Sympy_SolveRiccati_first_agent_target

October 3, 2021

0.1 Symbolic solution to the 2 by 2 Ricatti equation: targeting first agent

James Yu, 2 October 2021

[1]:
```
from sympy import *
```

The goal of this notebook is to construct an analytical solution to the algebraic matrix Ricatti equation of:

$$K_{ss} = \delta A'(K_{ss} - K_{ss}B(B'K_{ss}B + \frac{R}{\delta})^{-1}B'K_{ss})A + Q$$

We start by solving in the simple case where $\delta = 1$, $R = 0$ and $Q = I_n$ where $n = 2$ for now. We set this up as follows:

[2]:
```
K1, K2, K3, a11, a12, a2 = symbols("K1 K2 K3 a11 a12 a2")

K = Matrix(([[K1, K2], [K2, K3]]))
Q = eye(2)
A = Matrix(([[a11, a12], [0.5, 0.5]]))
B = Matrix(([[1 - a11 - a12], [0]]))
K
```

[2]:

$$\begin{bmatrix} K_1 & K_2 \\ K_2 & K_3 \end{bmatrix}$$

[3]:
```
A
```

[3]:

$$\begin{bmatrix} a_{11} & a_{12} \\ 0.5 & 0.5 \end{bmatrix}$$

[4]:
```
B
```

[4]:

$$\begin{bmatrix} -a_{11} - a_{12} + 1 \\ 0 \end{bmatrix}$$

If we plug these into the equation directly, we obtain:

[5]:
```
K_sol = simplify(A.T *(K - (K*B*(B.T * K * B).inv() * B.T * K)) * A + Q)
K_sol
```

[5]:

$$\begin{bmatrix} 0.25K_3 + 1.0 - \frac{0.25K_2^2}{K_1} & 0.25K_3 - \frac{0.25K_2^2}{K_1} \\ 0.25K_3 - \frac{0.25K_2^2}{K_1} & 0.25K_3 + 1.0 - \frac{0.25K_2^2}{K_1} \end{bmatrix}$$

Indeed both diagonals of this matrix are identical.

Not only that, but K_{ss} is a uniform matrix plus I_2.

This leads to a system of one equation in one unknown.

[6]:
```
K1_expr = simplify(expand(K_sol[0, 0]).subs(K3, K1))
K1_expr
```

[6]:

$$0.25K_1 + 1.0 - \frac{0.25K_2^2}{K_1}$$

[7]:
```
K2_expr = simplify(expand(K_sol[1, 0]).subs(K3, K1))
K2_expr
```

[7]:

$$0.25K_1 - \frac{0.25K_2^2}{K_1}$$

This indicates that $K_1 = K_2 + 1$.

[8]:
```
K2_expr = K2_expr.subs(K1, K2 + 1)
K2_expr
```

[8]:

$$-\frac{0.25K_2^2}{K_2 + 1} + 0.25K_2 + 0.25$$

[9]:
```
K2_solved = solve(K2_expr - K2, K2)
K2_solved
```

[9]: `[-0.809016994374947, 0.309016994374947]`

Suppose we take the positive solution.

So then we get:

[14]:
```
K2_sol = K2_solved[1]
K1_sol = K2_sol + 1
K_sol = Matrix(([[K1_sol, K2_sol], [K2_sol, K1_sol]]))
simplify(K_sol)
```

[14]:

$$\begin{bmatrix} 1.30901699437495 & 0.309016994374947 \\ 0.309016994374947 & 1.30901699437495 \end{bmatrix}$$

[15]:
```
K_sol_maybe = simplify(A.T *(K_sol - K_sol*B*(B.T * K_sol * B).inv() * B.T *␣
↪K_sol) * A + Q)
simplify(K_sol_maybe)
```

[15]:

$$\begin{bmatrix} \frac{1.713525491562422a_{11}^2+3.42705098312484a_{11}a_{12}-3.42705098312484a_{11}+1.71352549156242a_{12}^2-3.42705098312484a_{12}+1.71352549156242}{1.30901699437495a_{11}^2+2.61803398874989a_{11}a_{12}-2.61803398874989a_{11}+1.30901699437495a_{12}^2-2.61803398874989a_{12}+1.30901699437495} \\ \frac{0.404508497187474a_{11}^2+0.809016994374947a_{11}a_{12}-0.809016994374947a_{11}+0.404508497187474a_{12}^2-0.809016994374947a_{12}+0.404508497187474}{1.30901699437495a_{11}^2+2.61803398874989a_{11}a_{12}-2.61803398874989a_{11}+1.30901699437495a_{12}^2-2.61803398874989a_{12}+1.30901699437495} \end{bmatrix}$$

[16]:
```
factor(K_sol_maybe[0,0])
```

[16]:

1.30901699437495

[17]:
```
factor(K_sol_maybe[1,0])
```

[17]:

0.309016994374947

factor() factors polynomials and this indicates that the `K_sol_maybe` matrix is equal to the numerical matrix we obtained. This would seem to indicate that, for the given A and B, the solution is a single numeric matrix independent of A or B.

This is of course a result of the fact that the symbolic matrix on line 5 is not a function of A, meaning it is some sort of constant.