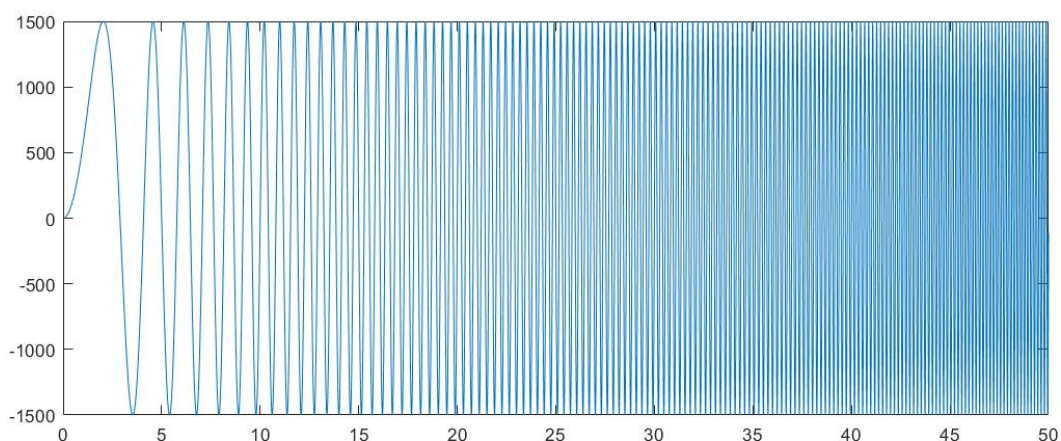


系统辨识及PID参数调节

系统辨识

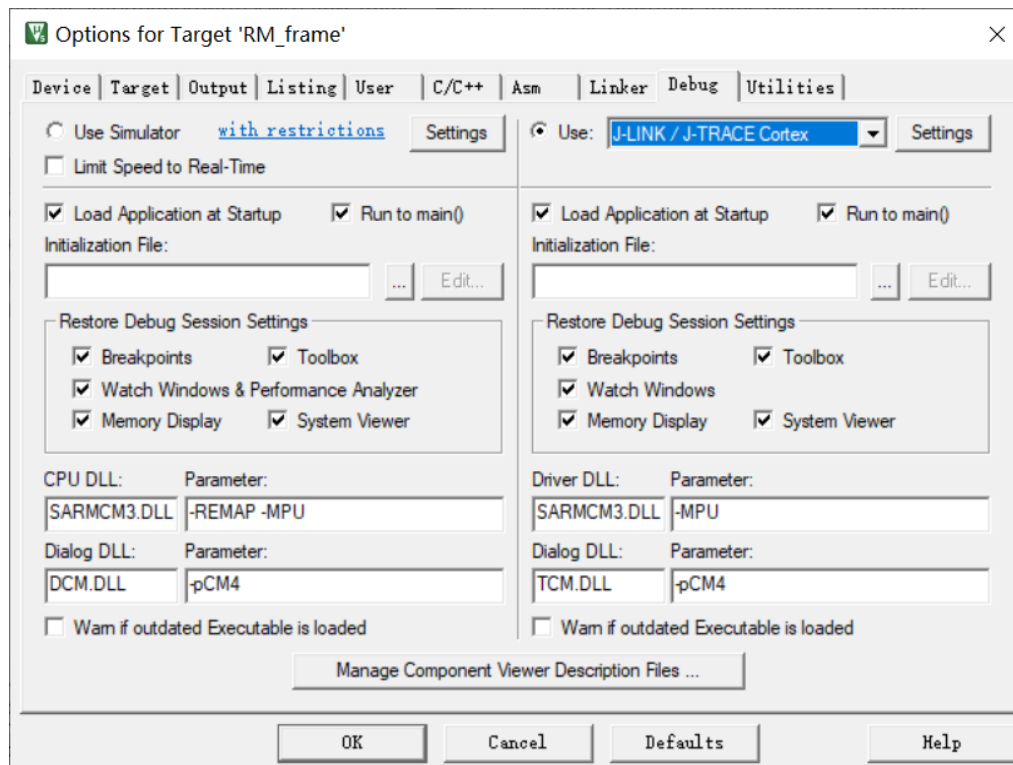
以云台6623电机为例

1. 利用MATLAB生成采样频率为500Hz，幅值为1500，从0Hz到10Hz的扫频信号，并生成为txt文件（程序：sweep_wave_script_txt.m）



