$$x_{t+1} = \begin{bmatrix} 1 & T \\ -k_2T & 1-k_1T \end{bmatrix} x_t + \begin{bmatrix} 0 \\ k_3T \end{bmatrix} u_t$$

$k_1 = k_2 = k_3 = 1$, $T = 0,1$, $x_0 = [5\ 1]^T$, $\hat{x}_0 = [6\ 0]^T$

$\boxed{1}$

$$P = Q + A^T P(\mathbb{I} + BR^{-1}B^T P)^{-1} A \quad (1),$$

$$(S + UTV)^{-1} = S^{-1} - S^{-1}U(T^{-1} + VS^{-1}U)^{-1}VS^{-1} \quad (2)$$

Let $U = B$, $T = R^{-1}$, $V = B^T P$. By applying (2) to (1) we then get:

$$P = Q + A^T P(\mathbb{I} - B(R + B^T PB)B^T P)A$$
$$= Q + A^T PA - A^T PB(R + B^T PB)B^T PA$$

$$\Rightarrow \underline{A^T PA - P - A^T PB(R + B^T PB)B^T PA + Q = 0} \quad \square$$

$\boxed{2}$ $y_t = [1\ 0]x_t$

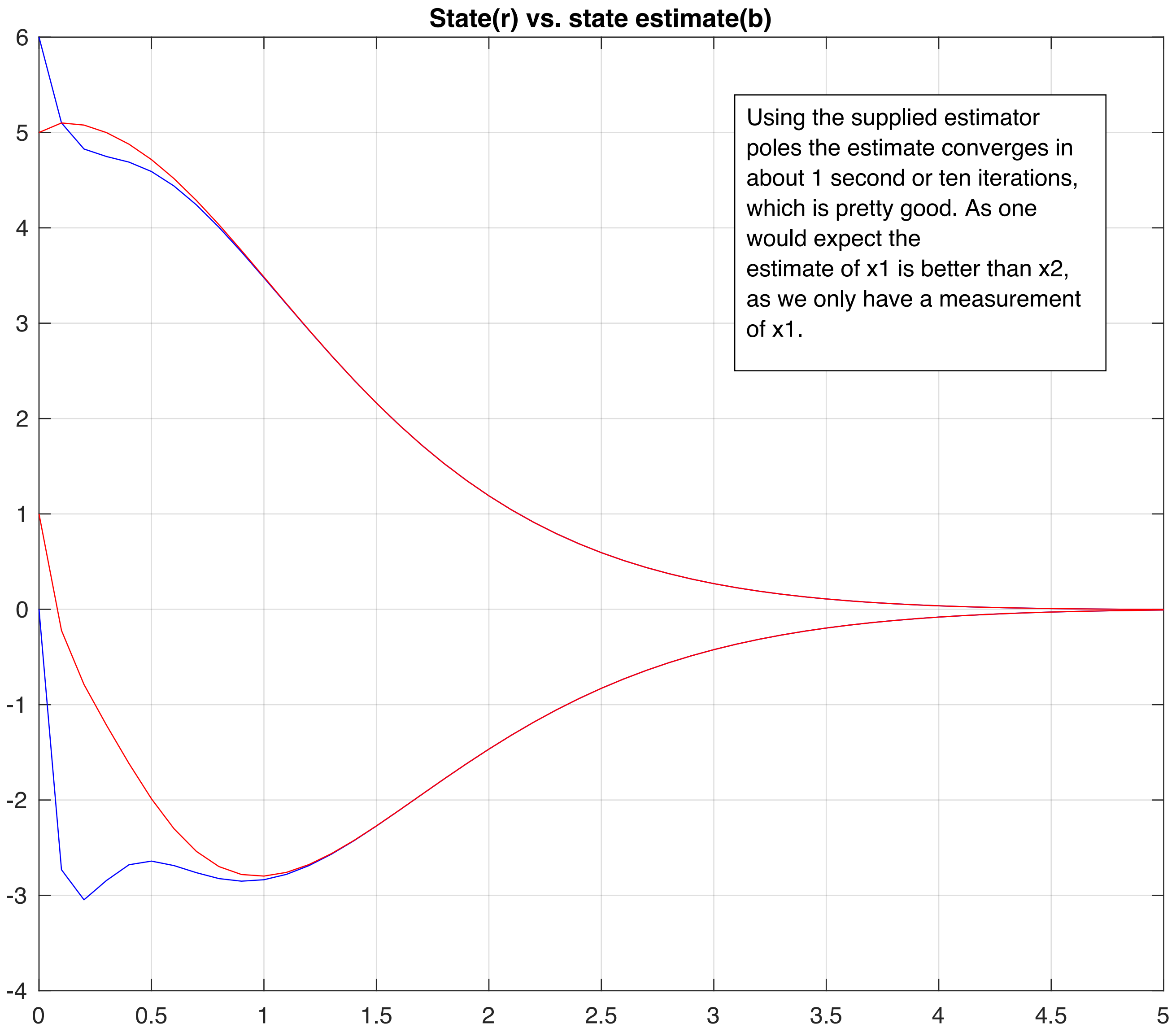
ⓐ $f(z) = \frac{1}{2}\sum_{t=0}^{\infty} \hat{x}_{t+1}^T Q \hat{x}_{t+1} + u_t^T R u_t$,

