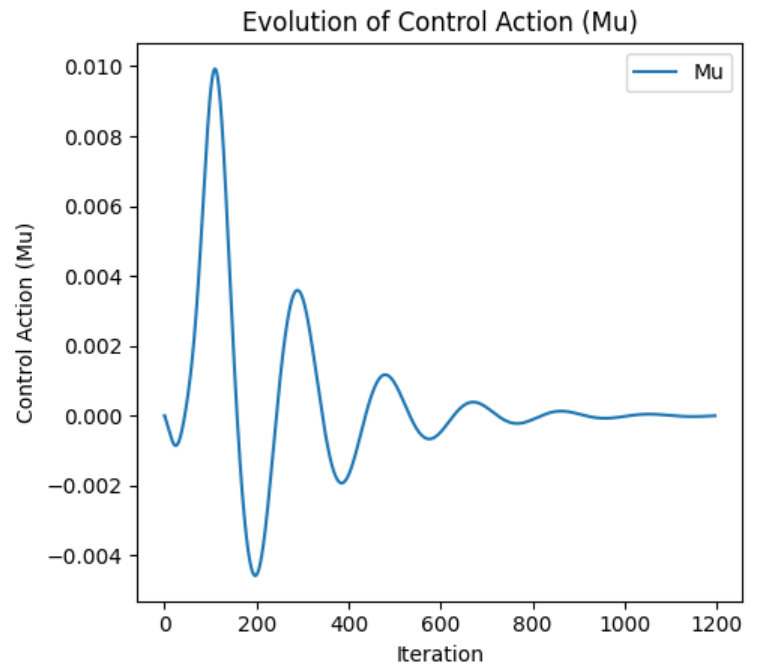
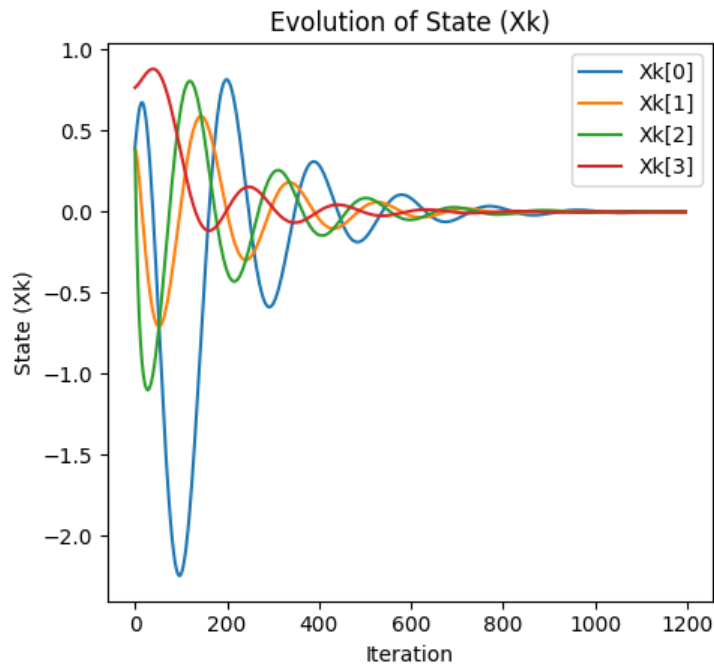


## Results

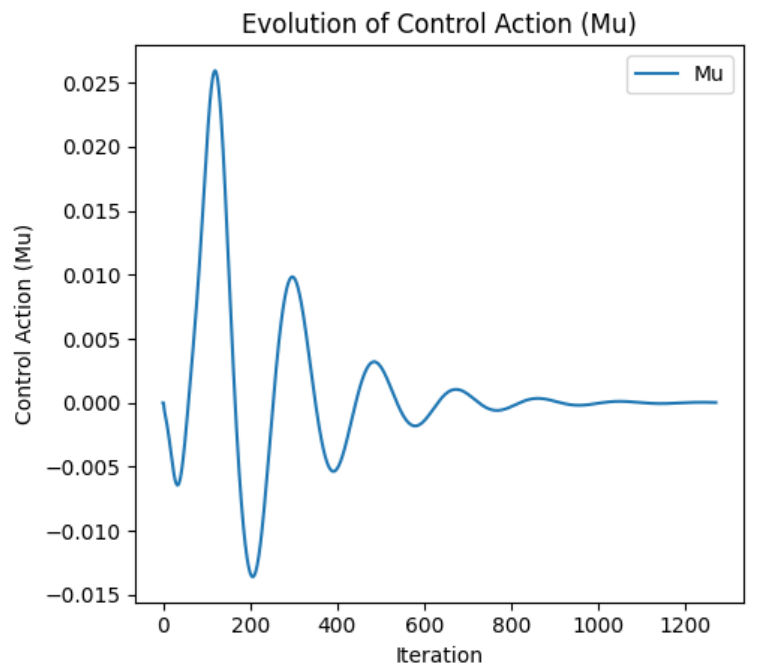
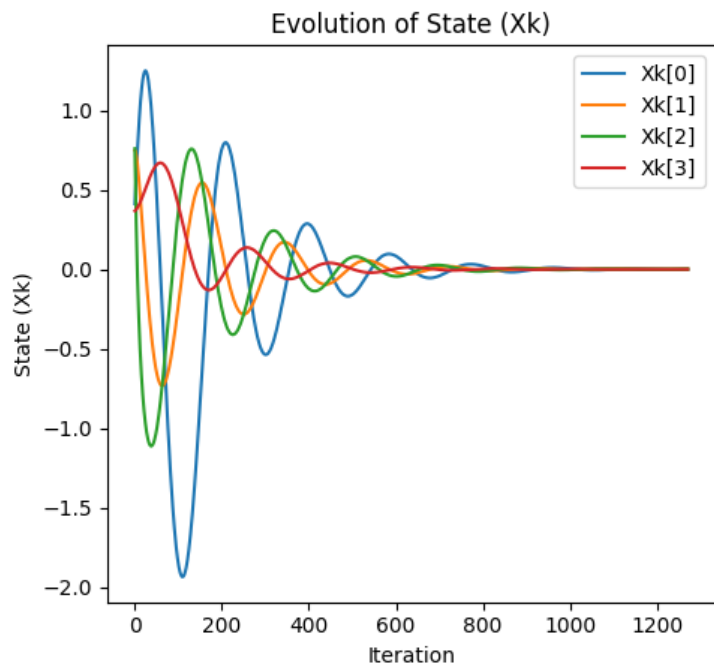
### Model Free