2. 利用生成的扫频信号作为 GMY.Intensity 的输入激励电机转动，使用 J-LINK 代替 ST-LINK 作为 Debugger，需要在 Settings 中检查一下连接。（[J-Link的SWD接线方式](#)）





3. 利用 Jscope 监测电流、角度和角速度输出值。此时电机应该**已经安装好了实际的负载**，因为系统辨识是需要得到该电机在工作状态下的传递函数，用这个数学模型来模拟实际情况，因此**需要在装好的车上进行系统辨识**，并且**要保证电机和所带负载在运动行程中没有受到机械限位的约束**。

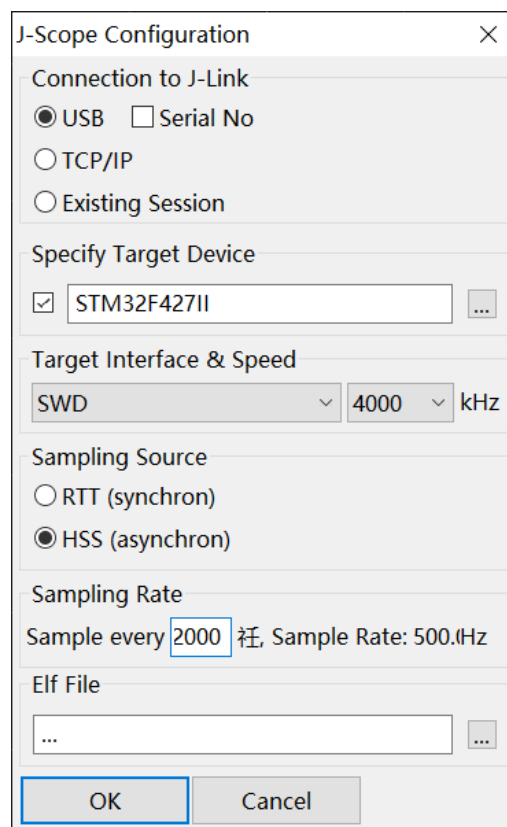
注意： 电流、角度和角速度对应的变量必须乘1000转化为整形才能被 Jscope 读取

```

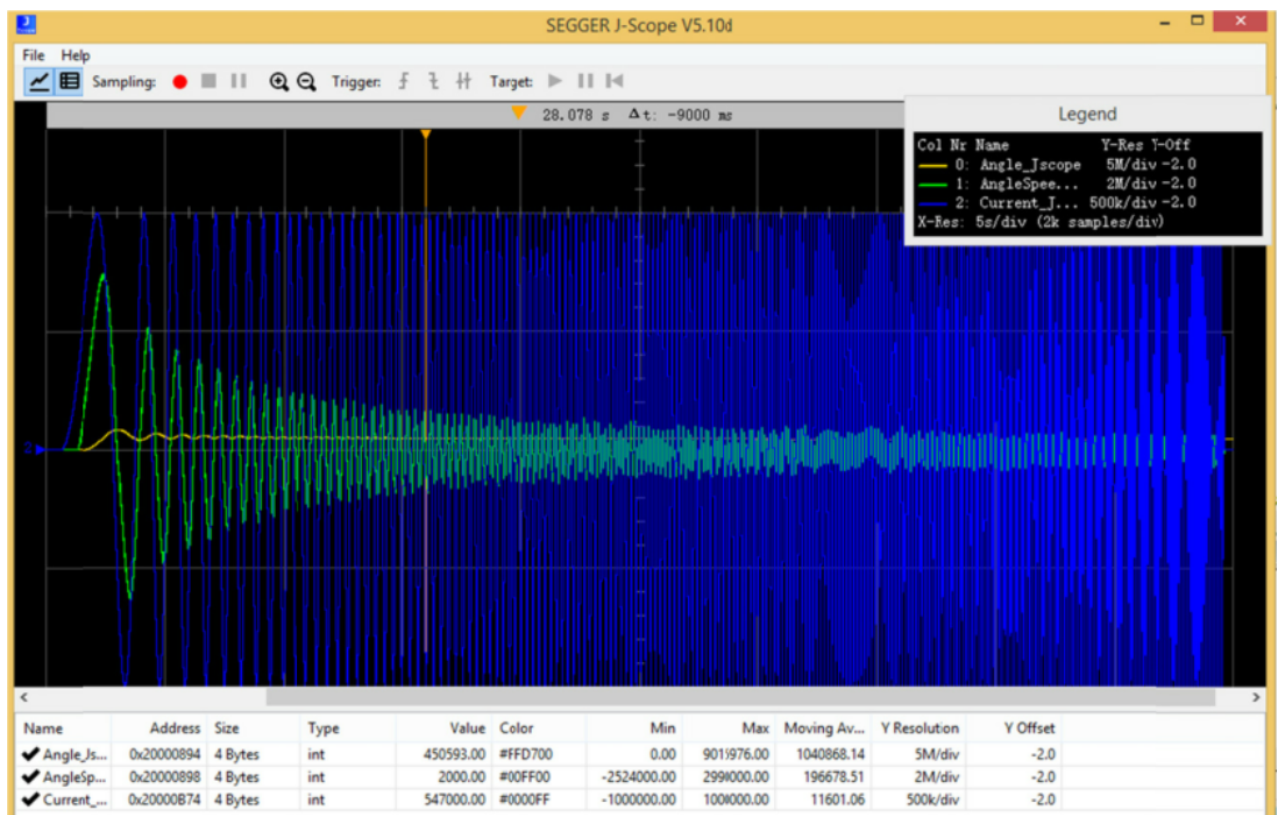
GMYtarget_int=(int)(GMY.Intensity*1000);
GMYAngleSpeed_int=(int)(imu.wz*1000);
GMYAngle_int=(int)(imu.yaw*1000);