$Q = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$, $R = 1$

The almighty matlab says:

$\underline{K = [1,0373\ \ 1,6498]}$,

$\underline{\lambda = 0,8675 \pm 0,0531i}$

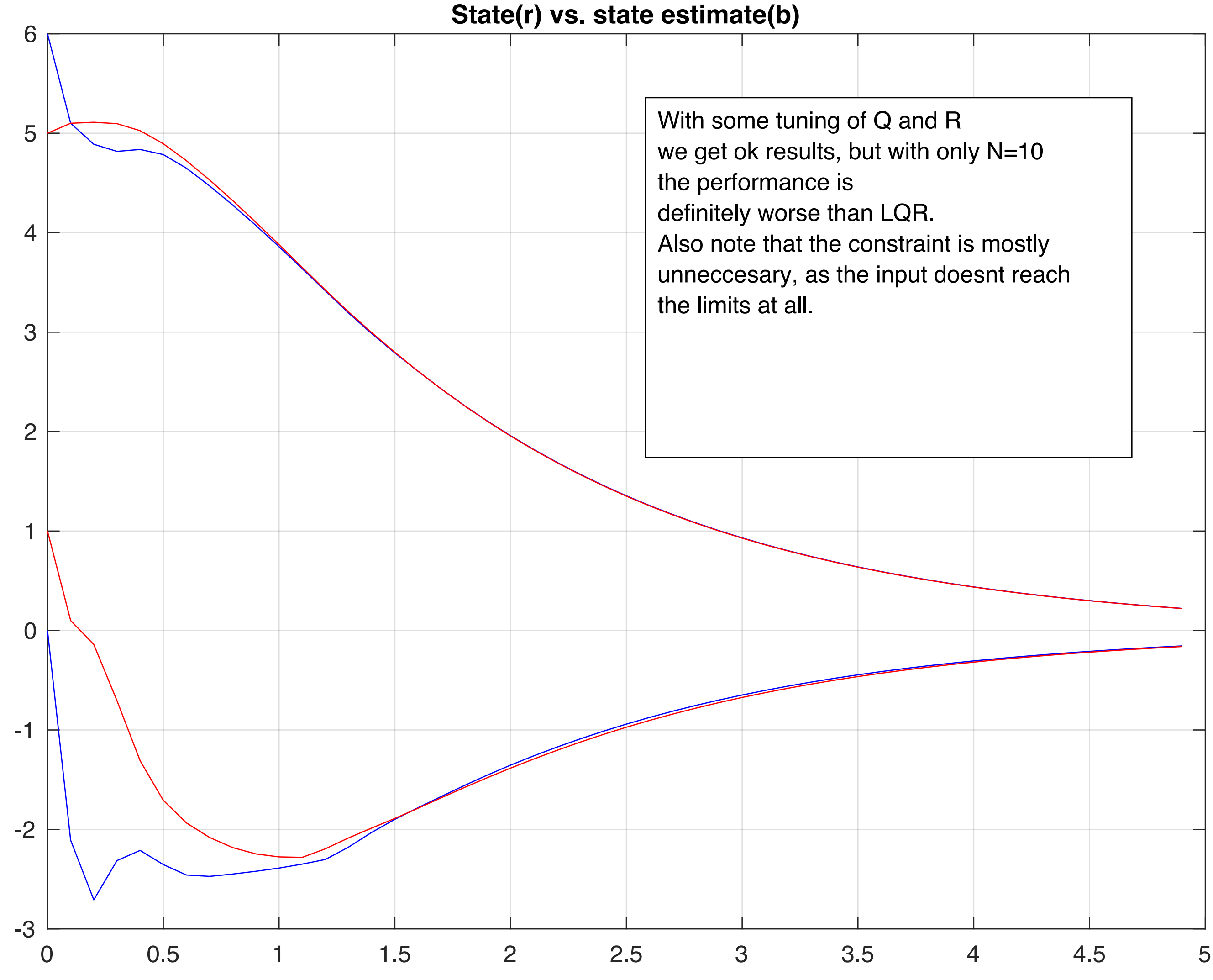

State(r) vs. state estimate(b)
Using the supplied estimator poles the estimate converges in about 1 second or ten iterations, which is pretty good. As one would expect the
