

# 系统辨识及其在 PID 参数调节上的应用-电控组

## 1. 系统辨识

(1) 利用 MATLAB 生成 50s 幅值为 1000<sup>1</sup>频率从 0Hz 到 10Hz 变化的扫频信号，作为电调的输入。如图 1 所示。

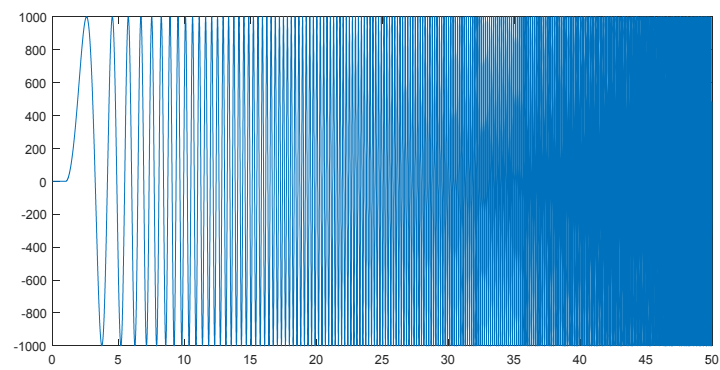


图 1 扫频电流信号

(2) 利用上一步生成的电流信号激励电机运转，再利用 Jscope 监测电流、角度和角速度的值。注意，电流、角度和角速度对应的变量必须转化成整形才能被 Jscope 读取。如图 2 所示，蓝线表示电流、绿线表示角速度、黄线表示角度。

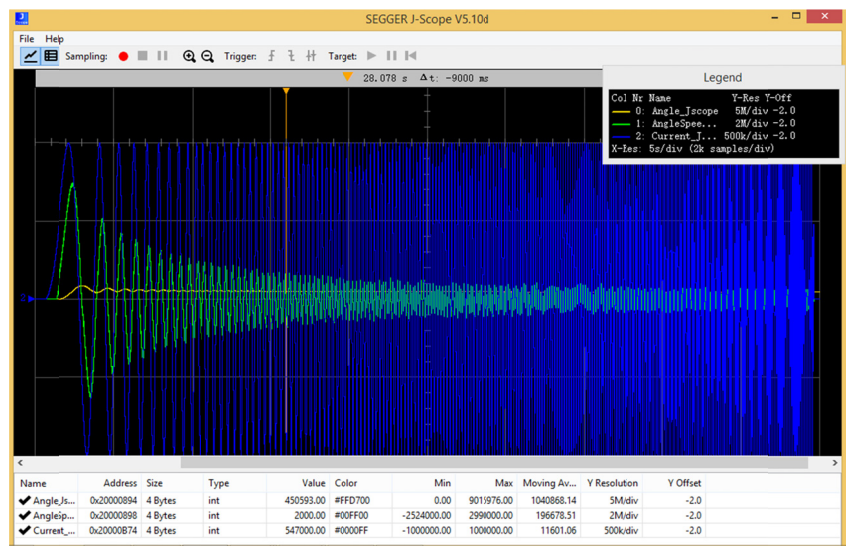
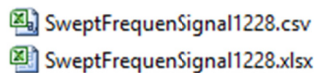


图 2 Jscope 监测得到的电机运转电流、角度和角速度（纵轴单位并没有实际意义，单位转换请看电机说明书）

<sup>1</sup> 单位不是 A，请查阅电调说明书。

(3) 将 Jscope 显示的电流、速度和角速度信号导出到 CSV 文件中并将 CSV 文件另存为 XLSX 格式的 Excel 文件，便于在 MATLAB 中操作。



(4) 由于信号保存在 xlsx 文件中为字符格式，因此利用 MATLAB 的相关符号操作函数把字符型变量转化成数值型变量，同时电流、角度和角速度要除以 1000，保证和真实测量情况一致。最后在 MATLAB 中显示的电流、角度和角速度如图 3 所示。

