

Hemuppgift 3, SF1851 Optimeringslära för E

Examinator: Per Enqvist, penqvist@math.kth.se.

Hemtalsassistent: Maja Karasalo, karasalo@math.kth.se.

I denna hemuppgift är det tillåtet att samarbeta i grupper om högst två personer. Begränsat samarbete med andra grupper är också tillåtet men måste nämnas i rapporten. En rapport per grupp ska lämnas in, och ni ska berätta med egna ord hur ni har löst problemet. Det är inte tillåtet att kopiera delar av annans grupps rapport! Svaren till frågorna i uppgiften ska finnas i den skriftliga huvudrapporten. Relevanta utskrifter av matlabkörningar och plottar ska bifogas till rapporten. Fullständig programkod ska finnas tillgänglig på begäran.

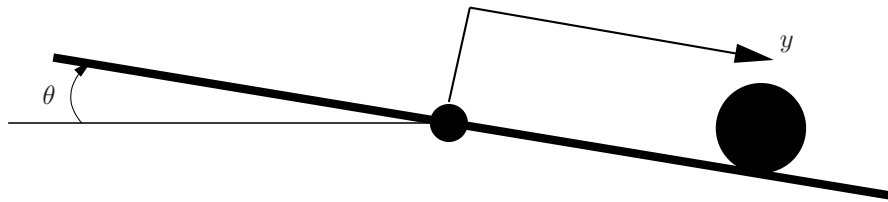
Skriv era namn, personnummer och emailadress på framsidan av rapporten.

Rapporten ska lämnas in (utskriven på papper) innan måndagen den 12:e oktober klockan 17.00. Notera att denna deadline är strikt.

I den här uppgiften skall du balansera en kula på en balk med hjälp av modelprediktiv reglering (MPC). Genom att variera tippvinkeln θ med en elektrisk motor skall kulan föras till balkens mittpunkt. Kulans avvikelse från balkens mittpunkt betecknas y och uppfyller följande differentialekvation

$$\ddot{y} = b \sin \theta \approx b\theta$$

där b är en konstant som beror på diverse systemparametrar. Här antar vi det normaliserade värdet $b = 1$. Om du vill veta mer om systemet och hur det modelleras så rekommenderar vi att du söker upp dokumenten [1, 4] på Internet.



Figur 1: Kulan antages rulla i ett spår längs balken och kan därför bara falla av balken då den når endera ändpunkten.

I fortsättningen kommer vi att kalla vinkeln ovan för u i stället för θ (eftersom styrvariabler ofta betecknas u inom reglertekniken).

Vi ska styra systemet med en datoralgoritm och behöver därför en diskrettidsbeskrivning av ovanstående differentialekvation. Vi använder följande tillståndsbeskrivning

$$\begin{aligned}\mathbf{z}_{k+1} &= \mathbf{\Phi}\mathbf{z}_k + \mathbf{\Gamma}u_k \\ y_k &= \mathbf{C}\mathbf{z}_k\end{aligned}\tag{1}$$

där T är samplingstiden, $u_k = u(kT) = \theta(kT)$, $y_k = y(kT)$, $\dot{y}_k = \dot{y}(kT)$,

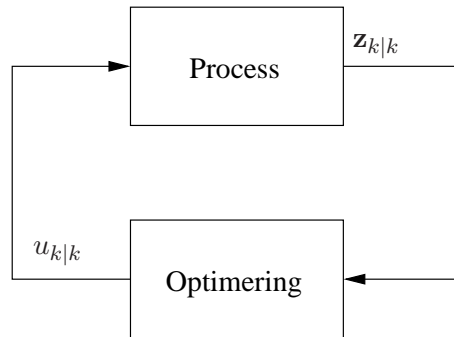
$$\mathbf{z}_k = \begin{bmatrix} y(kT) \\ \dot{y}(kT) \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \quad \mathbf{C} = [1 \quad 0]$$

Att detta är en rimlig approximation av den kontinuerliga dynamiken följer av att tillståndsformen i (1) är ekvivalent med följande differensekvation

$$\frac{1}{T^2}(y_{k+2} - 2y_{k+1} + y_k) = \frac{1}{2}(u_{k+1} + u_k),$$

där vänsterledet approximerar $\ddot{y}(t)$ medan högerledet approximerar $u(t)$.

