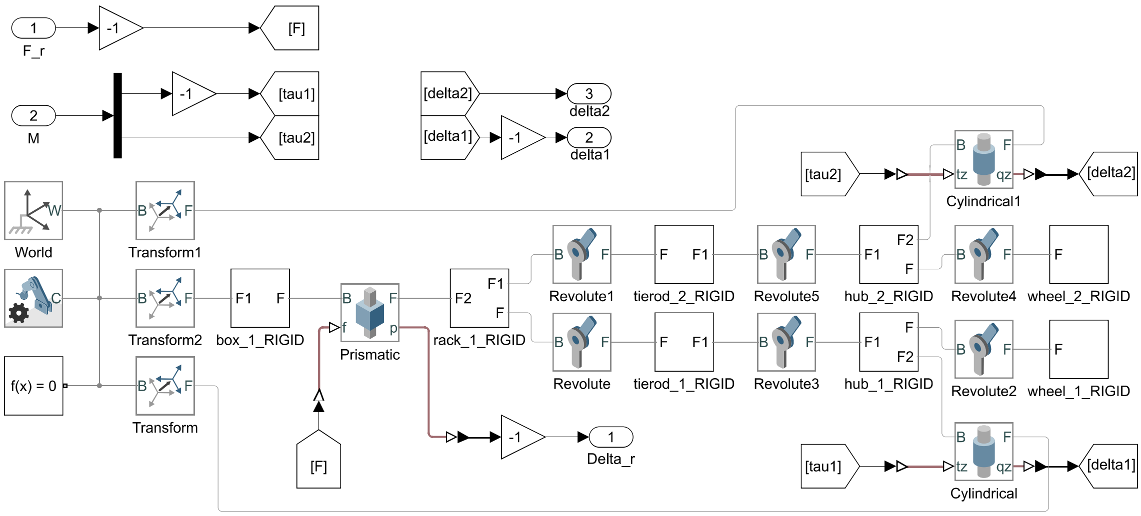
۴.۳.۱ ساخت مدل نرم‌افزاری سیستم فرمان

با توجه به مقادیر جدول ۴.۱ مدل اسمبلی [1] سیستم فرمان مطابق شکل زیر در نرم‌افزار سالیدورکس [2] ساخته شد و فولاد با چگالی $7.3g/cm^3$ برای بازوها انتخاب شد. سپس، از مدل اسمبلی فایل خروجی سیمولینک متلب [3] ساخته شد که پس از اضافه کردن حسگر و عملگر و اصلاح جهت مثبت حرکت آن‌ها سیستم فرمان مطابق شکل ۴.۴ در سیمولینک آماده شد.



شکل ۴.۳: نقشه قطعات مدل شده در اسمبلی سالیدورکس

[1]Assembly
[2]SolidWorks
[3]MATLAB Simulink



شکل ۴.۴: مدل سیمولینک سیستم فرمان ساخته شده از اسمبلی سالیدورکس

در مدل سیمولینک شکل فوق، F_r و M ورودی مدل، و Delta_r خروجی مدل است. همچنین، برای بررسی زاویه فرمان هر چرخ، delta۱ و delta۲ نیز به عنوان خروجی تعریف شده اند.

۲.۳.۴ مقادیر خواص جرمی سیستم فرمان

مقادیر مربوط به جرم و لختی دورانی قطعات سیستم فرمان که در مدل ریاضی (۲۷.۳) لازم است از مدل ساخته شده در سالیدورکس استخراج شده و در جدول زیر آمده است.

جدول ۲.۴: مقادیر مربوط به خواص جرمی سیستم فرمان

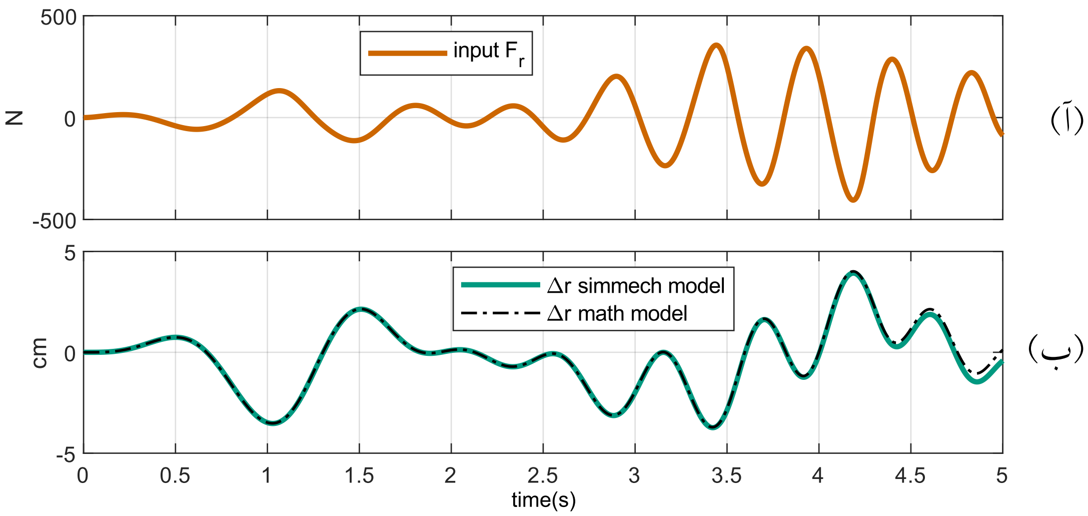
پارامتر	مقدار	واحد
m_2	1.12	kg
m_r	1.17	kg
I_1	776160	$kg\,mm^2$
I_2	24676	$kg\,mm^2$

۳.۳.۴ مقایسه مدل ریاضی ارائه شده با مدل نرم‌افزاری

در این بخش به هدف بررسی صحت معادلات (۲۷.۳) ، (۲۸.۳) ، (۲۹.۳) و (۳۰.۳) پس از جاگذاری مقادیر جدول‌های ۱.۴ و ۲.۴ در معادلات، در سه حالت مدل ریاضی و مدل نرم‌افزاری شکل ۴.۴ ، طی شبیه‌سازی در سیمولینک باهم مقایسه خواهند شد.

۱.۳.۳.۴ ورودی نیرو یکسان

درصورت اعمال نیرو مطابق نمودار (آ) در شکل ۵.۴ به هر دو مدل، خروجی جابه‌جایی رک بصورت نمودار (ب) خواهد بود. اگرچه در $t < 4s$ این نمودار یکسان بودن خروجی مدل‌ها را نشان می‌دهد، اما این معیار خوبی برای مقایسه مدل‌ها نیست، چرا که در این روش خطا هر قدم با خطاهای قبلی جمع می‌شود و درصورت وجود اختلاف هرقدر کوچک، با گذشت زمان در نهایت خروجی مدل‌ها متفاوت خواهد شد.



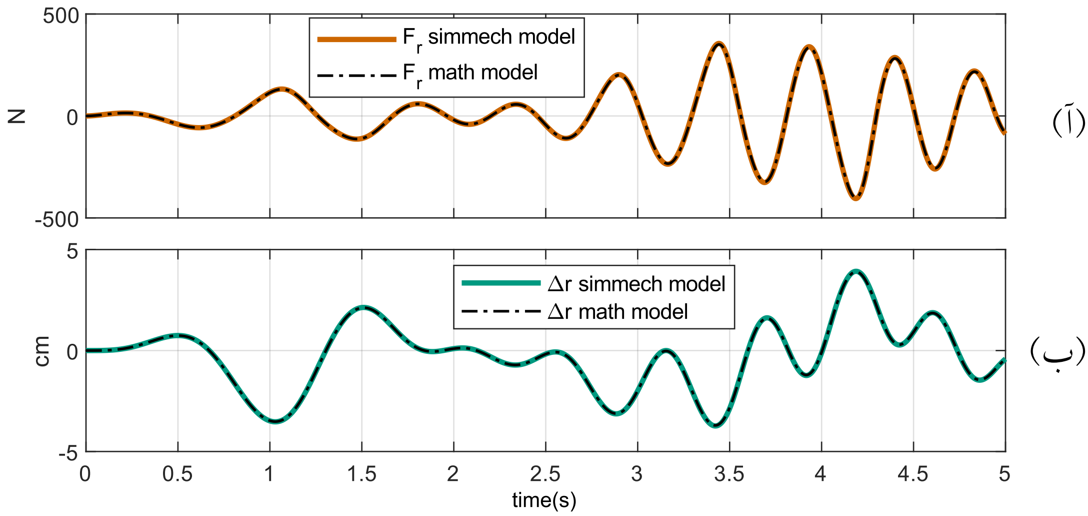
شکل ۵.۴: نمودار اعتبار سنجی حالت۱

۲.۳.۳.۴ ورودی حلقه بسته مجزا با کنترل‌کننده PD یکسان

برای برطرف کردن مشکل قسمت ۱.۳.۳.۴ دو کنترل‌کننده PD مجزا با ضرایب یکسان برای اعمال نیروی ورودی به مدل‌ها برای دنبال کردن ورودی مرجع (۲.۴) استفاده شدند. در شکل ۶.۴ نمودار (آ) نیروی ورودی و نمودار (ب) جابه‌جایی رک مدل ریاضی را با مدل مرجع مقایسه می‌کند. در این حالت گذر زمان باعث واگرا

شدن مقادیر ورودی یا خروجی نسبت به مدل مرجع نمی‌شود.

$$\Delta_{r_{ref}} = 0.046\sin(5t)\sin(t+\pi/3)\sin(t^2) \quad (۲.۴)$$

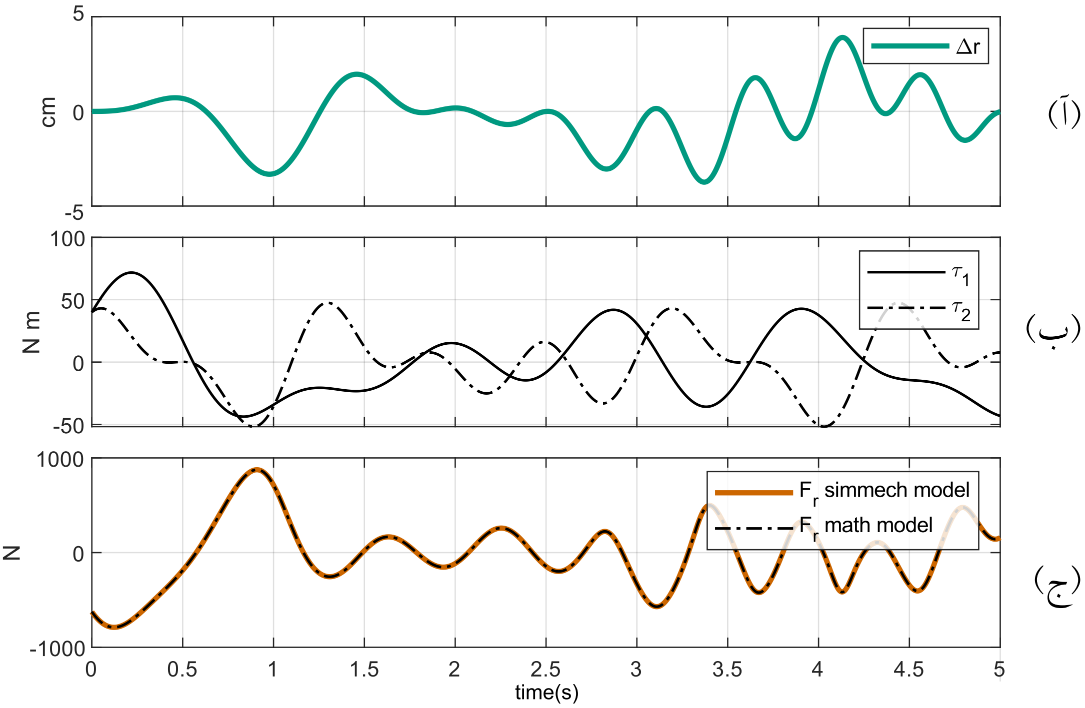


شکل ۶.۴: نمودار اعتبار سنجی حالت۲

۳.۳.۳.۴ ورودی مرجع و اغتشاش یکسان، محاسبه نیرو

برای کسب اطمینان بیشتر از صحت عملکرد مدل و بررسی اثر اغتشاش، در این بخش مدل معکوس (۲۳.۳) مورد بررسی قرار می‌گیرد. بنابراین، در کنار گشتاورهای اغتشاش، ورودی δ و مشتق اول و دوم آن متناظر $\Delta_{r_{ref}}$ با استفاده از روابط (۳۷.۳) به مدل‌ها وارد شده و خروجی نیروی رک خواهد بود. در شکل ۷.۴ نمودار (آ) ورودی مرجع ،و نمودار (ب) اغتشاش وارد شده به مدل‌ها را نشان می‌دهد. طبق نمودار (ج) نیروی محاسبه شده توسط هر دو مدل یکسان است.

با توجه به نتایج شبیه‌سازی‌های انجام شده که در شکل ۶.۴ و ۷.۴ آمده است، می‌توان نتیجه گرفت مدل ریاضی ارائه شده معتبر است. بنابراین در ادامه برای شبیه‌سازی سیستم فرمان از مدل (۲۷.۳) استفاده خواهد شد.



شکل ۷.۴: نمودار اعتبار سنجی حالت۳

۴.۴ عملکرد کنترل کننده

در این بخش با شبیه‌سازی سیستم حلقه بسته شکل ۸.۳ در سه حالت مختلف عملکرد کنترل‌کننده ارائه شده بررسی می‌شود. برای طراحی مشاهده‌کننده اغتشاش و کنترل‌کننده مدل تقریبی سیستم فرمان به فرم (۸۴.۳) و (۸۶.۳) لازم است. با استفاده از جعبه‌ابزار شناسایی سیستم متلب [1] مدل فضای حالت زیر برای ناحیه خطی سیستم فرمان (۲۷.۳) ($|\Delta_r| \leq 1cm$) بدست آمد:

$$\begin{cases} \dot{x}_n = \begin{bmatrix} -0.001588 & 4.256823 \\ -0.002479 & -0.010367 \end{bmatrix} x + \begin{bmatrix} 0.004649 \\ 0.019889 \end{bmatrix} u \\ y = \begin{bmatrix} 0.118814 & -0.027497 \end{bmatrix} x \end{cases} \tag{۳.۴}$$

[1]System Ident Toolbox

که $x = [\,\Delta_r\,,\,\dot{\Delta}_r]^T$ و $u = F_r$. درصورت تبدیل (۳.۴) به فرم کانونیکال داریم:

$$\begin{cases} \dot{x}_n = Ax + Bu \\ \\ y = Cx \end{cases} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \vec{a} \cdot x + bu \\ y = x_1 \end{cases} \tag{آ۴.۴}$$

$$A = \begin{bmatrix} 0 & 1 \\ -0.010571 & -0.011945 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.010064 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{ب۴.۴}$$

$$\vec{a} = \begin{bmatrix} -0.010571, -0.011945 \end{bmatrix}, b = 0.010064 \tag{ج۴.۴}$$

ملاحظه ۱.۴. با محاسبه ماتریس‌های کنترل‌پذیری و رویت‌پذیری و یا اجرای دستور *ctrb(A,B)* و *obsv(A,C)* در متلب، می‌توان نشان داد این ماتریس‌ها رتبه کامل بوده و مدل (۴.۴) کنترل‌پذیر و رویت‌پذیر است.

