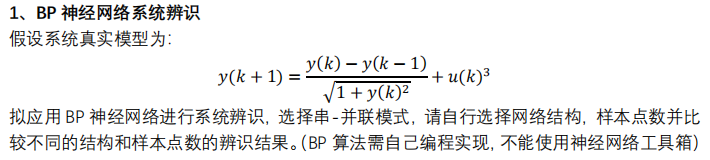
**EX1 BP神经网络系统辨识**  


采用单隐藏层BP网络进行系统辨识，BP网络反向传播模型见EX1\_BP.m

设制隐藏层节点数为10，训练10^4轮时，获得以下结果：

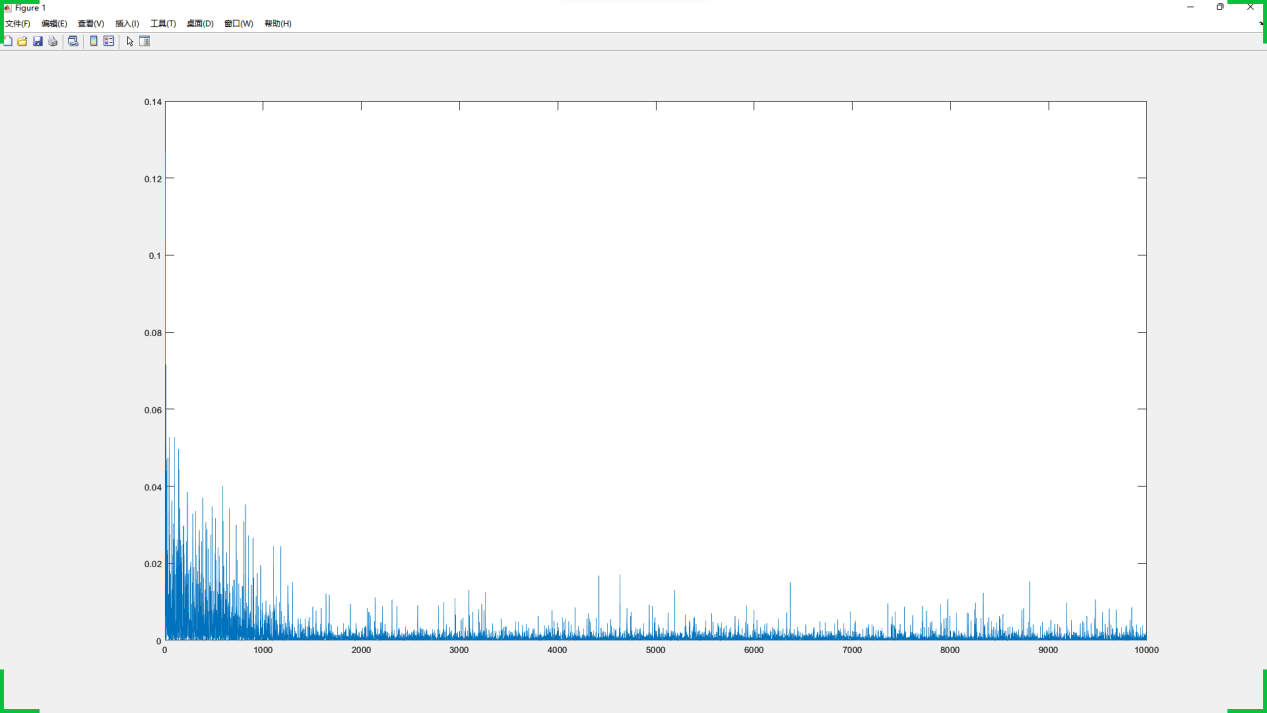


图 1 隐藏层节点数为10 损失函数 见EX1\_BP.m（下同）

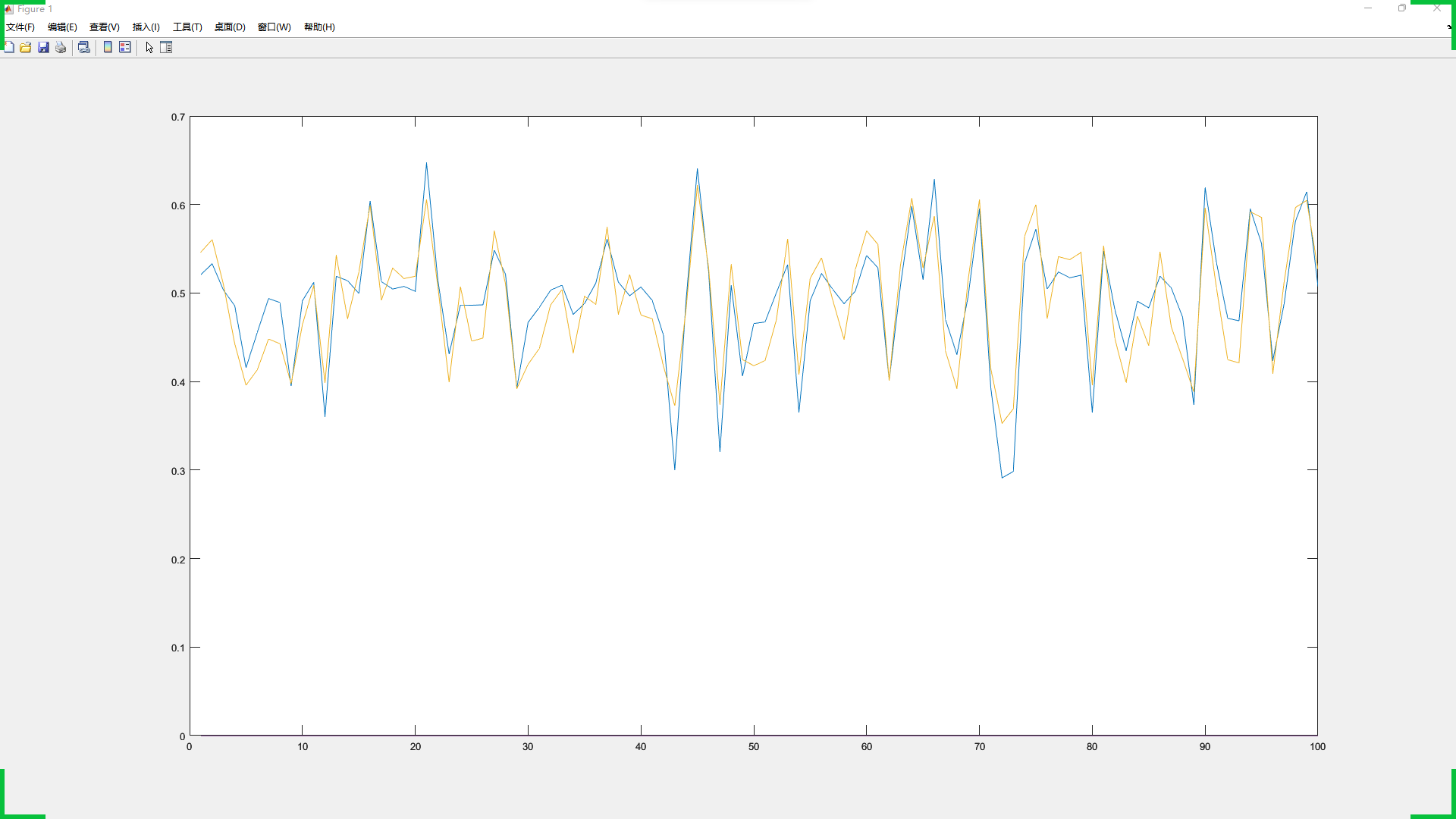


图 2 隐藏层节点数为10 真实系统与辨识系统比较 黄-真实系统 蓝-辨识系统 见EX1\_check.m（下同）