estimate of x1 is better than x2, as we only have a measurement of x1.
6
5
4
3
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

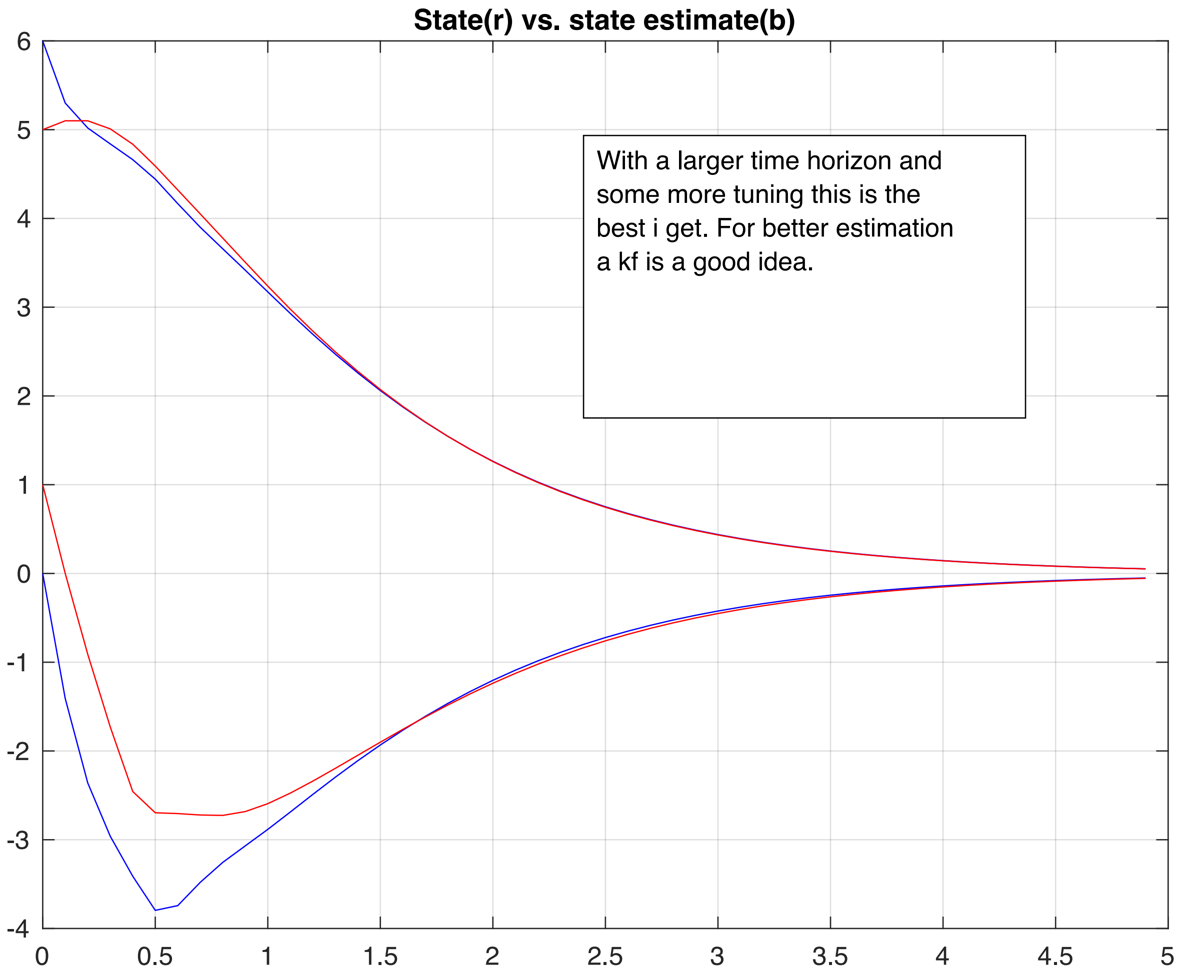
(c) $$\Phi = \begin{bmatrix} A-BK & BK \\ 0 & A-LC \end{bmatrix} = \begin{bmatrix} 1 & 0,1 & 0 & 0 \\ -0,204 & 0,735 & 0,104 & 0,165 \\ 0 & 0 & 0,1 & 0,1 \\ 0 & 0 & -1,609 & 0,9 \end{bmatrix}$$

