

Control Project

Suspension of Car



Present by

6530116821 Nattakit Chaijaroenmaitree

6532017721 Kantanat Phochanasombut

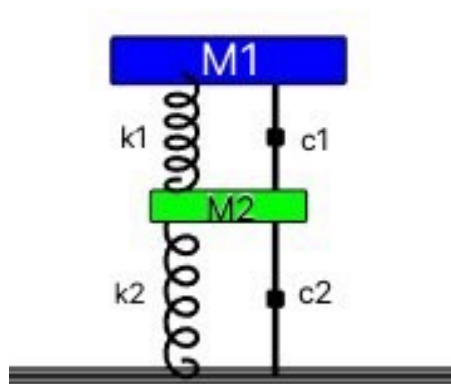
6532122921 Pun nawich Pongsriasswin

2103304 Automatic Control

ที่มาและความสำคัญ

ในปัจจุบันยานพาหนะจำนวนมากมีระบบกันสั่นสะเทือนเพื่อช่วยในการขับขี่ทั้งในด้านของความสะดวกสบายและในด้านความปลอดภัย เป้าหมายหลักของระบบกันสั่นสะเทือนมีสองข้อได้แก่ การควบคุมให้ยานพาหนะสามารถอยู่ภายใต้การควบคุมของคนขับตลอดเวลาเช่นในรถยนต์คือการทำให้ล้อติดถนน และการลดการสั่นสะเทือนของระบบทั้งหมดซึ่งก็คือการลดการกระจัดระหว่างส่วนประกอบต่างๆในระบบนี้ ซึ่งระบบกันสั่นสะเทือนมีสององค์ประกอบได้แก่ สปริงซึ่งมีหน้าที่ดูดซับแรงที่เกิดขึ้นต่อระบบและแดมเปอร์ซึ่งมีหน้าที่ดูดซับพลังงานที่เกิดขึ้นต่อระบบ ในระบบการสั่นสะเทือนที่นำมาทำการจำลองนี้ประกอบด้วยระบบสปริงและแดมเปอร์สองระบบได้แก่ สปริงแดมเปอร์ที่ลดการสั่นสะเทือนที่เกิดขึ้นจากพื้นถนนและเพิ่มการยึดเกาะระหว่างรถกับถนนซึ่งอยู่ระหว่างพื้นถนนกับแกนของระบบ กับสปริงแดมเปอร์ที่ช่วยลดการสั่นสะเทือนของห้องโดยสารซึ่งจะเชื่อมระหว่างแกนของระบบและห้องโดยสาร ในการจำลองการทำงานของระบบกันสั่นสะเทือนที่ประกอบไปด้วยสปริงแดมเปอร์นี้มีข้อมูลที่ป้อนเข้าระบบกันการสั่นสะเทือนคือ แรงที่เกิดขึ้นจากการขับขี่ และเป้าหมายในการออกแบบระบบกันการสั่นสะเทือนนี้คือการควบคุมการกระจัดของระบบและห้องโดยสารให้มีการเปลี่ยนแปลงตามที่ต้องการขณะที่เกิดการป้อนแรงเข้าสู่ระบบ

Physical Model: Suspension



Define

$M1 \rightarrow \frac{1}{4}$ Mass of car (kg)

$M2 \rightarrow$ Suspension Mass (kg)

$k1 \rightarrow$ Spring constant of Suspension System (N/m)

$k2 \rightarrow$ Spring constant of Wheel and tire (N/m)

$c1 \rightarrow$ Damping constant of Suspension System (Ns/m)

$c2 \rightarrow$ Damping constant of Wheel and tire (Ns/m)

$F \rightarrow$ Force Control Input

$W \rightarrow$ Disturbance from ground (m)

ให้ Input เป็น Force และ output เป็นระยะ x1-x2

Equation of Motion

$$\sum F = ma$$

$$M_1 \ddot{x}_1 = -c_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2) + F$$

$$M_2 \ddot{x}_2 = -c_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2) + c_2(\dot{W} - \dot{x}_2) - k_2(W - x_2) + F$$

Initial conditions are zero → Take Laplace

$$(M_1 s^2 + c_1 s + k_1)X_1(s) - (c_1 s + k_1)X_2(s) = F(s)$$

$$-(c_1 s + k_1)X_1(s) + (M_2 s^2 + (c_1 + c_2)s + (k_1 + k_2))X_2(s) = (c_2 s + k_2)W(s) - F(s)$$

$$\begin{bmatrix} (M_1 s^2 + c_1 s + k_1) & -(c_1 s + k_1) \\ -(c_1 s + k_1) & (M_2 s^2 + (c_1 + c_2)s + (k_1 + k_2)) \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} F(s) \\ (c_2 s + k_2)W(s) - F(s) \end{bmatrix}$$