设制隐藏层节点数为50，训练10^5轮时，获得以下结果：

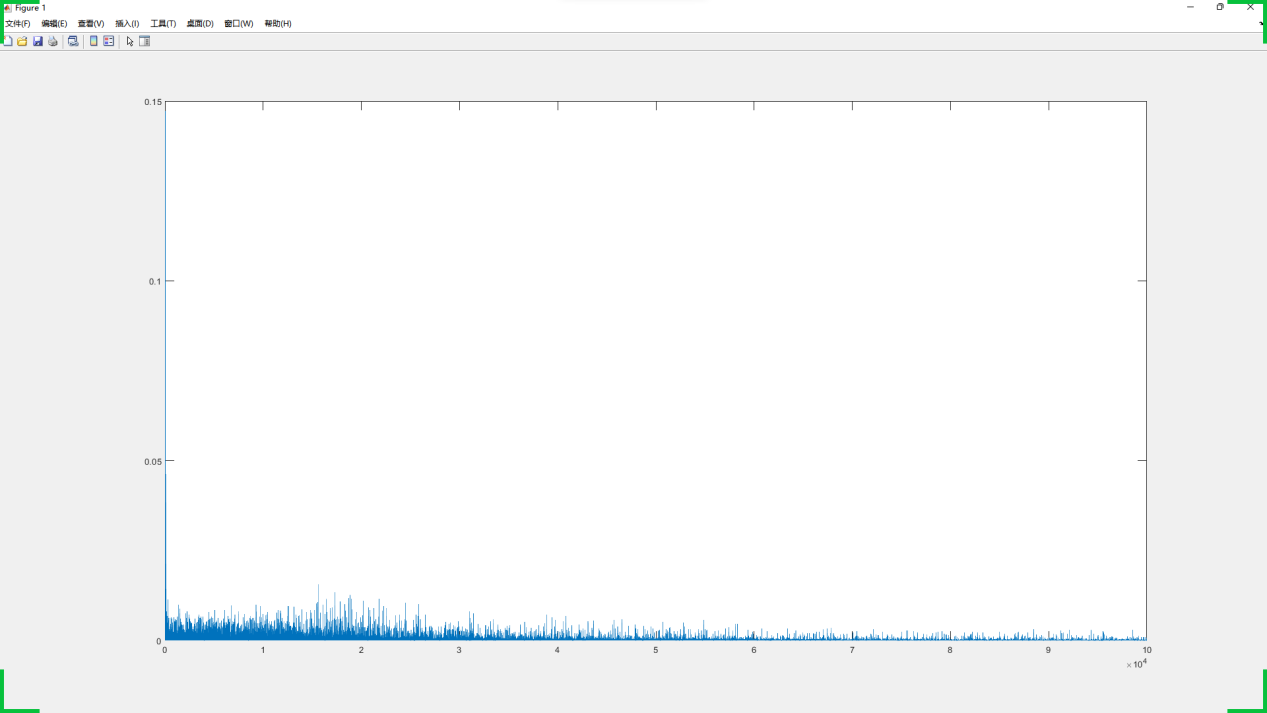
****

图 3 隐藏层节点数为50 损失函数

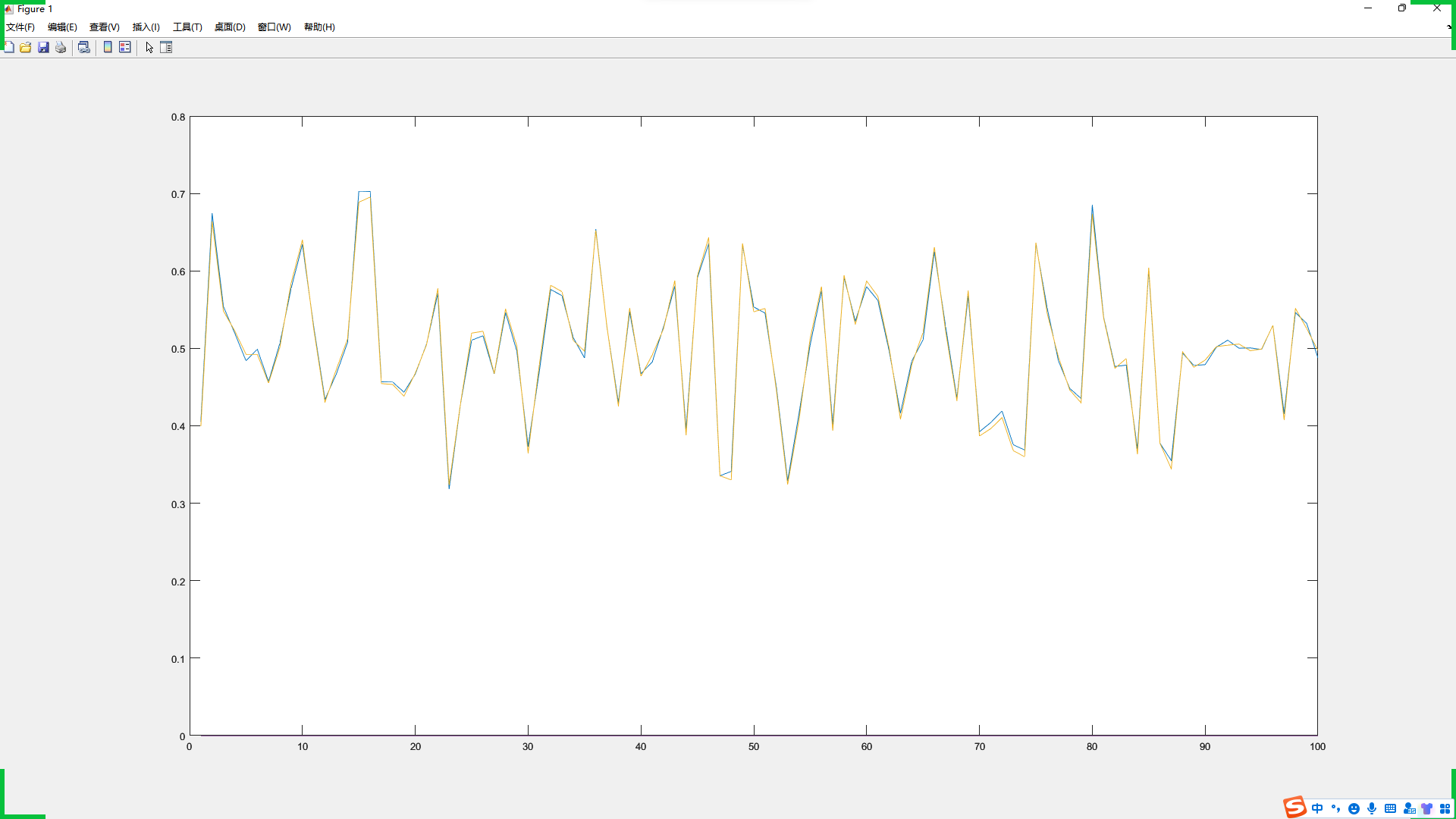


图 4 隐藏层节点数为50 真实系统与辨识系统比较 黄-真实系统 蓝-辨识系统

设制隐藏层节点数为250，训练10^6轮时，获得以下结果：

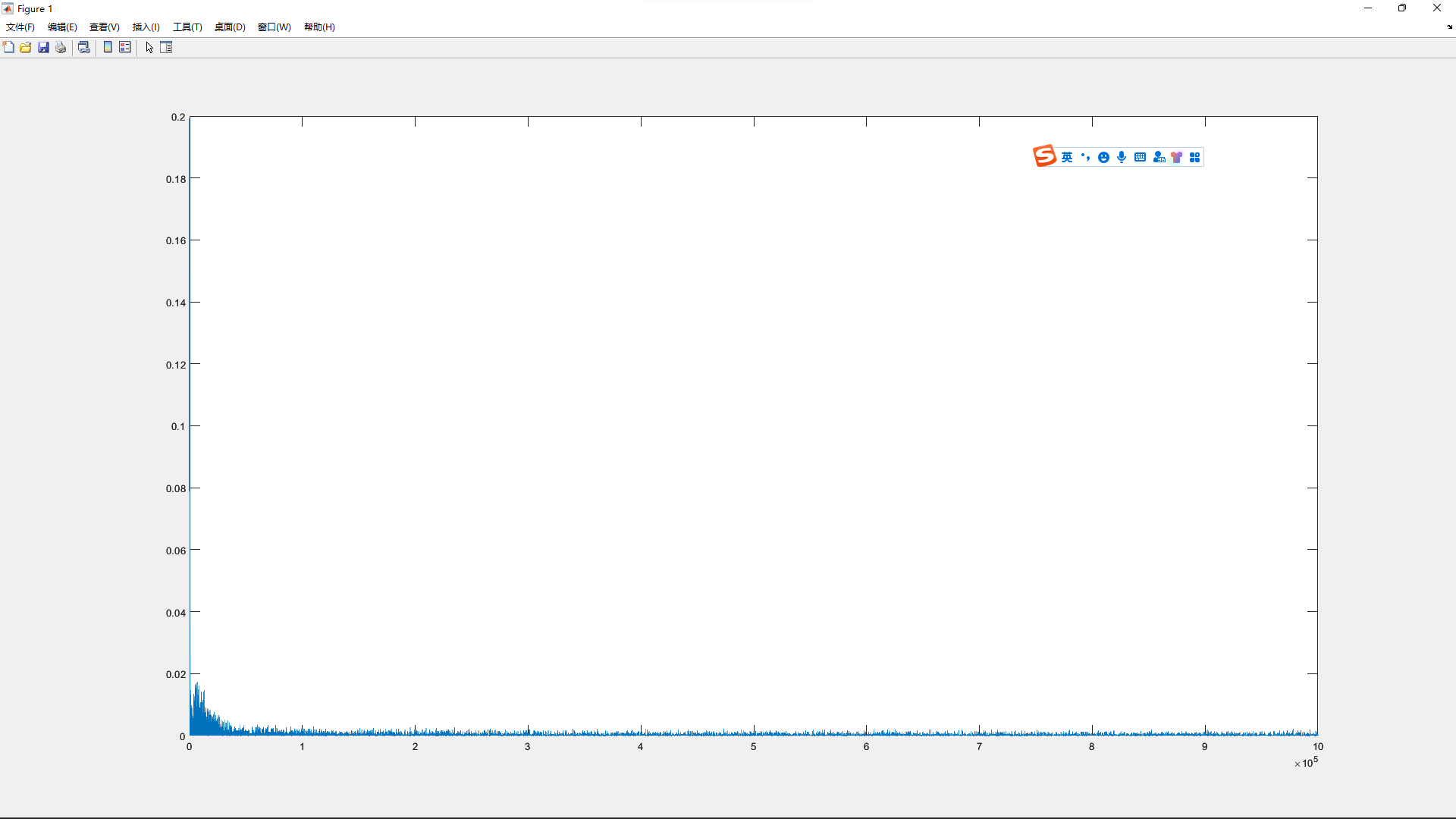


图 5 隐藏层节点数为250 损失函数

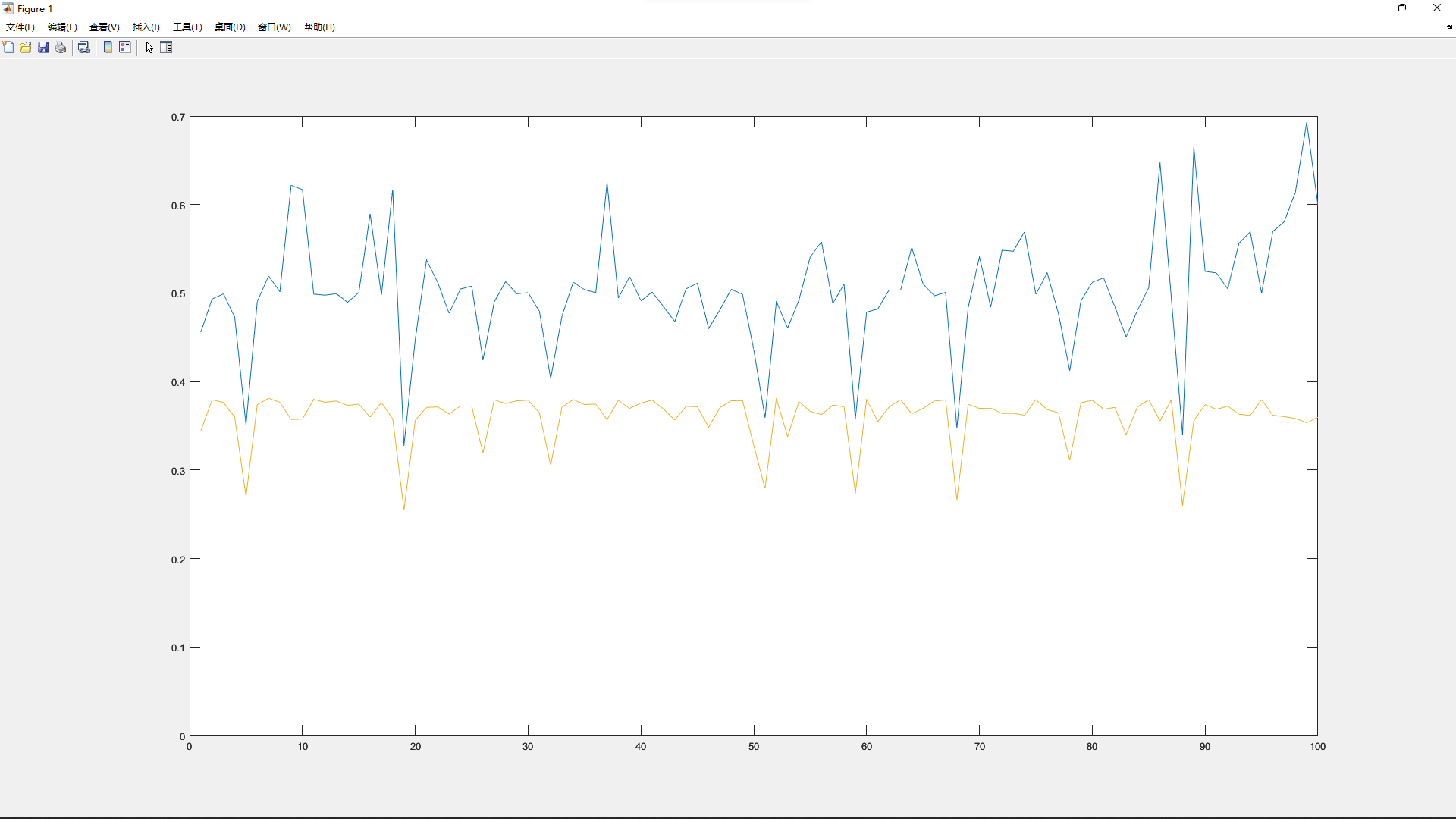
