Seja o sistema

$$\dot{x} = Ax + Bu$$

com o observador

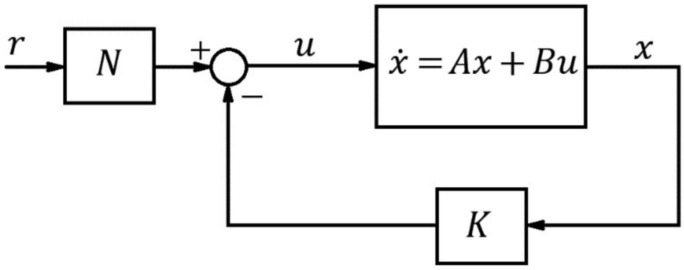
$$y = Cx + Du$$

Como apenas as variáveis do observador *y* estão disponíveis, e como variáveis do vetor dos estados, muitas vezes não tem significado físico, podemos escolher então como referência as variáveis do observador para controlá-las, ou seja seguir uma referência desejada, para tanto, criamos um vetor referência de mesma dimensão da observação. Portanto agora, o desejo de controle, não é apenas estabilizar o estado *x* na posição 0, mas sim estabilizar o sistema em um vetor específico *r* com a mesma dimensão do estado observado, isto é

$$r = \dim(y)$$

Para uma solução em estado estacionário com controle em malha fechada, *y* deve convergir para os valores de *r*.

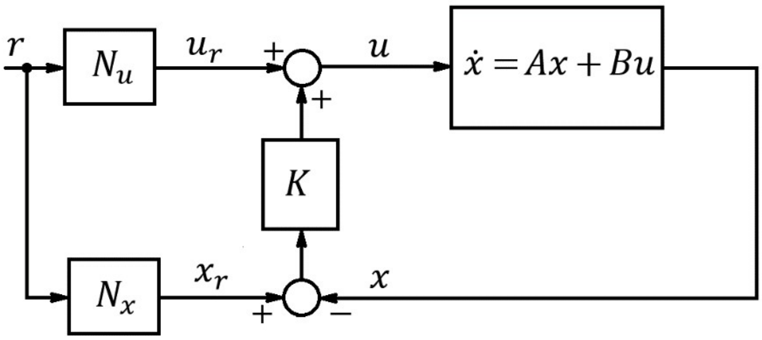
Propondo uma solução do forma



sendo *N* a matriz para compatibilizar dimensão da referência *r* na dimensão de *u*. Portanto o controlador pode ser equacionado como

$$u = Nr - Kx$$

Podemos separar a referência de controle, em duas partes da seguinte forma



Nesse caso o controle seria

$$u = u_r + K(x_r - x)$$

com

$$u_r = N_u r$$

e

$$x_r = N_x r$$

portanto

$$u = N_u r + K(N_x r - x)$$

Igualando as duas estratégias de controle temos

$$Nr - Kx = N_u r + K(N_x r - x)$$
$$Nr - Kx = N_u r + KN_x r - Kx$$
$$Nr = N_u r + KN_x r$$

ou

$$N = N_u + KN_x$$

precisamos agora determinar quem são N_u e N_x

para tanto vamos montar a dinâmica aumentada do sistema

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

na forma

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

como solução para o regime estacionário, ou seja $\dot{x} = \mathbf{0}$, $y = r$, $x = x_r$ e $u = u_r$, portanto

$$\begin{bmatrix} \mathbf{0} \\ r \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix}$$

substituindo x_r e u_r

$$\begin{bmatrix} \mathbf{0} \\ r \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} N_x r \\ N_u r \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} r = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} r$$

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix}$$

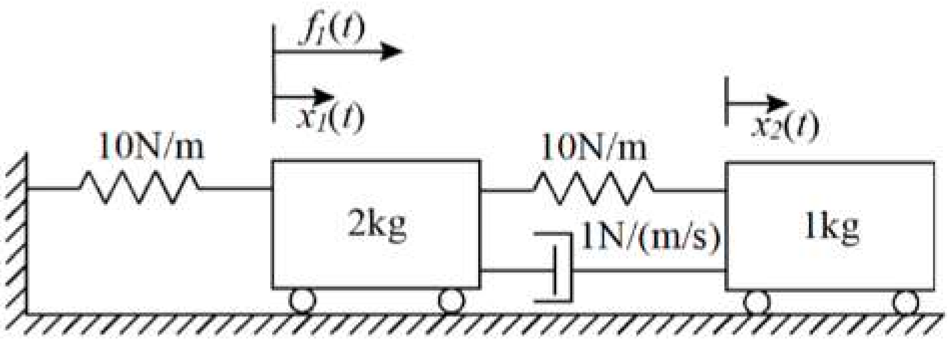
Portanto podemos determinar N_u e N_x pela seguinte equação

