Uniwersytet Zielonogórski
Instytut Sterowania i Systemów Informatycznych

Laboratorium Teorii Sterowania

Modelowanie niepewności i analiza odporności układów

Wprowadzenie

Skrzynka narzędziowa Robust Control Toolbox pozwala tworzyć niepewne elementy, takie jak parametry fizyczne, których wartości nie są znane dokładnie i połączyć te elementy w niepewnych modelach obiektów regulacji. Następnie można łatwo przeanalizować wpływ niepewności na jakość regulacji układu.

Dla przykładu rozważmy następujący model obiektu regulacji

$$P(s) = \frac{\gamma}{\tau s + 1}$$

gdzie

- $\gamma \in [3, 5]$
- τ ma nominalną wartość 0.5 z możliwością 30% zmiany wartości.

Model takiego obiektu modelujemy w następujący sposób (korzystamy z polecenia `ureal`):

```
gamma = ureal('gamma',4,'range',[3 5]);
tau = ureal('tau',.5,'Percentage',30);
P = tf(gamma,[tau 1])
```

Przypuśćmy, że mamy już zaprojektowany regulator C dla nominalnego układu ($\gamma = 4$, $\tau = 0.5$). Aby przekonać się jak zmiany wartości parametrów γ i τ wpływają na działanie obiektu regulacji i jakość regulacji układu zamkniętego należy w pierwszej kolejności utworzyć zamknięty układ regulacji (obiekt `CLP`)

```
KI = 1/(2*tau.Nominal*gamma.Nominal);
C = tf(KI,[1 0]);
CLP = feedback(P*C,1)
```

Możemy teraz utworzyć zbiór 20 losowych prawdopodobnych wartości niepewnych parametrów γ i τ i dla każdych z tych wartości narysować odpowiednie charakterystyki odpowiedzi skokowych obiektu i układu zamkniętego

```
subplot(2,1,1); step(usample(P,20)), title('Plant response (20 samples)')
subplot(2,1,2); step(usample(CLP,20)), title('Closed-loop response (20 samples)')
```

Model silnika DC z niepewnością parametrów

W celu lepszego zobrazowania możliwości modelowania niepewności, skupimy się na modelu sinika prądu stałego (DC) w modelu którego wybrane parametry nie będą dokładnie znane (lub będą mogły się zmieniać). Dodatkowo będziemy mogli badać wpływ niemodelowanej dynamiki obiektu. Przeprowadzimy analizę odporności (ang. *robustness*) układu regulacji na zmianę parametrów obiektu.

Nominalny model silnika prądu stałego definiowany jest przez następujące parametry

- rezystancję uzwojeń wirnika R,
- indukcyjność uzwojeń wirnika L,
- stałą elektromotoryczną Kb
- stałą mechaniczną Km
- liniową aproksymację tarcia lepkiego Kf
- moment bezwładności J.

Zakładamy, że rezystancja i indukcyjność uzwojeń silnika nie są dokładnie znane i mogą się zmieniać w granicach $\pm 40\%$ od ich wartości nominalnych. Kolejny raz użyjemy funkcji `ureal` do konstrukcji obiektów zawierających niepewność parametryczną

```
R = ureal('R',2,'Percentage',40);
L = ureal('L',0.5,'Percentage',40);
```

Dla uproszczenia zakładamy również, że parametry Kf i Kb mają nieznane choć niezmienne wartości - w praktyce jest to mało realistyczne założenie. Przyjmiemy, że nominalne wartości stałych Kf i Kb to 0.015 i zakres możliwych wartości jest pomiędzy 0.012 i 0.019.

```
K = ureal('K',0.015,'Range',[0.012 0.019]);
Km = K;
Kb = K;
```

Stała tarcia lepkiego Kf ma nominalną wartość 0.2 z możliwością zmian ±50% wokół tej wartości. W środowisku `Matlab` definiujemy ten parametr w następujący sposób:

```
Kf = ureal('Kf',0.2,'Percentage',50);
```

W silniku prądu stałego prąd (ang. *current*) w obwodzie elektrycznym oraz moment obrotowy są funkcjami przyłożonego napięcia (ang. *applied voltage*) i prędkości obrotowej (ang. *angular speed*). Utwórzmy zatem odpowiednią transmitancję opisującą relację pomiędzy tymi parametrami i przyjmijmy prędkość obrotową jako dodatkową zmienną wyjściową do późniejszych zastosowań

```
H = [1;0;Km] * tf(1,[L R]) * [1 -Kb] + [0 0;0 1;0 -Kf];
H.InputName = {'AppliedVoltage';'AngularSpeed'};
H.OutputName = {'Current';'AngularSpeed';'RotorTorque'};
```

Jak łatwo zauważyć transmitancja H reprezentuje obiekt o wielu wejściach i wielu wyjściach (MIMO) z niepewnościami parametrów.