با اعمال (۸۵.۳) به رابطه (۴.۴) مدل فضای حالت زمان گسسته به ازای نرخ نمونه برداری $h = 10\ ms$ برابر است با:

$$\begin{cases} x(k+1) = Fx(k) + Gu \\ \\ y(k) = Cx(k) \end{cases} \tag{آ۵.۴}$$

$$F = \begin{bmatrix} 1 & 0.01 \\ -0.000106 & 0.99988 \end{bmatrix}, G = \begin{bmatrix} 0.000006 \\ 0.000101 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{ب۵.۴}$$

با توجه به بیشینه‌ی اغتشاش، پس از سعی و خطا مقادیر انتخاب شده برای مشاهده‌کننده و کنترل‌کننده در جدول ۳.۴ آمده است.

ملاحظه ۲.۴. در شبیه‌سازی‌هایی که در ادامه آمده است از روابط (۴.۴) و (۵.۴) به عنوان تخمین مدل سیستم فرمان برای طراحی مشاهده‌کننده اغتشاش و کنترل‌کننده استفاده شده است، و مدل ریاضی محاسبه شده در بخش ۵.۳.۳ برای شبیه‌سازی سیستم فرمان استفاده شده است.

ملاحظه ۳.۴. رابطه (۶.۴) اغتشاش معادل وارد شده به سیستم از دید مشاهده‌کننده اغتشاش را بر حسب τ_1 و τ_2 بیان می‌کند. از این رابطه برای بررسی صحت تخمین اغتشاش در نمودارهای این فصل استفاده شده است.

$$d(t) = \frac{\dot{\delta}_1\tau_1 + \dot{\delta}_2\tau_2}{\dot{\Delta}_r} \quad (۶.۴)$$

جدول ۳.۴: مقادیر انتخاب شده برای پارامترهای مشاهده‌کننده و کنترل‌کننده

پارامتر	مقدار	پارامتر	مقدار
C_s	$[-10\,,\ -1]$	k	100
β'	10	ϵ	10
τ_{sc}	0.01	p_o	13
$\bar{\alpha}$	0.1	q_o	15
$\bar{\beta}$	0.1	β	5000

ملاحظه ۴.۴. مشاهده‌کننده و کنترل‌کننده استفاده شده در شبیه‌سازی‌ها با جاگذاری روابط (۴.۴ج) و (۵.۴ب) ، و مقادیر مشخص شده برای پارامترها در جدول ۳.۴ در معادلات (۸۷.۳) و (۸۸.۳) بدست می‌آید.

۱.۴.۴ شبیه‌سازی۱ - عملکرد درحضور اغتشاش و عدم‌قطعیت

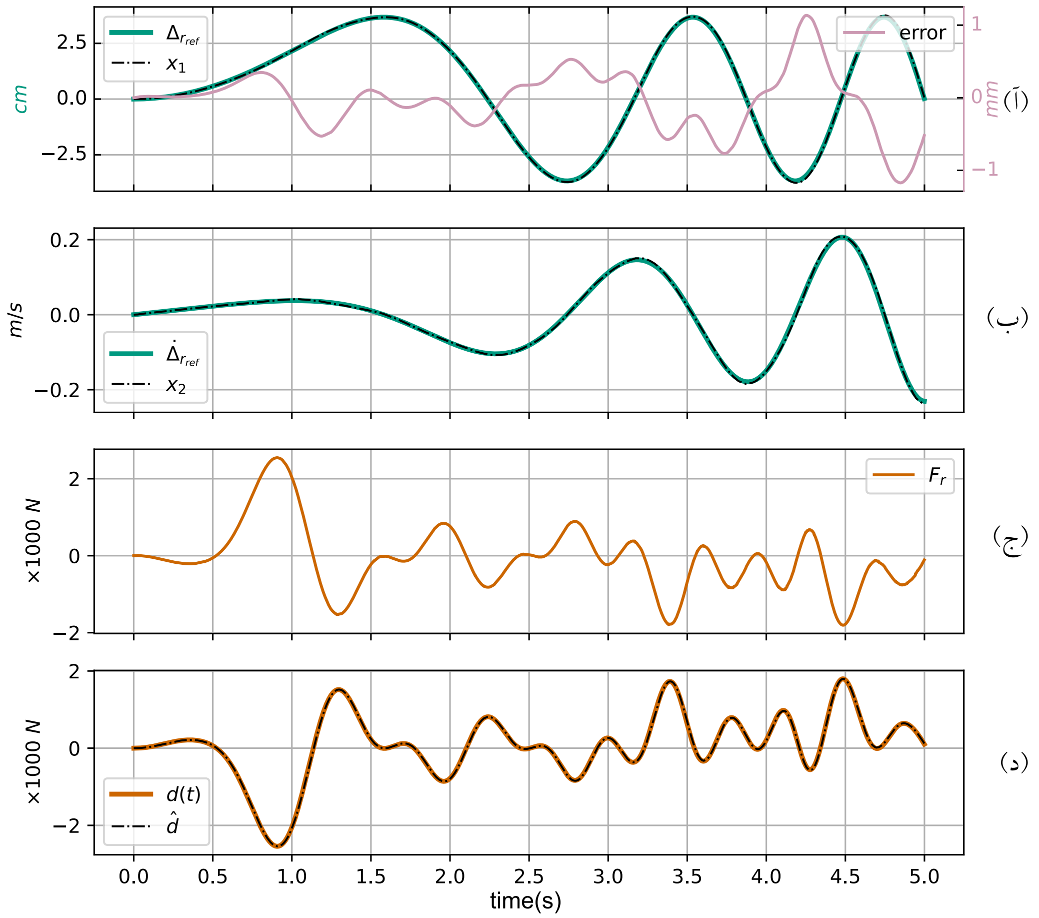
از آنجایی که معادله (۴.۴) برای ناحیه خطی سیستم فرمان صادق است، برای زاویه‌های $|\Delta_r| > 1cm$ بدلیل غیرخطی بودن معادلات (۲۴.۳) بین این دو سیستم اختلاف به وجود می‌آید. این اختلاف به وجود آمده به عنوان عدم قطعیت در سیستم خطی سازی شده در نظر گرفته می‌شود. همچنین، در این شبیه‌سازی گشتاور اغتشاش مطابق زیر به سیستم فرمان اعمال می‌شود:

$$\tau_1 = 100\sin(t^2) \times \big(\cos(3t) + \sin(5t) + \sin(7t)\big) \quad (۷.۴آ)$$

$$\tau_2 = 100\sin(t^2) \times \big(\cos(10t) + \cos(4t) + \sin(6t)\big) \quad (۷.۴ب)$$

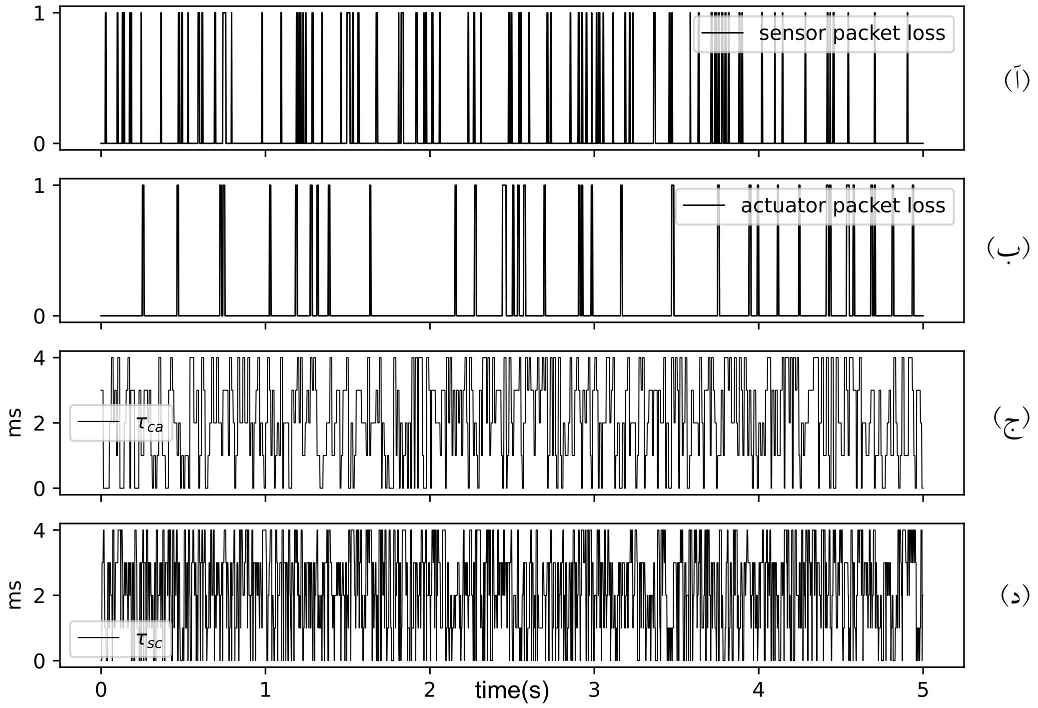
شکل ۸.۴ نتیجه‌ی ۵ ثانیه شبیه‌سازی سیستم حلقه بسته شکل ۸.۳ را به ازای ورودی مرجع (۸.۴) و تأخیر و اتلاف بسته شبکه شبیه‌سازی شده مطابق شکل ۹.۴ نشان می‌دهد.

$$\Delta_{r_{ref}} = 0.37 \sin\left(\frac{\pi}{5}t^2\right) \qquad (۸.۴)$$



شکل ۸.۴: نتایج شبیه‌سازی۱ - عملکرد در حضور اغتشاش و عدم قطعیت

باتوجه به نمودار (آ) و (ب) در شکل ۸.۴، کنترل‌کننده ارائه شده با عملکرد خوبی در حضور تأخیرهای نمایش داده شده در شکل ۹.۴ مرجع کنترلی را ردیابی کرده است. همچنین، با توجه به نمودار (ج) ورودی کنترلی دچار نوسان نشده است. نمودار (د) نیز عملکرد مشاهده‌کننده اغتشاش را در تخمین اغتشاش وارد شده و عدم قطعیت سیستم نشان می‌دهد. و از آنجایی که اغتشاش وارد شده نسبتا بسیار بزرگتر از تأثیر عدم قطعیت است نمودار $\hat{d}(t)$ تقریبا منطبق بر $d(t)$ است.

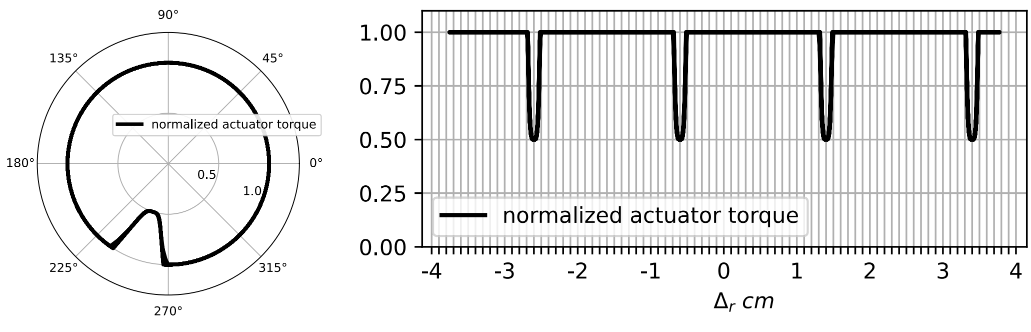


شکل ۹.۴: تأخیر و اتلاف بسته در شبیه‌سازی ۱ و ۲

شکل فوق میزان تأخیر و لحظه اتلاف بسته شبکه را بین حسگر و کنترل‌کننده ، و بین کنترل‌کننده و عملگر نشان می‌دهد؛ لازم به ذکر است این میزان تأخیر و اتلاف بسته اغراق آمیز بوده و به هدف بررسی عملکرد کنترل‌کننده در شرایط دشوار به این صورت در نظر گرفته شده است.

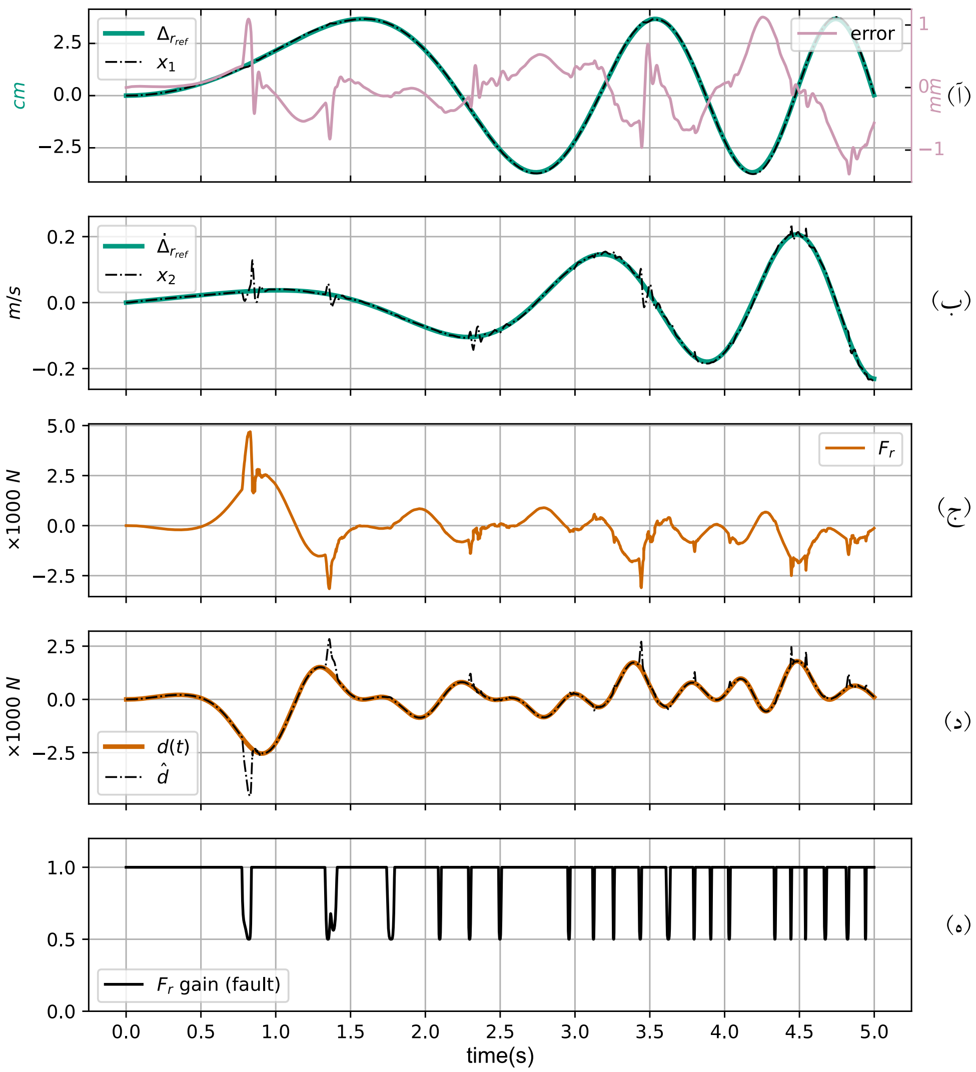
۲.۴.۴ شبیه‌سازی۲ - عملکرد تشخیص و جبران عیب عملگر

در این بخش عیب عملگر مطابق شکل ۱۰.۴ شبیه‌سازی شده است. و سایر شرایط مشابه شبیه‌سازی۱ است. با توجه به نمودار شکل مذکور عملگر دچار عیبی شده است که باعث می‌شود گشتاور پینیون و به تبعیت از آن ورودی F_r در موقعیت مشخصی 50% دچار افت شود. این عیب در واقعیت می‌تواند به دلایل مختلفی از جمله سوختن یک سیم‌پیچ روتور یا عیب در یاتاقان‌های جعبه دنده به وجود آید.



