

Föreläsning 9, Bestämning av tidsdiksreta överföringsfunktioner

Reglerteknik, IE1304

Innehåll

- 1 Kapitel 17.1. Inledning
- 2 Kapitel 17.1-2, 19.1. Diskretisering
- 3 Kapitel 17.3-17.4. Experimentell bestämning

Innehåll

- 1 Kapitel 17.1. Inledning
- 2 Kapitel 17.1-2, 19.1. Diskretisering
- 3 Kapitel 17.3-17.4. Experimentell bestämning

Hur bestämma den diskreta överföringsfunktionen för en process

Vi kommer i första hand betrakta det system vi vill styra, här kallad processen, inte regulatorer eller andra komponenter. Det finns två grupper av metoder att ta fram processens diskreta överföringsfunktion.

- 1 *Diskretisering*, omvandla den kontinuerliga överföringsfunktionen till diskret form. Förutsätter att den kontinuerliga överföringsfunktionen är känd.
- 2 *Experimentellt*, ge processen olika insignaler och mät dess utsignal. Beräkna en överföringsfunktion som korrekt beskriver sambandet mellan given insignal och mätt utsignal.

Innehåll

- 1 Kapitel 17.1. Inledning
- 2 Kapitel 17.1-2, 19.1. Diskretisering
- 3 Kapitel 17.3-17.4. Experimentell bestämning

Olika diskretiseringsmetoder

- Det går inte att hitta en diskret överföringsfunktion som exakt i alla avseenden beter sig som dess analoga motsvarighet. Det beror på att den diskreta komponenten "inte vet" vad som sker mellan samplingstidpunkterna.
 - Här följer några olika diskretiseringsmetoder som alla har sina för- och nackdelar.
- 1 *Impulsinvariant transform* Den diskreta överföringsfunktionen har samma impulssvar som den kontinuerliga.
 - 2 *Steginvariant transform* Den diskreta överföringsfunktionen har samma stegsvar som den kontinuerliga.
 - 3 *Rampinvariant transform* Den diskreta överföringsfunktionen har samma rampsvar som den kontinuerliga.

Olika diskretiseringsmetoder, forts

- 4 *Bilineär transform (Tustins approximation)* Bevarar stabilitet och minimumfas-egenskap. Definitionen för hur en Laplace-transform transformeras till en z-transform är att sätta $z = e^{sT}$, där T är samplingstiden, vilket ger $s = \frac{1}{T} \ln(z)$. Tustins approximation bygger på att detta samband approximeras med $s = \frac{2}{T} \cdot \frac{z-1}{z+1}$
- 5 *Euler-transform* Den enklaste metoden, tanken är att om vi känner värdet och derivatan i en viss punkt, $y(k)$, kan vi räkna ut ett approximativt värde för föregående punkt, $y(k-1)$ genom att gå samplingstiden T bakåt i derivatans riktning. Funktionen approximeras alltså med en rät linje under ett samplingsintervall. Detta görs genom att byta ut s i den kontinuerliga överföringsfunktionen mot $\frac{z-1}{zh}$.

Diskretisering av *processens* överföringsfunktion

- Steginvariant transform lämpar sig väl för diskretisering av en process överföringsfunktion eftersom processens insignal, vid diskret reglering, består av en serie steg.
- En tabell för steginvariant diskretisering finns på sid 314 (och 516).
- Om den sökta överföringsfunktionen inte kan hittas i tabellen kan vi göra en steginvariant transform med hjälp av formeln

$$H(z) = (1 - z^{-1}) \cdot Z \left[L^{-1} \left[\frac{G(s)}{s} \right]_{t=kh} \right]$$

- Bevis för formeln finns på sid 315-316.
- Övningsuppgifter: 17.1b-c, 17.2, 17.3, 17.4

Diskretisering av *regulatorns* överföringsfunktion

- Regulatorns insignal är inte en serie steg, varför steginvariant transform lämpar sig sämre här.
- Vanliga transformer för regulatorns överföringsfunktion är rampinvariant, bilineär och euler. Rampinvariant och bilineär ger bättre värden medans euler är lättare. Normalt är dock alla acceptabla för låga och medelhöga frekvenser, vilket är vad vi normalt är intresserade av.
- För att få en bra transform måste samplingsintervallet vara kort i förhållande till regulatorns tidskonstanter, kanske 10% av den dominerande tidskonstanten (dvs den som ger snabbast insvängning).
- Exempel på resultatet av olika transformer finns i bilden på sid 350.

Diskretisering med Matlab

- Matlab har en funktion, `c2d` , för att diskretisera en kontinuerlig överföringsfunktion.
- Syntaxen är `c2d(G, T, metod)` , där G är den kontinuerliga överföringsfunktionen, T samplingstiden och `metod` diskretiseringsmetoden. Möjliga värden för metoden är bland annat 'zoh' (zero order hold) som ger steginvariant diskretisering, 'foh' (first order hold) som ger rampinvariant diskretisering och 'tustin' som ger bilineär diskretisering.

