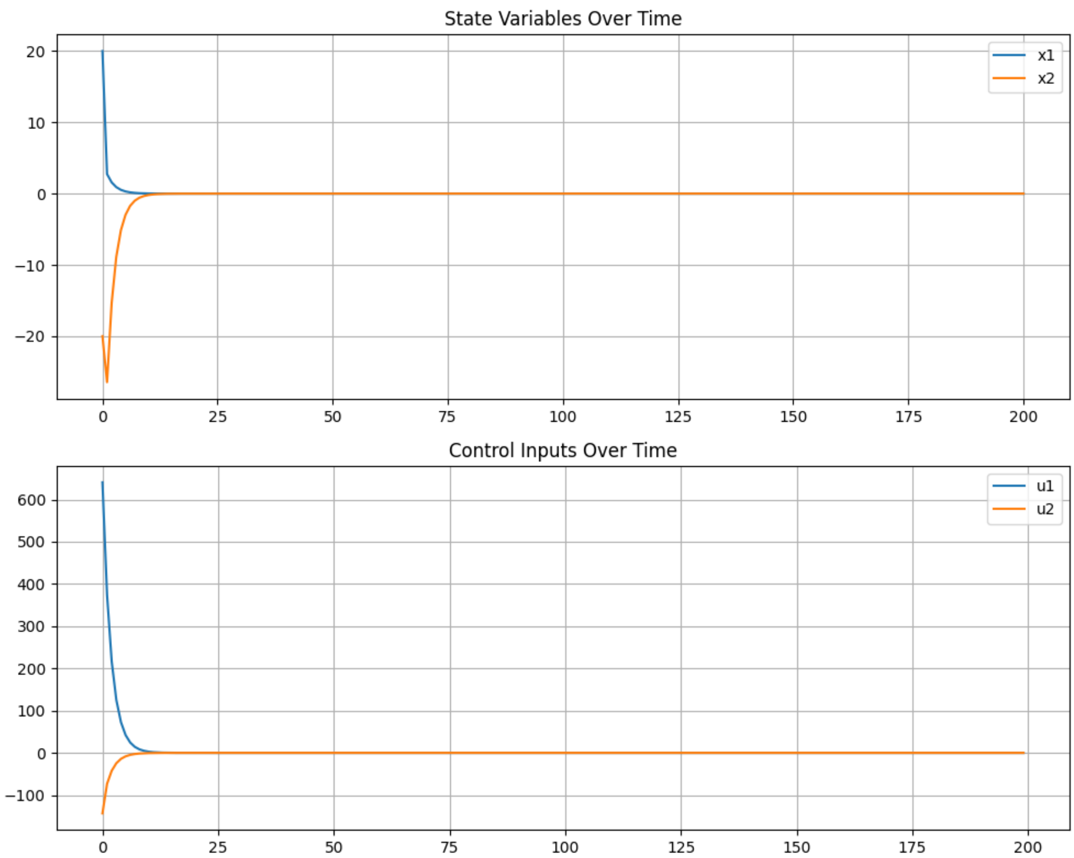
`mpc_r0.py`是基于DR_CAN在bilibili提供的Octave代码使用python语言复现的，其中的实现原理也是按照DR_CAN在视频中在假设系统参考r=0的情况下进行推导得到的目标函数，如式(1)所示

$$J = x_k^T M^T \bar{Q} M x_k + 2 x_k^T M^T \bar{Q} C U_k + U_k^T (C^T \bar{Q} C + \bar{R}) U_k \tag{1}$$

上式的推导过程简化如下：

$$J = \sum_{k}^{N-1} E_k^T Q E_k + U_k^T \bar{R} U_k + E_N^T F E_N \tag{2}$$



由于假设$r=0$，所以$e_k = y_k - r = x_k - r = x_k$，于是便得到了下式(具体推导过程在视频中有具体细节，此处省略)

$$J = X_k^T \bar{Q} X_k + U_k^T \bar{R} U_k \tag{3}$$

然后将式(4)代入上式，就会得到式(1)