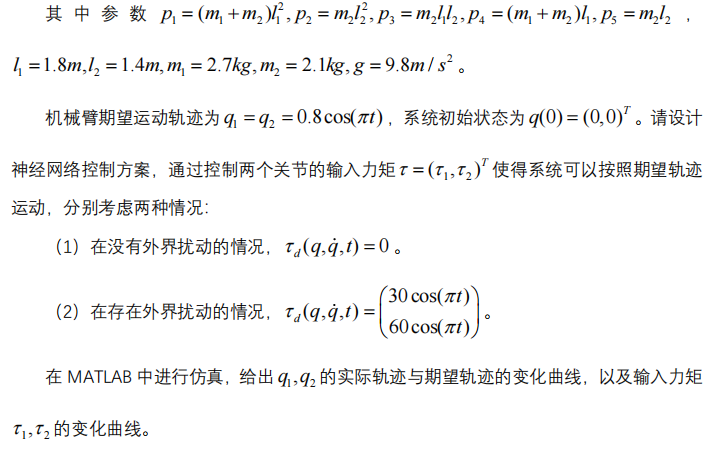
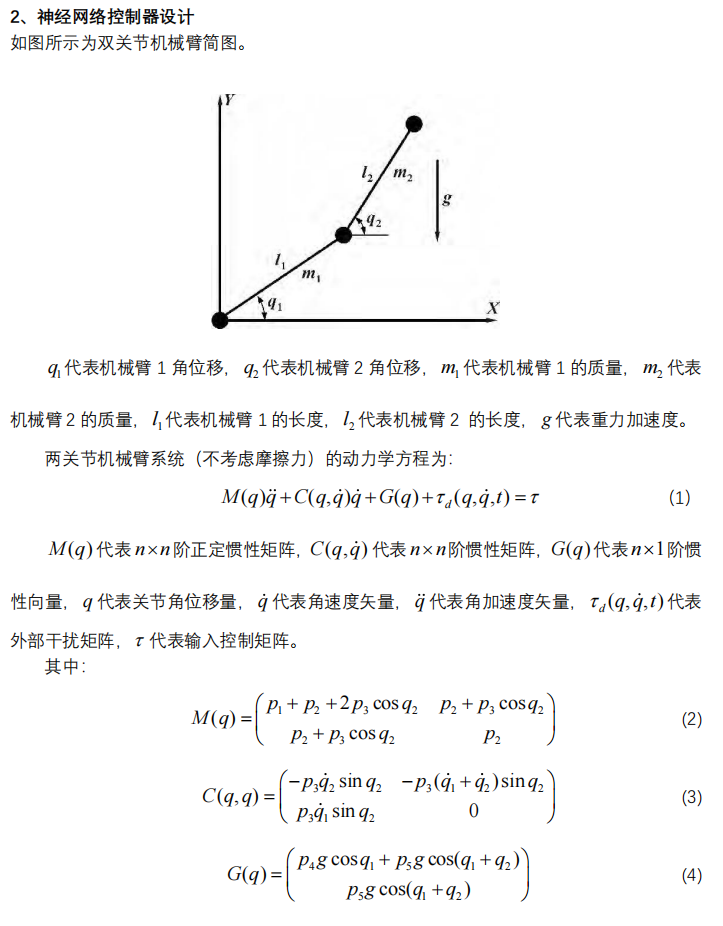


图 6 隐藏层节点数为10 真实系统与辨识系统比较 黄-真实系统 蓝-辨识系统

由以上三则设制不同的隐藏层节点数的BP网络表现可知：在隐藏层节点数为10时，网络特化性不足，其训练得到模型与真实模型输出存在一些偏差；当隐藏层节点数为250时，网络泛化性不足，其训练得到的模型与真实模型输出出入极大；当隐藏层节点数为50时，获得了比较好的辨识效果，其模型已经储存在EX1\_BPN\_model.mat中。