Vid modelprediktiv reglering löser man i varje samplingsintervall ett optimeringsproblem som definierar styrsignalen, se Figur 2.



Figur 2: I varje samplingsintervall mäter man tillståndet i processen (dvs \mathbf{z}_k ; kulans position och hastighet). Med hjälp av en optimeringsalgoritm bestämmer man sedan en styrsignal (dvs en ny tippvinkel) som appliceras på processen. Detta förfarande repeteras varje samplingstidpunkt.

I MPC algoritmen nedan inför vi en ny notation $\mathbf{z}_{k|k}$ och $u_{k|k}$, vilket är motiverat av att optimeringsproblemet definieras över en relativ tidsaxel med hjälp av predikterade framtida tillståndvariabler. Detta illustreras i Figur 3. Notationen $\mathbf{z}_{k|k}$ indikerar att vi mäter tillståndet \mathbf{z} vid tiden k , vilket sedan utgör begynnelsevärdet över den relativa tidsaxeln. De predikterade tillstånden betecknas $\mathbf{z}_{k+l|k}$, vilket skall tolkas som att detta tillstånd är en estimation av systemets framtida tillstånd vid tidpunkten l längs den relativa tidsaxeln givet att startvärdet var $\mathbf{z}_{k|k}$.

Den allmänna MPC algoritmen definieras av följande steg.

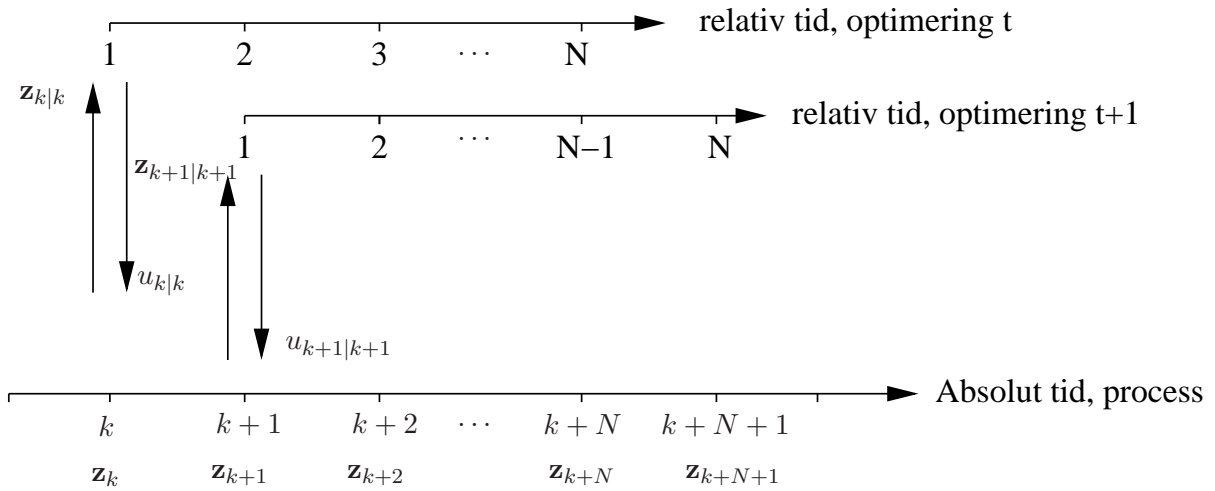
0. Sätt $k := 1$.

1. Mät $\mathbf{z}_{k|k}$.

2. Bestäm styrsignalerna $u_{k|k}, u_{k+1|k}, \dots, u_{k+N-1|k}$ mha ett optimeringsproblem.

3. Applicera $u_{k|k}$.

4. Om $k < M$, sätt $k := k + 1$ och gå tillbaka till 1.



Figur 3: Vid varje samplingstidpunkt mäts processens tillstånd. Detta tillstånd döps till $\mathbf{z}_{k|k}$ och utgör begynnelsevärde för ett optimeringsproblem som definieras över en relativ tidsaxel och involverar predikterade tillstånd $\mathbf{z}_{k+l|k}$ och motsvarande styrsignaler $u_{k+l|k}$.