From $AX = B \rightarrow X = A^{-1}$ where $A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ and simplify the left term to only F and W
by $\begin{bmatrix} F(s) \\ (c_2 s + k_2)W(s) - F(s) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & (c_2 s + k_2) \end{bmatrix} \begin{bmatrix} F(s) \\ W(s) \end{bmatrix}$

We get

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{1}{\det(A)} \begin{bmatrix} M_2 s^2 + c_2 s + k_2 & c_1 c_2 s^2 + (c_1 k_1 + c_2 k_1)s + k_1 k_2 \\ -M_1 s^2 & M_1 c_2 s^3 + (M_1 k_2 + c_1 c_2)s^2 + (c_1 k_2 + c_2 k_1)s + k_1 k_2 \end{bmatrix} \begin{bmatrix} F(s) \\ W(s) \end{bmatrix}$$

Where $\det(A) = (M_1 s^2 + c_1 s + k_1) \times (M_2 s^2 + (c_1 + c_2)s + (k_1 + k_2)) - (c_1 s + k_1) \times (c_1 s + k_1)$

Neglect Disturbance W = 0 for Plant 1

$$P_1 = \frac{X_1(s) - X_2(s)}{F(s)} = \frac{(M_1 + M_2)s^2 + c_2 s + k_2}{(M_1 s^2 + c_1 s + k_1) \times (M_2 s^2 + (c_1 + c_2)s + (k_1 + k_2)) - (c_1 s + k_1) \times (c_1 s + k_1)}$$

Neglect Disturbance F = 0 for Plant 2

$$P_2 = \frac{X_1(s) - X_2(s)}{W(s)} = \frac{-M_1 c_2 s^3 - M_1 k_2 s^2}{(M_1 s^2 + c_1 s + k_1) \times (M_2 s^2 + (c_1 + c_2)s + (k_1 + k_2)) - (c_1 s + k_1) \times (c_1 s + k_1)}$$

Variables assignments

Table 2 Model parameters

<i>Parameter</i>	<i>Vehicle #1</i>	<i>Vehicle #2</i>	<i>Vehicle #3</i>	<i>Unit</i>
Tyre stiffness	205,000	213,818	245,000	N/m
Tyre damping	100	100	100	N/m/s ²
Unsprung mass	30	50	55	kg
Suspension stiffness	24,200	21,600	28,400	N/m
Suspension damping	See Figure 7	See Figure 7	See Figure 7	N s/m
Sprung mass	225	360	450	kg
Inertial momentum	780	1150	1440	Kg m ⁴
Tread width	1.2	1.4	1.4	m
Length A	0.08	0.10	0.10	m
Length B	0.32	0.46	0.46	m
Length C	0.52	0.6	0.6	m

Table 5 Factors and levels combination

<i>Factor</i>	<i>Model Parameter</i>	<i>Level</i>	<i>Value</i>	<i>Unit</i>
A	Sprung mass	1	250	kg
		2	500	kg
B	Suspension stiffness	1	18,000	N/m
		2	25,000	N/m
C	Tyre stiffness	1	120,000	N/m
		2	210,000	N/m
D	Unsprung mass	1	40	kg
		2	60	kg