**EX2 神经网络控制器设计**



思路：使用配置好的PID的输入（即偏差、偏差的积分、偏差的倒数）输出（即控制量）作为训练集分别训练两个神经网络，应用于两个关节的控制

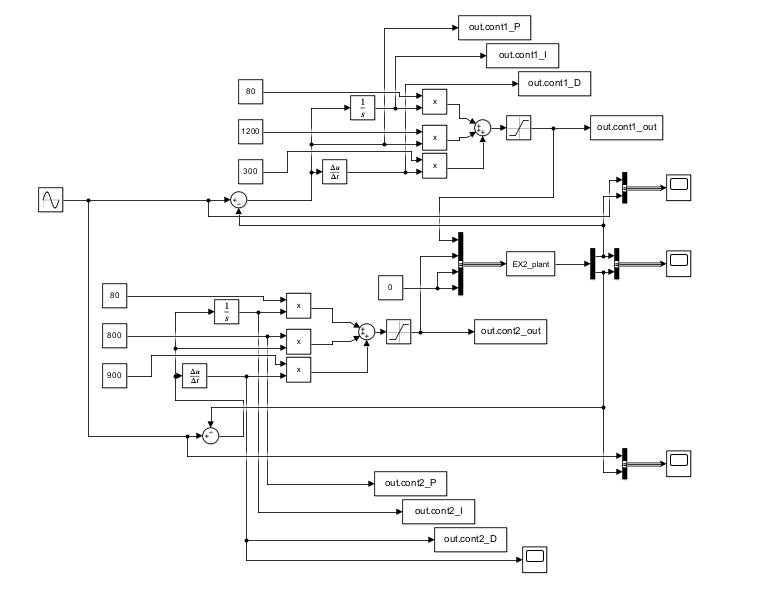
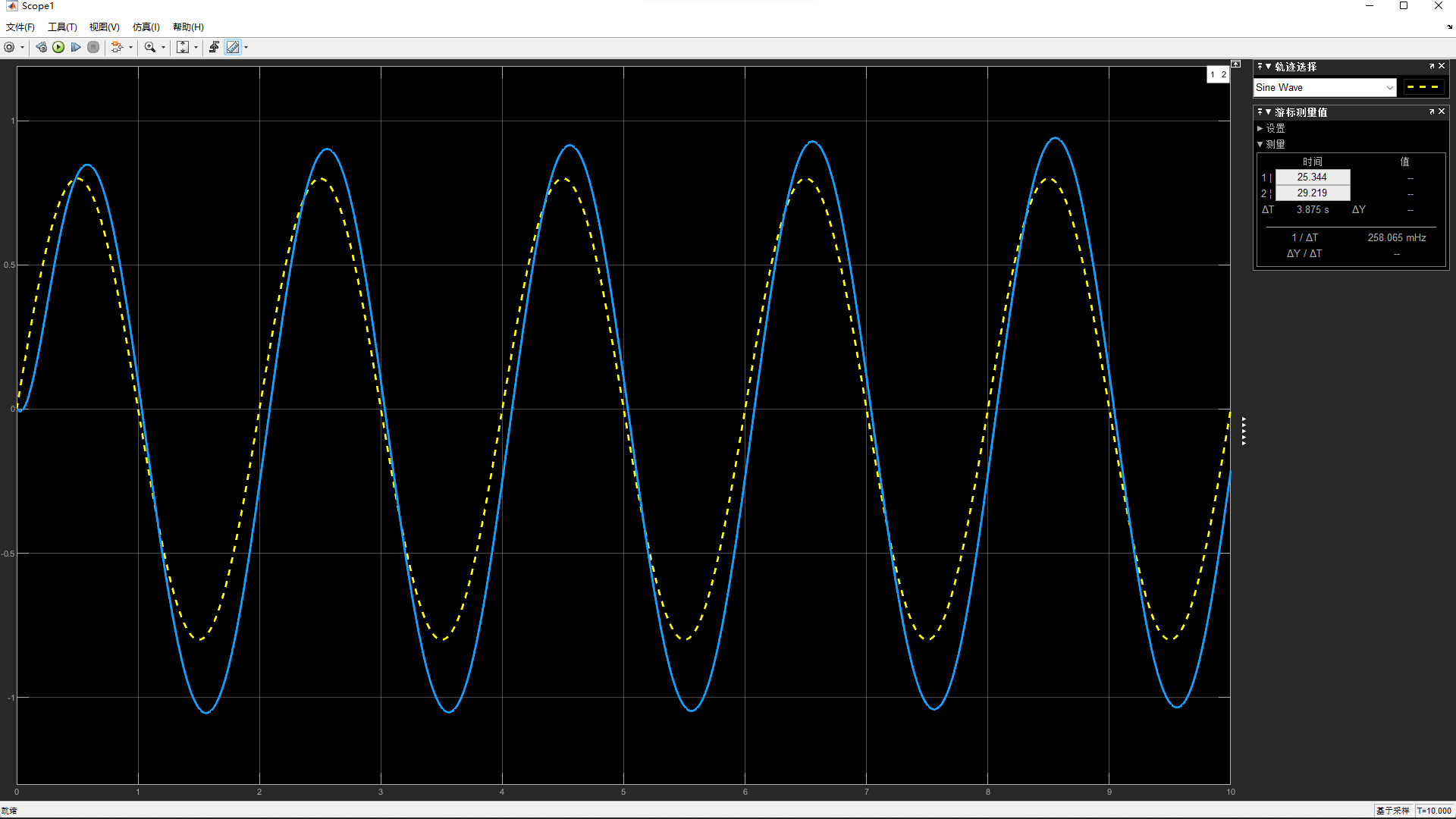