(ب) (آ)

شکل ۱۰.۴: (آ) عیب عملگر نسبت به موقعیت رک (ب) عیب عملگر نسبت به موقعیت پینیون



شکل ۱۱.۴: نتایج شبیه‌سازی۲ – عملکرد تشخیص و جبران عیب

نمودار (ه) در شکل ۱۱.۴ زمان بروز عیب در عملگر را نشان می‌دهد، و با توجه به نمودارهای (د) و (ج) مشاهده‌کننده اغتشاش وجود عیب را تشخیص داده و برای جبران آن نیروی ورودی را به نحوی افزایش داده است که با وجود عیب عملگر، عملکرد ردیابی کنترل‌کننده دچار مشکل نشود. با در نظر داشتن این موضوع که میزان عیب متناسب با مقدار ورودی کنترلی است، در بعضی لحظات که نمودار (ه) بروز عیب را نشان می‌دهد، تخمین عیب در نمودار (د) به دلیل بزرگ بودن مقیاس نمایش نمودار قابل مشاهده نیست. نهایتاً، نمودار (آ) و (ب) عملکرد ردیابی مرجع را در حضور عیب نشان می‌دهند.

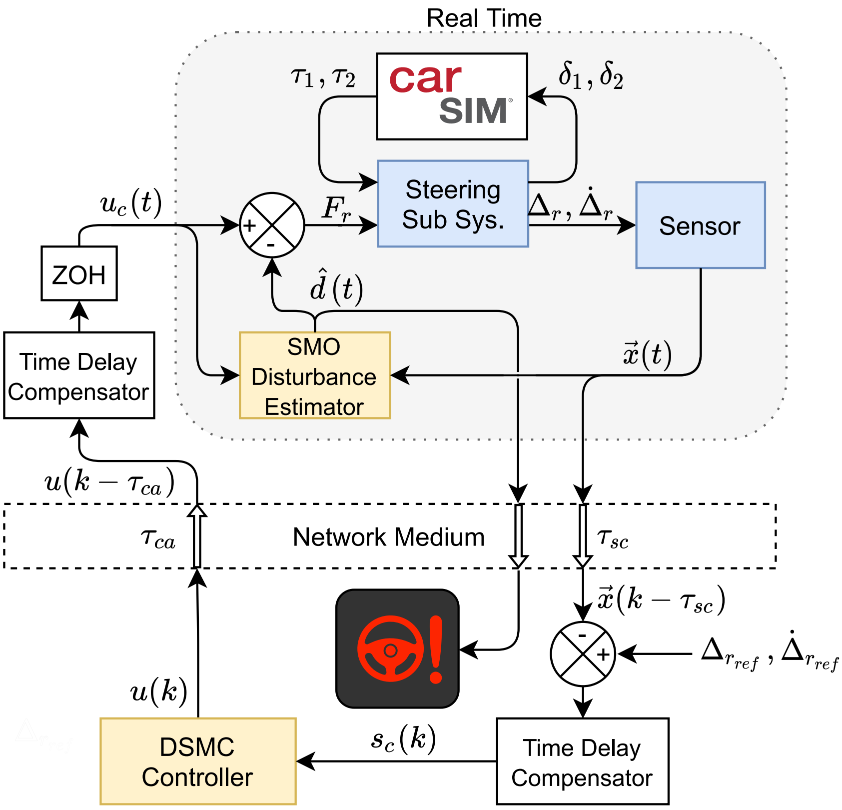
۳.۴.۴ عملکرد در شبیه‌سازی شرایط واقعی خودرو و جاده

در این بخش برای شبیه‌سازی مقادیر واقعی اغتشاش که از زمین به سیستم فرمان وارد می‌شود از نرم‌افزار شبیه‌ساز کارسیم [1] استفاده شده است. در شبیه‌سازی۳ مانور استاندارد تعویض لاین [2] و در شبیه سازی۴ رانندگی سریع در پیست مسابقه مشخص شده در شکل ۱۷.۴ برای به چالش کشیدن کنترل‌کننده در شرایط سخت و متنوع، شبیه‌سازی شده است.

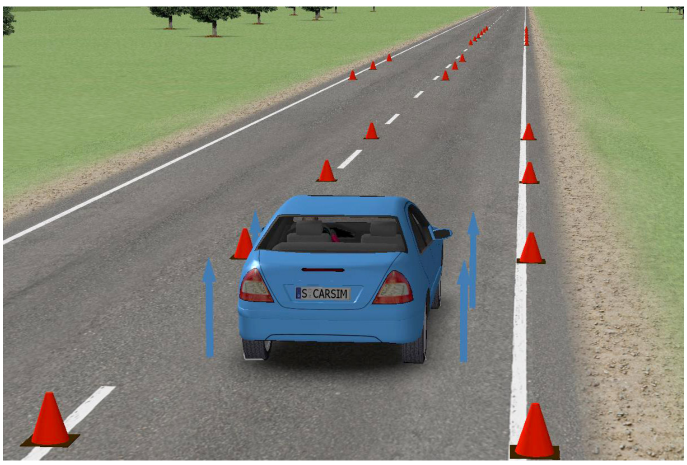
۱.۳.۴.۴ شبیه‌سازی۳ - تعویض لاین استاندارد

مسیر حرکت خودرو مطابق شکل ۱۳.۴ و نمودار (آ) ۱۵.۴ که مانور استاندارد برای بررسی فرمان پذیری خودرو است انتخاب شد. همچنین، همانطور که در شکل ۱۲.۴ آمده است از شبیه‌ساز کارسیم اغتشاش وارد شده به هر چرخ خروجی گرفته شد و در سیمولینک به مدل سیستم فرمان اعمال شد. به این ترتیب عملکرد کنترل‌کننده در شرایط نزدیک به واقعیت بررسی شده است که نتایج آن در شکل ۱۵.۴ قابل مشاهده است.

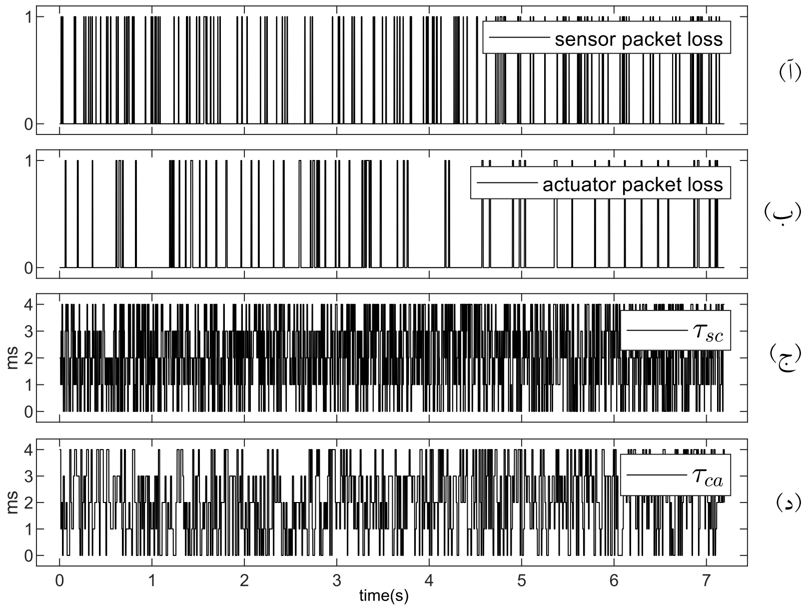
[1] CarSim
[2] ISO 3888-2: 2011. Passenger cars — Test track for a severe lane-change manoeuvre.



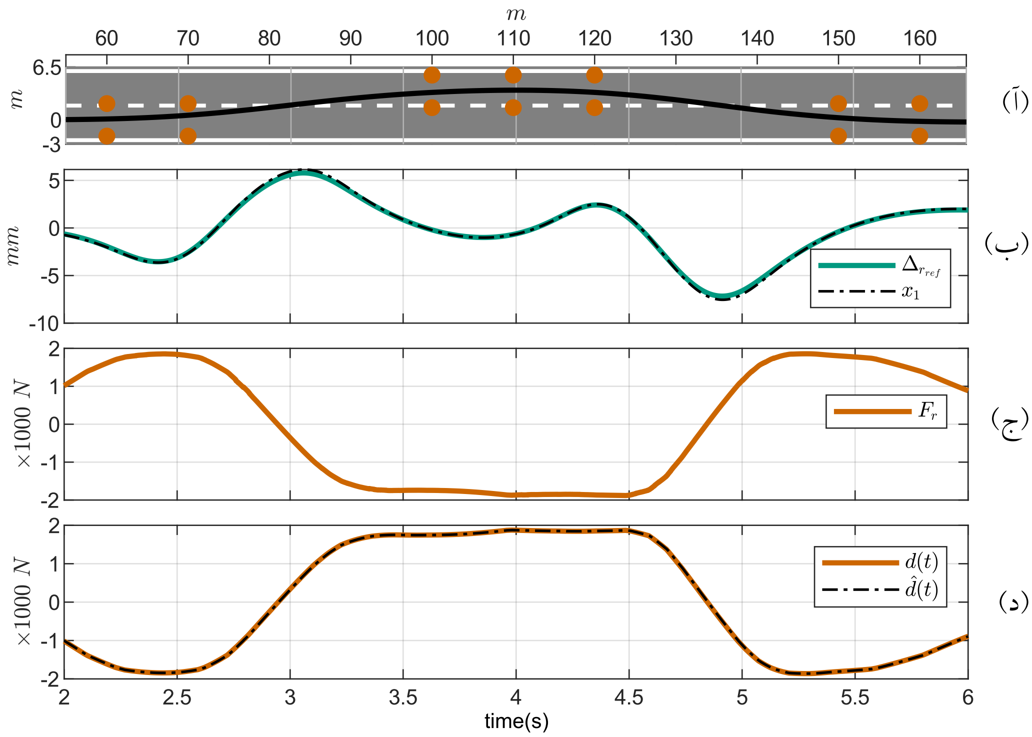
شکل ۱۲.۴: دیاگرام سیستم حلقه بسته برای شبیه‌سازی۳ و ۴



شکل ۱۳.۴: تصویر محیط کارسیم در شبیه‌سازی ۳



شکل ۱۴.۴: تأخیر و اتلاف بسته در شبیه‌سازی۳



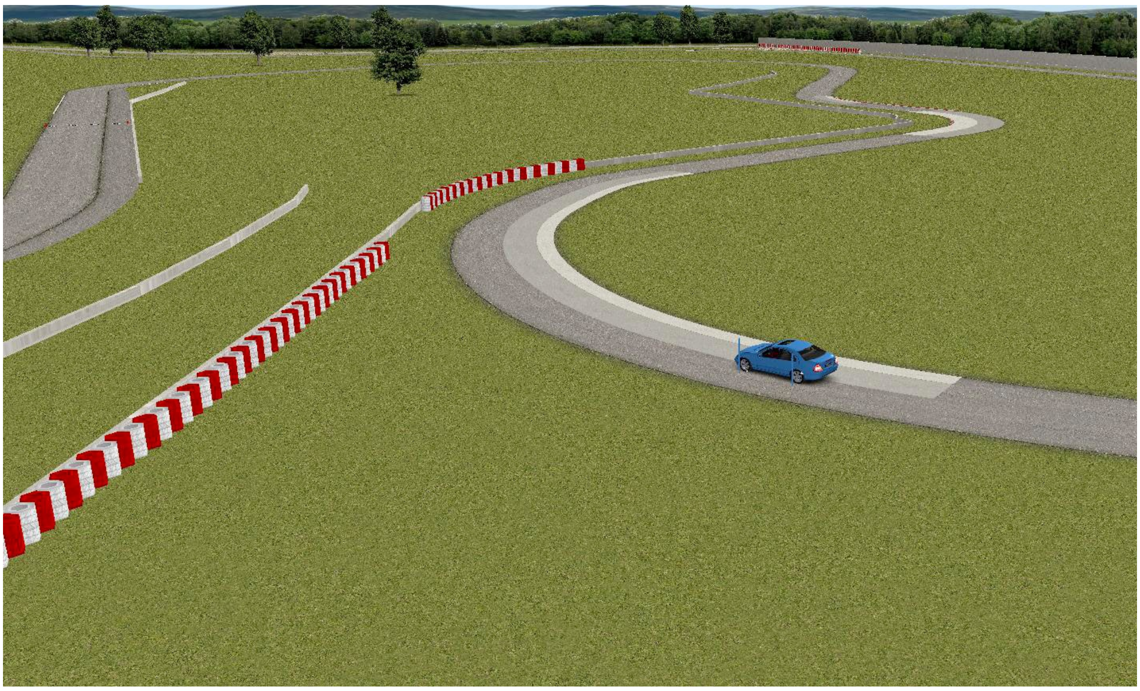
شکل ۱۵.۴: نتایج شبیه‌سازی۳ - عملکرد کنترل‌کننده در شرایط واقعی خودرو در مانور تعویض لاین

نمودار (ب) با نمایش جابجایی رک عملکرد ردیابی کنترل‌کننده، و نمودار (د) عملکرد تخمین اغتشاش را

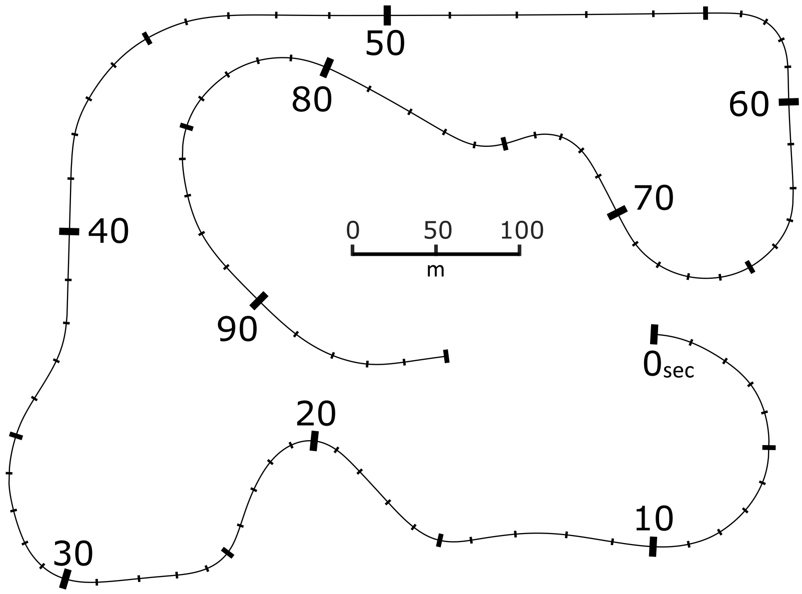
نشان می‌دهد. همچنین با توجه به نمودار (ج) ورودی کنترلی پیوسته و بدون نوسان است.