จากข้อมูลกำหนดให้

$$M1 = 1300/4$$

$$M2 = 225$$

$$k1 = 24200$$

$$k2 = 205000$$

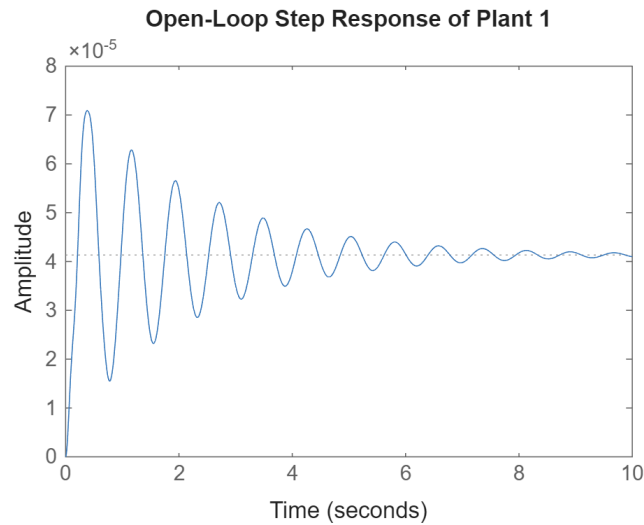
$$c1 = 350$$

$$c2 = 1500$$

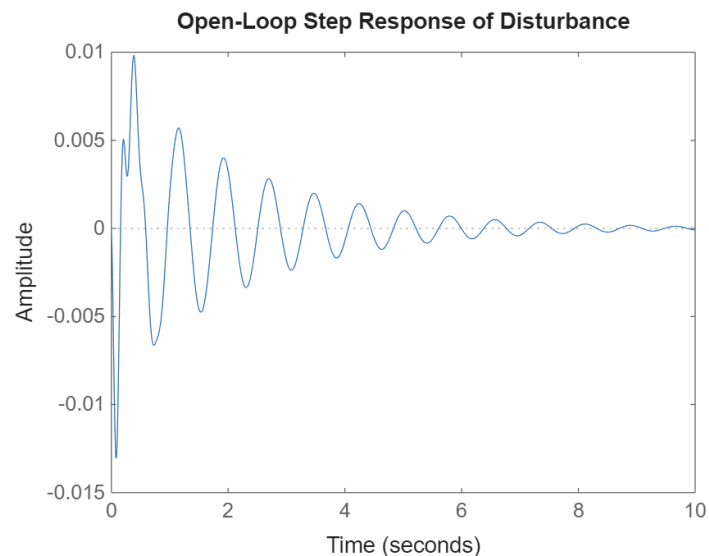
Design requirements

ออกแบบ Controller ให้ Settling Time น้อยกว่า 5 วินาที และ %Overshoot น้อยกว่า 25% เช่นเมื่อรถเคลื่อนที่ผ่านพื้นที่ขรุขระต่างจากระดับปกติ 0.01 m หรือ 1 cm จะมีการสั่นต่ำกว่า 0.25 cm

เริ่มโดยแทนค่าตัวแปรลงในสมการ Open loop Transfer P1 และ P2 จะได้ว่า DC-gain ของ Plant 1 เท่ากับ 4.1322×10^{-5} และ Plant 2 เท่ากับ 0 โดยพบว่า P1 มี poles คือ $-4.2003 \pm 31.7653i$ และ $-0.4493 \pm 8.1165i$ กับ zero คือ $-1.3636 \pm 19.2579i$



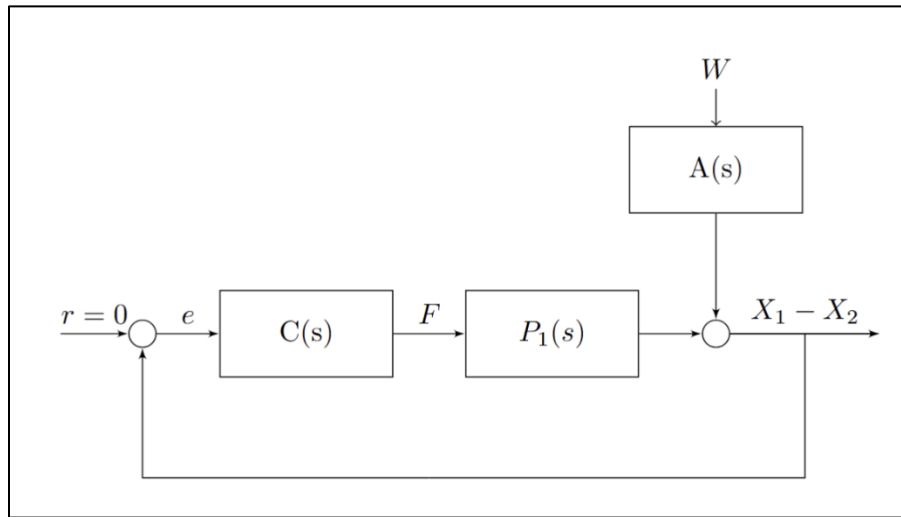
รูปที่ 1 Open Loop Step Response ของ P1 เมื่อให้ unit step 1(t)



รูปที่ 2 Open Loop Step Response ของ P2 เมื่อให้ unit step 0.01(t)

จากรูปที่ 1 และรูปที่ 2 สังเกตได้ว่าเมื่อรถเคลื่อนที่ผ่านรอยขรุขระต่างจากระดับพื้น 0.01 m หรือ 1 cm จะต้องใช้เวลาประมาณ 10 วินาที กว่าจะมีการสั่นที่น้อยลงและเข้าสู่ Steady State และมี Amplitude การสั่นสูงถึง 0.01 m เลย ดังนั้นจำเป็นต้องออกแบบ Controller เพื่อมาควบคุมระบบ

