Andreas Johansson
Institutionen för system- och rymdteknik
2021-10-26

R0001E Laboration 2

Reglerteknik med Matlab

Huvudsyftet med denna laboration är att introducera Matlab som verktyg för att arbeta med linjära dynamiska system. Övningarna görs på två tillämpningar med fordonsanknytning, farthållare och hjulupphängning.

Redovisning

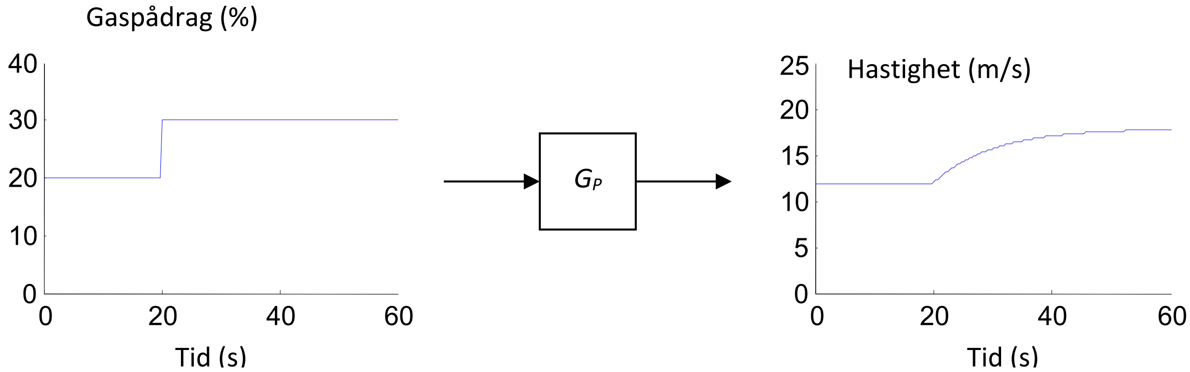
Resultaten från de punkter som markerats med stjärna * redovisas i form av en kortfattad rapport som laddas upp på kurshemsidan. Rapporten behöver inte uppfylla några formalia förutom att skrivas med egna ord och hela meningar samt att resultaten illustreras med figurer.

Bil med farthållare

En mycket enkel modell av en bil är överföringsfunktionen

$$G_P(s) = \frac{0{,}6}{10s + 1}$$

från gaspådrag till hastighet. Denna modell säger att om man ökar gaspådraget med *n* % så kommer hastigheten att öka med 0,6*n* m/s och att det kommer att ta 10 sek. innan 63% av ökningen är genomförd. Detta är naturligtvis en förenkling som bara gäller inom begränsade områden av hastighet och gaspådrag.



Som farthållare till bilen ska en PI-regulator användas, dvs

$$G_R(s) = K\left(1 + \frac{1}{T_I s}\right)$$

Mata in överföringsfunktioner

Det är mycket praktiskt att använda sig av skript, dvs att skriva in sina Matlabkommandon i en fil för att sedan kunna exekvera dem på en gång.

1. Öppna skript-editorn genom att i Matlab-fönstret välja *New Script* (knappen längst upp till vänster). Skriv in Matlab-kommandona som behövs för att lösa uppgifterna nedan i skriptet, avsluta rader med semikolon för att undvika att resultatet ekas på skärmen. Spara skriptet

under namnet `lab3.m`. Tryck *Run* (gröna triangeln högst upp i mitten) när du vill testa kod som du skrivit.

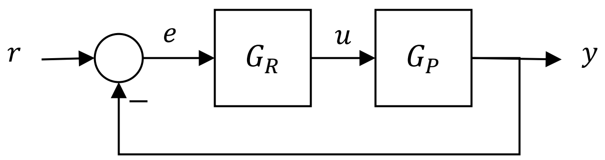
Överföringsfunktioner på formen

$$G(s) = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_n}{a_0 s^n + a_1 s^{n-1} + \cdots + a_n}$$

kan skrivas in i Matlab med funktionen tf enligt `G = tf([b0 b1 … bn],[a0 a1 … an]);`

2. Skriv in bilens överföringsfunktion i Matlab enligt `GP = tf(0.6,[10 1])`. Observera att täljaren bara har en koefficient och kan därför skrivas utan hakparenteser.
3. Sätt farthållarens överföringsfunktion på gemensam nämnare så att den blir en kvot mellan två 1:a-gradspolynom, dvs på formen
$$G_R(s) = \frac{b_1 s + b_0}{a_1 s + a_0}$$
där $a_0,\ a_1, b_0,\ b_1$ beror på K och T_I.
4. Mata in farthållarens överföringsfunktion i Matlab under namnet `GR` med parametervärdena $K = 20$ och $T_I = 2{,}5$. Observera att både täljare och nämnare är 1:a-gradspolynom, dvs har två koefficienter (även om en av nämnarens koefficienter är 0).