۴.۴.۳.۲ شبیه‌سازی ۴ - رانندگی سریع در پیست مسابقه

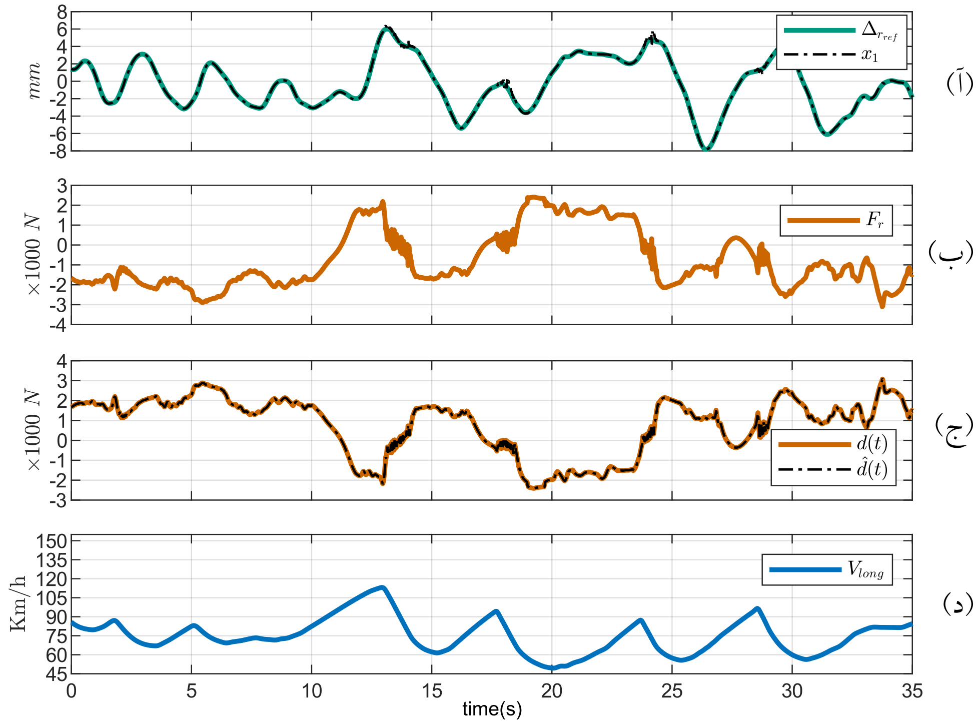
در آخرین شبیه‌سازی مسیر پیچیده‌ی مشخص شده در شکل ۴.۱۷ به منظور بررسی کردن عملکرد کنترل‌کننده برای ردیابی زاویه فرمان مطلوب در شرایط متنوع سرعت و شتاب خودرو انتخاب شده است. نمودار (آ) در شکل ۴.۱۸ موقعیت رک را نسبت به جابه‌جایی مطلوب رک، نمودار (ب) ورودی کنترلی F_r و نمودار (ج) عملکرد تخمین اغتشاش وارد شده از زمین به سیستم فرمان را نسبت به زمان نمایش می‌دهد. همچنین، در نمودار (د) سرعت خودرو نسبت به زمان نشان داده شده است. برای درک بهتر این نمودارها به زمان مشخص شده روی نمودار مسیر حرکت در شکل ۴.۱۷ توجه کنید. به منظور واضح بودن نمودارها خروجی شبیه‌سازی در سه قسمت زمانی رسم شده اند؛ شکل ۴.۱۸ مقادیر ذکر شده را از ابتدا تا لحظه‌ی ۳۵ثانیه، شکل ۴.۱۹ همان نمودارها را برای لحظات ۳۰ تا ۶۵ثانیه، و شکل ۴.۲۰ برای ۶۰ تا ۹۵ثانیه که پایان شبیه‌سازی است نمایش می‌دهد.



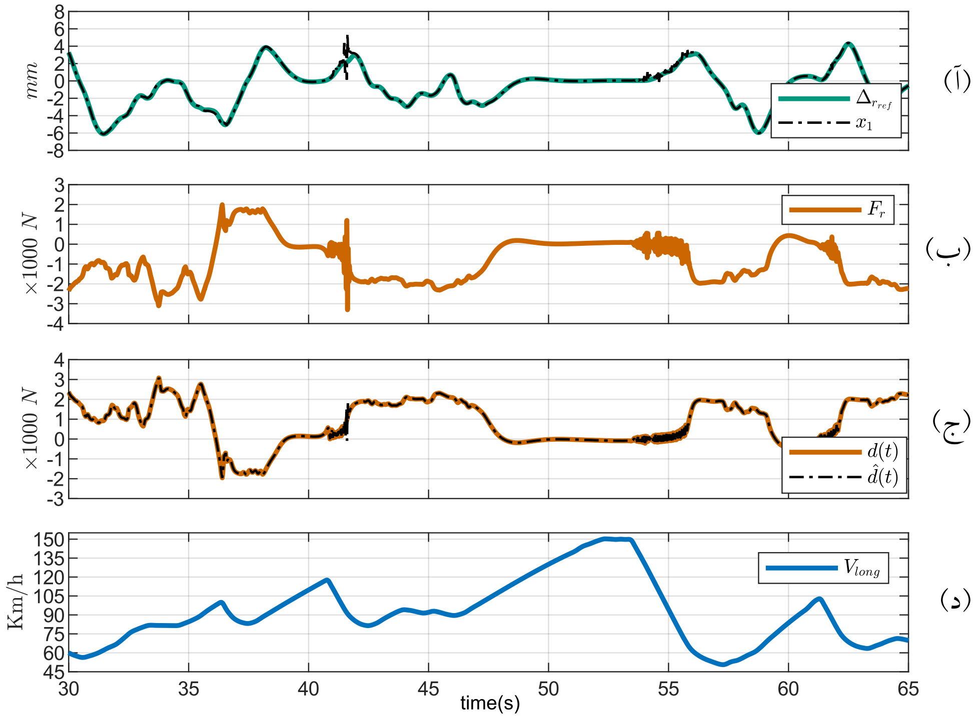
شکل ۴.۱۶: تصویر محیط کارسیم در شبیه‌سازی ۴



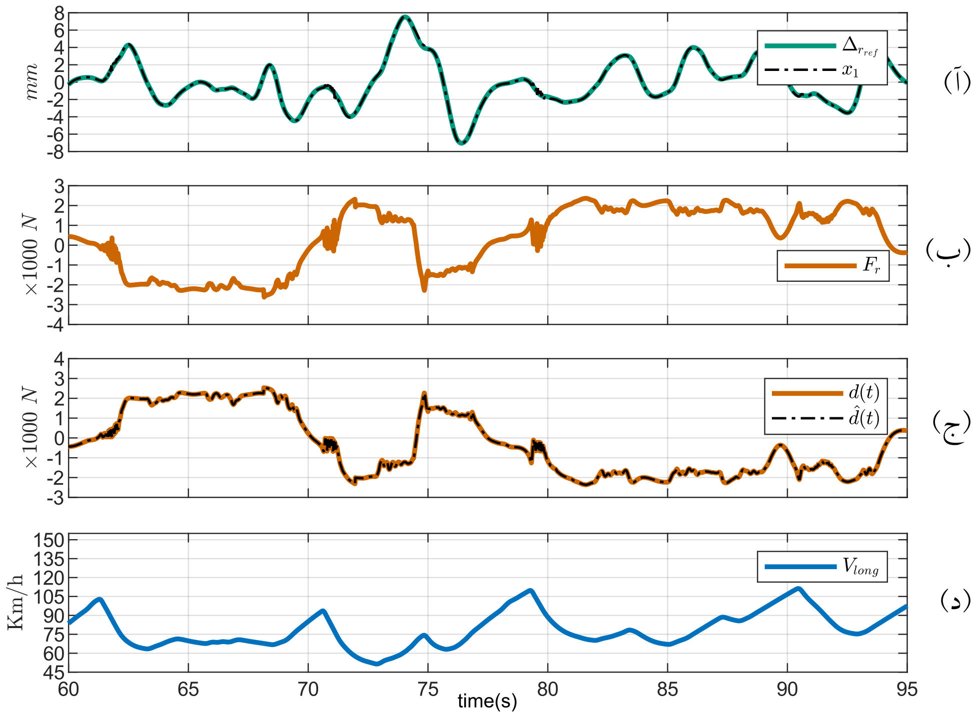
شکل ۱۷.۴: مسیر حرکت نسبت به زمان



شکل ۱۸.۴: نتایج شبیه‌سازی۴ از لحظه ۰ تا ۳۵ ثانیه



شکل ۱۹.۴: نتایج شبیه‌سازی۴ از لحظه ۳۰ تا ۶۵ ثانیه



شکل ۲۰.۴: نتایج شبیه‌سازی۴ از لحظه ۶۰ تا ۹۵ ثانیه

۵.۴ جمع بندی

در این فصل پس از اعتبارسنجی مدل ریاضی ارائه شده برای دینامیک سیستم فرمان، طی چهار شبیه‌سازی عملکرد کنترل‌کننده در شرایط مختلف بررسی شد. با توجه به نتایج شبیه‌سازی ۱، ۳ و ۴ کنترل‌کننده عملکرد خوبی در دفع اغتشاش و جبران عدم قطعیت‌های سیستم در حضور تأخیر شبکه و اتلاف بسته دارد. همچنین، شبیه‌سازی۲ مشخصاً تحمل‌پذیری کنترل‌کننده در برابر عیب عملگر را نشان داد.

فصل ۵

بحث و نتیجه‌گیری

۱.۵ جمع بندی

با توجه به اهمیت قابلیت اطمینان سیستم هدایت-با-سیم، در این پژوهش کنترل‌کننده تحمل‌پذیر عیب با قابلیت جبران عدم قطعیت مدل و دفع اغتشاشات خارجی برای سیستم هدایت-با-سیم تحت شبکه ارائه شد و با در نظر گرفتن تأخیر شبکه پایداری آن اثبات شد. همچنین، سعی شد حداکثر دقت در مدلسازی و شبیه‌سازی برای بررسی عملکرد روش ارائه شده استفاده شود.

در این تحقیق ملاحظات مربوط به طراحی سیستم فرمان بیان شد و برنامه کامپیوتری ذکر شده در پیوست آ.۱ جهت طراحی هندسه سیستم فرمان آماده شد. پس از آن معادلات غیرخطی حاکم بر دینامیک سیستم فرمان با فرض دو بعدی بودن بازو و مفصل‌ها از روش لاگرانژ و مختصات تعمیم یافته استخراج و برای حل عددی به فرم ماتریسی در برنامه کامپیوتری به زبان پایتون نوشته شد (مطابق پیوست آ.۲) . از این مدل به عنوان سیستم تحت کنترل در شبیه‌سازی‌ها استفاده شده است. همچنین، با استفاده از نرم‌افزار شناسایی سیستم متلب [1] مدل ساده‌سازی شده متناظر با ناحیه خطی سیستم فرمان به هدف استفاده در طراحی کنترل‌کننده و مشاهده‌کننده به دست آمد.

الگوریتم کنترلی ارائه شده شامل یک مشاهده‌کننده مدلغزشی زمان محدود [2] برای تخمین و جبران عیب و

[1] MATLAB System Identification Toolbox
[2] Terminal Sliding Mode Observer

اغتشاش خارجی، و یک کنترل‌کننده مدلغزشی زمان گسسته [۳] تحت شبکه برای ردیابی سیگنال مرجع است.

به منظور جبران تأخیر به وجود آمده در سیگنال‌های شبکه برای محاسبه سطح لغزش و سیگنال کنترلی از روش تخمین تیران استفاده شد و پایداری مشاهده‌کننده و کنترل‌کننده در شرایط کلی با روش لیاپانوف اثبات شد. سپس، برای شبیه‌سازی مشاهده‌کننده برنامه‌ی کامپیوتری آ.۳ و برای شبیه‌سازی کنترل‌کننده و تأخیر شبکه و اتلاف بسته برنامه‌ی آ.۴ به زبان پایتون نوشته شد.

در پایان، عملکرد الگوریتم کنترلی ارائه شده در چهار شبیه‌سازی با شرایط مختلف بررسی شد که در ادامه بحث و بررسی می‌شود.

۲.۵ نوآوری

با توجه به این که سیستم هدایت-با-سیم نسبتاً جدید است، تحقیق‌های محدودی به اثر تأخیر شبکه روی سیستم هدایت-با-سیم و کنترل تحمل‌پذیر عیب برای آن پرداخته‌اند، و تعداد زیادی از آن‌ها با فرض عدم وجود تأخیر پایداری سیستم حلقه بسته را نشان داده‌اند. یکی از نوآوری‌های این پژوهش استفاده از کنترل‌کننده ارائه شده در [۶۱] و ترکیب آن با مشاهده‌کننده اغتشاش ارائه شده در [۱۴] برای سیستم هدایت-با-سیم با در نظر گرفتن تأخیر شبکه و اتلاف بسته است.

همانطور که در دیاگرام شکل ۸.۳ آمده است، مشاهده‌کننده اغتشاش بصورت بی‌درنگ اجرا می‌شود در حالی که فرکانس اجرای دستورات کنترل‌کننده محدود به نرخ نمونه برداری شبکه است. این معماری یکی از نوآوری‌های این پایان‌نامه محسوب می‌شود که برای سیستم هدایت-با-سیم کاربردی است و باعث افزایش کارایی تخمین و جبران اغتشاش و عیب می‌شود.

۳.۵ بحث و بررسی نتایج

شبیه‌سازی‌های انجام شده در بخش ۴.۴ عملکرد الگوریتم کنترلی را مورد آزمایش قرار داد.