Diskretisering med Matlab, forts

- Följande matlab-kommandon visar den analoga överföringsfunktionen, steginvariant diskretisering och tustin-diskretisering i samma fönster. Använd förstoringsglaset i fönstret med graferna för att zooma in så syns skillnaden bättre.

```
G=tf(3,[2 1])
```

```
step(G)
```

```
hold
```

```
GZOH = c2d(G, 0.1, 'zoh')
```

```
step(GZOH)
```

```
GTUST = c2d(G, 0.1, 'tustin')
```

```
step(GTUST)
```

Övningsuppgifter

- Eventuellt 19.2a-b, 19.4b,d,e

Innehåll

- 1 Kapitel 17.1. Inledning
- 2 Kapitel 17.1-2, 19.1. Diskretisering
- 3 Kapitel 17.3-17.4. Experimentell bestämning

För- och nackdelar med experimentell bestämning

- Experimentell bestämning innebär att vi ger processen olika insignaler och mäter dess utsignal. Därefter beräknar vi en överföringsfunktion som korrekt beskriver sambandet mellan given insignal och mätt utsignal.
- Fördelen med experimentell bestämning jämfört med diskretisering av en analog modell är att det inte är nödvändigt att ställa upp en matematisk modell för processen.
- Nackdelen med experimentell bestämning jämfört med diskretisering av en analog modell är att vi måste “provköra” processen med varierande indata, även indata som simulerar oönskade tillstånd. Detta kanske inte är möjligt med en komplex eller dyr process.

Minsta kvadratmetoden

- 1 Ansätt en differensekvation som beskriver processen,
 $y(k) = ay(k-1) + bu(k-1)$, där u är processens insignal och y dess utsignal.
- 2 Ge processen en viss insignal och mät dess utsignal.
- 3 Nu har vi följande ekvationssystem, där $y(1)$, $y(2)$, $y(3)$, $u(1)$, $u(2)$ och $u(3)$ är kända. Löser vi ekvationssystemet får vi konstanterna a och b , och därmed har vi överföringsfunktionen.

$$\begin{cases} y(3) = ay(2) + bu(2) \\ y(2) = ay(1) + bu(1) \end{cases}$$

Minsta kvadratmetoden, forts

- Proceduren på föregående sida fungerar tyvärr inte eftersom processen alltid är utsatt för störningar och därför varierar dess överföringsfunktion. Varje gång proceduren upprepas får vi olika värden på a och b .
- Lösningen är att ta fram väsentligt mer än två par av in- och utsignal och sedan beräkna ett “medelvärde” av a och b .
- Med minsta kvadratmetoden beräknas a och b så att summan, J , av kvadraten på felen, $e(k)$, minimeras, $J = \sum_{k=1}^n e(k)^2$, där $e(k) = y(k) - ay(k-1) - bu(k-1)$ och n är antalet ekvationer.

Att lösa ett överbestämt ekvationssystem

- Antag att vi tar fram 8 par av in-/utsignal. Det ger följande överbestämda (fler ekvationer än obekanta) ekvationssystem.

$$\begin{cases} y(8) = ay(7) + bu(7) \\ y(7) = ay(6) + bu(6) \\ y(6) = ay(5) + bu(5) \\ y(5) = ay(4) + bu(4) \\ y(4) = ay(3) + bu(3) \\ y(3) = ay(2) + bu(2) \\ y(2) = ay(1) + bu(1) \\ y(1) = ay(0) + bu(0) \end{cases}$$

Att lösa ett överbestämt ekvationssystem, forts

- Om vi ställer upp det på matrisform, se sid 320, kan vi uttrycka det som $\mathbf{y} = \mathbf{A} \cdot \mathbf{r}$. Lösningen enligt minsta kvadratmetoden blir då $\mathbf{r} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$, där $\mathbf{A}^T$ är transponatet och $\mathbf{A}^{-1}$ inversen till $\mathbf{A}$.
- I Matlab löses den överbestämda ekvationen enligt minsta kvadratmetoden med kommandot $r = A \backslash y$

Praktiska aspekter på minsta kvadratmetoden

Hur många mätvärden? Desto fler desto bättre. Vid normala störningar är några hundra eller några tusen lämpligt.

Vilket ordningstal ska differensekvationen ha? För lågt ger onödigt grov uppskattning, för högt ger överanpassning till mätdataserien, vilket innebär att differensekvationen stämmer sämre med andra mätdataserier. Två till fyra är vanliga gradtal, en metod att bestämma gradtal visas i bilden på sid 324.

Hur ska insignalen väljas? Det är viktigt att insignalen varierar så mycket som möjligt, vitt brus är lämpligt.

Hur långt ska samplingsintervallet vara? Väsentligt mindre än procesens tidskonstant. Ett bra värde är 10% av processens dominerande tidskonstant.

Övningsuppgifter

- Eventuellt 17.8