Silnik zazwyczaj porusza pewne ciało materialne poprzez wał napędowy, którego dynamiczne cechy są relacją zastosowanego momentu obrotowego na zmiany prędkości kątowej wału. W przypadku ciał sztywnych (ang. *rigid body*) relacja ta jest stała. Bardziej realistycznym założeniem jest to,że model może zawierać nieznane tłumione rezonanse. Użyjemy obiektu `ultidyn` aby zamodelować modelować niepewną liniową i niezmienną w czasie dynamikę. Wartość nominalna bezwładności ciała sztywnego jest ustawiony na 0.02, a my dodamy 15% niepewności dynamiki w postaci multiplikatywnej:

```
J = 0.02*(1 + ultidyn('Jlti',[1 1],'Type','GainBounded','Bound',0.15,...
   'SampleStateDim',4));
```

W tym momencie możliwe jest powiązanie prędkości kątowej (`AngularSpeed`) będącej wejściem input z momentem obrotowym wału (`RotorTorque`) stanowiącym wyjście poprzez moment bezwładności J, który jest niepewnym parametrem. Użyjemy to tego polecenia `lft`. Prędkość kątowa wału (`AngularSpeed`) jest dana znaną zależnością równą `RotorTorque/(J*s)`, dlatego aby uzyskać taką zależność musimy połączyć 3-cie wyjście obiektu `H` (tj. `RotorTorque`) z drugim wejściem (tj. `AngularSpeed`). W wyniku takiej operacji uzyskujemy układ z jednym wejściem (`AppliedVoltage`) i dwoma wyjściami (`Current` i `AngularSpeed`).

```
Pall = lft(H, tf(1,[1 0])/J);
```

Do dalszych rozważań wybieramy tylko prędkość kątową wału (*AngularSpeed*) czyli wyjście numer 2.

```
P = Pall(2,:)
```

Otrzymujemy zatem układ `P`, który ma jedno wejście i jedno wyjście (SISO). W celu przeprowadzenia analizy układu regulacji, przyjmiemy, że zastosowany regulator jest postaci

```
Cont = tf(84*[.233 1],[.0357 1 0]);
```

Regulator ten został uzyskany w standardowej procedurze syntezy regulatora (np. z użyciem narzędzi `SISOtool`) dla nominalnego modelu obiektu sinika DC.

W pierwszej kolejności porównajmy odpowiedzi skokowe nominalnego modelu sinika DC (kolor czerwony) z różnymi odpowiedziami modelu niepewnego (kolor niebieski)

```
clf
step(P.NominalValue,'r-+',usample(P,20),'b',3)
legend('Nominal','Samples')
```

Możemy również porównać charakterystyki Bode'go obiektu nominalnego (kolor czerwony) i przykładowych niepewnych modeli (kolor niebieski) silnika DC.

```
om = logspace(-1,2,80);
Pg = ufrd(P,om);
bode(usample(Pg,25),'b',Pg.NominalValue,'r-+');
legend('Samples','Nominal')
```

W dalszej kolejności przeprowadzimy analizę jakości regulacji układu zamkniętego. Jako wstęp do tej analizy sprawdzimy wielkości zapasu fazy i wzmocnienia układu nominalnego.

```
margin(P.NominalValue*Cont)
```

Jak widać, nominalny układ wykazuje dużą odporność na potencjalną niepewność parametrów (PM=54.3 deg, GM=10.5). Do analizy klasycznych marginesów stabilności możemy zastosować polecenie `loopmargin`, które, w szczególności, jest bardzo użyteczne gdy analizujemy układy MIMO.

W przypadku analizy układów z niepewnościami (a takim jest nasz obiekt silnika prądu stałego) możemy użyć funkcji `wcmargin` aby określić najgorszą możliwą (ang. *worst-case*) wartość zapasu fazy i wzmocnienia. Funkcja `wcmargin` wyznacza najgorszą możliwą zmianę zapasu i fazy dla każdej pary wejść i wyjść (gdy analizujemy układy MIMO). Analiza najgorszego przypadku pokazuje możliwą degradację wartości zapasów wzmocnienia i fazy, które dla tego przypadku mogą zmieniać się odpowiednio od 11 dB i 59 deg (w przypadku modelu nominalnego) do 1.2dB i 8 deg w przypadku występowania wszystkich najgorszych wartości niepewnych parametrów w modelu sinika DC (a jest ich 5 jak pamiętamy).

```
wcmarg = wcmargin(Pg,Cont);
```