图 7 训练数据生成仿真模型 见EX2\_Simu.slx



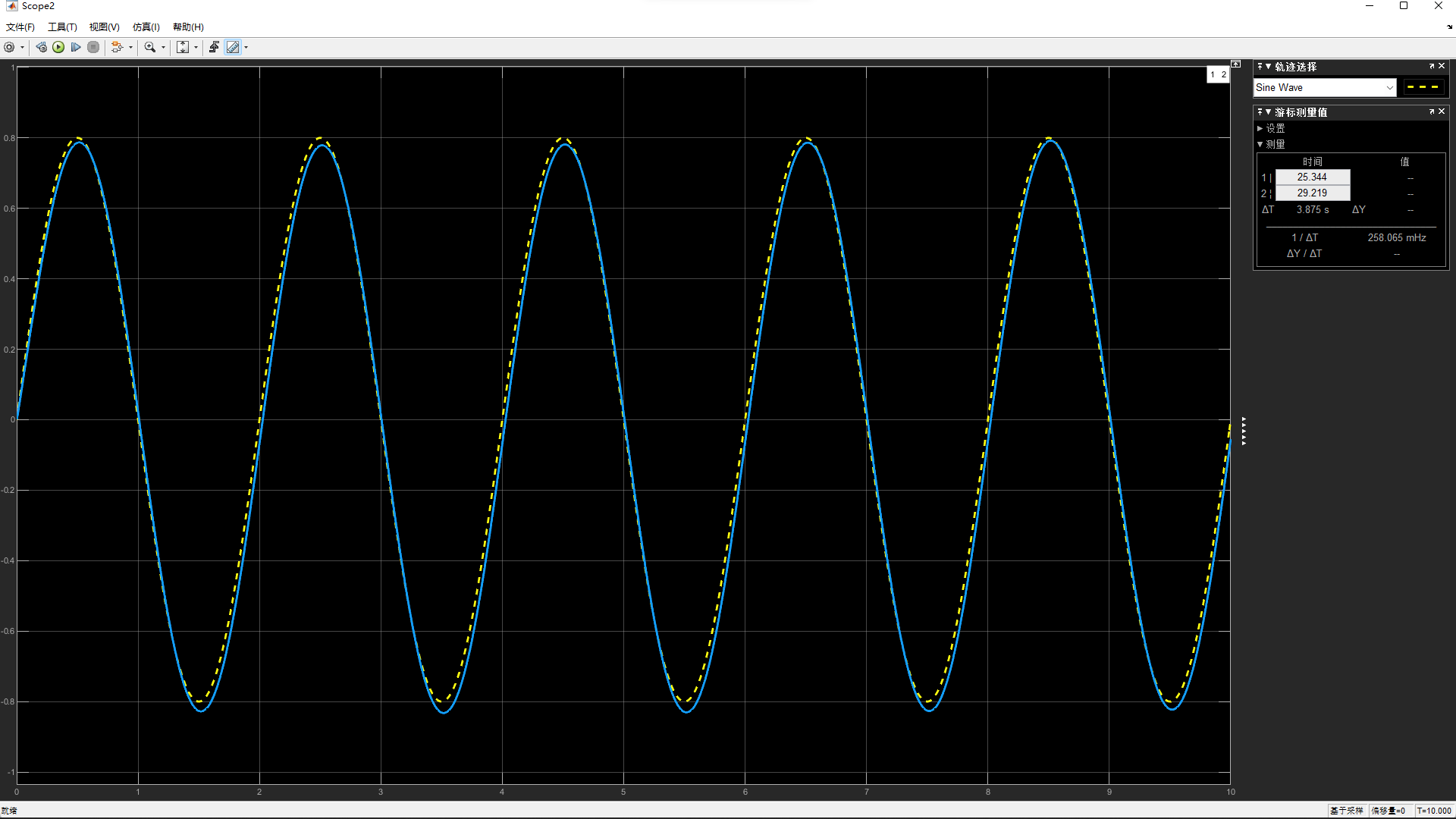


图 8 PID控制器控制下的系统输出 黄虚-期望输出 蓝实-实际输出

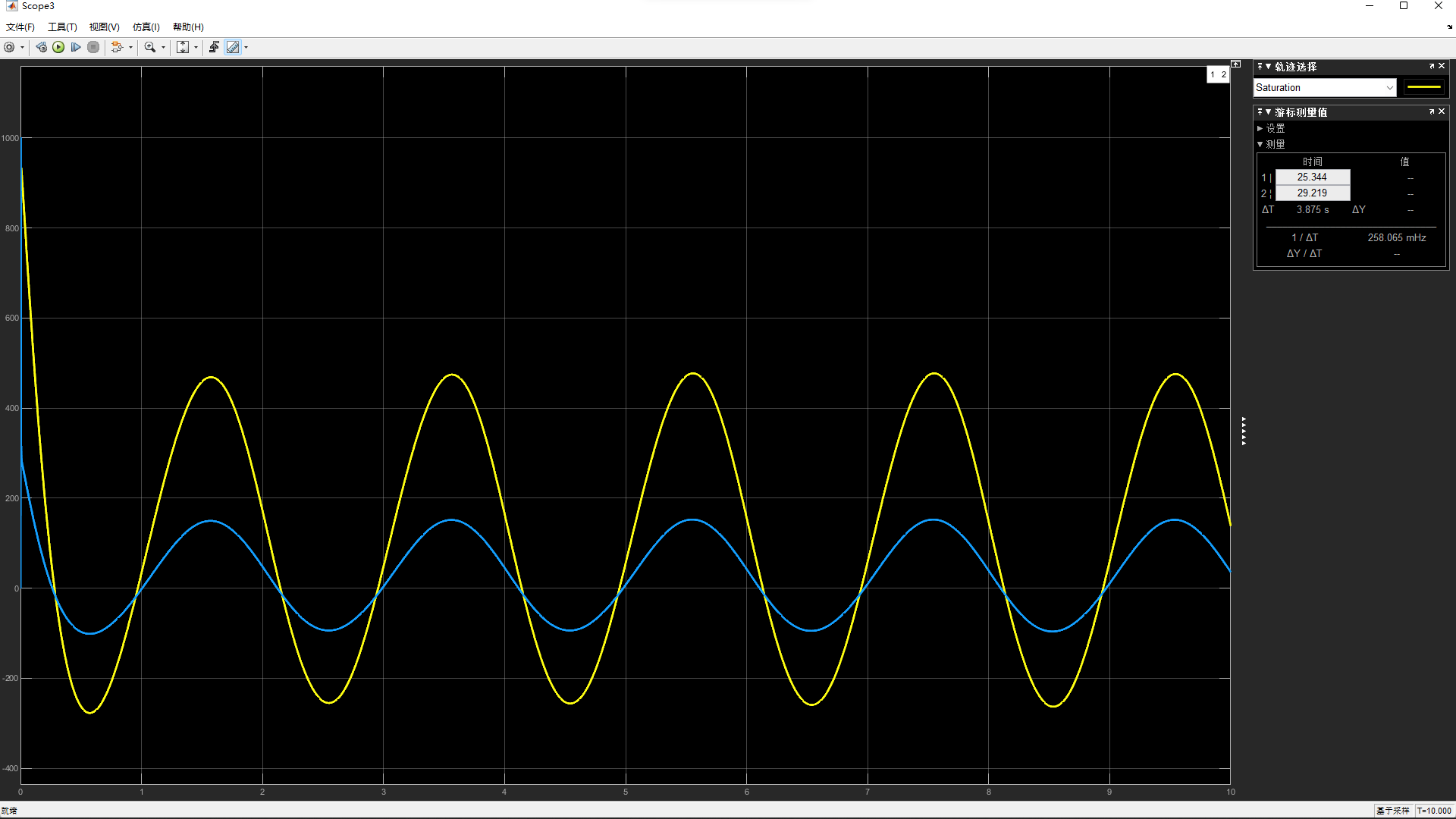
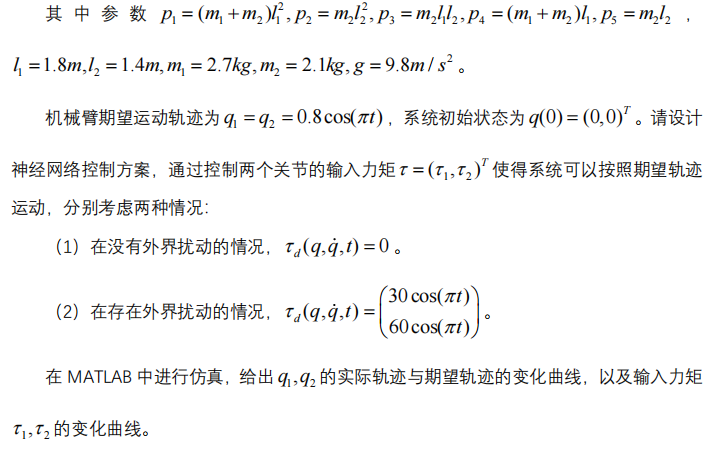


图 9 PID控制器控制下两个关节的力矩输出 黄实-关节1力矩 蓝实-关节2力矩

脚本EX2\_dataManagement.m可以整理仿真数据为可用的数据集

之后可以使用神经网络拟合模块利用这一数据集训练神经网络

训练模型为单隐藏层，隐藏层节点数均为20，训练得到模型的可以输出至Simulink模块作为控制器



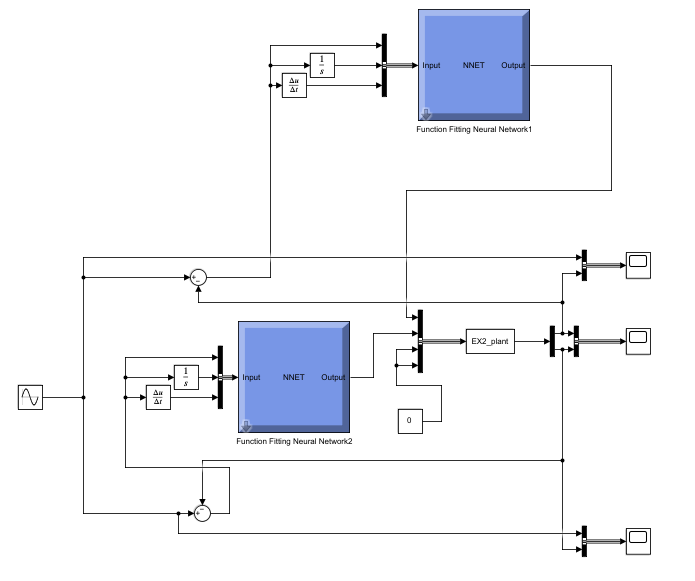
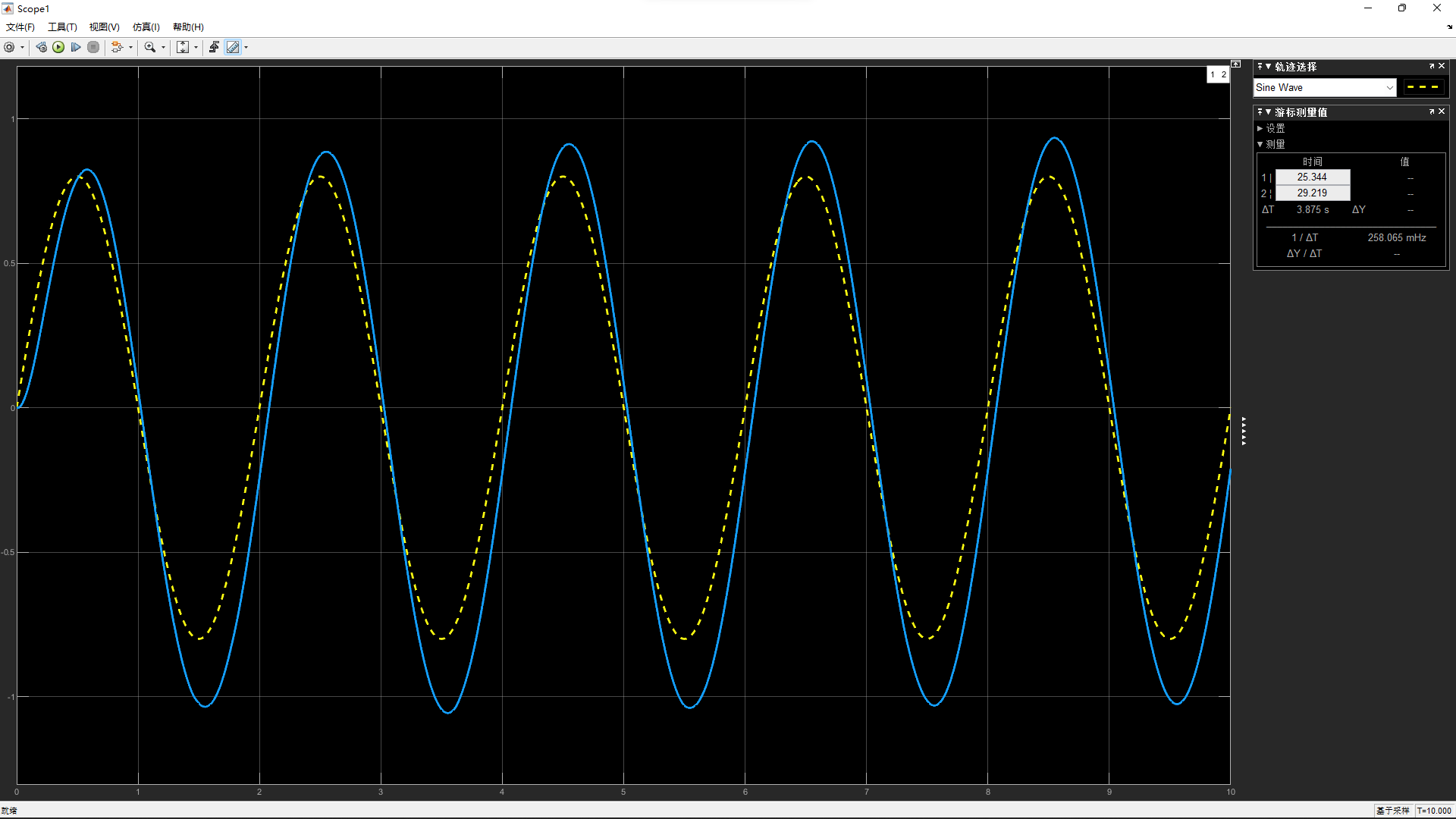