Koppla ihop system



Farthållare och bil ska nu kopplas ihop enligt figuren ovan där r är börvärde (önskad hastighet) och y är ärvärde. Då ett reglersystem kopplas på detta sätt så är kretsöverföringen G_K seriekopplingen av regulator och process (dvs $G_R G_P$) medan servosystemet G_{TOT} är slutna systemet från r till y.

5. Ta fram servosystemets överföringsfunktion (från börvärde till ärvärde) i blockschemat ovan.

Ihopkoppling av system i Matlab görs enkelt genom att skriva in uttrycket för överföringsfunktionen.

6. Beräkna kretsöverföringen, dvs genom att seriekoppla bilen och farthållaren enligt `GK=GP*GR`. Beräkna även servosystemets överföringsfunktion `GTOT` genom att skriva in uttrycket du fick fram i punkt 5.

Analysera linjära system i LTI Viewer

Många olika analysmetoder för linjära system finns tillgängliga i Matlab och flera av dem är samlade i verktyget LTI Viewer (LTI står för *Linear Time-Invariant*).

7. Skriv `ltiview` för att starta LTI Viewer. Välj *Import* under menyn *File*. Här bör nu finnas en lista med system som finns i workspace. Välj `GP` och `GTOT` (använd ctrl-tangenten för att välja flera system) och tryck OK.

Matlab plottar nu både stegsvaret för bilen och för slutna systemet med bil och farthållare. Flera analysverktyg finns nu tillgängliga genom att högerklicka på ritytan (inte på kurvorna) och gå till *Characteristics* i menyn som kommer fram.

8. *Använd *Characteristcs* för att ta fram:
   - Processens statiska förstärkning

- Slutna systemets översväng och insvängningstid

  Numeriska värden får man fram genom att peka med markören på punkterna som dyker upp på kurvan. Ibland misslyckas Matlab med att beräkna värden. Då får man istället zooma in på kurvan och avläsa för hand.

Även frekvenssvar kan analyseras med LTI Viewer.

9. Välj *New Linear System Analyzer* under menyn *File* för att starta ett nytt fönster och importera kretsöverföringen. Högerklicka på ritytan och välj *Plot Types* och alternativet *Bode*.
10. *Använd *Characteristics* för att ta fram fasmarginalen.
11. Starta ytterligare ett fönster och importera servosystemet. Välj *Bode Magnitude* under *Plot Types*. Amplitudförstärkningen uttrycks i dB. För att ändra detta till absoluta tal, välj *Properties…* och sedan, under fliken *Units*, välj *Magnitude:* ***absolute*** och *Scale:* ***log scale***.
12. *Använd *Characteristics* för att ta fram resonanstoppen.

Dimensionera regulatorer med trial-and-error

Som framgick i punkt 8, så har slutna systemets stegsvar en översväng. Detta är troligen till nackdel för passagerarnas komfort eftersom bilens beteende blir något ryckigt. Översvängar har ofta samband med integralverkan och kan minskas genom att integraltiden ökas. Med hjälp av LTI Viewer kan vi prova ut en integraltid som gör att vi undviker översvängen.

13. Välj en ny integraltid och skriv in regulatorn på nytt under namnet `GR1` (Obs att integraltiden påverkar både täljare och nämnare i GR). Beräkna även ny kretsöverföring och servosystem för den nya regulatorn med namnen `GK1`, `GTOT1`.
14. *Importera `GK1` och `GTOT1` i dina ltiviewer-fönster. Prova dig fram till du hittar en integraltid som gör att översvängen försvinner. Kontrollera resultatet genom att under *Edit*-menyn välja *Refresh systems* i LTI Viewer.