Closed Loop Control Diagram

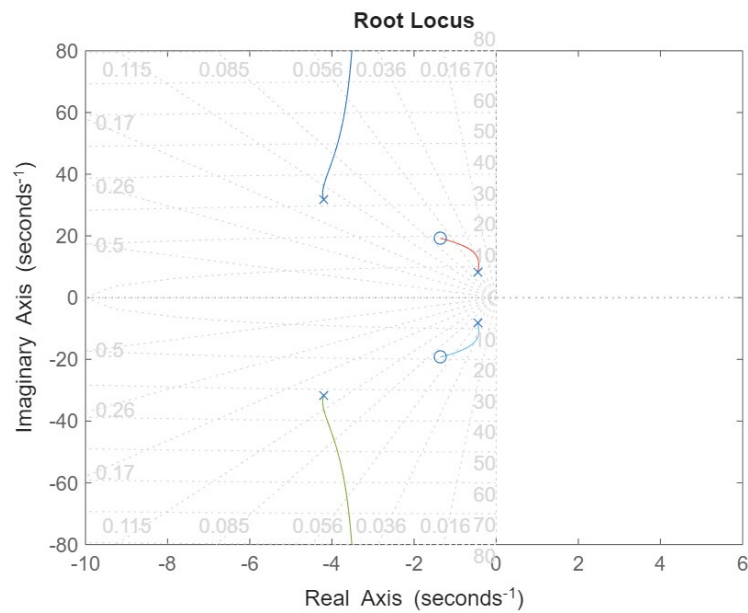


รูปที่ 3 ไดอะแกรมระบบปิด (Closed Loop Diagram)

เมื่อ $P_2 = P_1 A$

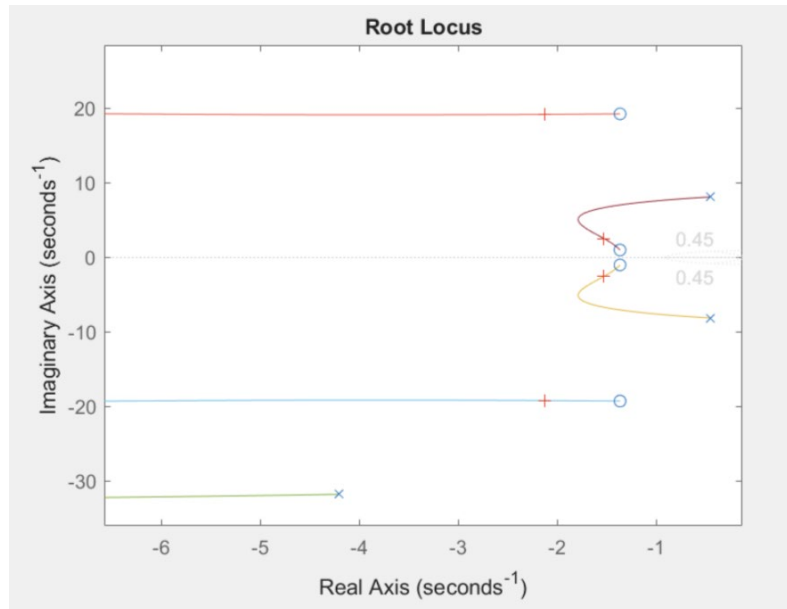
Root Locus

วิธีแรกที่น่ามาหา Controller คือ Root Locus โดยจะอาศัยการเพิ่มตำแหน่งของ pole และ zero เพื่อให้ Step Response เป็นไปตามที่เรากำหนด และดังรูปที่ 4



รูปที่ 4 Root Locus ของ Plant 1

ออกแบบระบบ Plant 1 เพื่อหารายละเอียดของ Controller โดยสังเกตได้ว่า Dominated Pole ของ Open loop อยู่ใกล้กับ Imaginary axis มากเกินไปทำให้ระบบไม่เสถียร ดังนั้นจึงต้องเติม Complex Zero เข้าไปเพื่อ Cancel คู่ Pole ดังกล่าว จากนั้นเพื่อดึงให้ Locus ของระบบออกจาก Zero เดิมจึงเติม Pole ไปทางซ้ายเพื่อให้ผ่าน Time Specification โดย $\%Mp \leq 1 - \frac{\zeta}{0.6}$; $\zeta \leq 0.6$ ซึ่งจากการกำหนดให้ %Overshoot ต่ำกว่า 25 เปอร์เซนต์จะได้ว่า $\zeta \geq 0.45$ ดังนั้นจึงเลือกให้ $C = \frac{0.59s^2 + 0.95s + 1}{(s-30)(s-20)}$ จากนั้นใช้ rlocfind(C*P1) เพื่อหาค่า k ที่จุดนั้น แสดงได้ในรูปที่ 5 และ 6