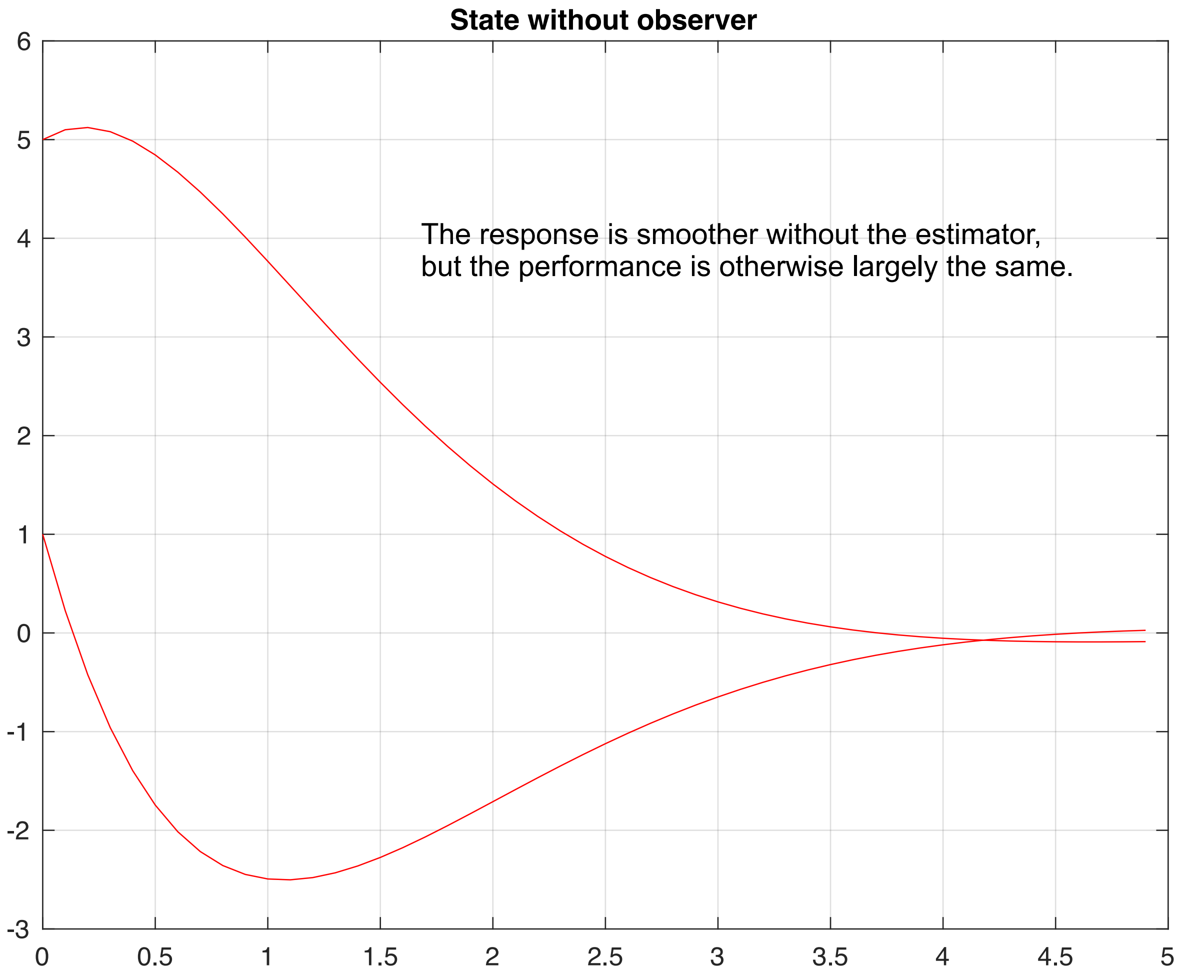
$$\lambda_{\Phi} = 0,8675 \pm 0,0531i, \; 0,5 \pm 0,03i$$