$$X_k = M x_k + C U_k \tag{4}$$

但是如果$r \neq 0$，那么$e_k = x_k - r$，这里假设r是固定的，由于

$$e_k = x_k - r$$

$$e_{k+1} = x_{k+1} - r$$

$$x_{k+1} = A * x_k + B * u_k$$

$$\text{于是，} e_{k+1} = A * e_k + B * u_k + (A - I) * r$$

$$\text{(这里的} * \text{表示矩阵相乘)}$$

那么，就会得到系统误差矩阵，如下

$$E_k = M * e_k + C * u_k + T * r \tag{5}$$

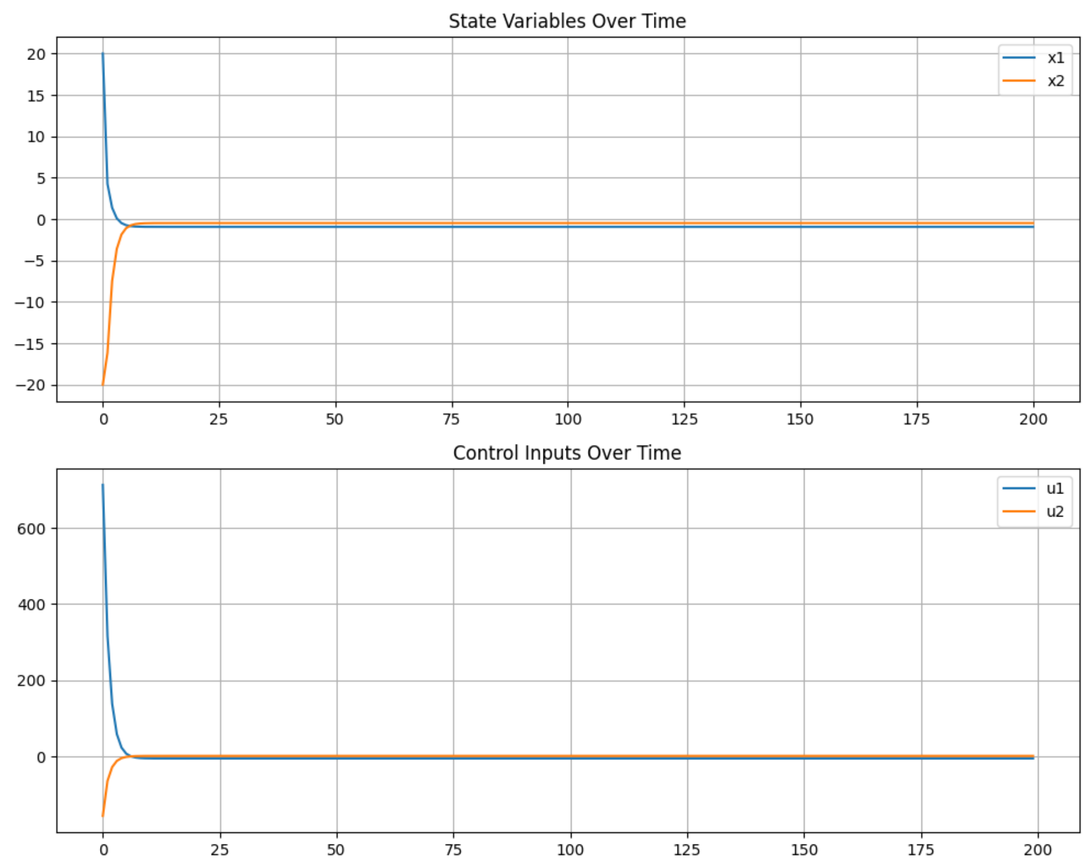
这里的M, C与之前的相同，T矩阵如下

$$T = \begin{bmatrix} 0 \\ A - I \\ A * (A - I) \\ A^2 * (A - I) \\ \vdots \\ A^{N-1} * (A - I) \end{bmatrix} \tag{6}$$

将代入到式(6)，得到目标函数如下

$$J = U_k^T (C^T \bar{Q} C) U_k + 2 * x_k^T M^T \bar{Q} C U_k + 2 * r^T T^T \bar{Q} C U_k + x_k^T M^T \bar{Q} M x_k + 2 * x_k^T M^T \bar{Q} T r + r^T$$

最终在`mpc.py`中实现



在程序`mpc.py`中设置的系统期望为$[10, 10]$，但是从图中可以看出并没有趋于10，推导过程应该是没有错的，目前推测是$r \neq 0$时的其他假设或者前提存在问题，正在解决中...，如果您感兴趣的话希望可以在 issue 中交流一下