States of Model-Free RL



### Model-Based

States of Model-based RL



# Code

## Model Free

```
# Model-free Value Iteration
```

```
#Wxx 4*4  Wxu 4*1
```

```
#Wux 1*4  Wuu 1*1
```

```
Wc = np.ones((5,5))
```

```
## Wa
```

```
Wa=np.zeros((1, 5))
```

```
# Q,R > 0
```

```
Q= np.random.rand(4, 4)
```

```
R= np.random.rand(1, 1)
```

```
# Use small random initial conditions x0
```

```
## Xk 4*1 then Uk 1*1
```

```
Xk0 = np.random.random((5, 1))
```

```
Zk = np.zeros((1, 5))
```

```
Zk1 = np.zeros((1, 5))
```

```
Mu = Wa @ Xk0
```

```
Xk_list = [Xk0]
```

```
Mu_list = [Mu]
```

```
Wa_list = [Wa]
```

```
Wc_list = [Wc]
```

```
k = 0
```

```
while True:
```

```
    #Update the Wa
```

```
    Wuu_c = np.array([[Wc_list[k][-1,-1]])
```

```
    Mu = Wa_list[k] @ Xk_list[k]
```

```
    Mu_list.append (Mu)
```

```
    Mu_desire = - np.block([[np.linalg.inv(Wuu_c),Wc_list[k][-1,:-1]]) @ Xk_list[k]
```

```
    Wa_list.append( Wa_list[k] - alpha_a * ((Mu) - (Mu_desire)) @ Xk_list[k].T ) )
```

```
    X = np.block([[A @ Xk_list[k][:-1] + B @ Mu_list[k][[-1]]], [Mu]])
```

```
    Xk_list.append( X )
```

```
    V = 0.5 * ( Xk_list[k].T @ Wc_list[k] @ Xk_list[k] )
```

```
    Vk1 = 0.5 * ( Xk_list[k+1].T @ Wc_list[k] @ Xk_list[k+1] )
```

```
    V_desire = 0.5 * ( Xk_list[k][:-1].T @ Q @ Xk_list[k][:-1] + Mu_list[k][[-1]].T @ R @ Mu_list[k][[-1]] ) + Vk1
```

```
    Wc_list.append( Wc_list[k] - alpha_c * ( (V - V_desire) * ( Xk_list[k] @ Xk_list[k].T ) ) )
```

```
    if np.array_equal(Wc_list[k+1] , Wc_list[k]) == True:
```

```
        print(k)
```

```
        break
```

```
    k = k + 1
```

1195

It reaches converged after 1195 iteration.

## Model-Based

```
# Model Based Value Iteration

## Wc
Wc = np.ones((4,4))

## Wa
Wa=np.zeros((1, 4))

# Q,R > 0
Q= np.random.rand(4, 4)
R= np.random.rand(1, 1)

# Use small random initial conditions x0
Xk0 = np.random.random((4, 1))

Zk = np.zeros((1, 4))
Zk1 = np.zeros((1, 4))

Mu = Wa @ Xk0

Xk_list = [Xk0]
Mu_list = [Mu]
Wa_list = [Wa]
Wc_list = [Wc]

k = 0
while True:
    #Update the Wa
    Wu_c = np.array([[Wc_list[k][-1,-1]]])

    Mu = Wa_list[k] @ Xk_list[k]
    Mu_list.append (Mu)
    Mu_desire = - ( np.linalg.inv(R) @ B.T @ Wc_list[k] @ Xk_list[k] )

    Wa_list.append( Wa_list[k] - alpha_a * ((Mu) - (Mu_desire)) @ Xk_list[k].T )

    X = A @ Xk_list[k] + B @ Mu_list[k]
    Xk_list.append( X )

    V = 0.5 * ( Xk_list[k].T @ Wc_list[k] @ Xk_list[k] )
    Vk1 = 0.5 * ( Xk_list[k+1].T @ Wc_list[k] @ Xk_list[k+1] )
    V_desire = 0.5 * ( Xk_list[k].T @ Q @ Xk_list[k] + Mu_list[k].T @ R @ Mu_list[k] ) + Vk1

    Wc_list.append( Wc_list[k] - alpha_c * ( (V - V_desire) * ( Xk_list[k] @ Xk_list[k].T ) ) )

    if np.array_equal(Wc_list[k+1] , Wc_list[k]) == True:
        print(k)
        break

    k = k + 1
```

1270

It reaches converged after 1270 iteration.