شبیه‌سازی۱ در بخش ۱.۴.۴ عملکرد ردیابی مرجع در حضور اغتشاش و تأخیر شبکه را بررسی کرد. همچنین،

[3] Descrete Time Sliding Mode Controller

با توجه به استفاده از مدل ساده سازی شده در طراحی کنترل‌کننده و مشاهده‌کننده اغتشاش در نواحی غیرخطی سیستم فرمان مقدار قابل توجهی عدم قطعیت در مدل اعمال شده است.

شبیه‌سازی۲ در بخش ۲.۴.۴ تحمل‌پذیری کنترل‌کننده نسبت به عیب عملگر، که یکی از عیب‌های محتمل سیستم هدایت-با-سیم است، را نشان داد. در این شبیه‌سازی فرض شده است که گشتاور عملگر جعبه فرمان در موقعیت‌هایی دچار افت می‌شود. و نشان داده شد که مشاهده‌کننده اغتشاش مقدار این عیب را به خوبی تخمین زده و با افزایش توان ورودی به عملگر آن را جبران می‌کند.

در شبیه‌سازی۳ و ۴ در بخش ۳.۴.۴ با استفاده از نرم‌افزار کارسیم [1] و سیمولینک متلب، دینامیک خودرو و جاده شبیه‌سازی شده است و گشتاور اغتشاش بدست آمده از خروجی نرم‌افزار به سیستم فرمان اعمال شد. به این ترتیب عملکرد الگوریتم کنترلی ارائه شده در شرایط واقعی خودرو بررسی شد. در شبیه‌سازی۳ مانور استاندارد تعویض لاین با سرعت ثابت و در شبیه‌سازی۴ به منظور به چالش کشیدن کنترل‌کننده در شرایط دشوار مسیر حرکت یک پیست مسابقه و با سرعت و شتاب مختلف شبیه‌سازی شد.

نتایج به دست آمده در شبیه‌سازی‌ها عملکرد مناسب الگوریتم کنترلی را با توجه به اهداف طراحی تأیید می‌کنند. بنابر نتایج، کنترل‌کننده طراحی شده قادر به ردیابی مناسب سیگنال مرجع در حضور عدم قطعیت مدل، اغتشاش، تأخیر شبکه و عیب عملگر است. لازم به ذکر است هرکدام از این موارد دارای پیش شرط‌هایی بوده و یا کنترل‌کننده تا حد مشخصی قادر به جبران آن‌ها است که در فرضیات طراحی در فصل ۳ عنوان شده است.

۴.۵ محدودیت‌ها و پیشنهادات

- در این پژوهش پارامترهای مربوط به طراحی مشاهده‌کننده و کنترل‌کننده با روش سعی و خطا انتخاب شده‌اند؛ در صورت استفاده از روش‌های بهینه‌سازی مثل الگوریتم ژنتیک پیش‌بینی می‌شود که عملکرد آن‌ها بهبود یابد.

- در این پژوهش به منظور ساده سازی روابط، عملگر جعبه فرمان ایده‌آل فرض شده است؛ درصورت استفاده از مدل مناسب برای درنظر گرفتن دینامیک عملگر شبیه‌سازی‌ها یک قدم به واقعیت نزدیکتر خواهند شد.

[1] CarSim

- در این تحقیق احتمال وقوع تأخیر و اتلاف بسته در شبکه معلوم فرض شده است. بنابراین، قبل از استفاده از این روش باید حد بالای احتمال وقوع آن‌ها مشخص شود و بر اساس آن پارامترهای کنترل‌کننده طراحی شوند.

- از آنجایی که شبیه‌سازی حس فرمان [1] مستلزم ساخت واحد غربیلک فرمان و تحقیق مفصل دیگری است در این پژوهش لحاظ نشده است، اما از نظر تئوری می‌توان تخمین اغتشاش را، که توسط مشاهده‌کننده اغتشاش در این پژوهش محاسبه می‌شود، پس از انتقال از طریق شبکه به واحد غربیلک فرمان مستقیماً برای شبیه‌سازی حس فرمان استفاده کرد. برای اطلاعات بیشتر در این مورد توصیه می‌شود به ادبیات موضوع که در فصل ۲ اشاره شد مراجعه کنید.

- همچنین، در این پایان‌نامه فقط مشاهده‌کننده اغتشاش زمان محدود است، و ردیابی مرجع بصورت زمان محدود صورت نگرفته است؛ برای توسعه‌ی الگوریتم کنترلی ارائه شده می‌توان از کنترل‌کننده زمان محدود استفاده کرد.

- در آخر این پایان‌نامه محدود به شبیه‌سازی کامپیوتری است و بسیاری از محدودیت‌های سیستم واقعی را درنظر نمی‌گیرد، پیاده‌سازی عملی این سیستم بصورت آزمایشگاهی برای اعتبارسنجی الگوریتم ارائه شده و توسعه‌ی آن برای سیستم هدایت–با–سیم بسیار کاربردی خواهد بود.

[1] Steering Feel

مراجع

[1] Akira, ITO and Hayakawa, Yoshikazu. Design of fault tolerant control system for electric vehicles with steer-by-wire and in-wheel motors. *IFAC Proceedings Volumes*, 46(21):556–561, 2013.

[2] Altby, Alexander and Majdandzic, Davor. Design and implementation of a fault-tolerant drive-by-wire system. *Chalmers University of Technology*, 2014.

[3] Anwar, Sohel and Chen, Lei. Analytical redundancy based fault tolerant control of a steer-by-wire system. In *ASME 2006 International Mechanical Engineering Congress and Exposition*, pages 297–306. American Society of Mechanical Engineers, 2006.

[4] Anwar, Sohel and Chen, Lei. An analytical redundancy-based fault detection and isolation algorithm for a road-wheel control subsystem in a steer-by-wire system. *IEEE Transactions on Vehicular Technology*, 56(5):2859–2869, 2007.

[5] Anwar, Sohel and Niu, Wei. Analytical redundancy based predictive fault tolerant control of a steer-by-wire system using nonlinear observer. In *2010 IEEE International Conference on Industrial Technology*, pages 477–482. IEEE, 2010.

[6] Anwar, Sohel and Niu, Wei. A nonlinear observer based analytical redundancy for predictive fault tolerant control of a steer-by-wire system. *Asian Journal of Control*, 16(2):321–334, 2014.

[7] Avizienis, Algirdas, Laprie, Jean-Claude, and Randell, Brian. Fundamental concepts of computer system dependability. In *Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments*, pages 1–16. Citeseer, 2001.

[8] Bartoszewicz, Andrzej and Leśniewski, Piotr. Reaching law approach to the sliding mode control of periodic review inventory systems. *IEEE Transactions on Automation Science and Engineering*, 11(3):810–817, 2014.

[9] Başlamişli, S Çağlar, Köse, İ Emre, and Anlaç, Günay. Handling stability improvement through robust active front steering and active differential control. *Vehicle System Dynamics*, 49(5):657–683, 2011.

[10] Bertacchini, Alessandro, Pavan, Paolo, Tamagnini, Luca, and Fergnani, Lorenzo. Control of brushless motor with hybrid redundancy for force feedback in steer-by-wire applications. In *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, pages 6–pp. IEEE, 2005.

[11] Blanke, Mogens, Kinnaert, Michel, Lunze, Jan, Staroswiecki, Marcel, and Schröder, J. *Diagnosis and fault-tolerant control*, volume 2. Springer, 2006.

[12] Boada, BL, Boada, MJL, and Diaz, V. Fuzzy-logic applied to yaw moment control for vehicle stability. *Vehicle System Dynamics*, 43(10):753–770, 2005.

[13] Cetin, A. E., Adli, M. A., Barkana, D. E., and Kucuk, H. Compliant control of steer-by-wire systems. In *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 636–643, July 2009.

[14] Chen, Mou, Wu, Qing-Xian, and Cui, Rong-Xin. Terminal sliding mode tracking control for a class of siso uncertain nonlinear systems. *ISA transactions*, 52(2):198–206, 2013.

[15] Dhahri, Slim, Sellami, Anis, and Hmida, Faycal Ben. Robust sensor fault detection and isolation for a steer-by-wire system based on sliding mode observer. In *2012 16th IEEE Mediterranean Electrotechnical Conference*, pages 450–454. IEEE, 2012.

[16] Ding, Nenggen and Taheri, Saied. An adaptive integrated algorithm for active front steering and direct yaw moment control based on direct lyapunov method. *Vehicle System Dynamics*, 48(10):1193–1213, 2010.

[17] El Messoussi, Wissam, Pagès, Olivier, and El Hajjaji, Ahmed. Four-wheel steering vehicle control using takagi-sugeno fuzzy models. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6. IEEE, 2007.

[18] Führer, Thomas and Schedl, Anton. The steer-by-wire prototype implementation: Realizing time triggered system design, fail silence behavior and active replication with fault-tolerance support. *SAE transactions*, pages 646–653, 1999.

[19] Gadda, Christopher D, Laws, Shad M, and Gerdes, J Christian. Generating diagnostic residuals for steer-by-wire vehicles. *IEEE transactions on control systems technology*, 15(3):529–540, 2007.

[20] Gadda, Christopher D, Yih, Paul, and Gerdes, J Christian. Incorporating a model of vehicle dynamics in a diagnostic system for steer-by-wire vehicles. In *Proceedings of AVEC*, volume 4, pages 779–784, 2004.

[21] Gadda, Christopher David. *Optimal fault-detection filter design for steer-by-wire vehicles*. Stanford University, 2009.

[22] Gao, Tianyi, Yin, Shen, Qiu, Jianbin, Gao, Huijun, and Kaynak, Okyay. A partial least squares aided intelligent model predictive control approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99):1–9, 2017.

[23] Ginsberg, Jerry H. *Advanced engineering dynamics*. Cambridge University Press, 1998.

[24] Haggag, Salem, Rosa, Aristoteles, Huang, Kevin, and Cetinkunt, Sabri. Fault tolerant real time control system for steer-by-wire electro-hydraulic systems. *Mechatronics*, 17(2-3):129–142, 2007.

[25] Härkegård, Ola. *Backstepping and control allocation with applications to flight control*. PhD thesis, Linköpings universitet, 2003.

[26] Hasan, Mohammad S and Anwar, Sohel. Sliding mode observer based predictive fault diagnosis of a steer-by-wire system. *IFAC Proceedings Volumes*, 41(2):8534–8539, 2008.

[27] Hasan, Mohammad Sharif-ul and Anwar, Sohel. Sliding mode observer and long range prediction based fault tolerant control of a steer-by-wire equipped vehicle. Technical report, SAE Technical Paper, 2008.

[28] Hashemi, Ali. Model-based system fault diagnosis utilizing adaptive threshold with application to automotive electrical systems. 2011.

[29] Hayama, Ryouhei, Higashi, Masayasu, Kawahara, Sadahiro, Nakano, Shirou, and Kumamoto, Hiromitsu. Fault-tolerant automobile steering based on diversity of steer-by-wire, braking and acceleration. *Reliability Engineering & System Safety*, 95(1):10–17, 2010.

[30] He, L, Chen, GY, and Zheng, HY. Fault tolerant control method of dual steering actuator motors for steer-by-wire system. *International Journal of Automotive Technology*, 16(6):977–987, 2015.

[31] He, L, Zong, C.-F, Tian, C.-W, Wu, R.-J, and Zhang, T.-W. Dc motor fault diagnosis and fault tolerance control method for steer-by-wire car. 41:608–612, 05 2011.

[32] He, Lei, Zong, Changfu, Chen, Shuang, and Wang, Chang. The tri-core fault-tolerant control for electronic control unit of steer-by-wire system. Technical report, SAE Technical Paper, 2011.

[33] He, Lei, Zong, Changfu, Zhao, Honghui, and Yu, Zhixin. The dual-core fault-tolerant control for electronic control unit of steer-by-wire system. In *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, volume 4, pages 436–439. IEEE, 2010.

[34] Ho, Lok Man and Ossmann, Daniel. Fault detection and isolation of vehicle dynamics sensors and actuators for an overactuated x-by-wire vehicle. In *53rd IEEE Conference on Decision and Control*, pages 6560–6566. IEEE, 2014.

[35] Hu, Zhongyi, Zhang, Fengdeng, and Wei, Zhiqiang. Research on fault tolerant strategy and reliability of steering-by-wire. *International Journal of Modeling and Optimization*, 6(2):106, 2016.

[36] Huang, Chao, Naghdy, Fazel, and Du, Haiping. Delta operator-based fault estimation and fault-tolerant model predictive control for steer-by-wire systems. *IEEE Transactions on Control Systems Technology*, 26(5):1810–1817, 2017.

[37] Huang, Chao, Naghdy, Fazel, and Du, Haiping. Fault tolerant sliding mode predictive control for uncertain steer-by-wire system. *IEEE transactions on cybernetics*, (99):1–12, 2017.

[38] Huang, Chao, Naghdy, Fazel, and Du, Haiping. Delta operator-based model predictive control with fault compensation for steer-by-wire systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99):1–16, 2018.

[39] Huang, Chao, Naghdy, Fazel, and Du, Haiping. Observer-based fault-tolerant controller for uncertain steer-by-wire systems using the delta operator. *IEEE/ASME Transactions on Mechatronics*, 23(6):2587–2598, 2018.

[40] Huang, Chao, Naghdy, Fazel, Du, Haiping, and Huang, Hailong. Fault tolerant steer-by-wire systems: An overview. *Annual Reviews in Control*, 47:98–111, 2019.

[41] Hudha, Khisbullah, Ahmad, Fauzi, Kadir, Zulkiffli Abd, Jamaluddin, Hishamuddin, et al. Pid controller with roll moment rejection for pneumatically actuated active roll control (arc) suspension system. In *PID Control, Implementation and Tuning*. IntechOpen, 2011.

[42] Im, Jae Sung, Ozaki, Fuminori, Yeu, Tae Kyeong, and Kawaji, Shigeyasu. Model-based fault detection and isolation in steer-by-wire vehicle using sliding mode observer. *Journal of mechanical science and technology*, 23(8):1991–1999, 2009.

[43] Isermann, Rolf. Mechatronic systems—innovative products with embedded control. *Control Engineering Practice*, 16(1):14–29, 2008.

[44] Ito, Akira and Hayakawa, Yoshikazu. Practical fault-tolerant control to protect steer-by-wire systems against sensor faults. In *2015 IEEE Conference on Control Applications (CCA)*, pages 1895–1900. IEEE, 2015.

[45] Jazar, Reza N. *Vehicle dynamics: theory and application*. Springer, 2017.

[46] Jiang, Jin and Yu, Xiang. Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in control*, 36(1):60–72, 2012.

[47] Ke, Fan, Li, Zhijun, Xiao, Hanzhen, and Zhang, Xuebo. Visual servoing of constrained mobile robots based on model predictive control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1428–1438, 2016.

[48] Kheirandish, Azadeh, Husain, A Rashid, Kazemi, M Saeed, Gatavi, Ehsan, and Ahmad, M Noh. Robust fault detection and isolation of steer by wire system under various class of fault and system uncertainties. In *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, pages 1–4. IEEE, 2011.