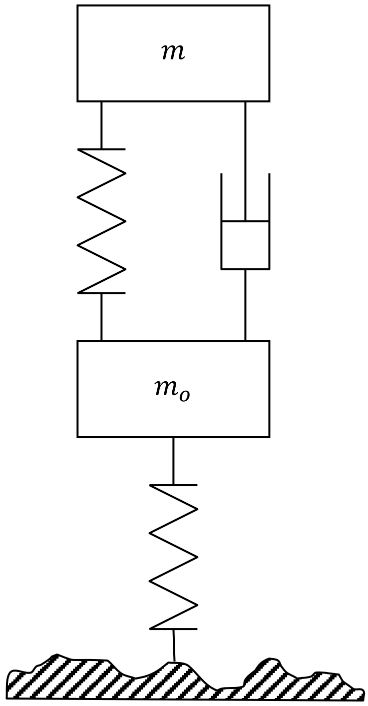
När vi har fått bort översvängen så bör detta också avspeglas i ändrad fasmarginal och resonanstopp, vilket vi nu ska undersöka.

15. *Starta en ny ltiviewer och importera `GK` och `GK1`. Plotta kretsöverföringarnas Nyquistkurvor genom att byta Plot type till Nyquist. Använd *Characteristics* för att ta fram fasmarginalen. Zooma in i figuren (knappen med ett plustecken i ett förstoringsglas) så att enhetscirkeln fyller ut ritytan så att det framgår var fasmarginalen är avläst. Hur har fasmarginalen ändrats med den nya integraltiden?
16. *I din ltiviewer för slutna systemets Bodediagram, avläs nya och gamla resonanstoppen för att se hur den ändrats.

Hjulupphängning

Det är inte bara reglerkretsar som kan beskrivas som återkopplade system som vi nu ska illustrera med en kvartsbilsmodell vilket är en enkel modell av fjädringen i en bil. Den bygger på två massor vilka utgörs av en fjärdedel av bilens vikt (m) samt den ofjädrade vikten per hjul (dvs hjul och hjulupphängning sammantaget, m_o). Dessa massor är sammankopplade med en fjäder och en dämpare, vilka tillsammans utgör bilens fjädringssystem. Hjul och hjulupphängning är, i sin tur, sammankopplade med vägen genom en ytterligare fjäder som representerar fjädringen i bilens däck. Om man låter y, z och v representera karossens, hjulets och vägytans vertikala positioner relativt sina jämviktspunkter så kan man ta fram följande kraftbalans för karossen

$$my'' = -k_f(y - z) - b(y' - z') - F \quad (1)$$

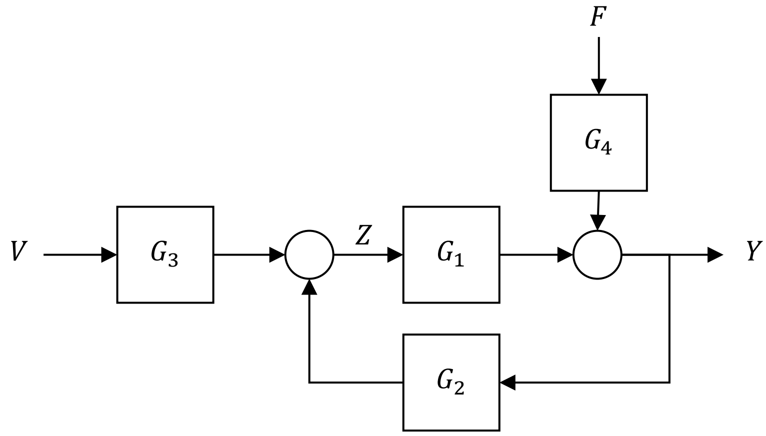


och för den ofjädrade vikten:

$$m_o z'' = k_f(y - z) + b(y' - z') - k_d(z - v) \quad (2)$$

där f är en extern kraft (tex. från passagerare) medan konstanterna k_f och k_d är fjäderkonstanter för fjädring respektive däck och b är dämparens konstant. Antag att bilens vikt är 1200 kg och att den ofjädrade vikten (per hjul) är 50 kg. För att ta reda på k_f och k_d har man gjort ett experiment då en person som väger 100 kg sätter sig mitt i bilen. Resultatet blir att karossen sjunker 11 mm och att däcken trycks ihop 1 mm.