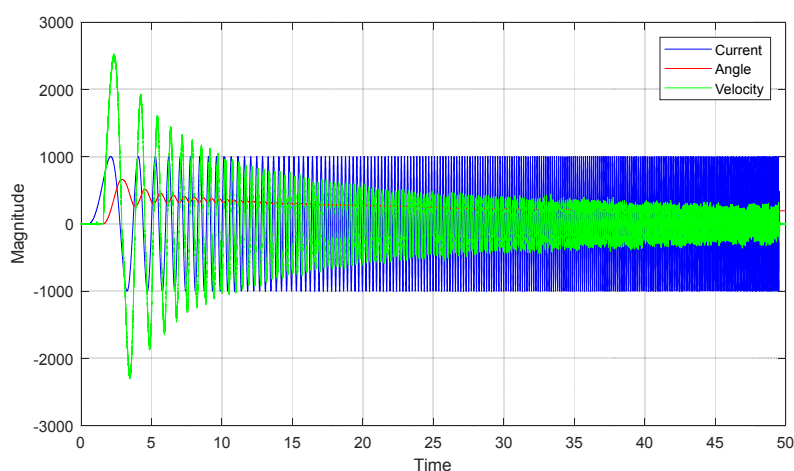


图 3 利用 MATLAB 绘制的电流、角度和角速度

(5) 利用 MATLAB 中的 System Identification APP，将电流作为输入，角速度作为输出辨识电机的传递函数模型，在本例中传递函数的分母设为 3 阶，分子设为 2 阶。辨识过程如图 4 所示。

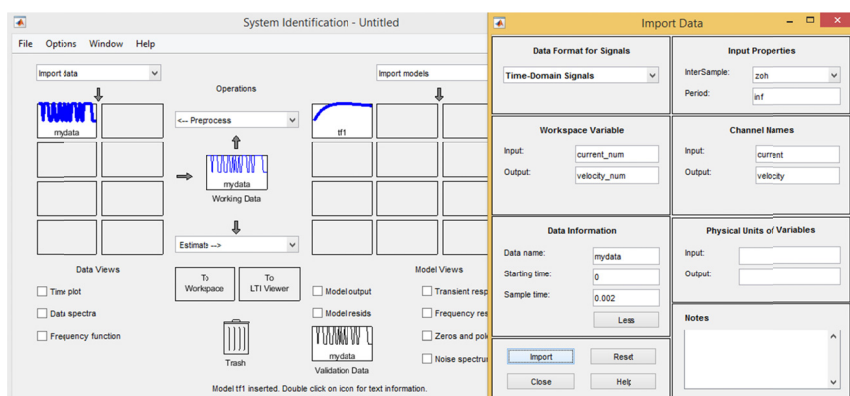


图 4 System Identification 应用包以及电流信号和角速度信号的导入

(6) 进行模型验证。从时域图来看，辨识模型的输出和电机真实输出的拟合度为 81.22%。整体和局部结果图放大图见图 5，其中黑线为实际输出，蓝线为辨识模型输出。