图 10 神经网络控制系统无干扰仿真模型

在没有干扰的情况下，上述系统控效果为：



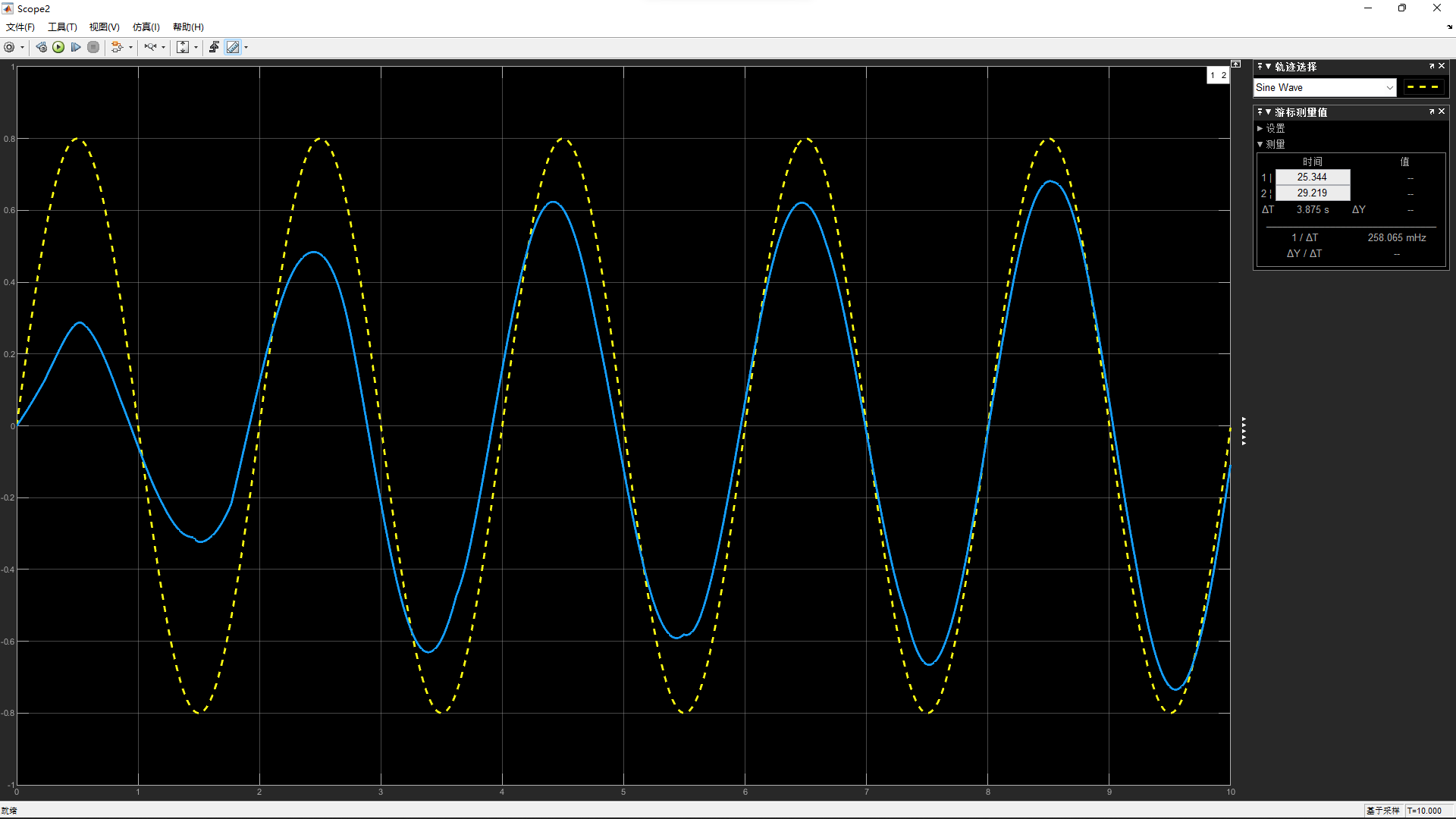


图 11 神经网络控制器控制下的系统输出 黄虚-期望输出 蓝实-实际输出

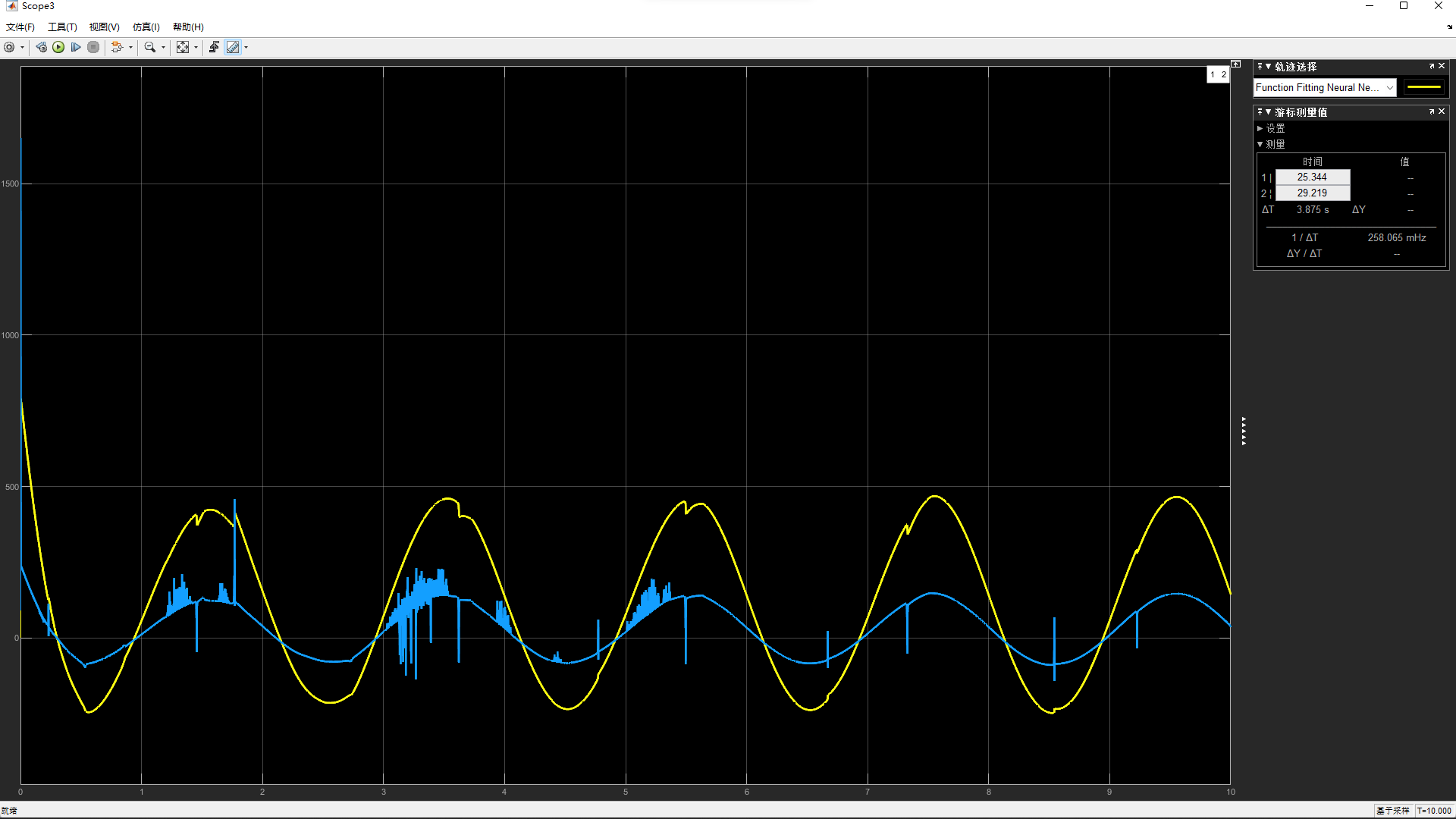
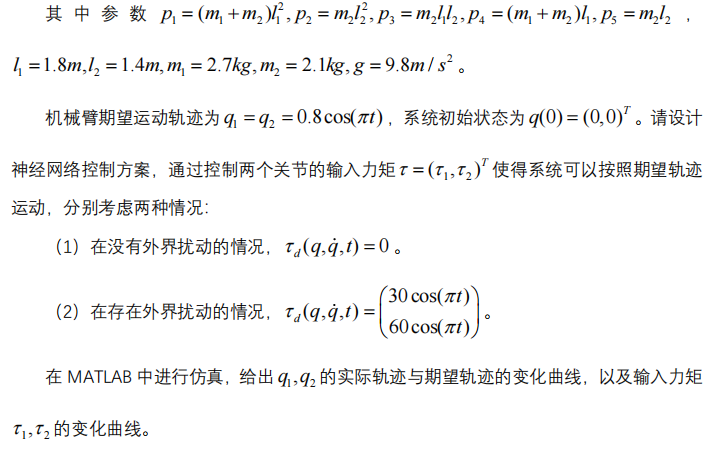


图 12 神经网络控制器控制下两个关节的力矩输出 黄实-关节1力矩 蓝实-关节2力矩

