



# hpm\_motor 库使用说明

---

先楫半导体《hpm\_motor库使用说明》

## 目录

<b>1</b>	<b>简介 .....</b>	<b>4</b>
<b>2</b>	<b>电机库的添加.....</b>	<b>4</b>
2.1	库内容 .....	4
2.2	如何添加库.....	4
<b>3</b>	<b>库功能说明 .....</b>	<b>5</b>
3.1	轨迹规划算法.....	6
3.1.1	功能简介 .....	6
3.1.2	接口说明 .....	7
3.1.3	使用方法 .....	8
3.1.4	配置示例 .....	10
3.2	惯量辨识算法.....	12
3.2.1	功能简介 .....	13

版本:

日期	版本号	说明
2022-5-13	1.0	初版

# 1 简介

hpm\_motor 是一个针对电机控制核心算法的库文件，旨在通过不断地更新完善，模块化  
管理具体功能算法；对用户来说不用去关心底层如何实现，加速用户程序开发。

该库目前仅支持在 windows 操作系统下使用。

## 2 电机库的添加

### 2.1 库内容

HPM 电机库包含如下内容：

- 包含函数声明，宏声明，结构体声明的头文件
- 包含初始化配置函数和实时运行函数的.a 文件
- 一个 demo 示例，展示 HPM MOTOR 库使用。
- 文档

### 2.2 如何添加库

如需使用库里面的一些功能，请先将库添加工程下引用，具体引用流程如下：

(1) 将 hpm\_motor 文件添加到 middleware 分支下



图 1

## (2) 工程引用 hpm\_motor 库

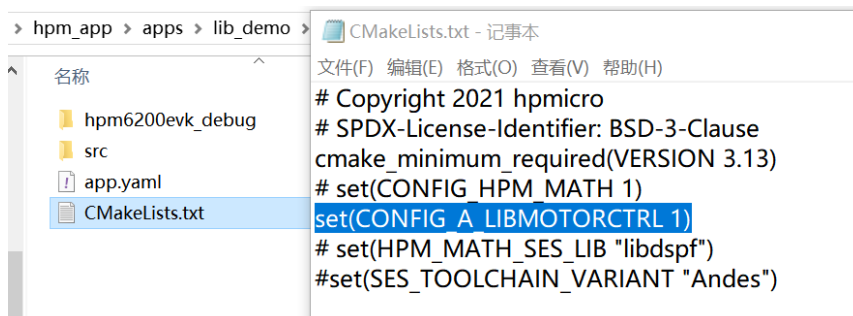


图 2

## (3) 新建工程，打开工程后可以看到 hpm\_motor 库被工程引用

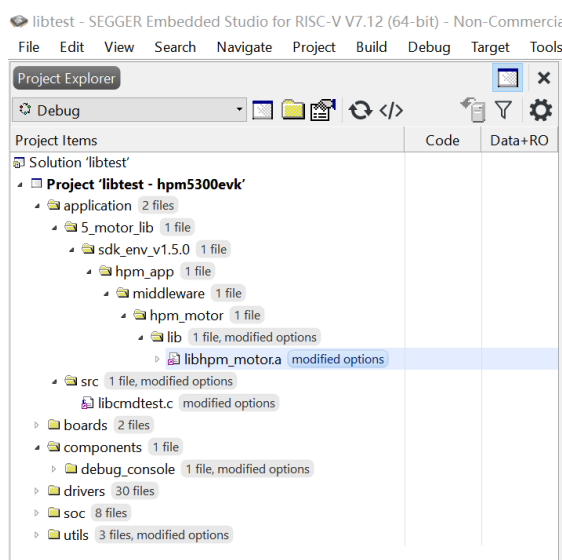


图 3

## (4) 工程编译

这里要说明一下：hpm\_motor 库文件使用，同时支持 gcc 以及 andes 工具链。

# 3 库功能说明

当前 1.0 版电机库，支持轨迹规划算法函数接口调用，其他算法功能会迭代更新完善。用

户使用之前请先调用版本号函数接口获取版本号，与.a 文件显示的版本号核对。

## 3.1 轨迹规划算法

电机在启停时，速度存在阶跃的情况。电机速度的突变可能会导致电流过载。所以通常用加减速控制算法来规划电机速度。

libhpm\_motor.a 中使用的轨迹规划算法是 S 型曲线，其核心是通过可变加加速度大小来控制加速度的变化，保证加速度不突变，使得加减速的速度曲线平滑，以此达到电机平稳运行的目的。