[49] Lanigan, Patrick E, Kavulya, Soila, Narasimhan, Priya, Fuhrman, Thomas E, and Salman, Mutasim A. Diagnosis in automotive systems: A survey. *Last accessed Sept*, 10:2011, 2011.

[50] Li, Chenfeng, Li, Hui, Chen, Yuzhong, Dong, Honglei, Zhao, Xun, and Xiao, Lingyun. Model-based sensor fault detection and isolation method for a vehicle dynamics control system. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 231(2):147–160, 2017.

[51] Mammar, Said and Koenig, Damien. Vehicle handling improvement by active steering. *Vehicle system dynamics*, 38(3):211–242, 2002.

[52] Masrur, M Abul. System level reliability issues and their enhancement in drive-by-wire (dbw) systems. *Fault Tolerant Drive by Wire Systems: Impact on Vehicle Safety and Reliability*, page 29, 2012.

[53] Mihály, András and Gáspár, Péter. Reconfgurable fault-tolerant control of in-wheel electric vehicles with steering system failure. *IFAC-PapersOnLine*, 48(26):49–54, 2015.

[54] Moon, SW, Ji, YK, Huh, KS, Cho, DG, and Park, JH. A method of fault diagnosis for steer-bywire system's sensor using analytical redundancy. In *Proceedings of Korean Society of Automotive Engineers Conference*, pages 442–447, 2005.

[55] Onoda, Yuichi, Onuma, Yutaka, Goto, Takeshi, and Sugitani, Tatsuo. Design concept and advantages of steer-by-wire system. Technical report, SAE Technical Paper, 2008.

[56] Pimentel, Juan R. An architecture for a safety-critical steer-by-wire system. Technical report, SAE Technical Paper, 2004.

[57] Pinello, Claudio, Carloni, Luca P, and Sangiovanni-Vincentelli, Alberto L. Fault-tolerant distributed deployment of embedded control software. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(5):906–919, 2008.

[58] Pisu, Pierluigi, Serrani, Andrea, You, Song, and Jalics, Laci. Adaptive threshold based diagnostics for steer-by-wire systems. *Journal of dynamic systems, measurement, and control*, 128(2):428–435, 2006.

[59] Segel, Leonard. Theoretical prediction and experimental substantiation of the response of the automobile to steering control. *Proceedings of the Institution of Mechanical Engineers: Automobile Division*, 10(1):310–330, 1956.

[60] Setiawan, Joga Dharma, Safarudin, Mochamad, and Singh, Amrik. Modeling, simulation and validation of 14 dof full vehicle model. In *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009*, pages 1–6. IEEE, 2009.

[61] Shah, Dipesh and Mehta, Axay. Discrete-time sliding mode controller subject to real-time fractional delays and packet losses for networked control system. *International Journal of Control, Automation and Systems*, 15(6):2690–2703, 2017.

[62] Song, De-yu, Li, Qiang, Zou, Feng-lou, and Yuan, Bin. Fault-tolerant control architecture for steering-by-wire system. In *2008 Second International Symposium on Intelligent Information Technology Application*, volume 1, pages 677–681. IEEE, 2008.

[63] Sreedhar, Rajiv, Fernandez, B, and Masada, GY. Robust fault detection in nonlinear systems using sliding mode observers. In *Proceedings of IEEE International Conference on Control and Applications*, pages 715–721. IEEE, 1993.

[64] TIAN, Cheng-wei, ZONG, Chang-fu, WANG, Xiang, JIANG, Guo-bin, and HE, Lei. Sensor fault tolerance control method for steer-by-wire car [j]. *Journal of Jilin University (Engineering and Technology Edition)*, 1, 2010.

[65] Wada, Nobutaka, Fujii, Kosuke, and Saeki, Masami. Reconfigurable fault-tolerant controller synthesis for a steer-by-wire vehicle using independently driven wheels. *Vehicle System Dynamics*, 51(9):1438–1465, 2013.

[66] Wang, Rongrong and Wang, Junmin. In-wheel motor fault diagnosis for electric ground vehicles. In *ASME 2010 Dynamic Systems and Control Conference*, pages 133–140. American Society of Mechanical Engineers, 2010.

[67] Wang, Rongrong and Wang, Junmin. Fault-tolerant control for electric ground vehicles with independently-actuated in-wheel motors. *Journal of Dynamic Systems, Measurement, and Control*, 134(2):021014, 2012.

[68] Wang, Rongrong and Wang, Junmin. Passive actuator fault-tolerant control for a class of overactuated nonlinear systems and applications to electric vehicles. *IEEE Transactions on Vehicular Technology*, 62(3):972–985, 2012.

[69] Wei, Yanling, Qiu, Jianbin, and Fu, Shasha. Mode-dependent nonrational output feedback control for continuous-time semi-markovian jump systems with time-varying delay. *Nonlinear Analysis: Hybrid Systems*, 16:52–71, 2015.

[70] Wilwert, Cédric, Song, YeQiong, Simonot-Lion, Françoise, Clément, Thomas, et al. Evaluating quality of service and behavioral reliability of steer-by-wire systems. In *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 03TH8696)*, volume 1, pages 193–200. IEEE, 2003.

[71] Wu, Jing and Chen, Tongwen. Design of networked control systems with packet dropouts. *IEEE Transactions on Automatic control*, 52(7):1314–1319, 2007.

[72] Yao, Yixin and Daugherty, Brian. Control method of dual motor-based steer-by-wire system. Technical report, SAE Technical Paper, 2007.

[73] Yu, Xinghuo and Zhihong, Man. Fast terminal sliding-mode control design for nonlinear dynamical systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(2):261–264, 2002.

[74] Zhang, Han and Zhao, Wanzhong. Two-way h∞ control method with a fault-tolerant module for steer-by-wire system. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 232(1):42–56, 2018.

[75] Zhang, Jian, Swain, Akshya Kumar, and Nguang, Sing Kiong. *Robust Observer-Based Fault Diagnosis for Nonlinear Systems Using MATLAB®*. Springer, 2016.

[76] Zhang, Youmin and Jiang, Jin. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.

[77] Zheng, Bing, Altemare, Cliff, and Anwar, Sohel. Fault tolerant steer-by-wire road wheel control system. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1619–1624. IEEE, 2005.

[78] Zheng, Bing and Anwar, Sohel. Fault-tolerant control of the road wheel subsystem in a steer-by-wire system. *International Journal of Vehicular Technology*, 2008, 2008.

[79] Zheng, Shuibo, Tang, Houjun, Han, Zhengzhi, and Zhang, Yong. Controller design for vehicle stability enhancement. *Control Engineering Practice*, 14(12):1413–1421, 2006.

[80] Zong, Changfu, Xiang, Haiou, He, Lei, and Sha, Fei. Study on control method of dual-motor for steer-by-wire system. In *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 2890–2893. IEEE, 2012.

[81] Zou, Fenglou, Song, Deyu, Li, Qiang, and Yuan, Bin. A new intelligent technology of steering-by-wire system by variable structure control with sliding mode. In *2009 International Joint Conference on Artificial Intelligence*, pages 857–860. IEEE, 2009.

پیوست آ

برنامه‌های کامپیوتری

در این بخش برنامه‌های اصلی استفاده شده در شبیه‌سازی‌های این پژوهش بطور کامل برای علاقه‌مندان پیوست شده است.

آ.۱ برنامه طراحی هندسه سیستم فرمان

این برنامه در Jupyter Notebook نوشته شده است، لذا به ترتیب هر قسمت برنامه داخل یک سلول است.

برنامهٔ آ.۱: کد پایتون طراحی هندسه سیستم فرمان سلول۱

```
import numpy as np
from numpy import sin,cos,arcsin,arccos,tan,arctan,pi
from matplotlib import pyplot as plt
# from sympy import *
%matplotlib inline
D  = 1.600 #meter
W  = 2.650 #meter
L1 = 0.14  # assume that the length of Steering-Arm is
   fixed (or is designed beforehand)
#Ideal Values
def outter_delta(inner_delta):
    return arctan( 1/(D/W+1/tan(inner_delta)) )

x = np.linspace(0.0001,45*pi/180,100)
y = outter_delta(x)
with plt.style.context("ggplot"):
    plt.rcParams['figure.figsize'] = [10, 5]
    plt.plot(x*180/pi,y*180/pi,label='Ideal')
    plt.xlabel(r'$\delta_{inner}$',fontsize=18)
    plt.ylabel(r'$\delta_{outter}$',fontsize=18)
    plt.axis('equal')
    plt.legend()
    plt.show()
```

برنامهٔ آ.۲: کد پایتون طراحی هندسه سیستم فرمان سلول۲

```
import matplotlib.animation

%matplotlib notebook
plt.rcParams["animation.html"] = "jshtml"
plt.rcParams['figure.dpi'] = 150
plt.ioff()
fig, ax = plt.subplots()

def rotate(arr,theta):
    R = np.array([[cos(theta),-sin(theta)],[sin(theta),cos(
       theta)]])
```

```python
    return np.matmul(R,arr)

def tire(x0,y0,theta,draw_per_line=True):
    t = 0.15
    r = 0.5
    p = rotate(np.array([[-t/2,-t/2,t/2,t/2,-t/2],[r/2,-r/2,-r/2,r/2,r/2]]),theta)
    plt.plot(p[0]+x0,p[1]+y0,color = 'black')
    if draw_per_line:
        p = rotate(np.array([[-W/(sin(theta)+0.00001)-1.5,0],[0,0]]),theta)
        plt.plot(p[0]+x0,p[1]+y0,color='coral')
    plt.axis('equal')
    plt.axis('off')

def animate(t):
    plt.cla()
    inner_delta = (t+10)/40
    tire(-D/2,W/2,inner_delta)
    tire(-D/2,-W/2,0,False)
    tire(D/2,-W/2,0)
    tire(D/2,W/2,outter_delta(inner_delta))
    plt.ylim(-W,W)
    plt.xlim(-10,D/1.5)

matplotlib.animation.FuncAnimation(fig, animate, frames=15)
```

برنامهٔ آ.۳: کد پایتون طراحی هندسه سیستم فرمان سلول۳

```python
#Rack and Pinion steering system design
class rack_pinion_steering_sys:
    table = []
    alpha = []
    L1 = L1
    L2 = []
    L3 = []
    d = []
    D = D
    theta_max = []
    phi2 = []
    def __init__(self, alpha = 17.5*pi/180, d = 0.2, L2 = 0.5):
        self.alpha = alpha
```

```
        self.d = d
        self.L2 = L2
        phi_0 = arcsin((d-L1*cos(alpha))/L2)
        self.L3 = self.D-2*(  L2*cos(phi_0) + L1*sin(alpha)
            );
        self.theta_max = arccos(d/(L1+L2))
        self.phi2 = phi_0

    def get_states(self,delta):
        theta = delta + self.alpha
        phi = arcsin((self.d-self.L1*cos(theta))/self.L2)
        C = self.D - (self.L1*sin(theta)+self.L2*cos(phi)+
           self.L3)
        gamma = arccos((self.L1**2+self.L2**2-self.d**2-C
           **2)/(2*self.L1*self.L2))
        while True:
            phi2 = arcsin((self.d-self.L1*sin(gamma - self.
               phi2))/self.L2)
            if abs(phi2-self.phi2)<1e-10:
                self.phi2 = phi2
                break
            self.phi2 = phi2
        theta2 = arcsin((self.L2*cos(self.phi2)-C)/self.L1)
        delta2 = theta2 + self.alpha
        return np.array([delta2,theta,phi,theta2,self.phi2
           ])

    def get_y2(self,x):
        y2 = np.zeros_like(x)
        for i,xx in enumerate(x):
            y2[i],_,_,_,_ = self.get_states(xx)
        return y2
```

برنامهٔ آ.۴: کد پایتون طراحی هندسه سیستم فرمان سلول۴

```
mySteer = rack_pinion_steering_sys()

import ipywidgets as widgets
# import warnings
# warnings.filterwarnings("ignore")
%matplotlib notebook
x = np.linspace(0.0001,45*pi/180,100)
y = outter_delta(x)
```

```
y2 = mySteer.get_y2(x)
%matplotlib notebook
with plt.style.context("default"):
    fig = plt.figure()
    ax = fig.add_subplot(1,1,1)
    ax.plot(x*180/pi,x*180/pi,label='parallel',color='gray')
    main, = ax.plot(x*180/pi,y*180/pi,label='Ideal')
    mine, = ax.plot(x*180/pi,y2*180/pi,'-.',label='rack and pinion',color='k')
    ax.set_xlabel(r'$\delta_{inner}$',fontsize=14)
    ax.set_ylabel(r'$\delta_{outter}$',fontsize=14)
    ax.legend()
    ax.grid()

def update_plot(alpha,d=mySteer.d,L2=mySteer.L2):
    mySteer.__init__(alpha,d,L2)
    y2 = mySteer.get_y2(x)
    mine.set_ydata(y2*180/pi)
    fig.canvas.draw()

def fnc1(val):
    tmp = val['new']*pi/1800
    update_plot(tmp,mySteer.d,mySteer.L2)
def fnc2(val):
    tmp = val['new']/100.0
    update_plot(mySteer.alpha,tmp,mySteer.L2)
def fnc3(val):
    tmp = val['new']/100.0
    update_plot(mySteer.alpha,mySteer.d,tmp)

slider1 = widgets.IntSlider(value=175, min=-250, max=250, continuous_update=False)
slider2 = widgets.IntSlider(value=20, min=-20, max=30, continuous_update=False)
slider3 = widgets.IntSlider(value=50, min=10, max=70, continuous_update=False)
slider1.observe(fnc1, names="value")
slider2.observe(fnc2, names="value")
slider3.observe(fnc3, names="value")
display(slider1,slider2,slider3)
```

برنامهٔ آ.۵: کد پایتون طراحی هندسه سیستم فرمان سلول۵

```python
plt.rcParams["animation.html"] = "jshtml"
plt.rcParams['figure.dpi'] = 300
plt.ioff()
fig2,_ = plt.subplots()

L2,L3,d = mySteer.L2,mySteer.L3,mySteer.d

def draw_bars(theta,phi,theta2,phi2):
    X = np.array([-D/2])
    Y = np.array([W/2])
    X = np.append(X,X[-1]+L1*sin(theta))
    Y = np.append(Y,Y[-1]-L1*cos(theta))
    X = np.append(X,X[-1]+L2*cos(phi))
    Y = np.append(Y,Y[-1]-L2*sin(phi))
    X = np.append(X,X[-1]+L3)
    Y = np.append(Y,Y[-1]+0)
    X = np.append(X,X[-1]+L2*cos(phi2))
    Y = np.append(Y,Y[-1]+L2*sin(phi2))
    X = np.append(X,X[-1]-L1*sin(theta2))
    Y = np.append(Y,Y[-1]+L1*cos(theta2))
    plt.plot(X,Y,marker='.',color = 'blue',linewidth=1,markersize=3,mfc='k',mec='gray')
def animate(t):
    plt.cla()
    inner_delta = (t+20)/80
    [outter_delta,theta,phi,theta2,phi2] = mySteer.get_states(inner_delta)
    tire(-D/2,W/2,inner_delta)
    tire(-D/2,-W/2,0,False)
    tire(D/2,-W/2,0)
    tire(D/2,W/2,outter_delta)
    draw_bars(theta,phi,theta2,phi2)
    plt.ylim(-W,W)
    plt.xlim(-10,D/1.5)
matplotlib.animation.FuncAnimation(fig2, animate, frames=15)
```