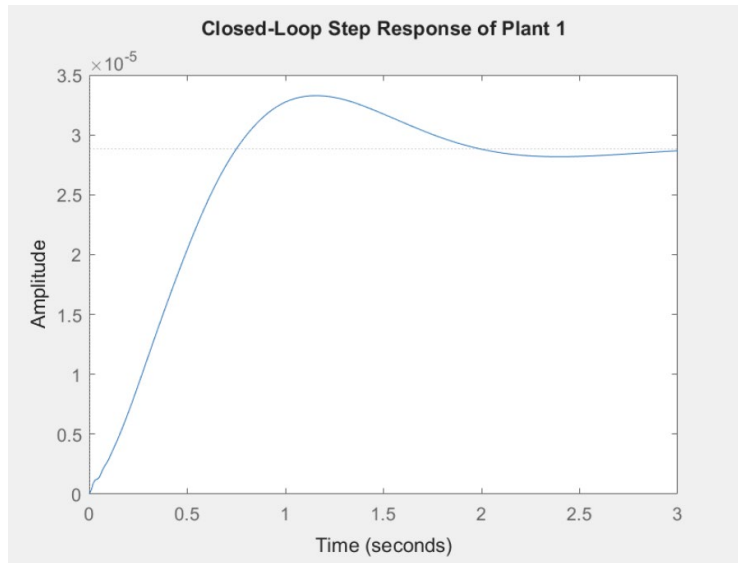


รูปที่ 5 Root Locus ของ Plant 1 หลังเปลี่ยนแปลงค่า

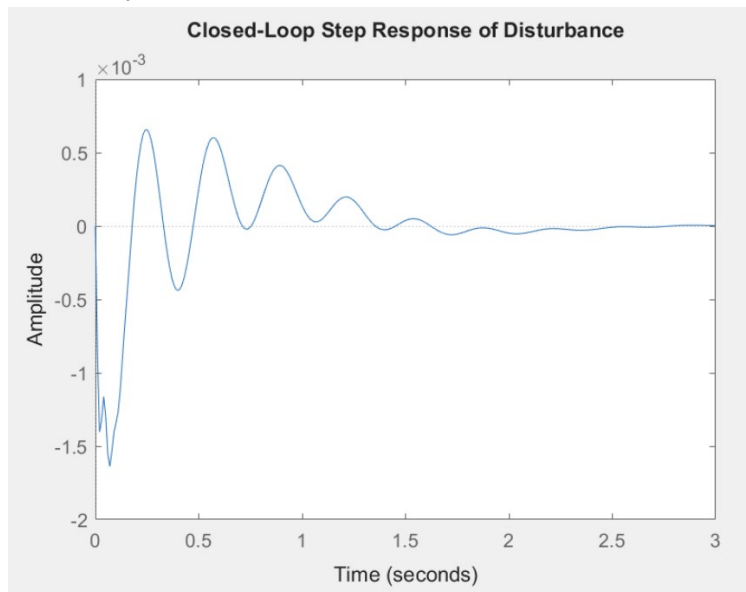
```
selected_point = -1.5010 + 2.5408i
k = 6.2814e+06
```

รูปที่ 6 ค่า k ที่หาจากการใช้ rlocfind(C*P1)

พบว่าจุดที่เลือกมีค่า k คือ 6.2814×10^6 นำค่า C และ k ไปพลอตเพื่อหา Closed-Loop Step Response ของ Plant 1 และ Disturbance โดยแสดงได้ในรูปที่ 7 และ 8 โดยเห็นได้ว่าทั้งคู่สามารถผ่าน Time Requirements ได้ โดยเฉพาะการจัดการกับ Disturbance ขนาด 1 cm มี Amplitude สูงสุดประมาณ 1.5 mm และสามารถเข้าสู่ Steady State ภายใน 3 วินาที



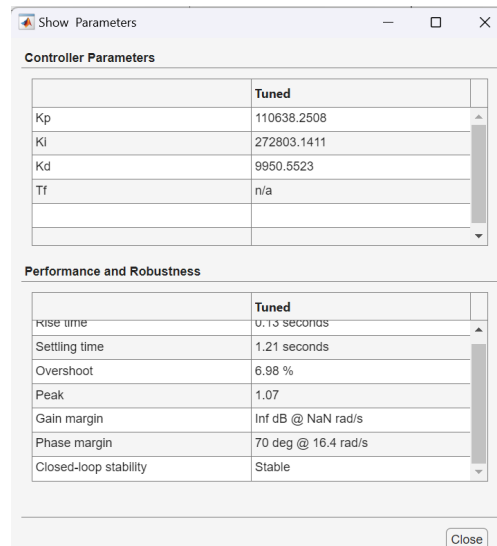
รูปที่ 7 Closed-Loop Step Response ของ Plant 1



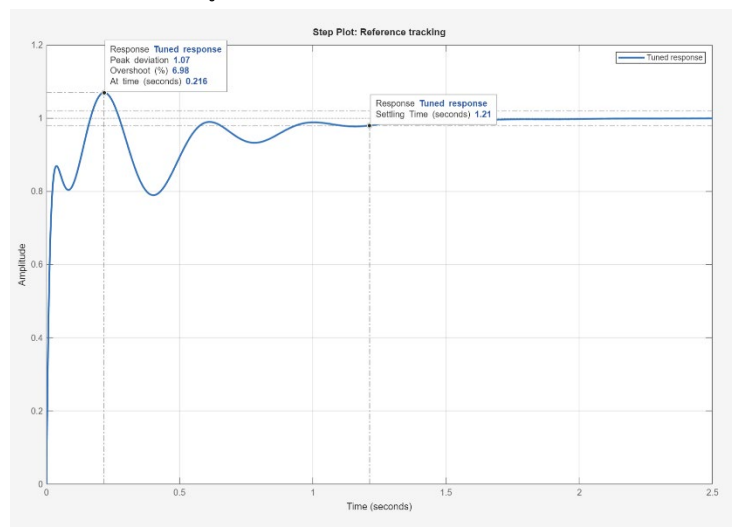
รูปที่ 8 Closed-Loop Step Response ของ Disturbance (W