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix}$$

E então a matriz *N* pode ser determinada por

$$N = N_u + KN_x$$

Lembrando que o matriz zero ‘$\mathbf{0}$’ da equação acima possui o mesmo número de colunas que o vetor *r* e o número de linhas do vetor *x*. E a matriz identidade ‘$\boldsymbol{I}$’ é uma matriz quadrada com as mesmas linhas e colunas da dimensão de *r*, isto é *y*, pois *y* e *r* possuem as mesmas dimensões.



Para o problema anterior temos

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

com

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -10 & 5 & -0.5 & 0.5 \\ 10 & -10 & 1 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \qquad D = [0]$$

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{I} = 1$$

portanto

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix}$$

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -10 & 5 & -0.5 & 0.5 & 0.5 \\ 10 & -10 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

```
import numpy as np
m1=2.0; m2=1.0; k1=10.0; k2=10.0; c1=0; c2=1.0;
A = np.array([[0,0,1,0],[0,0,0,1],[-(k1+k2)/m1,k2/m1,-(c1+c2)/m1,c2/m1],[k2/m2,-k2/m2,c2/m2,-c2/m2]])
B = np.array([[0],[0],[1/m1],[0]])
C = np.array([[0,1,0,0]])
D = np.array([[0]])
AB = np.concatenate([A,B],axis=1)
CD = np.concatenate([C,D],axis=1)
ABCD = np.concatenate([AB,CD])
np.linalg.inv(ABCD)
```

```
array([[-0.1,  0.1, -0. ,  0.1,  1. ],
       [-0. , -0. , -0. , -0. ,  1. ],
       [ 1. ,  0. ,  0. ,  0. ,  0. ],
       [ 0. ,  1. ,  0. ,  0. ,  0. ],
       [-1. ,  1. ,  2. ,  2. , 10. ]])
```

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} -0.1 & 0.1 & 0 & 0.1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 2 & 2 & 10 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 10 \end{bmatrix}$$

$$N_x = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad N_u = 10$$

E a matriz N pode ser obtida por

$$N = N_u + KN_x$$

com K o ganho de controle calculado anteriormente

$$K = \begin{bmatrix} 105.92 & -64.72 & 29 & 17.08 \end{bmatrix}$$

$$N = 10 + \begin{bmatrix} 105.92 & -64.72 & 29 & 17.08 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$N = 51.2$$

Simulando o controle para uma referência em posição da massa 2 igual para 0.5m

```
import numpy as np

def RK4(f): return lambda x, u, dt:(lambda dx1:(lambda dx2:(lambda dx3:(lambda
dx4:(dx1+2*dx2+2*dx3+dx4)/6)(dt*f(x+dx3,u)))(dt*f(x+dx2/2,u)))(dt*f(x+dx1/2,u)))(dt*f(x,u))
dx = RK4(lambda x, u: A@x + B@u)

m1=2.0; m2=1.0; k1=10.0; k2=10.0; c1=0; c2=1.0;
A = np.array([[0,0,1,0],[0,0,0,1],[-(k1+k2)/m1,k2/m1,-(c1+c2)/m1,c2/m1],[k2/m2,-k2/m2,c2/m2,-c2/m2]])
B = np.array([[0],[0],[1/m1],[0]])
C = np.array([[0,1,0,0]])
D = np.array([[0]])
K = np.array([[105.92,-64.72,29,17.08]])
N = np.array([[51.2]])

t, tf, dt, u, x, r = 0, 10, .01, np.array([[0]]), np.array([[0.5],[1],[0],[0]]), np.array([[.5]])
X, U, T = x, u, t

for i in range(int((tf-t)/dt)):
    t, x = t + dt, x + dx(x,u,dt)
    u = N@r-K@x
    X, U, T = np.append(X,x,axis=1), np.append(U,u,axis=1), np.append(T,t)

Y = C@X+D@U

import matplotlib.pyplot as plt
plt.plot(T,Y[0],'k')
plt.xlabel('tempo (s)')
plt.ylabel('x2 (m)')
plt.grid(True)
plt.show()
```