De predikterade tillstånden erhålls genom att rekursivt lösa (1) i framåtriktning

$$\begin{aligned}
 \mathbf{z}_{k+1|k} &= \Phi \mathbf{z}_{k|k} + \Gamma u_{k|k} \\
 \mathbf{z}_{k+2|k} &= \Phi \mathbf{z}_{k+1|k} + \Gamma u_{k+1|k} \\
 &\vdots \\
 \mathbf{z}_{k+N|k} &= \Phi \mathbf{z}_{k+N-1|k} + \Gamma u_{k+N-1|k}
 \end{aligned} \tag{2}$$

Syftet med optimeringsproblemet är att driva kulan mot balkens mittpunkt utan att använda för mycket styrenergi. Ett naturligt optimeringskriterium för att uppnå detta syfte är följande kvadratiske optimeringsproblem

$$\begin{aligned}
 \text{minimera} \quad & q_f |\mathbf{z}_{k+N|k}|^2 + \sum_{l=0}^{N-1} (q |\mathbf{C} \mathbf{z}_{k+l|k}|^2 + r u_{k+l|k}^2) \\
 \text{då} \quad & \mathbf{z}_{k+l+1|k} = \Phi \mathbf{z}_{k+l|k} + \Gamma u_{k+l|k} \quad l = 0, 1, \dots, N-1
 \end{aligned} \tag{3}$$

där q_f , q , och r är positiva parametrar som beskriver kostnaden. Termerna i summan syftar till att begränsa storleken på kulans avvikelse från origo samt storleken på styrsignalen under tidsintervallet $[k, k+N-1]$. Den första termen syftar till att göra såväl kulans position som hastighet liten vid diskreta tiden $k+N$. Notera att $\mathbf{z}_{k|k}$ är känd medan övriga storheter $\mathbf{z}_{k+1|k}, \dots, \mathbf{z}_{k+N|k}, u_{k|k}, \dots, u_{k+N-1|k}$ är variabler vid optimeringen. Efter optimeringen använder vi bara $u_{k|k}$. Det kan tyckas onödigt att optimera över en massa variabler som sedan inte används, men orsaken är att optimeringsproblemet ger en bättre beskrivning av systemet ju längre tidshorisont N man betraktar under optimeringen. Anledningen till att man upprepar optimeringen varje samplingsstillfälle är att varje mätning $\mathbf{z}_{k|k}$ ger ny information om systemets tillstånd och på så sätt kompenserar man för störningar och modellfel som alltid förekommer i praktiska tillämpningar. För utförligare beskrivningar av MPC tekniken hänvisar vi till [3, 2].

I de följande uppgifterna skall du implementera kvadratiska optimeringsproblem för MPC styrning av kulan på balken. I appendix finns Matlabkod som kan definiera strukturen på din kod. Det finns dessutom några tips för hur du kan konstruera matriser. Din uppgift blir att definiera matriserna som specificerar optimeringsproblemet samt att experimentera med algoritmen.

1. Skriv om optimeringsproblemet (3) på formen

$$\begin{aligned} &\text{minimera} \quad \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} \\ &\text{då} \quad \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq} \end{aligned}$$

där $\mathbf{x} = \left(\mathbf{z}_{k+1|k}^T \quad \dots \quad \mathbf{z}_{k+N|k}^T \quad u_{k|k} \quad \dots \quad u_{k+N-1|k} \right)^T$ (notera att $\mathbf{z}_{k|k}$ är given och är ingen variabel i optimeringsproblemet). Notera att högerledet \mathbf{b}_{eq} beror av senaste uppmätta tillståndet $\mathbf{z}_{k|k}$ och måste uppdateras i varje iteration av MPC algoritmen (se kod i appendix). Lös problemet med Matlab för följande parametervärden:

$\mathbf{z}_{0|0} = (0.5 \quad 1)^T$, $r = 1$, samplingstiden $T = 0.1$ samt

- (a) $N = 5$, $q_f = 5$, $q = 2$,
- (b) $N = 10$, $q_f = 5$, $q = 2$,
- (c) $N = 10$, $q_f = 10$, $q = 5$.

Vilka slutsatser kan ni dra vad gäller konvergens till origo?