```

Jscope 的配置如下，Sample Rate 是 500Hz，因此前面是 2000；Elf File 就是我们程序编译生成的 RM_frame.axf 文件：

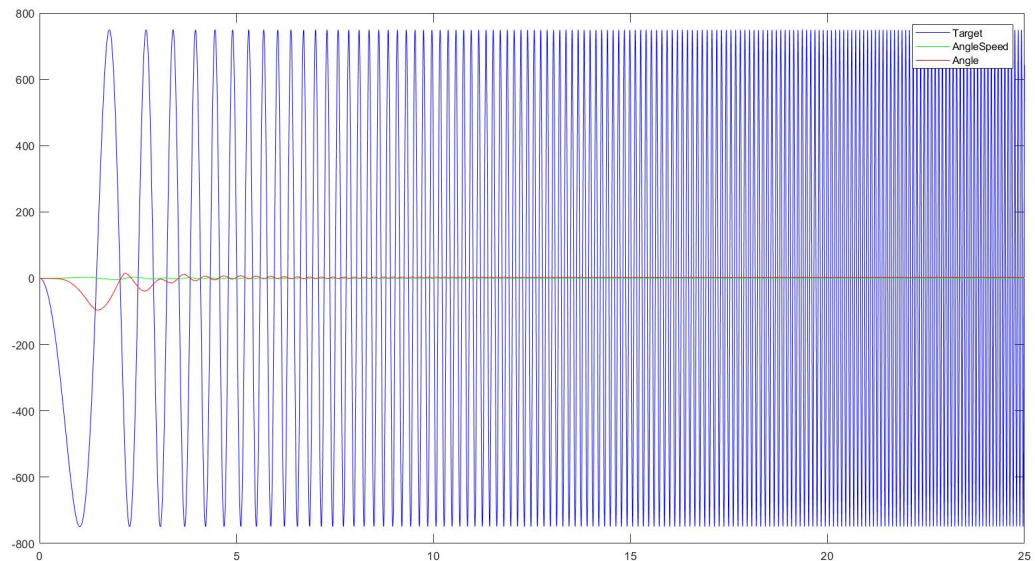


Jscope 的输出形状大致如下图，蓝色为输入的扫频信号，绿色为角速度值，黄色为角度值：



- 将 Jscope 的数据导出到CSV中，在Excel 365中用数据导入功能，将CSV文件转化为 .xlsx 格式，并在Excel中对数据进行必要的预处理，找出一个合适的测量段范围。例如，在 mydata.m 文件中所用的 0126_1502.xlsx 数据的测量段是 B3083:D15582，从扫频起点开始，正好是12500个采样值，也就是一次扫频信号的输出结果，其中三列分别是扫频信号、角速度、角度。