PID Controller

วิธีสองที่นำมาหา Controller คือการใช้ PID โดยมีการใช้ pidtuner และทำการปรับให้ Plant 1 สามารถผ่านข้อกำหนดได้ โดยพบว่าได้ค่า $K_p = 110638$, $K_d = 9950$ และ $K_i = 272803$ แสดงในรูปที่ 9 และ Closed-Loop Step Response ของ Plant 1 แสดงในรูปที่ 10

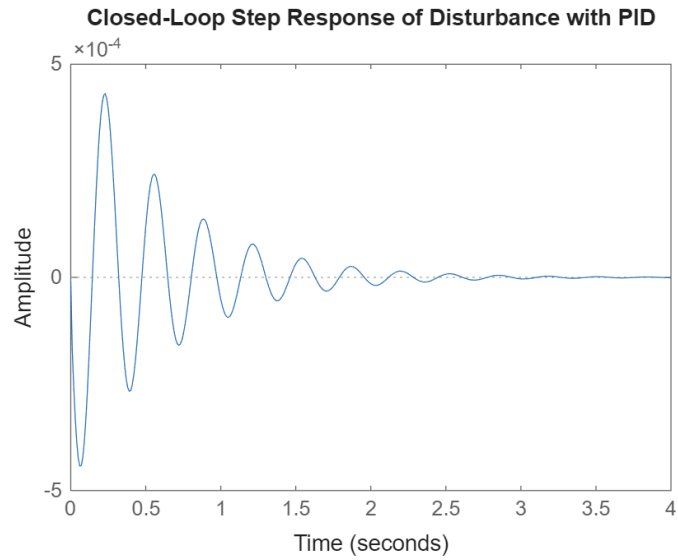


รูปที่ 9 ค่าคงที่ของ PID Controller



รูปที่ 10 Step Response ของ Plant 1

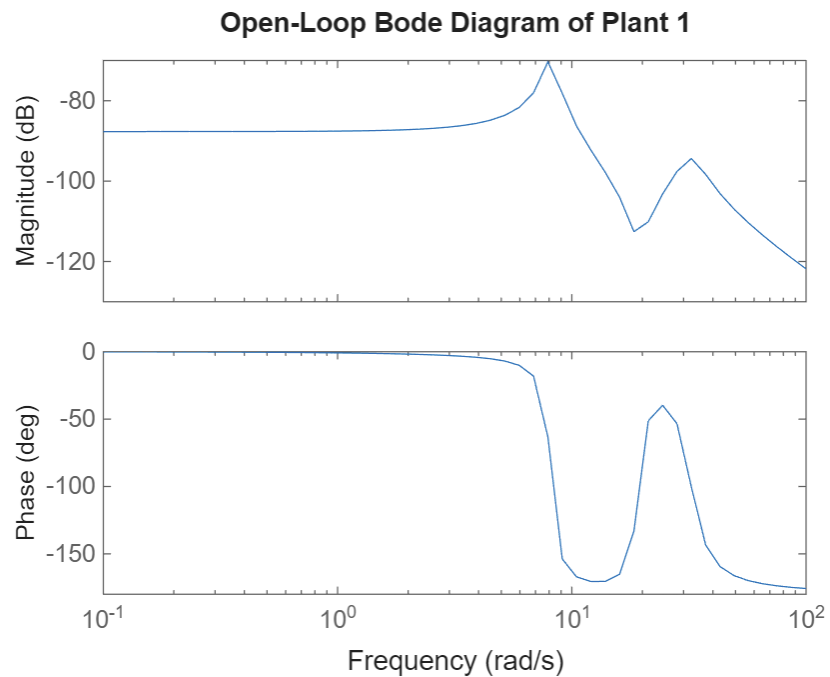
จากนั้นนำค่า K_p , K_d , และ K_i ที่ได้มาใส่ใน Controller และ Feedback ของ Disturbance ซึ่งได้ผลว่าเมื่อมีการรบกวน 1 cm เข้ามาจะมีการสั่น Amplitude สูงสุดประมาณ 0.4 mm และเข้าสู่ Steady State ภายใน 4 วินาที แสดงในรูปที่



รูปที่ 11 Close Loop Step Response ของ Disturbance (W ด้วย PID

Bode Plot

วิธีสามที่นำมาหา Controller คือการใช้ Bode Plot และ Lead Control วิธีนี้สามารถนำมาใช้ได้เพราะระบบมี Stability ตั้งแต่ Open Loop โดยเริ่มต้นด้วยการวาด Bode Plot ของ Open Loop เพื่อตรวจสอบหาตำแหน่งความถี่ (ω) ที่ทำให้ Phase เท่ากับ 180 แสดงในรูปที่ 12