Checked this in MATLAB

[3] $-4 \leq u_t \leq 4$


State(r) vs. state estimate(b)
With some tuning of Q and R
we get ok results, but with only N=10
the performance is
definitely worse than LQR.
Also note that the constraint is mostly
unneccesary, as the input doesnt reach
the limits at all.
6
5
4
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

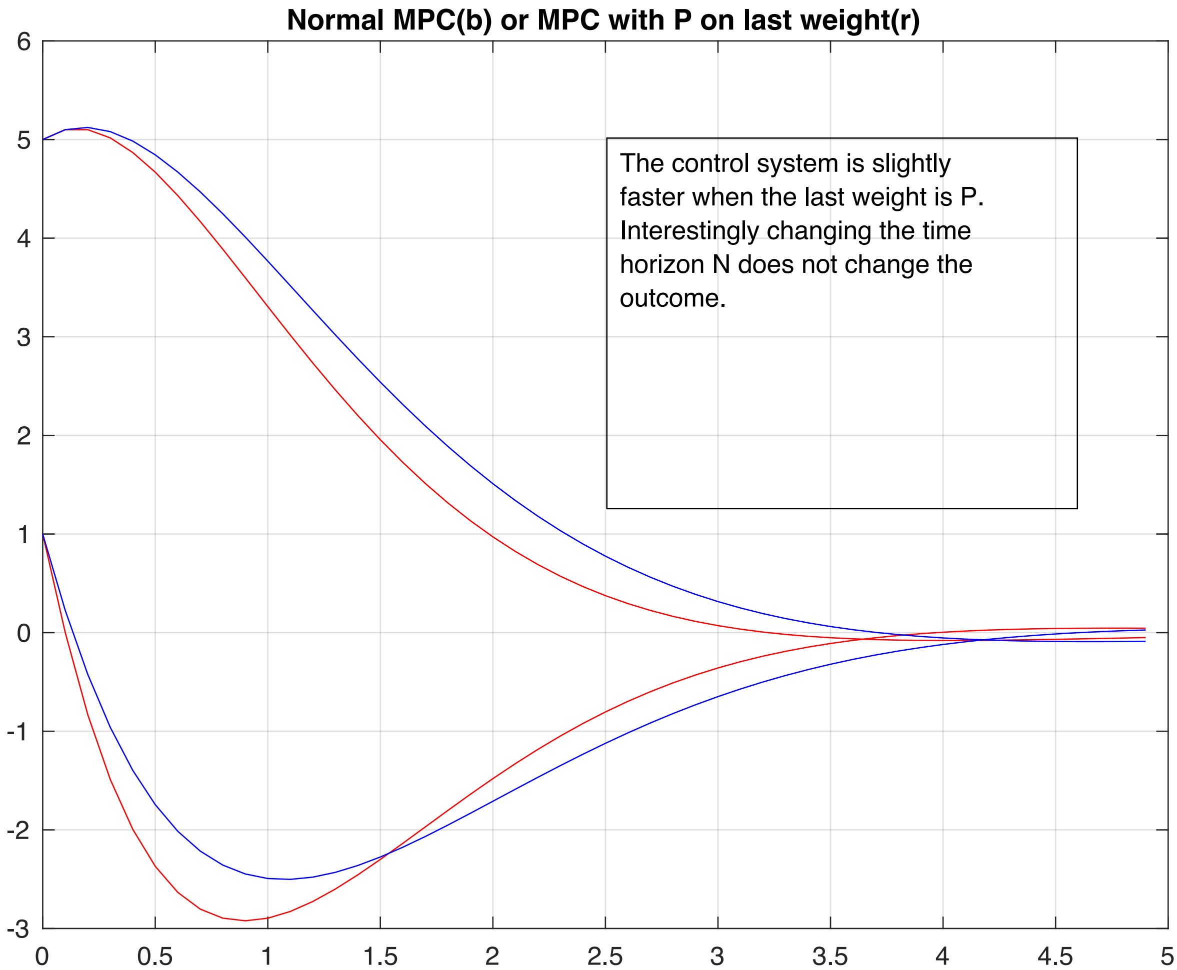

State(r) vs. state estimate(b)
With a larger time horizon and
some more tuning this is the
best i get. For better estimation
a kf is a good idea.
6
5
4
3
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