آ.۲ برنامه شبیه‌سازی دینامیک سیستم فرمان

برنامهٔ آ.۶: کد پایتون - کلاس سیستم فرمان

```
import Output_functions as f
import numpy as np

class STEERING_SYSTEM:
    dt = []
    a = 17.5*np.pi/180
    L1 =0.14
    L2 =0.5
    L3 =0.5246808715635702
    d  =0.2
    D  =1.6
    lock_margin = 10*np.pi/180
    max_lock_angle = np.arccos(d/(L1+L2)) - lock_margin - a
    min_lock_angle = []
    x, xd, xdd = [], [], []
    y, yd, ydd = [], [], []
    z, zd, zdd = [], [], []
    w, wd, wdd = [], [], []

    def __init__(self,dt):
        self.dt = dt
        self.xd = 0
        self.xdd = 0
        self.x = self.max_lock_angle
        self.apply_constraints()
        self.min_lock_angle = -self.z
        self.x = 0
        self.apply_constraints()


    def apply_constraints(self):
        self.y = f.Y(self.x)
        self.w = f.W(self.x,self.y)
        self.z = f.Z(self.x,self.y,self.w)
        self.yd   = f.Yd(self.x,self.xd,self.y)
        self.zd = f.Zd(self.x,self.xd,self.y,self.yd,self.z
            ,self.w)
```

```
        self.wd = f.Wd(self.x,self.xd,self.y,self.yd,self.z,self.w)
        self.ydd = f.Ydd(self.x,self.xd,self.xdd,self.y,self.yd)
        self.zdd = f.Zdd(self.x,self.xd,self.xdd,self.y,self.yd,self.ydd,self.z,self.zd,self.w,self.wd)
        self.wdd = f.Wdd(self.x,self.xd,self.xdd,self.y,self.yd,self.ydd,self.z,self.zd,self.w,self.wd)

    def get_delta_r(self):
        return f.x_to_dr(self.x,self.xd,self.xdd,self.y,self.yd,self.ydd)

    def deltaX_to_x(self,dX):
        self.x   = f.dr_to_x_0(dX[0],self.y)
        self.xd  = f.dr_to_x_1(dX[1],self.x,self.y,self.yd)
        self.xdd = f.dr_to_x_2(dX[2],self.x,self.xd,self.y,self.yd,self.ydd)


    def step_xdd(self,M,F):
        F += self.self_aligning_T()
        param = np.array([self.ydd,self.xd**2,self.yd**2,self.zdd,self.wdd,self.zd**2,self.wd**2,M[0],M[1],F],dtype=float)
        xdd = np.dot(f.Xdd(self.x,self.y,self.z,self.w).reshape([10,]),param)
        self.xdd = 0.5*self.xdd + 0.5*xdd
        self.xd += self.xdd*self.dt
        self.x  += 0.5*self.xdd*self.dt**2 + self.xd*self.dt
        self.lock()
        self.apply_constraints()

    def get_F(self,M):
        if (self.lock()):
            self.F = 0
            self.apply_constraints()
            return self.F
        self.apply_constraints()
        param = np.array([self.xdd,self.ydd,self.xd**2,self.yd**2,self.zdd,self.wdd,self.zd**2,self.wd**2,M
```

```
            [0],M[1]])
        self.F = np.dot(f.F(self.x,self.y,self.z,self.w),
           param)
        return self.F


    def lock(self):
        if self.x <= self.min_lock_angle:
            self.xdd = 0
            self.xd = 0
            self.x = self.min_lock_angle
            return True
        if self.x >= self.max_lock_angle:
            self.xdd = 0
            self.xd = 0
            self.x = self.max_lock_angle
            return True
        return False
```

برنامهٔ آ.۷: کد پایتون توابع کمکی functions Output

```
from numpy import array,sin,cos,tan,arcsin,arccos,arctan,
   sqrt,pi

alpha=17.5*pi/180
L_1=0.14
L_2=0.5
L_3=0.5246808715635702
d=0.2
D=1.6
m_2=1.12
m_r=1.17
I_1=0.776160160573 2
I_2=0.024676

def F(x, y, z, w):
    return (array([[(-1/2*L_1*L_2*m_2*sin(alpha + x + y) -
       L_1*L_2*m_r*sin(y)*cos(alpha + x))*sin(alpha + x)/(-
       L_2*sin(y)*sin(alpha + x) + L_2*cos(y)*cos(alpha + x
       )) + (I_1 + L_1**2*m_2 + L_1**2*m_r*cos(alpha + x)
       **2)*cos(y)/(-L_1*sin(y)*sin(alpha + x) + L_1*cos(y)
       *cos(alpha + x)), (I_2 + (1/4)*L_2**2*m_2 + L_2**2*
       m_r*sin(y)**2)*sin(alpha + x)/(-L_2*sin(y)*sin(alpha
```

```
         + x) + L_2*cos(y)*cos(alpha + x)) + (-1/2*L_1*L_2*
        m_2*sin(alpha + x + y) - L_1*L_2*m_r*sin(y)*cos(
        alpha + x))*cos(y)/(-L_1*sin(y)*sin(alpha + x) + L_1
        *cos(y)*cos(alpha + x)), -L_1**2*m_r*sin(alpha + x)*
        cos(y)*cos(alpha + x)/(-L_1*sin(y)*sin(alpha + x) +
        L_1*cos(y)*cos(alpha + x)) + (-1/2*L_1*L_2*m_2*cos(
        alpha + x + y) + L_1*L_2*m_r*sin(y)*sin(alpha + x))*
        sin(alpha + x)/(-L_2*sin(y)*sin(alpha + x) + L_2*cos
        (y)*cos(alpha + x)), L_2**2*m_r*sin(y)*sin(alpha + x
        )*cos(y)/(-L_2*sin(y)*sin(alpha + x) + L_2*cos(y)*
        cos(alpha + x)) + (-1/2*L_1*L_2*m_2*cos(alpha + x +
        y) - L_1*L_2*m_r*cos(y)*cos(alpha + x))*cos(y)/(-L_1
        *sin(y)*sin(alpha + x) + L_1*cos(y)*cos(alpha + x)),
         (1/2)*L_1*L_2*m_2*sin(alpha - z)*sin(-alpha + w + z
        )/(-L_2*sin(w)*sin(alpha - z) + L_2*cos(w)*cos(alpha
         - z)) + (I_1 + L_1**2*m_2)*cos(w)/(-L_1*sin(w)*sin(
        alpha - z) + L_1*cos(w)*cos(alpha - z)), -1/2*L_1*
        L_2*m_2*sin(-alpha + w + z)*cos(w)/(-L_1*sin(w)*sin(
        alpha - z) + L_1*cos(w)*cos(alpha - z)) - (I_2 +
        (1/4)*L_2**2*m_2)*sin(alpha - z)/(-L_2*sin(w)*sin(
        alpha - z) + L_2*cos(w)*cos(alpha - z)), (1/2)*L_1*
        L_2*m_2*sin(alpha - z)*cos(-alpha + w + z)/(-L_2*sin
        (w)*sin(alpha - z) + L_2*cos(w)*cos(alpha - z)),
        -1/2*L_1*L_2*m_2*cos(w)*cos(-alpha + w + z)/(-L_1*
        sin(w)*sin(alpha - z) + L_1*cos(w)*cos(alpha - z)),
        -cos(y)/(-L_1*sin(y)*sin(alpha + x) + L_1*cos(y)*cos
        (alpha + x)), -cos(w)/(-L_1*sin(w)*sin(alpha - z) +
        L_1*cos(w)*cos(alpha - z))]]))


def Xdd(x, y, z, w):
    return (array([[-2*L_1*(I_2 + (1/4)*L_2**2*m_2 + L_2
        **2*m_r*sin(y)**2)*sin(alpha + x)/(2*I_1*L_2*cos(y)
        - L_1**2*L_2*m_2*sin(alpha + x)*sin(alpha + x + y) +
         2*L_1**2*L_2*m_2*cos(y) - 2*L_1**2*L_2*m_r*sin(y)*
        sin(alpha + x)*cos(alpha + x) + 2*L_1**2*L_2*m_r*cos
        (y)*cos(alpha + x)**2) - 2*(-1/2*L_1*L_2*m_2*sin(
        alpha + x + y) - L_1*L_2*m_r*sin(y)*cos(alpha + x))*
        cos(y)/(2*I_1*cos(y) - L_1**2*m_2*sin(alpha + x)*sin
        (alpha + x + y) + 2*L_1**2*m_2*cos(y) - 2*L_1**2*m_r
        *sin(y)*sin(alpha + x)*cos(alpha + x) + 2*L_1**2*m_r
        *cos(y)*cos(alpha + x)**2), 2*L_1**2*m_r*sin(alpha +
```

```
 x)*cos(y)*cos(alpha + x)/(2*I_1*cos(y) - L_1**2*m_2
*sin(alpha + x)*sin(alpha + x + y) + 2*L_1**2*m_2*
cos(y) - 2*L_1**2*m_r*sin(y)*sin(alpha + x)*cos(
alpha + x) + 2*L_1**2*m_r*cos(y)*cos(alpha + x)**2)
- 2*L_1*(-1/2*L_1*L_2*m_2*cos(alpha + x + y) + L_1*
L_2*m_r*sin(y)*sin(alpha + x))*sin(alpha + x)/(2*I_1
*L_2*cos(y) - L_1**2*L_2*m_2*sin(alpha + x)*sin(
alpha + x + y) + 2*L_1**2*L_2*m_2*cos(y) - 2*L_1**2*
L_2*m_r*sin(y)*sin(alpha + x)*cos(alpha + x) + 2*L_1
**2*L_2*m_r*cos(y)*cos(alpha + x)**2), -2*L_1*L_2
**2*m_r*sin(y)*sin(alpha + x)*cos(y)/(2*I_1*L_2*cos(
y) - L_1**2*L_2*m_2*sin(alpha + x)*sin(alpha + x + y
) + 2*L_1**2*L_2*m_2*cos(y) - 2*L_1**2*L_2*m_r*sin(y
)*sin(alpha + x)*cos(alpha + x) + 2*L_1**2*L_2*m_r*
cos(y)*cos(alpha + x)**2) - 2*(-1/2*L_1*L_2*m_2*cos(
alpha + x + y) - L_1*L_2*m_r*cos(y)*cos(alpha + x))*
cos(y)/(2*I_1*cos(y) - L_1**2*m_2*sin(alpha + x)*sin
(alpha + x + y) + 2*L_1**2*m_2*cos(y) - 2*L_1**2*m_r
*sin(y)*sin(alpha + x)*cos(alpha + x) + 2*L_1**2*m_r
*cos(y)*cos(alpha + x)**2), -1/2*L_1*L_2*m_2*(-2*L_1
*sin(y)*sin(alpha + x)*sin(alpha - z) + 2*L_1*sin(
alpha - z)*cos(y)*cos(alpha + x))*sin(-alpha + w + z
)/(-2*I_1*L_2*sin(w)*sin(alpha - z)*cos(y) + 2*I_1*
L_2*cos(w)*cos(y)*cos(alpha - z) + L_1**2*L_2*m_2*
sin(w)*sin(alpha + x)*sin(alpha - z)*sin(alpha + x +
 y) - 2*L_1**2*L_2*m_2*sin(w)*sin(alpha - z)*cos(y)
- L_1**2*L_2*m_2*sin(alpha + x)*sin(alpha + x + y)*
cos(w)*cos(alpha - z) + 2*L_1**2*L_2*m_2*cos(w)*cos(
y)*cos(alpha - z) + 2*L_1**2*L_2*m_r*sin(w)*sin(y)*
sin(alpha + x)*sin(alpha - z)*cos(alpha + x) - 2*L_1
**2*L_2*m_r*sin(w)*sin(alpha - z)*cos(y)*cos(alpha +
 x)**2 - 2*L_1**2*L_2*m_r*sin(y)*sin(alpha + x)*cos(
w)*cos(alpha + x)*cos(alpha - z) + 2*L_1**2*L_2*m_r*
cos(w)*cos(y)*cos(alpha + x)**2*cos(alpha - z)) + (
I_1 + L_1**2*m_2)*(2*sin(y)*sin(alpha + x)*cos(w) -
2*cos(w)*cos(y)*cos(alpha + x))/(-2*I_1*sin(w)*sin(
alpha - z)*cos(y) + 2*I_1*cos(w)*cos(y)*cos(alpha -
z) + L_1**2*m_2*sin(w)*sin(alpha + x)*sin(alpha - z)
*sin(alpha + x + y) - 2*L_1**2*m_2*sin(w)*sin(alpha
- z)*cos(y) - L_1**2*m_2*sin(alpha + x)*sin(alpha +
x + y)*cos(w)*cos(alpha - z) + 2*L_1**2*m_2*cos(w)*
cos(y)*cos(alpha - z) + 2*L_1**2*m_r*sin(w)*sin(y)*
```

```
sin(alpha + x)*sin(alpha - z)*cos(alpha + x) - 2*L_1
**2*m_r*sin(w)*sin(alpha - z)*cos(y)*cos(alpha + x)
**2 - 2*L_1**2*m_r*sin(y)*sin(alpha + x)*cos(w)*cos(
alpha + x)*cos(alpha - z) + 2*L_1**2*m_r*cos(w)*cos(
y)*cos(alpha + x)**2*cos(alpha - z)), -1/2*L_1*L_2*
m_2*(2*sin(y)*sin(alpha + x)*cos(w) - 2*cos(w)*cos(y
)*cos(alpha + x))*sin(-alpha + w + z)/(-2*I_1*sin(w)
*sin(alpha - z)*cos(y) + 2*I_1*cos(w)*cos(y)*cos(
alpha - z) + L_1**2*m_2*sin(w)*sin(alpha + x)*sin(
alpha - z)*sin(alpha + x + y) - 2*L_1**2*m_2*sin(w)*
sin(alpha - z)*cos(y) - L_1**2*m_2*sin(alpha + x)*
sin(alpha + x + y)*cos(w)*cos(alpha - z) + 2*L_1**2*
m_2*cos(w)*cos(y)*cos(alpha - z) + 2*L_1**2*m_r*sin(
w)*sin(y)*sin(alpha + x)*sin(alpha - z)*cos(alpha +
x) - 2*L_1**2*m_r*sin(w)*sin(alpha - z)*cos(y)*cos(
alpha + x)**2 - 2*L_1**2*m_r*sin(y)*sin(alpha + x)*
cos(w)*cos(alpha + x)*cos(alpha - z) + 2*L_1**2*m_r*
cos(w)*cos(y)*cos(alpha + x)**2*cos(alpha - z)) + (
I_2 + (1/4)*L_2**2*m_2)*(-2*L_1*sin(y)*sin(alpha + x
)*sin(alpha - z) + 2*L_1*sin(alpha - z)*cos(y)*cos(
alpha + x))/(-2*I_1*L_2*sin(w)*sin(alpha - z)*cos(y)
 + 2*I_1*L_2*cos(w)*cos(y)*cos(alpha - z) + L_1**2*
L_2*m_2*sin(w)*sin(alpha + x)*sin(alpha - z)*sin(
alpha + x + y) - 2*L_1**2*L_2*m_2*sin(w)*sin(alpha -
 z)*cos(y) - L_1**2*L_2*m_2*sin(alpha + x)*sin(alpha
 + x + y)*cos(w)*cos(alpha - z) + 2*L_1**2*L_2*m_2*
cos(w)*cos(y)*cos(alpha - z) + 2*L_1**2*L_2*m_r*sin(
w)*sin(y)*sin(alpha + x)*sin(alpha - z)*cos(alpha +
x) - 2*L_1**2*L_2*m_r*sin(w)*sin(alpha - z)*cos(y)*
cos(alpha + x)**2 - 2*L_1**2*L_2*m_r*sin(y)*sin(
alpha + x)*cos(w)*cos(alpha + x)*cos(alpha - z) + 2*
L_1**2*L_2*m_r*cos(w)*cos(y)*cos(alpha + x)**2*cos(
alpha - z)), -1/2*L_1*L_2*m_2*(-2*L_1*sin(y)*sin(
alpha + x)*sin(alpha - z) + 2*L_1*sin(alpha - z)*cos
(y)*cos(alpha + x))*cos(-alpha + w + z)/(-2*I_1*L_2*
sin(w)*sin(alpha - z)*cos(y) + 2*I_1*L_2*cos(w)*cos(
y)*cos(alpha - z) + L_1**2*L_2*m_2*sin(w)*sin(alpha
+ x)*sin(alpha - z)*sin(alpha + x + y) - 2*L_1**2*
L_2*m_2*sin(w)*sin(alpha - z)*cos(y) - L_1**2*L_2*
m_2*sin(alpha + x)*sin(alpha + x + y)*cos(w)*cos(
alpha - z) + 2*L_1**2*L_2*m_2*cos(w)*cos(y)*cos(
alpha - z) + 2*L_1**2*L_2*m_r*sin(w)*sin(y)*sin(
```

```
alpha + x)*sin(alpha - z)*cos(alpha + x) - 2*L_1**2*
L_2*m_r*sin(w)*sin(alpha - z)*cos(y)*cos(alpha + x)
**2 - 2*L_1**2*L_2*m_r*sin(y)*sin(alpha + x)*cos(w)*
cos(alpha + x)*cos(alpha - z) + 2*L_1**2*L_2*m_r*cos
(w)*cos(y)*cos(alpha + x)**2*cos(alpha - z)), -1/2*
L_1*L_2*m_2*(2*sin(y)*sin(alpha + x)*cos(w) - 2*cos(
w)*cos(y)*cos(alpha + x))*cos(-alpha + w + z)/(-2*
I_1*sin(w)*sin(alpha - z)*cos(y) + 2*I_1*cos(w)*cos(
y)*cos(alpha - z) + L_1**2*m_2*sin(w)*sin(alpha + x)
*sin(alpha - z)*sin(alpha + x + y) - 2*L_1**2*m_2*
sin(w)*sin(alpha - z)*cos(y) - L_1**2*m_2*sin(alpha
+ x)*sin(alpha + x + y)*cos(w)*cos(alpha - z) + 2*
L_1**2*m_2*cos(w)*cos(y)*cos(alpha - z) + 2*L_1**2*
m_r*sin(w)*sin(y)*sin(alpha + x)*sin(alpha - z)*cos(
alpha + x) - 2*L_1**2*m_r*sin(w)*sin(alpha - z)*cos(
y)*cos(alpha + x)**2 - 2*L_1**2*m_r*sin(y)*sin(alpha
 + x)*cos(w)*cos(alpha + x)*cos(alpha - z) + 2*L_1
**2*m_r*cos(w)*cos(y)*cos(alpha + x)**2*cos(alpha -
z)), 2*cos(y)/(2*I_1*cos(y) - L_1**2*m_2*sin(alpha +
 x)*sin(alpha + x + y) + 2*L_1**2*m_2*cos(y) - 2*L_1
**2*m_r*sin(y)*sin(alpha + x)*cos(alpha + x) + 2*L_1
**2*m_r*cos(y)*cos(alpha + x)**2), -(2*sin(y)*sin(
alpha + x)*cos(w) - 2*cos(w)*cos(y)*cos(alpha + x))
/(-2*I_1*sin(w)*sin(alpha - z)*cos(y) + 2*I_1*cos(w)
*cos(y)*cos(alpha - z) + L_1**2*m_2*sin(w)*sin(alpha
 + x)*sin(alpha - z)*sin(alpha + x + y) - 2*L_1**2*
m_2*sin(w)*sin(alpha - z)*cos(y) - L_1**2*m_2*sin(
alpha + x)*sin(alpha + x + y)*cos(w)*cos(alpha - z)
+ 2*L_1**2*m_2*cos(w)*cos(y)*cos(alpha - z) + 2*L_1
**2*m_r*sin(w)*sin(y)*sin(alpha + x)*sin(alpha - z)*
cos(alpha + x) - 2*L_1**2*m_r*sin(w)*sin(alpha - z)*
cos(y)*cos(alpha + x)**2 - 2*L_1**2*m_r*sin(y)*sin(
alpha + x)*cos(w)*cos(alpha + x)*cos(alpha - z) + 2*
L_1**2*m_r*cos(w)*cos(y)*cos(alpha + x)**2*cos(alpha
 - z)), -2*L_1*L_2*sin(y)*sin(alpha + x)/(2*I_1*L_2*
cos(y) - L_1**2*L_2*m_2*sin(alpha + x)*sin(alpha + x
 + y) + 2*L_1**2*L_2*m_2*cos(y) - 2*L_1**2*L_2*m_r*
sin(y)*sin(alpha + x)*cos(alpha + x) + 2*L_1**2*L_2*
m_r*cos(y)*cos(alpha + x)**2) + 2*L_1*cos(y)*cos(
alpha + x)/(2*I_1*cos(y) - L_1**2*m_2*sin(alpha + x)
*sin(alpha + x + y) + 2*L_1**2*m_2*cos(y) - 2*L_1
**2*m_r*sin(y)*sin(alpha + x)*cos(alpha + x) + 2*L_1
```

```
        **2*m_r*cos(y)*cos(alpha + x)**2)]]))


def Y(x):
    return (-arcsin((L_1*cos(alpha + x) - d)/L_2))


def Z(x, y, w):
    return (alpha + arcsin((-D + L_1*sin(alpha + x) + L_2*cos(w) + L_2*cos(y) + L_3)/L_1))


def W(x, y):
    return (2*arctan((2*L_2*d - sqrt(-L_1**4 + 2*L_1**2*L_2**2 + 2*L_1**2*d**2 + 2*L_1**2*(-D + L_1*sin(alpha + x) + L_2*cos(y) + L_3)**2 - L_2**4 + 2*L_2**2*d**2 + 2*L_2**2*(-D + L_1*sin(alpha + x) + L_2*cos(y) + L_3)**2 - d**4 - 2*d**2*(-D + L_1*sin(alpha + x) + L_2*cos(y) + L_3)**2 - (-D + L_1*sin(alpha + x) + L_2*cos(y) + L_3)**4))/(-L_1**2 + L_2**2 - 2*L_2*(-D + L_1*sin(alpha + x) + L_2*cos(y) + L_3) + d**2 + (-D + L_1*sin(alpha + x) + L_2*cos(y) + L_3)**2)))


def Yd(x, xd, y):
    return (L_1*xd*sin(alpha + x)/(L_2*cos(y)))


def Zd(x, xd, y, yd, z, w):
    return ((L_1*xd*cos(alpha + x) - L_2*yd*sin(y))*cos(w)/(L_1*cos(alpha + w - z)))


def Wd(x, xd, y, yd, z, w):
    return (-(L_1*xd*cos(alpha + x) - L_2*yd*sin(y))*sin(alpha - z)/(L_2*cos(alpha + w - z)))


def Ydd(x, xd, xdd, y, yd):
    return ((L_1*(xd**2*cos(alpha + x) + xdd*sin(alpha + x)) + L_2*yd**2*sin(y))/(L_2*cos(y)))