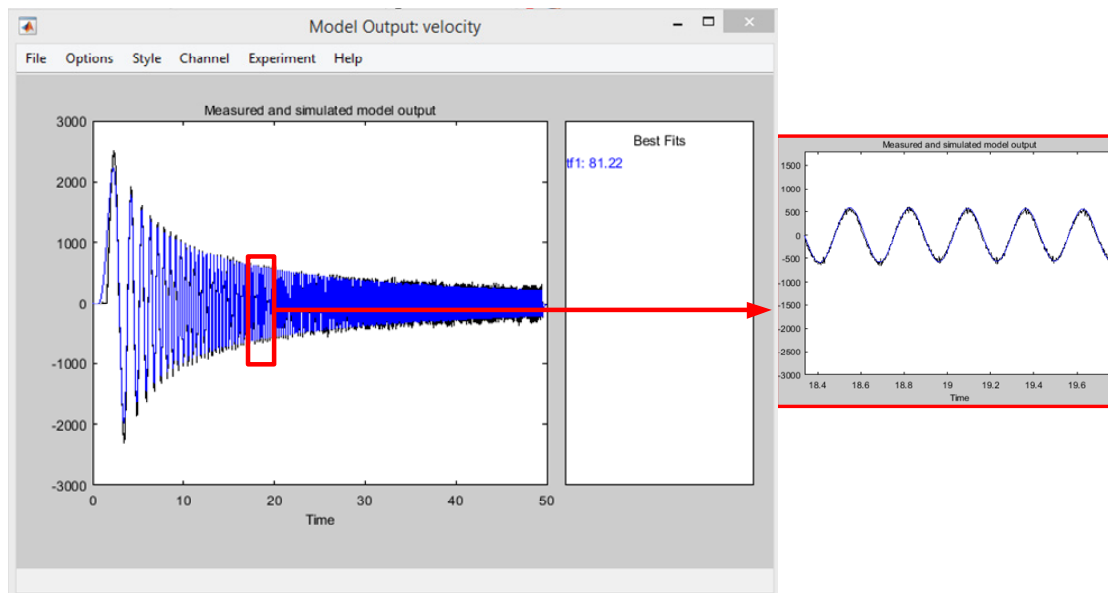


图 5 系统辨识结果时域图

从频域图（图 6）上看，系统的频率响应和和传递函数的频率响应一致性比较高。

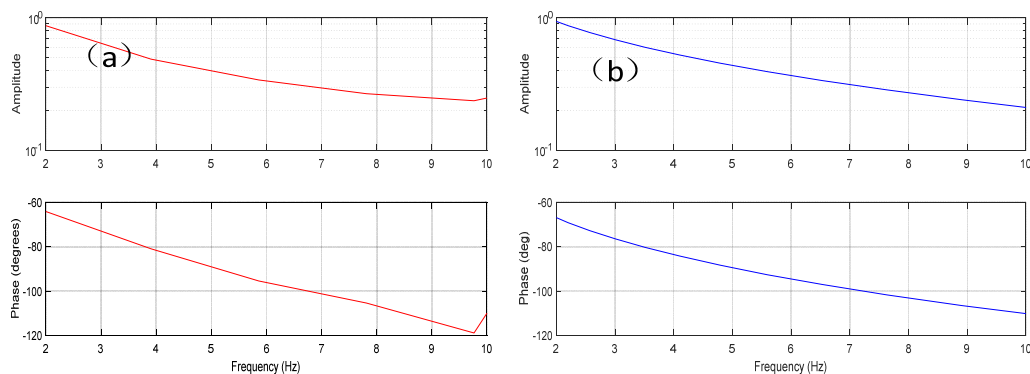


图 6 系统辨识结果频域图：（a）真实系统的频响函数；（b）传递函数的伯德图

(7) 传递函数的导出。将 tf1 拖到 to workspace 框图上，即可将 tf1（transfer function 1）导入到 MATLAB 的工作空间，如图 7 所示。最终的传递函数表达式见图 8。

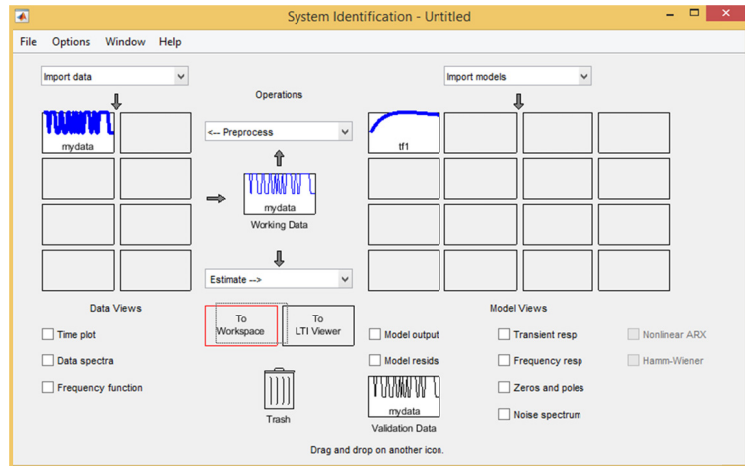


图 7 传递函数导入到工作空间

```

From input "current" to output "velocity":
      0.00625 z^-1 - 0.006075 z^-2
-----
      1 - 2.748 z^-1 + 2.505 z^-2 - 0.7577 z^-3

Name: tf1
Sample time: 0.002 seconds