相比之下，从系统输出可以看出，关节1的神经网络控制效果和PID控制基本一致令人比较满意，关节2瞬态响应过程时间相对较长，并不是十分理想。从控制器力矩输出来看，神经网络控制器与PID控制器相比整体区别不大，但关节2控制器输出存在很多不稳定的突变也影响了关节1控制器的输出，关节2控制器在控制过程开始时的极限值也略大于PID控制器设制的饱和值，但对整体响应影响不大。



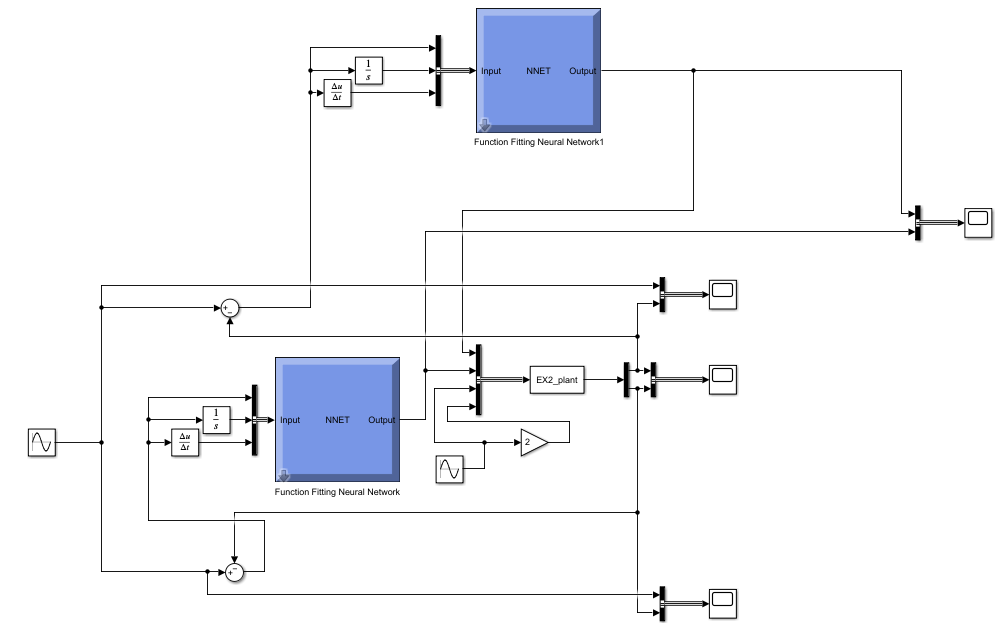
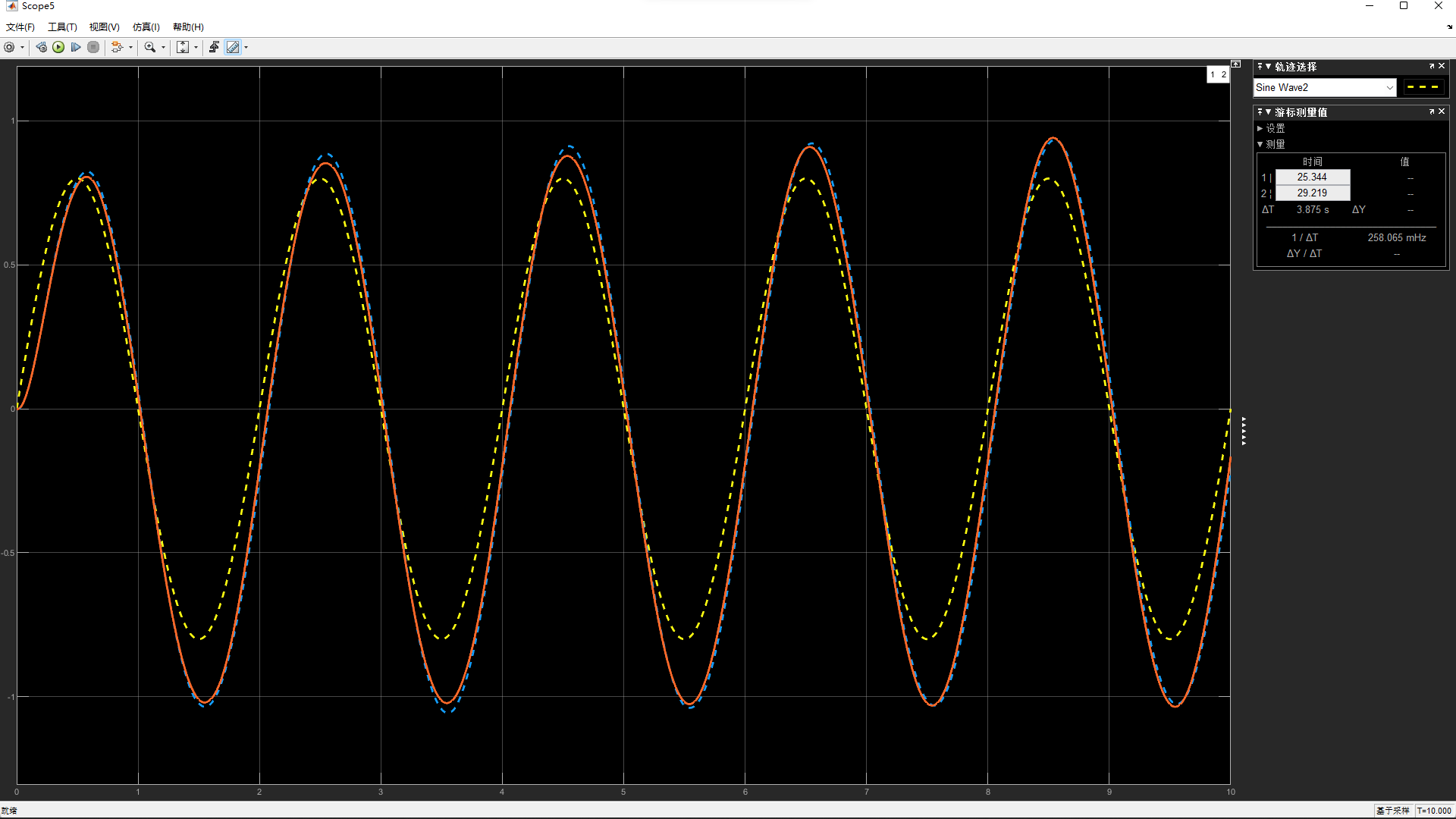