Jak pamiętamy, funkcja czułości, lub inaczej transmitancja uchybowa, $S(j\omega)$, gdzie

$$S = (1 + P * C)^{-1}$$

jest podstawowym wskaźnikiem jakości regulacji układów zamkniętych. Z tego właśnie powodu wykreślimy funkcje czułości układu nominalnego i porównamy go z układem z niepewnościami

```
S = feedback(1,P*Cont);
bodemag(usample(S,20),'b',S.Nominal,'r-+');
legend('Samples','Nominal')
```

W dziedzinie czasu, funkcja czułości wskazuje efektywność eliminacji skokowych zakłóceń. Zobrazujmy to poniższymi wykresami

```
step(usample(S,20),'b',S.Nominal,'r-+',3);
title('Disturbance Rejection')
legend('Samples','Nominal')
```

W naszej analizie możemy również użyć funkcji `wcgain` aby wyznaczyć najgorszą możliwą wartość funkcji czułości układu z niepewnościami (czyli poszukujemy maksymalnej wartości wzmocnienia pośród wszystkich czestotliwości). Alternatywnie możemy użyć funkcji `wcsens` w celu wyznaczenia tej wartości.

```
Sg = ufrd(S,om);
[maxgain,worstuncertainty] = wcgain(Sg);
maxgain
```

Korzystając z funkcji `usubs` możemy dokonać podstawienia wartości najgorszego możliwego przypadku pojawienia się niepewności (dostępnego w zmiennej `worstuncertainty`) do funkcji czułości `S`. W rezultacie otrzymujemy funkcję czułości `Sworst` dla najgorszego przypadku. należy tutaj zaznaczyć, że największe wzmocnienie funkcji `Sworst` odpowiada dolnej granicy wyznaczonej z użyciem funkcji `wcgain`.

```
Sworst = usubs(S,worstuncertainty);
Sgworst = frd(Sworst,Sg.Frequency);
norm(Sgworst,inf)
maxgain.LowerBound
```

Porównajmy teraz odpowiedzi skokowe funkcji czułości układu nominalnego i najgorszego możliwego przypadku wartości niepewności w modelu silnika

```
step(Sworst,'b',S.NominalValue,'r-+',6);
title('Disturbance Rejection')
legend('Worst-case','Nominal')
```

Ostatecznie, wyświetlmy wykresy Bode'go (tylko wzmocnienie) funkcji czułości dla układu nominalnego i przypadku najgorszego. Zaobserwuj, że maksymalna wartość funkcji `Sworst` występuje przy częstotliwości `maxgain.CriticalFrequency`

```
bodemag(Sg.NominalValue,'r-+',Sgworst,'b');
legend('Worst-case','Nominal')
hold on
semilogx(maxgain.CriticalFrequency,20*log10(maxgain.LowerBound),'g*')
hold off
```

Zadania

1. Zapoznaj się z możliwościami środowiska Matlab do analizowania wpływu niepewności parametrów. Szczegółowo zapoznaj się z powyższym przykładem analizy niepewności. Zwróć szczególna uwagę na funkcje umożliwiające definiowanie zakresów niepewności. Spróbuj zmienić zakres niepewności wybranych parametrów i dokonaj analizy wpływu tych zmian na jakość regulacji układu zamkniętego

2. Dla nominalnego obiektu silnika DC zaprojektuj regulator PID i porównaj jakość regulacji otrzymago układu regulacji z zamieszczonym przykładem

3. Załóżmy, że dany jest pewien obiekt mechaniczny `P` składający się z dwóch mas i sprężyny. Jego model może zostać zbudowany w środowisku Matlab w następujący sposób

   ```
   k = ureal('k',1,'percent',20); % sprezyna o stalej sprezysrosci k=1 + niepewnosc
   m1 = ureal('m1',1,'percent',20); % masa m1=1kg + niepewnosc
   m2 = ureal('m2',1,'percent',20); % masa m2=1kg + niepewnosc
   G1 = 1/s^2/m1;
   G2 = 1/s^2/m2;
   F = [0;G1]*[1 -1]+[1;-1]*[0,G2]
   P = lft(F,k)  % nasz obiekt do analizy
   ```

   Dla tego obiektu został dobrany następujący regulator

   ```
   s = zpk('s');
   C = 100*ss((s+1)/(.001*s+1))^3;
   ```

   Bazując na przedstawionym przykładzie dokonaj analizy jakości regulacji układu zamkniętego. W dalszej kolejności zbadaj dokładnie wpływ niepewności parametrów - użyj polecenia

   ```
   [StabilityMargin,Udestab,REPORT] = robuststab(T);
   ```

   gdzie `T` oznacza transmitancję układu zamkniętego.