17. Använd informationen ovan för att beräkna värden på fjäderkonstanterna k_f och k_d. *Tips*: Siffrorna gäller för steady-state, vilket erhålls genom att sätta alla derivator i modellen till 0. Var noga med förflyttningarnas riktning. De värden på k_f och k_d som erhålls måste vara positiva.
18. Laplacetransformera kvartsbilsmodellen. Lös ut $Y(s)$ ur (1) och $Z(s)$ ur (2) och hitta överföringsfunktionerna G_1, G_2, G_3och G_4 så att den kan formuleras med nedanstående blockdiagram



19. Gör en blockschemareduktion på ovanstående blockschema och ta fram överföringsfunktionen från $V(s)$ till $Y(s)$ (observera att återkopplingen är positiv).

20. Antag att *b*=0. Skriv in överföringsfunktionerna G_1, G_2 och G_3 (G_4 kommer inte att behövas) med hjälp av funktionen `tf`. Använd Matlab för att beräkna överföringsfunktionen från v till y och kalla den `K`. Applicera funktionen `minreal` på resultatet för att få ner ordningstalet till 4.

Identifiera och dimensionera processparametrar med trial-and-error

För att ta reda på dämparkonstanten har man gjort ett experiment där man kör över en trottoarkant i hög hastighet. Detta ger kvartsbilens stegsvar och det framkom att översvängen i karossens vertikala läge y var ca 25%.

21. Starta en LTI viewer och importera `K`. Stegsvaret bör nu vara mycket oscillativt eftersom $b = 0$ (dvs ingen dämpning).
22. *Hitta ett värde på *b* som ger 25% översväng. Detta görs enklast genom att ändra på *b* i skriptet `lab2.m` och sedan köra om det (med gröna triangeln) och göra *Refresh system* i LTI

viewer. *Tips*: Det krävs ganska stora värden på b (flera tusen) för att minska oscillationen, men alltför stora värden kommer istället att ge oscillation i däcket.

Karossens vertikala acceleration y" är viktig för passagerarnas komfort. I Laplaceplanet blir accelerationen $s^2Y(s)$. På ojämna vägar är det fördelaktigt att minska toppvärdet på y" vilket kan åstadkommas genom att justera bilens dämpare.

23. Spara ditt erhållna `K` under ett nytt namn, genom att skriva `K0=K;` (inte i skriptet utan direkt på kommandoraden) och importera det i LTI Viewer tillsammans med `K`. På det sättet har du kvar det att jämföra med när du i nästa uppgift gör ändringar i `K`.
24. Beräkna överföringsfunktionen från v till y" genom att multiplicera `K` med överföringsfunktionen s^2, implementerad med funktionen `tf`. Kalla resultatet `K2`. Spara även en kopia under namnet `K20` och importera både `K2` och `K20` i en ny LTI Viewer.
25. *Hitta ett värde på b som gör att toppvärdet på y" minskar med ca 20% jämfört med värdet du fick fram i punkt 22. Hur ändras beteendet i y när du ändrar b på detta sätt?

Simulera beteende i tidsplanet

Simulera beteendet för dynamiska system (även olinjära) kan göras bla med Simulink, som är ett grafiskt verktyg under Matlab. I denna lab ska vi istället använda oss av funktionen `lsim` som kan användas för att simulera linjära system. Vi ska använda funktionen för att undersöka hur bilen beter sig då vi kör på en serie tjälskott som vi antar kan beskrivas som en sinusfunktion.

26. Generera en 2 sek. lång tidsvektor med intervallet 0,01 sek genom att skriva `t=0:0.01:2;`
27. Generera en sinusformad insignal med vinkelfrekvens 30 rad/s och amplitud 0,1 m med `v=0.1*sin(30*t);`
28. *Simulera bilens beteende före och efter ändringen av b med `lsim(K,K0,v,t);`

Du bör nu ha fått en plot med tre sinusfunktioner, där den som har amplitud =0,1 är insignalen, dvs vägens profil. De två andra är bilens vertikala läge när den kör över tjälskotten för de två olika valen av b.

29. *Gör ett Bodediagram för `K` och `K0` i en LTI Viewer. Hur kan man i detta Bodediagram utläsa vilken amplitud svängningarna i plotten från punkt 28 borde ha? Med vilket av de två valen av b vore det bekvämare att åka över dessa tjälskott? *Tips*: Genom att klicka på en kurva får man fram värdet i den punkten. Det går även att zooma in genom att välja knappen med ett förstoringsglas och ett plustecken och sedan ruta in det område man vill zooma in med vänster musknapp.