รูปที่ 12 Open-Loop Bode Diagram ของ Plant 1

เห็นได้ว่าบริเวณที่มี การเปลี่ยน Phase ลงมาคือประมาณ 12 ดังนั้นจึงควรเพิ่มค่าบริเวณดังกล่าว และยังมี Phase Margin (PM) เยอะจะทำให้ค่า %Overshoot มีค่าน้อย ดังนั้นจึงเลือกเพิ่มไปอีก 120 องศา ผ่านการคำนวณค่า a ซึ่งเป็นค่าคงที่บอกถึงระยะห่างระหว่าง Pole และ Zero สำหรับการเพิ่ม Phase เข้าไป 120 องศา ตามสมการที่ 1

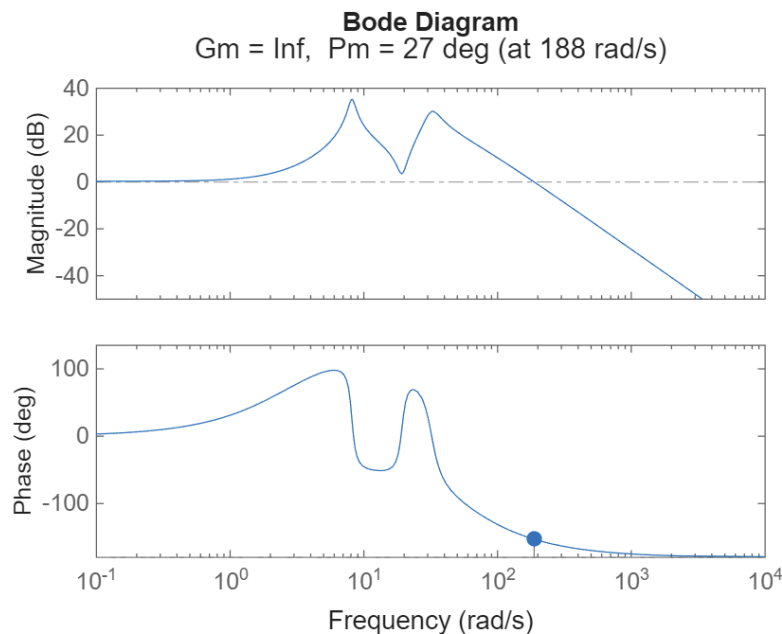
$$a = \frac{1 - \sin 60^\circ}{1 + \sin 60^\circ}$$

จากนั้นให้คำนวณ T และ aT ตามสมการที่ 2 และ 3 ในการหาตำแหน่งความถี่ตัด (Cut Off Frequency) เพื่อให้ Phase ที่เพิ่มอยู่ตรงตำแหน่งความถี่ที่ต้องการแทนด้วยตัวแปร W ในการคำนวณนี้เท่ากับ 12

$$T = \frac{1}{W\sqrt{a}}$$

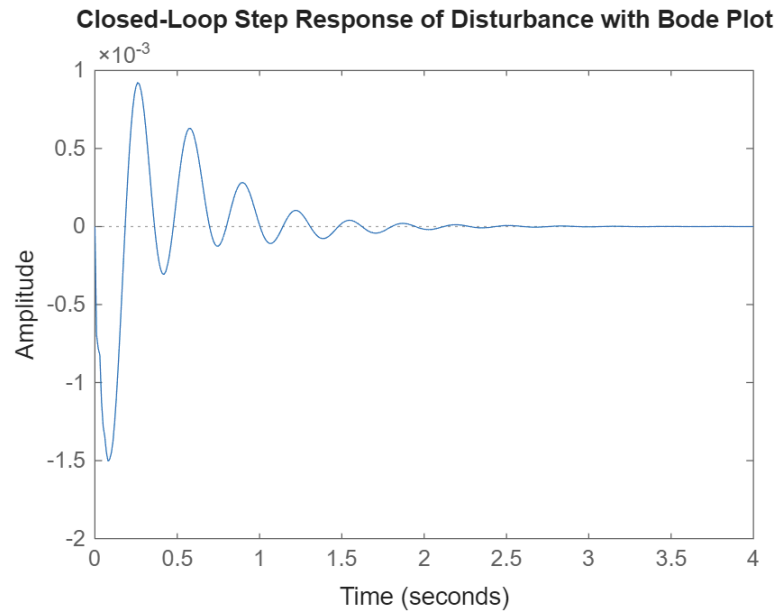
$$aT = \frac{\sqrt{a}}{W}$$

นำค่าตัวแปรทั้งหมดเข้าไปสร้าง Controller Transfer Function ตามสมการที่ 4 และเพิ่ม Gain เข้าไปในระบบโดยทำให้อัตราขยายเพิ่มจาก -80 dB เป็น 0 ด้วยการคูณ K เท่ากับ 25,000 เข้าไป ในหน่วย dB ด้วย $20\log(K)$ และหา Bode Plot อีกครั้งแสดงในรูปที่ 13