图 13 神经网络控制系统无干扰仿真模型 见EX2\_Simu\_NNController.slx



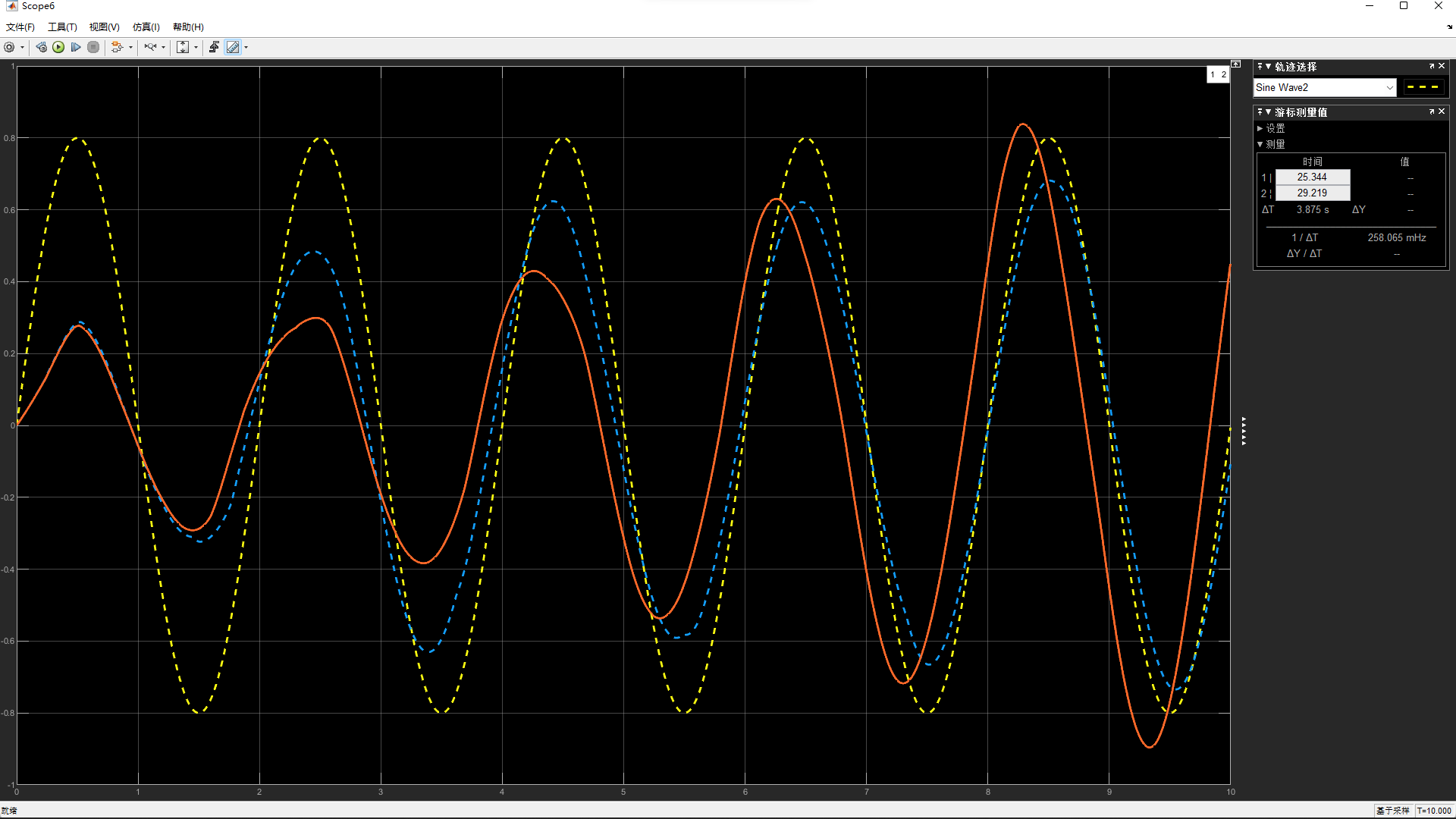


图 14 神经网络控制器控制下系统输出 黄虚-期望输出 蓝虚-无干扰输出 橙实-有干扰输出

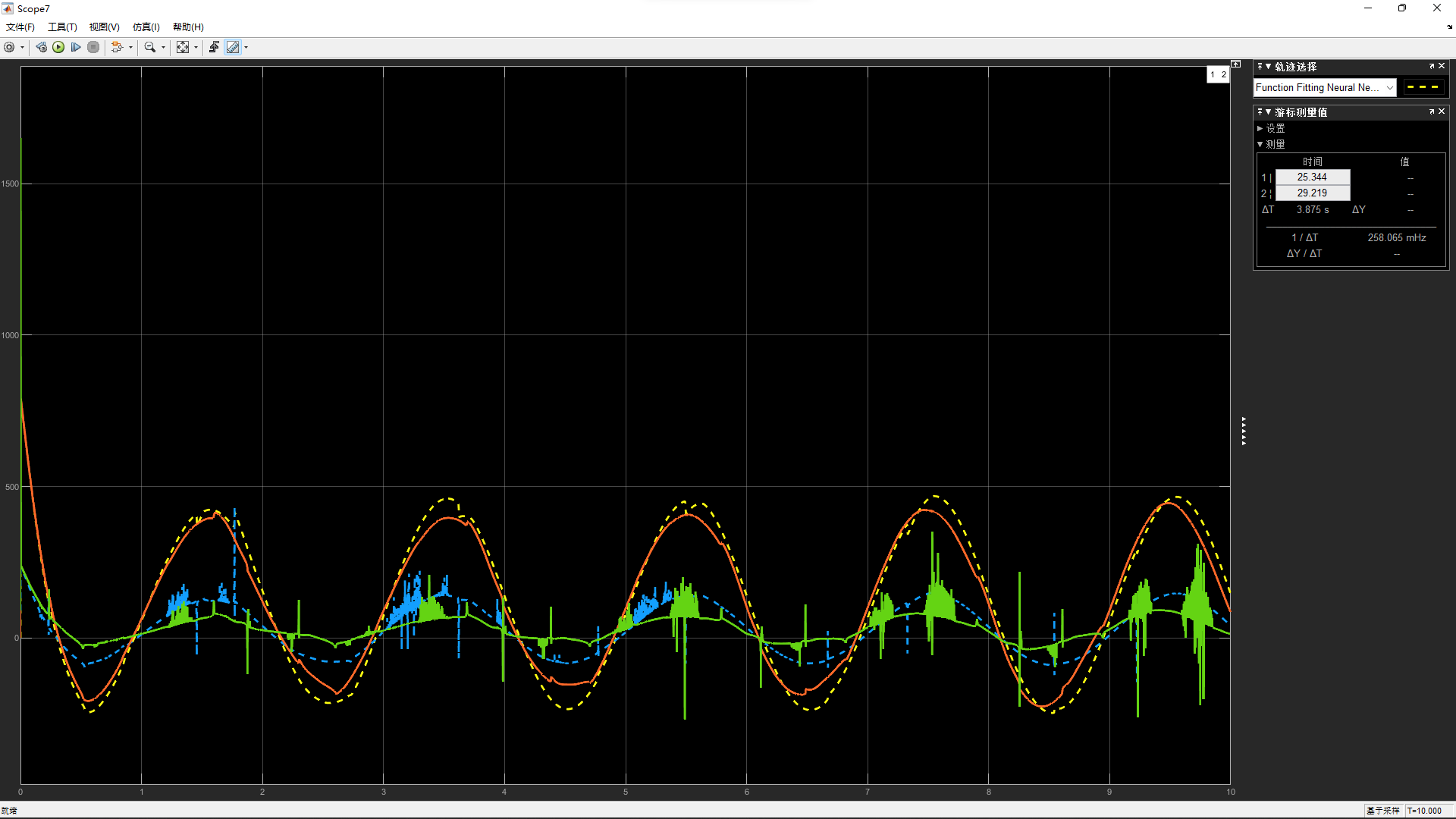


图 15 神经网络控制器控制下两个关节的力矩输出 黄虚-无干扰关节1力矩 蓝虚-无干扰关节2力矩 橙实-有干扰关节1力矩 绿实-有感染关节2力矩

相比之下，从系统输出来看，干扰力矩并没有给关节1的输出产生明显影响，但是却明显使得关节2的暂态过程时间增加并产生了一个提前相位。从力矩输出来看，干扰力矩整体上减少了控制器力矩输出。