注意：此处三列数据均要除以1000，变换回 double 值。

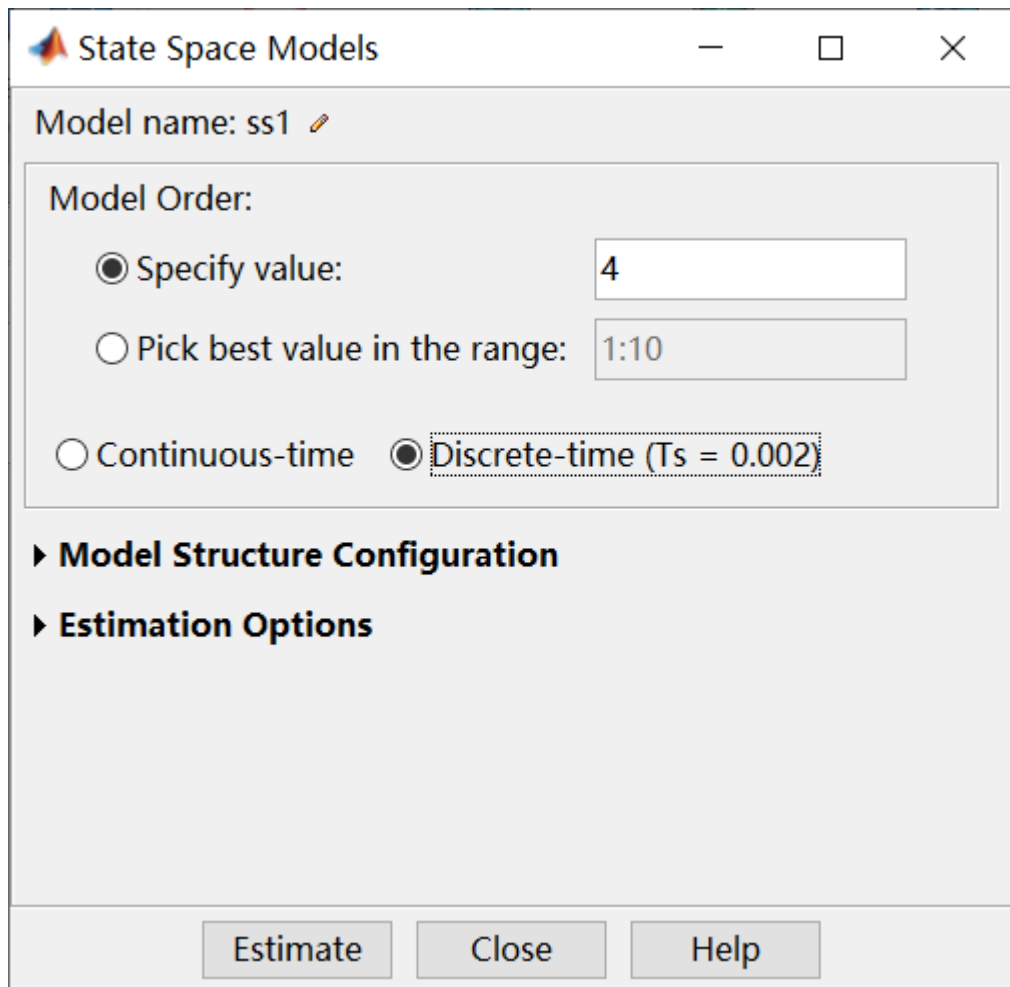


上图的输出结果和 Jscope 上意义相同，只不过我使用了相反的扫频信号（当时受到了另一侧的机械限位，以后可以直接用正的扫频信号）

5. 打开MATLAB的 System Identification 工具箱。

- Import data 中选择 Time domain data，在弹出的对话框中输入：