รูปที่ 13 Bode Diagram ใหม่

Bode diagram ใหม่นั้นได้แสดงให้เห็นว่ามี PM = 27 องศา ที่ความถี่ 188 rad/s และเมื่อหา Step Response จะได้ว่าระบบสามารถเข้าสู่สถานะ Steady ได้ในเวลา 3 วินาที และมี Amplitude Overshoot สูงสุดเท่ากับ 1.5 mm จาก Step Input 1 cm



รูปที่ 14 Closed Loop Step Response Response ของ Disturbance (W ด้วย Bode plot

สรุปผล

จากการจำลองโมเดลระบบป้องกันการสั่นสะเทือนพบว่าการออกแบบ Controller ที่จะทำให้ระบบเป็นไปตาม Design requirements โดยใช้สามวิธีได้แก่ Root Locus, PID และ Bode Plot ให้ผลที่ต่างกัน ภายใต้ Time specification เดียวกันคือ Settling Time น้อยกว่า 5 วินาที และ %Overshoot น้อยกว่า 25% โดยสนใจไปที่ Step Response ของ W หรือ Disturbance เมื่อตัวรถเคลื่อนที่ผ่านผิวขรุขระต่างจากระดับปกติ 1 cm

วิธี Root Locus จะใช้การเพิ่มตำแหน่งของ Poles และ Complex Zero โดย $C = \frac{0.59 s^2 + 0.95 s + 1}{(s-30)(s-20)}$ และอาศัย function rlocfind(ใน MATLAB ซึ่งได้ค่า k ของระบบเท่ากับ 6.2814×10^6 และได้ผลลัพธ์ Amplitude สูงสุดประมาณ 1.5 mm และสามารถเข้าสู่ Steady State ภายใน 3 วินาที

วิธีที่ใช้ PID เพื่อหา Controller ที่เหมาะสมในการออกแบบระบบนี้จะนำค่า Kp, Ki และ Kd มาจากโปรแกรม pidtuner จาก MATLAB จะได้ค่า Kp = 110638, Ki = 272803 และ Kd = 9950 ซึ่งจะทำให้ระบบสามารถแสดงค่าตาม requirements โดย Amplitude สูงสุดประมาณ 0.4 mm และเข้าสู่ Steady State ภายใน 4 วินาที

วิธี Bode Plot สำหรับการจำลองระบบนี้ด้วยการเพิ่ม Phase Margin ที่บริเวณความถี่เท่ากับ 12° เพื่อนำไปหาความถี่ตัดแล้วนำตัวแปรไปสร้าง Controller Transfer Function ที่มีค่า K เท่ากับ 25,000 พบว่าได้ค่า Phase Margin ของระบบนี้เท่ากับ 27 deg โดย Amplitude สูงสุดประมาณ 1.5 mm และเข้าสู่ Steady State ภายใน 3 วินาที

อ้างอิง

Calvo, J. A., Diaz, V., & Roman, J. L. (2005). Establishing inspection criteria to verify the dynamic behaviors of the vehicle suspension system by a platform vibrating test bench. *International Journal of Vehicle Design*, 38(4), 290. <https://doi.org/10.1504/ijvd.2005.007623>

Tilbury, Prof. D. (n.d.. Control Tutorial. Control tutorials for MATLAB and Simulink – Home. <https://ctms.engin.umich.edu/CTMS/index.php?aux=Home>

ลิงค์ GitHub พร้อมวิธีใช้งาน

NuchPurnawichP / Auto_Control_Project

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Auto_Control_ProjectPublic

File

Unwatch

Fork

Star

main • 1 branch 0 Tags

Q. Go to file

Add file • Code

About

NuchPurnawichP Update README.md · d4432c3 · 5 days ago · 11 Comments

Animation.mp4animationlast week

Control_Project.mlxAdd PID source code · last week

Project Control Suspension of Car.pdfProject Report · 5 days ago

README.mdUpdate README.md · 5 days ago

draw_MBSK2.mAnimation and Graph · last week

README

Repository Guideline: Auto_Control_Project

This guideline provides an overview of the files in the repository and instructions on how to use them.

Repository Contents

1. Animation.mp4

- Location: Root directory
- Description: This file contains an animation showcasing the control system's performance or behavior. It can be used for visualizing the results of the implemented PID controller or other control systems.

2. Control_Project.mlx

- Location: Root directory
- Description: MATLAB Live Script containing the PID controller implementation. It includes:
 - Code for simulating the control system.

Releases

No releases published.
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

MATLAB 100.0%

https://github.com/NuchPunnawichP/Auto_Control_Project/blob/main/Animation.mp4