2. Balken har i praktiken begränsad längd och vi måste införa bivillkor i optimeringsproblemet för att försäkra oss om att kulan inte trillar av balken. Normalt finns det också begränsningar på styrsignalens storlek vilket ger ytterligare bivillkor. Det resulterande optimeringsproblemet kan formuleras som

$$\begin{aligned} &\text{minimera} \quad q_f |\mathbf{z}_{k+N|k}|^2 + \sum_{l=0}^{N-1} (q |\mathbf{C} \mathbf{z}_{k+l|k}|^2 + r u_{k+l|k}^2) \\ &\text{då} \quad \begin{cases} \mathbf{z}_{k+l+1|k} = \mathbf{\Phi} \mathbf{z}_{k+l|k} + \mathbf{\Gamma} u_{k+l|k} & l = 0, \dots, N-1 \\ -1 \leq y_{k+l|k} \leq 1, & l = 1, \dots, N \\ -1.5 \leq u_{k+l|k} \leq 1.5, & l = 0, \dots, N-1 \end{cases} \end{aligned} \quad (4)$$

och din uppgift är att skriva om det på formen

$$\begin{aligned} &\text{minimera} \quad \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} \\ &\text{då} \quad \begin{aligned} &\mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq} \\ &\mathbf{x}_{\text{lower}} \leq \mathbf{x} \leq \mathbf{x}_{\text{upper}} \end{aligned} \end{aligned}$$

samt att lösa det med Matlab för följande parametervärden:

$\mathbf{z}_{0|0} = (0.5 \quad 1)^T$, $r = 1$, samplingstiden $T = 0.1$ samt

- (a) $N = 5$, $q_f = 5$, $q = 2$,
- (b) $N = 10$, $q_f = 5$, $q = 2$,
- (c) $N = 10$, $q_f = 10$, $q = 5$.

Jämför resultaten med uppgift 1 och kommentera.

Dokumentation

En kortfattad rapport skall lämnas in i ett exemplar per grupp om högst två personer. Följande skall finnas med

1. Härledning av blockmatriserna för optimeringsproblemen.
2. Kortfattad diskussion om simuleringarna. Vilka slutsatser kan du dra om hur parametervärdena (N , q_f och q) påverkar konvergensen. Ni får gärna experimentera med fler parametervärden.
3. Matlabplottar. Använd kommandot `subplot` för att erhålla samtliga tre experimenten i respektive uppgift i en figur.
4. Matlabkoden.

Tips för Matlabimplementeringen

Två tips som kan förenkla genereringen av dina blockmatriser.

1. Blockdiagonala matriser kan skapas med kommandot `blkdiag(A,B)`.
2. Kommandot `kron` kan vara användbart. Allmänt gäller att

$$\text{kron}(A, B) = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \vdots & & & \\ a_{n1}B & a_{n2}B & \dots & a_{nn}B \end{bmatrix}$$

Referenser

- [1] LTH Department of Automatic Control. The ball and beam lab process. <http://www.control.lth.se/education/laboratory/bommen.html>.
- [2] M. Morari et. al. Model predictive control course. <http://control.ee.ethz.ch/>.
- [3] Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice. a survey. *Automatica*, 25(3):335–348, 1989.
- [4] P. Wellstead. Ball and beam 1: Basics. www.control-systems-principles.co.uk/whitepapers/ball-and-beam1.pdf.

Lycka till!

Matlabkod

```
%
%----- Basic system model
clear;
T=0.1;
Phi=[1 T;0 1];
Gam=[T^2/2;T];
C=[1 0];
n=size(Phi,1);
m=size(Gam,2);
%
%----- Parameters -----
qf=5;
q=2;
r=1;
N=5;
z0=[0.5;1];
%----- Define matrices and vectors for the QP -----
%
This is for you to do!
%
%----- MPC algorithm -----
M=100; %time horizon
Aineq=[];
bineq=[];
% Om inga gränser på variablerna: xlow=[]; xupp= [];
zt=z0;
yvec=[];
uvec=[];
options=optimset('largescale','off');
for flcnt1=1:M
    beq=AA*zt; % Matrisen AA definierar hur det senast mätta tillståndet zt
               % bestämmer högerledet i likhetsbivillkoret.
    x=quadprog(H,f,Aineq,bineq,Aeq,beq,xlow,xupp,[],options);
    ut=x(n*N+1); % Plocka ut variabeln  $u_{\{k|k\}}$ 
    zt=Phi*zt+Gam*ut;
    yvec=[yvec;C*zt];
    uvec=[uvec;ut];
end
tvec=T*(1:1:M)';
subplot(3,1,1) %For the other two sets of parameters you should change
               %the third index to 2 and 3, respectively.
plot(tvec,yvec,'-',tvec,uvec,'--')
grid
```