Niech dany będzie obiekt liniowy o równaniu:

Z warunkiem początkowym:

Niech będzie dany również wskaźnik jakości o postaci:

Gdzie: **Q –** macierz symetryczna dodatnio określona o wymiarach nxn;

**R –** macierz symetryczna dodatnio określona o wymiarach pxp.

Należy wyznaczyć wektor starowania u, spełniający równanie (1) z warunkiem początkowym (2), który minimalizuje wskaźnik jakości (3). Zadaniem będzie wyznaczyć macierz K wektora sterowania u jako funkcji wektora stanu x, czyli

Powyższe zadanie w literaturze nazywamy **syntezą optymalnego regulatora**.

Korzystając z metody mnożników Lagrange’a lub zasady maksimum Pontriagina można wykazać, że w przypadku obiektu sterowania liniowego i stacjonarnego sterowanie optymalne u jest dane wzorem

Gdzie:

 (5)

**P –** macierz symetryczna i rozwiązanie algebraiczne równania Riccatiego.

Wówczas minimalna wartość wskaźnika jakości wynosi:

Z powyższych rozważań wynika następujący tok postępowania przy wyznaczaniu sterowania optymalnego:

* Mając dane macierze A, B, Q i R wyznaczamy macierz P z równania (6);
* Poszukiwane sterowanie optymalne u wyznaczymy z zależności (4), dla wyznaczonej z równania (5) macierzy K.

Jeżeli macierz **A-BK** jest stabilna otrzymujemy zawsze poprawne rozwiązanie.

# Przykład

Należy wyznaczyć sterowanie u obiektem liniowym stacjonarnym:

Z zadanymi warunkami początkowymi [x1(0) = x0 = 8], [x2(0) = 0]

Minimalizujące wskaźniki jakości o postaci

Dla wartości:

***Listing Maple'a:***

> restart:

> with(plots):

> with(student):

> with(linalg):

> A:=linalg[matrix](2,2,[0,1,-8,-0.3]);

> x:=array([x1,x2]);

> B:=array([0,1]);

> Q:=linalg[matrix](2,2,[1,1,1,3]);

> R:=0.03;

> P:=linalg[matrix](2,2,[p11,p12,p12,p22]);

> BT:=transpose(B);

> AT:=transpose(A);

> C:=evalm(B\*1/R\*BT);

> Riccati:=evalm(P&\*A+AT&\*P+P&\*C&\*P-Q);

> r1:=Riccati[1,1]=0;

> r2:=Riccati[1,2]=0;

> r3:=Riccati[2,1]=0;

> r4:=Riccati[2,2]=0;

> eqnt:={r1,r2,r4};

> wzmoc:=solve(eqnt);

> wzmoc1:=subs(wzmoc[3]);

> p11:=subs(wzmoc1,p11);

> p12:=subs(wzmoc1,p12);

> p22:=subs(wzmoc1,p22);

> P:=linalg[matrix](2,2,[p11,p12,p12,p22]);

> K:=evalm(-1/R\*BT&\*P&\*x);

> E:=eigenvals(A+1/R\*B&\*BT&\*P);

> u :=evalm(1/R\*BT&\*P&\*array([x1(t),x2(t)]));

Lub dla punktu 2 dla r = 7

> u :=evalm(1/R\*BT&\*P&\*array([x1(t),x2(t)-7]));

> with(plots):

> r1:=diff(x1(t),t)=A[1,1]\*x1(t)+A[1,2]\*x2(t)+B[1]\*u;

> r2:=diff(x2(t),t)=A[2,1]\*x1(t)+A[2,2]\*x2(t)+B[2]\*u;

> wp:=x1(0)=8,x2(0)=0;

> Gw:=dsolve({r1,r2,wp},{x1(t),x2(t)},type=numeric);

> odeplot(Gw,[t,x1(t)],0..10);

> odeplot(Gw,[t,x2(t)],0..10);

> odeplot(Gw,[t,u],0..10);

> odeplot(Gw,[x1(t),x2(t)],0..10);

*Otrzymane wyniki:*

***Listing Matlab'a:***

A=[0 1; -8 -0.3];

B=[0 ; 1];

Q=[1 1;1 3];

R=0.03;

[K,P,E]=lqr(A,B,Q,R) %wywolanie procedury

K

P

E

sys=ss(A-B\*K,eye(2),eye(2),eye(2));

t=0:0.01:10;

x=initial(sys,[8;0],t);

x1=[1 0]\*x';

x2=[0 1]\*x';

xx=[x1;x2];

u\_opt=-K\*xx;

subplot(2,2,1); plot(t,x1); grid on; xlabel('t[s]'); ylabel('x1[m]'),

subplot(2,2,2); plot(t,x2); grid on; xlabel('t[s]'); ylabel('x2[m/s]');

subplot(2,2,3); plot(t,u\_opt); grid on; xlabel('t[s]'); ylabel('u\_\_opt[-]');

subplot(2,2,4); plot(x1,x2); grid on; xlabel('x1[m]'); ylabel('x2[m/s]');

*Otrzymane wyniki:*