- Estimate 中选择 State Space Models，默认采用四阶的，修改为离散时间，点击 Estimate，可以得到一份辨识报告，适配度在99.96%左右：

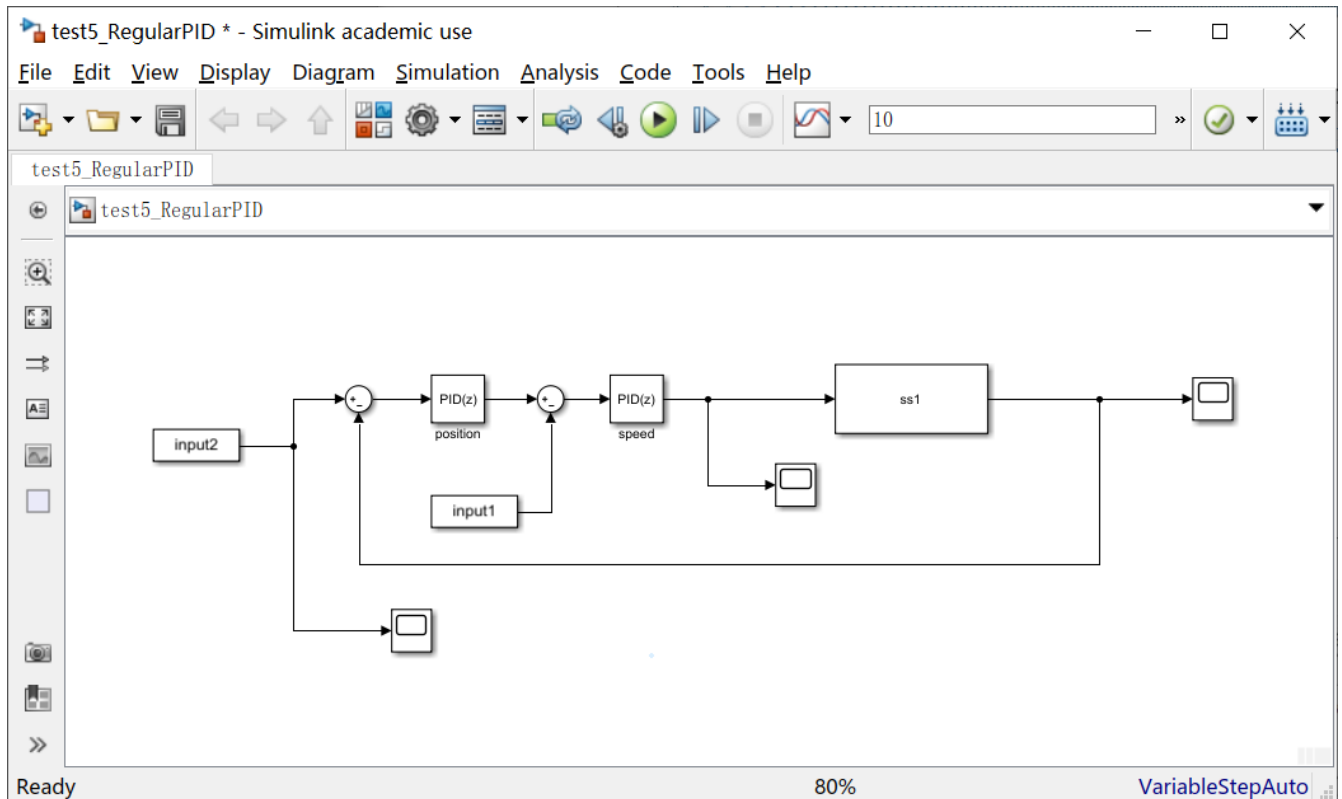


- 此时 Model Views 中就生成了一个模型 ss1，将其拖拽到 To workspace 控件上，即可在工作区看到这个模型，即为辨识到的系统状态空间表达式。

