Sympy_SolveRiccati_second_agent_target

October 3, 2021

0.1 Symbolic solution to the 2 by 2 Ricatti equation: targeting second 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, a1, a2 = symbols("K1 K2 K3 a1 a2")

K = Matrix(([[K1, K2], [K2, K3]]))
Q = eye(2)
A = Matrix(([[a1, 1 - a1], [a2, a2]]))
B = Matrix(([[0], [1 - 2*a2]]))
K
```

[2]:
$$\begin{bmatrix} K_1 & K_2 \\ K_2 & K_3 \end{bmatrix}$$

[3]:
```
A
```

[3]:
$$\begin{bmatrix} a_1 & 1 - a_1 \\ a_2 & a_2 \end{bmatrix}$$

[4]:
```
B
```

[4]:
$$\begin{bmatrix} 0 \\ 1 - 2a_2 \end{bmatrix}$$

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]:

$$\begin{bmatrix} K_1 a_1^2 - \frac{K_2^2 a_1^2}{K_3} + 1 & \frac{a_1\left(-K_1 K_3 a_1 + K_1 K_3 + K_2^2 a_1 - K_2^2\right)}{K_3} \\ \frac{a_1\left(-K_1 K_3 a_1 + K_1 K_3 + K_2^2 a_1 - K_2^2\right)}{K_3} & K_1 a_1^2 - 2K_1 a_1 + K_1 - \frac{K_2^2 a_1^2}{K_3} + \frac{2K_2^2 a_1}{K_3} - \frac{K_2^2}{K_3} + 1 \end{bmatrix}$$

This leads to a system of three equations in three unknowns because of the symmetry, so we need to solve these all at the same time.

[6]:
```
K1_expr = simplify(expand(K_sol[0, 0]))
K1_expr
```

[6]:

$$K_1 a_1^2 - \frac{K_2^2 a_1^2}{K_3} + 1$$

[7]:
```
K2_expr = simplify(expand(K_sol[1, 0]))
K2_expr
```

[7]:

$$\frac{a_1\left(K_1 K_3 (1 - a_1) + K_2^2 a_1 - K_2^2\right)}{K_3}$$

[8]:
```
K3_expr = simplify(expand(K_sol[1, 1]))
K3_expr
```

[8]:

$$\frac{-K_2^2 a_1^2 + 2K_2^2 a_1 - K_2^2 + K_3\left(K_1 a_1^2 - 2K_1 a_1 + K_1 + 1\right)}{K_3}$$

[9]:
```
K1_solved = solve(K1_expr - K1, K1)[0]
K1_solved
```

[9]:

$$\frac{K_2^2 a_1^2 - K_3}{K_3\left(a_1^2 - 1\right)}$$

[10]:
```
K2_expr = simplify(K2_expr.subs(K1, K1_solved))
K2_expr
```

[10]:

$$\frac{a_1\left(-K_2^2 + K_3\right)}{K_3\left(a_1 + 1\right)}$$

[11]:
```
K3_expr = simplify(K3_expr.subs(K1, K1_solved))
K3_expr
```

[11]:

$$\frac{K_2^2 a_1 - K_2^2 + 2K_3}{K_3\left(a_1 + 1\right)}$$

[12]:
```
K2_solved = solve(K2_expr - K2, K2)
K2_solved[0]
```

[12]:

$$\frac{-K_3\left(a_1 + 1\right) + \sqrt{K_3\left(K_3 a_1^2 + 2K_3 a_1 + K_3 + 4a_1^2\right)}}{2a_1}$$

This has two solutions but the other one, which is:

[13]:
```
K2_solved[1]
```

[13]:

$$-\frac{K_3\left(a_1+1\right)+\sqrt{K_3\left(K_3a_1^2+2K_3a_1+K_3+4a_1^2\right)}}{2a_1}$$

is negative. So we take the first.

[14]:
```
K3_expr = simplify(K3_expr.subs(K2, K2_solved[0]))
K3_expr
```

[14]:

$$\frac{K_3a_1^2-K_3+2a_1^2-a_1\sqrt{K_3\left(K_3a_1^2+2K_3a_1+K_3+4a_1^2\right)}+\sqrt{K_3\left(K_3a_1^2+2K_3a_1+K_3+4a_1^2\right)}}{2a_1^2}$$

[15]:
```
K3_solved = solve(K3_expr - K3, K3)
K3_solved[0]
```

[15]:

$$a_1^2-a_1+\left(1-a_1\right)\sqrt{a_1^2+1}+1$$

[16]:
```
K3_solved[1]
```

[16]:

$$a_1^2-a_1+\left(a_1-1\right)\sqrt{a_1^2+1}+1$$

Again, only the first of these is positive. So then we get:

[17]:
```
K3_sol = K3_solved[0]
K2_sol = simplify(K2_solved[0].subs(K3, K3_sol))
K1_sol = simplify(K1_solved.subs(K2, K2_sol).subs(K3, K3_sol))
K_sol = Matrix(([[K1_sol, K2_sol], [K2_sol, K3_sol]]))
simplify(K_sol)
```

[17]:

$$\left[\begin{matrix}\frac{-a_1^2+a_1+(a_1-1)\sqrt{a_1^2+1}+\frac{\left(a_1^3-a_1^2\sqrt{a_1^2+1}+\sqrt{a_1^2+1}-\sqrt{2a_1^6-2a_1^5\sqrt{a_1^2+1}+3a_1^4-2a_1^3\sqrt{a_1^2+1}-2a_1^3+2a_1^2\sqrt{a_1^2+1}+3a_1^2+2\sqrt{a_1^2+1}+2}+1\right)^2}{4}-1}{\left(a_1^2-1\right)\left(a_1^2-a_1-(a_1-1)\sqrt{a_1^2+1}+1\right)} & -a_1^3+a_1^2\sqrt{a_1^2 \\ \frac{-a_1^3+a_1^2\sqrt{a_1^2+1}-\sqrt{a_1^2+1}+\sqrt{2a_1^6-2a_1^5\sqrt{a_1^2+1}+3a_1^4-2a_1^3\sqrt{a_1^2+1}-2a_1^3+2a_1^2\sqrt{a_1^2+1}+3a_1^2+2\sqrt{a_1^2+1}+2}-1}{2a_1} & \end{matrix}\right.$$

[18]:
```
K_sol.subs(a1, 0.9)
```

[18]:

$$\left[\begin{matrix}4.60743547812971 & 0.400826164236634\\0.400826164236634 & 1.04453624047074\end{matrix}\right]$$

[19]:
```
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)
```

[19]:

$$\left[\begin{matrix}-4a_1^7+4a_1^6\sqrt{a_1^2+1}+8a_1^6-8a_1^5\sqrt{a_1^2+1}-13a_1^5+11a_1^4\sqrt{a_1^2+1}+2a_1^4\sqrt{2a_1^6-2a_1^5\sqrt{a_1^2+1}+3a_1^4-2a_1^3\sqrt{a_1^2+1}-2a_1^3+2a_1^2\sqrt{a_1^2+1}+3a_1^2+2\sqrt{a_1^2+1}+2}+16a_1^4- \\ -4a_1^7+4a_1^6\sqrt{a_1^2+1}+8a_1^6-8a_1^5\sqrt{a_1^2+1}-9a_1^5+7a_1^4\sqrt{a_1^2+1}+2a_1^4\sqrt{2a_1^6-2a_1^5\sqrt{a_1^2+1}+3a_1^4-2a_1^3\sqrt{a_1^2+1}-2a_1^3+2a_1^2\sqrt{a_1^2+1}+3a_1^2+2\sqrt{a_1^2+1}+2}+4a_1^4- \end{matrix}\right.$$

[20]:

```
K_sol_maybe.subs(a1, 0.9)
```

[20]:

$$\begin{bmatrix} 4.60743547812968 & 0.400826164236632 \\ 0.400826164236632 & 1.04453624047074 \end{bmatrix}$$

[21]:

```
import numpy as np
for i in np.linspace(0.0001, 0.9999, 100):
    mat1 = K_sol.subs(a1, i)
    mat2 = K_sol_maybe.subs(a1, i)
    for j in range(2):
        for k in range(2):
            if round(mat1[j, k], 6) != round(mat2[j, k], 6):
                print("ERROR FOUND")

print("computation finished")
```

```
computation finished
```

These seem to be equal for various values of a_1 but this is not really a usable analytical solution.