```

图 8 传递函数的表达式

## 2. PID 参数调节

(1) PID 参数的整定。将传递函数 tf1 导入到 MATLAB 的 PID Tuner APP，并利用它找到相对较好的 PID 控制器的参数（Kp、Ki 和 Kd），如图 9 所示。为了保证 MATLAB 的 PID 算法和板卡程序中的 PID 算法一致，需要对控制器的特性进行设置，如图 10 所示，Integral Formula 和 Derivative Formula 均为 Backward Euler。

(2) 将 Kp、Ki 和 Kd 的数值赋给电机的控制程序对应的变量，然后烧录程序到板卡上，观察电机转速的实际控制效果如图 11 所示，从图中可以看出基于系统辨识的 MATLAB 辅助 PID 参数调节方法能较快地找到合适的 PID 控制器参数，极大地提高程序调试效率。

文档编辑：易思成。感谢简心语的联调，感谢张芳园、刘文通的协助！

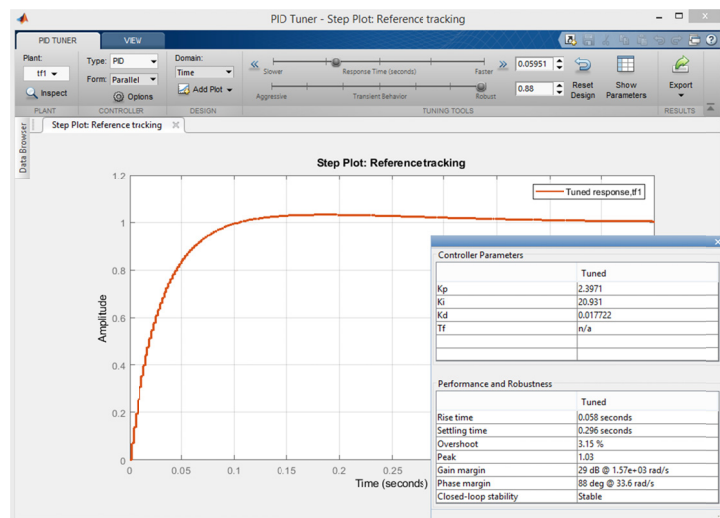


图 9 PID Tuner 应用包的操作

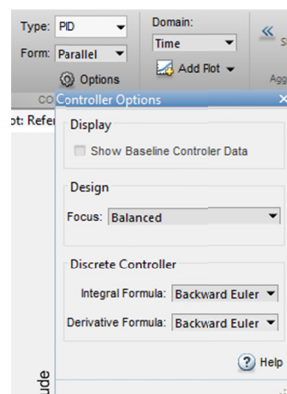


图 10 控制器设置

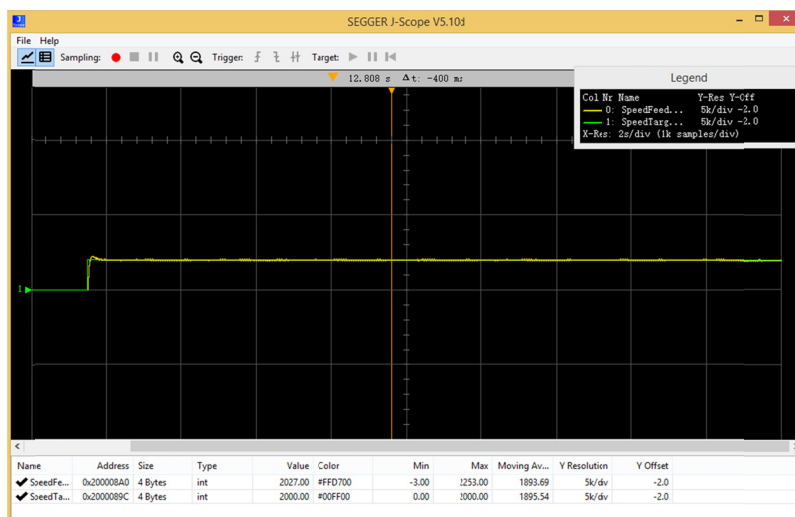


图 11 PID 参数整定后，PID 控制效果图（绿线是期望角速度，黄线是实际角速度）