注意：可以使用扫频数据和角速度数据按照上述过程也辨识一个模型，用于 mydataplot.m 中进行测试比对。

结合Simulink调节PID参数

Simulink模型是 test5_RegularPID.slx 文件。其中 ss1 就是刚辨识出来的模型，input1 是时间和角速度的增广，input2 是时间和角度的增广，这两个在 mydataplot.m 中生成。该Simulink模型由速度环和位置环组成，位置环的输出作为速度环的输入，将其与实际角速度进行差分。



- 调节PID参数时，先调节速度环的PID：

Block Parameters: speed

PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PID** Form: **Parallel**

Time domain:
☐ Continuous-time
☒ Discrete-time

Discrete-time settings
 Integrator method: **Forward Euler**
 Sample time (-1 for inherited): **0.002**

Main | PID Advanced | Data Types | State Attributes

Controller parameters

Source: **internal** [Compensator formula](#)

Proportional (P): **68.2624729425209**

Integral (I): **224.342953487122**

Derivative (D): **0.0680381299890337**

☐ Use filtered derivative

Filter coefficient (N): **192.316032733822**

Select Tuning Method: **Transfer Function Based (PID Tuner App)** **Tune...**

Initial conditions

Source: **internal**

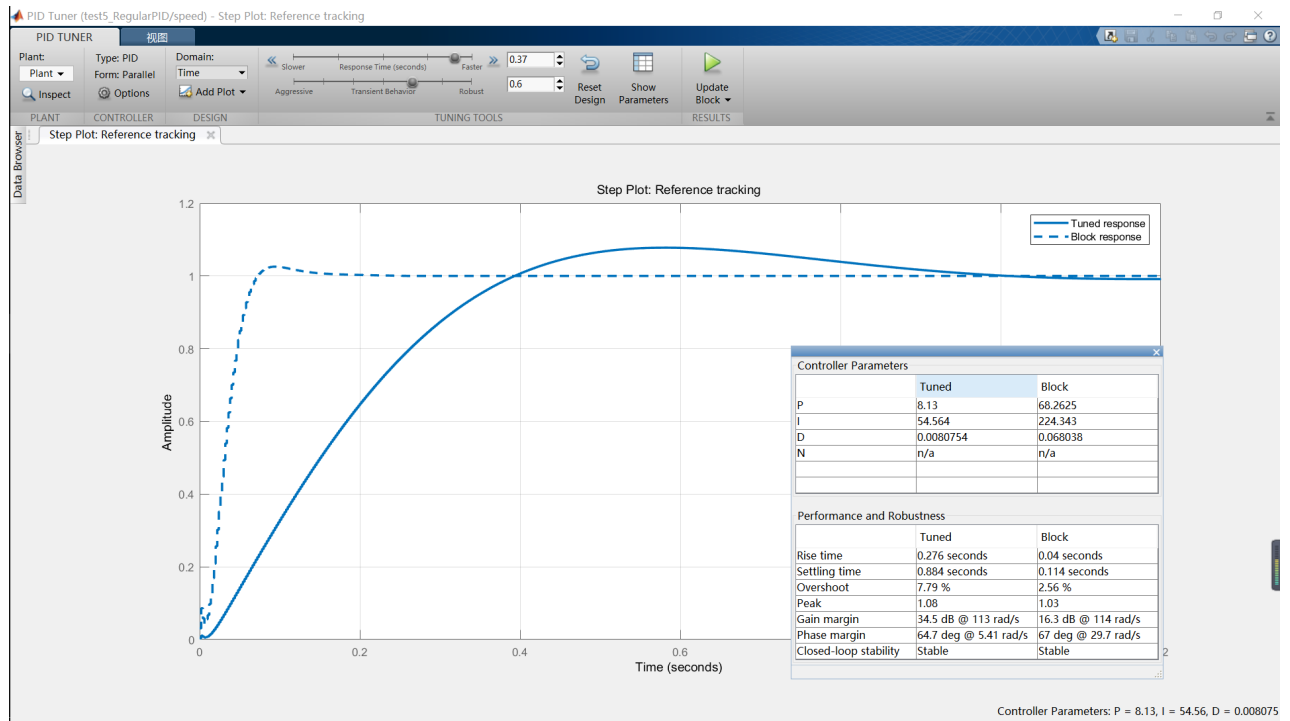
Integrator: **0**

Differentiator: **0**

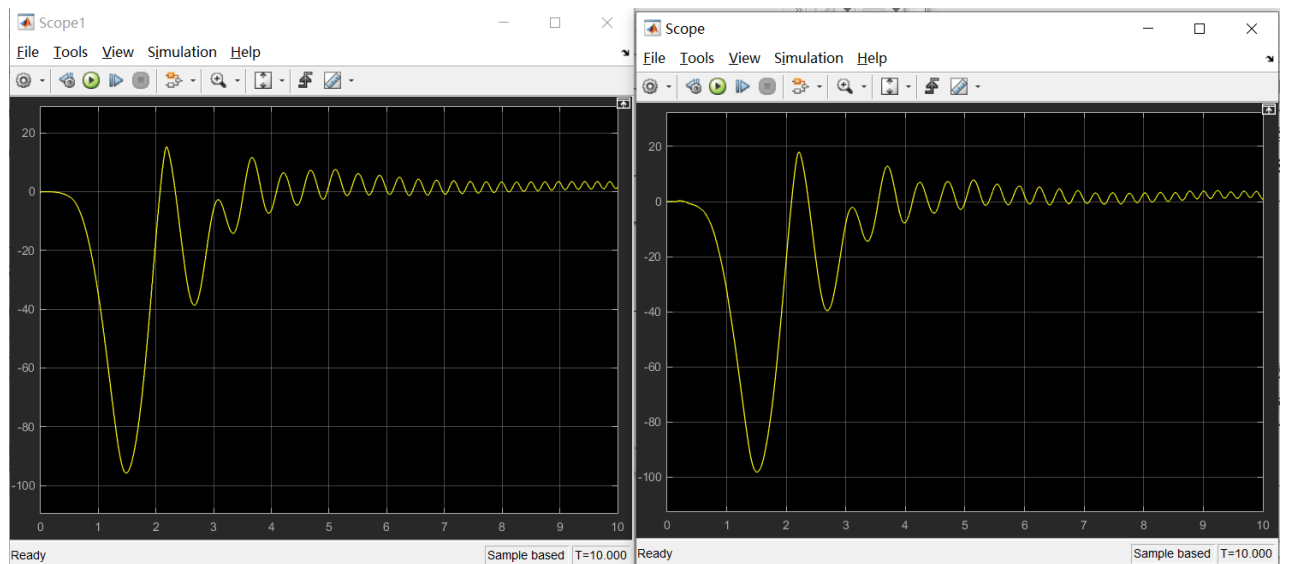
$$P + I \cdot T_s \frac{1}{z-1} + D \cdot \frac{1}{T_s} \frac{z-1}{z}$$

OK Cancel Help Apply

- 点击 **Tune...**，打开 **PID Tuner App**，利用该工具箱调节PID，如下；



- 调节好速度环之后，输出PID参数，点击 **Update Block**，即可在 **Block Parameters** 中看到调整好的PID参数；
- 位置环PID同理；
- 最后输入、输出的Scope如下图（本例前段数据有异常，只要输入输出相似就好）。



得到的速度PID和位置PID写入frame的相应电机PID参数中，即可得到较好的效果，接下来在此数据基础上微调即可。