```

```python

def Zdd(x, xd, xdd, y, yd, ydd, z, zdd, w, wd):
    return ((-L_1*xd**2*sin(alpha + x)*cos(w) + L_1*xdd*cos(w)*cos(alpha + x) - L_1*zdd**2*sin(alpha + w - z) - L_2*wd**2 - L_2*yd**2*cos(w)*cos(y) - L_2*ydd*sin(y)*cos(w))/(L_1*cos(alpha + w - z)))


def Wdd(x, xd, xdd, y, yd, ydd, z, zdd, w, wd):
    return ((L_1*xd**2*sin(alpha + x)*sin(alpha - z) - L_1*xdd*sin(alpha - z)*cos(alpha + x) + L_1*zdd**2 + L_2*wd**2*sin(alpha + w - z) + L_2*yd**2*sin(alpha - z)*cos(y) + L_2*ydd*sin(y)*sin(alpha - z))/(L_2*cos(alpha + w - z)))


def dr_to_x_0(dr, y):
    return (-alpha + arcsin(((1/2)*D - L_2*cos(y) - 1/2*L_3 + dr)/L_1))


def dr_to_x_1(drd, x, y, yd):
    return ((L_2*yd*sin(y) + drd)/(L_1*cos(alpha + x)))


def dr_to_x_2(drdd, x, xd, y, yd, ydd):
    return ((L_1*xd**2*sin(alpha + x) + L_2*yd**2*cos(y) + L_2*ydd*sin(y) + drdd)/(L_1*cos(alpha + x)))


def x_to_dr(x, xd, xdd, y, yd, ydd):
    return (array([[-1/2*D + L_1*sin(alpha + x) + L_2*cos(y) + (1/2)*L_3], [L_1*xd*cos(alpha + x) - L_2*yd*sin(y)], [-L_1*xd**2*sin(alpha + x) + L_1*xdd*cos(alpha + x) - L_2*yd**2*cos(y) - L_2*ydd*sin(y)]]))
```

آ.۳ برنامه مشاهده‌کننده اغتشاش

برنامهٔ آ.۸: کد پایتون - کلاس مشاهده کننده اغتشاش

```
import numpy as np
from numpy import matmul as mm

class SMO:
    dt = []
    A, B, C = [], [], []
    k=100
    beta, eps = 5000, 10
    po, qo = 13, 15
    z, d_hat = 0, 0

    def __init__(self,dt,A,B,C):
        self.dt = dt
        self.A = A
        self.B = B
        self.C = C

    def SMO(self,z,xn,fx,gx,u):
        so = z - xn
        tmp = -self.k*so -self.eps*np.abs(so)**(self.po/
           self.qo)*np.sign(so) -self.beta*np.arctanh(so
           /10) -np.abs(fx)*np.arctanh(so)
        z_dot = tmp + gx*u
        d_hat = tmp - fx
        return z_dot,d_hat

    def step_obs(self,x,u):
        z_dot, d_hat = self.SMO(self.z,x[1],mm(self.A[1],x)
           ,self.B[1],u)
        self.z += z_dot*self.dt
        return d_hat/self.B[1]
```

آ.۴ با برنامه کنترل‌کننده مدلغزشی زمان گسسته با تأخیر شبکه و اتلاف بسته

برنامهٔ آ.۹: کد پایتون - کلاس کنترل کننده

```python
import numpy as np
from numpy import matmul as mm

class CONTROLLER:
    Hist = []
    dt = []
    F = []
    G = []
    C = []
    h = []
    Q = []
    Cs = []
    si = []
    delay=0
    delay2=0
    sensor_packet_lost = 0
    actuator_packet_lost = 0
    P_sensor_packet_lost = 0.1
    P_actuator_packet_lost = 0.1
    sample_instant = 0
    actuation_instant = 0
    alpha_bar = P_sensor_packet_lost
    d_hat_sens = 0
    xk =  np.array([[0.0],[0.0]])
    xk_1 =  np.array([[0.0],[0.0]])
    u = 0
    uk = 0
    dk_1 = 0
    dk_2 = 0
    rand = []
    randi = []
    def __init__(self,dt,N,F,G,C,h,Cs,si):
        np.random.seed(0)
        self.dt = dt
        self.F = F
```

```
        self.G = G
        self.C = C
        self.h = h
        self.Cs = Cs
        self.si = si
        self.tau_sc = 1/2
        self.zeta = self.tau_sc/(self.tau_sc+1)
        self.n = np.int(self.h//self.dt)
        self.i = 0
        N = int(N)
        self.rand = np.random.rand(2,N)
        self.randi = np.random.randint(0,int(self.n//2),(2,N))
        return

    def get_delays(self):
        return np.array([self.delay,self.delay2,self.sensor_packet_lost,self.actuator_packet_lost])

    def q(self,Sk):
        return self.si/(self.si+np.linalg.norm(Sk))

    def CONTROLLER(self,xk,xk_1):
        Sk = mm(self.Cs,xk)-self.zeta*mm(self.Cs,xk_1)

        H = (1-self.alpha_bar)*mm(self.Cs,self.F)
        I = self.zeta*(1-self.alpha_bar)*self.zeta*self.Cs
        J = 1 - self.q(Sk)
        K = self.alpha_bar*self.Cs
        L = self.zeta*self.alpha_bar*self.Cs

        uk = -(mm(H,xk) -mm(I,xk) +mm(K,xk) -mm(L,xk_1) -J) / (mm(self.Cs,self.G)*(1-self.alpha_bar))
        return uk


    def step_controller(self,err):
        if self.i==self.sample_instant: # sensor
            self.sensor_packet_lost = 0
            if self.rand[0,self.i] > self.P_sensor_packet_lost:
                self.xk_1 = self.xk
```

```python
                self.xk = err
            else:
                self.sensor_packet_lost = 1
            self.sample_instant -= self.delay
            self.delay = self.randi[0,self.i]
            self.sample_instant += self.n//2+self.delay

        if self.i%self.n == 0: # controller
            self.uk = self.CONTROLLER(self.xk,self.xk_1)

        if self.i==self.actuation_instant: # actuator
            self.actuator_packet_lost = 0
            if self.rand[1,self.i] > self.
               P_actuator_packet_lost:
                self.u = self.uk
            else:
                self.actuator_packet_lost = 1
            self.actuation_instant -= self.delay2
            self.delay2 = self.randi[1,self.i]
            self.actuation_instant += self.n+self.delay2

        self.i+=1

        return self.u
```