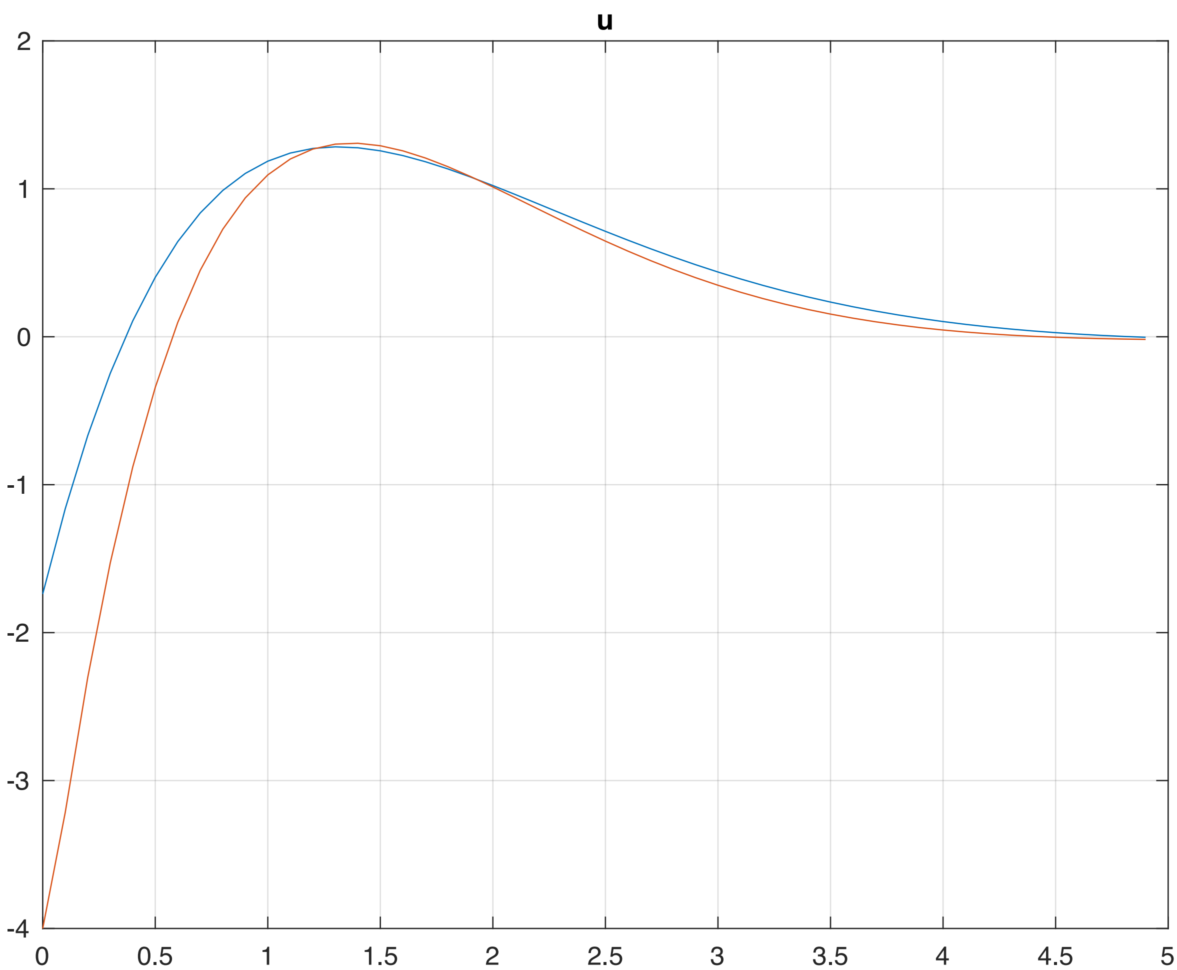

State without observer
The response is smoother without the estimator,
but the performance is otherwise largely the same.
6
5
4
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Task 4

```
P =

   28.5963    8.3981
    8.3981   12.8302
```


Normal MPC(b) or MPC with P on last weight(r)
The control system is slightly faster when the last weight is P. Interestingly changing the time horizon N does not change the outcome.
6
5
4
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


u
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5