### 3.1.1 功能简介

S 型曲线主要由加速段 ( $T_a$ )，匀速段 ( $T_v$ )，减速段 ( $T_d$ ) 三大部分组成。加速段又分为加加速 ( $T_{j1}$ )、匀加速、减加速 ( $T_{j1}$ )，减速度段分为加减速 ( $T_{j2}$ )、匀减速和减减速 ( $T_{j2}$ )，一共七段如图 1 所示，其中， $v_{max}$  为系统最大加速， $v_0$  为起始速度， $v_1$  为终止速度， $a_{max}$  为系统最大加速度， $a_{min}$  为系统最大减速度， $j_{max}$  为系统最大加加速度， $j_{min}$  为系统最大减减速度。

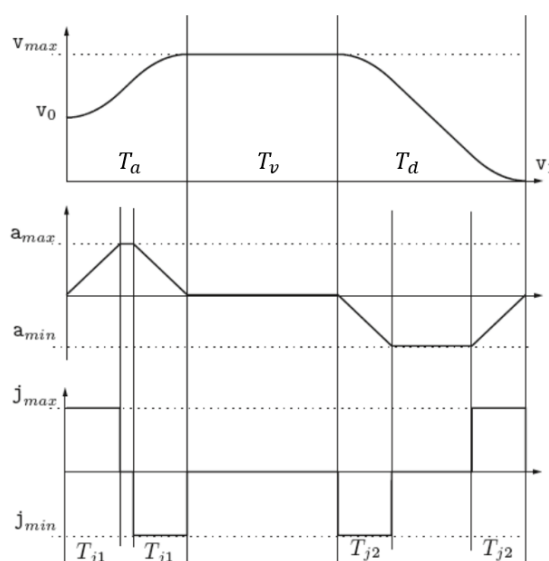


图 4 S 型规划下速度/加速度/加加速度

### 3.1.2 接口说明

#### (1) 入口参数

	名称	说明
轨迹生成配置参数	CMDGENE_POSCFG_PARA	位置模式下
	CMDGENE_VELCFG_PARA	速度模式下
中间变量	CMDGENE_USER_PARA	
输出变量	CMDGENE_OUTPUT_PARA	位置/速度/加速度/加加速度序列

#### (2) 结构体成员

结构体		描述	单位
CMDGENE_POSCFG_PARA	q0	起始位置	unit
	q1	绝对运动位置	unit
	v0	起始速度, 一般设置为 0	unit/s
	v1	终止速度, 一般设置为 0	unit/s
	vmax	最大速度	unit/s
	amax	最大加速度	unit/s /s
	jmax	最大加加速度	unit/s /s/s
	cyclecnt	往返次数	/
	cycletype	运动类型	/
	dwelltime	往返停歇时间	ms
	isr_time_s	更新位置序列的中断时间	s
CMDGENE_VELCFG_PARA	q0	起始位置	unit
	Tv	匀速时间	s
	v0	起始速度, 一般设置为 0	unit/s
	v1	终止速度, 一般设置为 0	unit/s
	vmax	最大速度	unit/s
	amax	最大加速度	unit/s/s
	jmax	最大加加速度	unit/s/s
	isr_time_s	更新速度序列的中断时间	s
CMDGENE_OUTPUT_PARA	poscmd	位移指令	unit
	velcmd	速度指令	unit/s
	acccmd	加速度指令	unit/s/s

注：单位 unit 一般为 r, count

### (3) 函数接口

函数接口名称	说明	调用状态
pos_cmd_gene	位置模式下, 位置/速度序列更新	中断函数内调用
vel_cmd_gene	速度模式下, 速度序列更新	中断函数内调用
cmd_gene_disable	轨迹规划重置	轨迹规划结束后重置

## 3.1.3 使用方法

轨迹规划算法分为以下几步：

Step1：添加库头文件。C 头文件 “libhpm\_motor.h” 包含库使用的函数声明和结构体声明。在主 C 文件的顶部添加以下行：

```
#include "libhpm_motor.h"
```

.h 文件在 hpm\_app\middleware\hpm\_motor\inc 下。

Step2：在“ {ProjectName}-Main.c” 文件中定义轨迹规划结构体全局变量。

```
CMDGENE_PARA cmdpar=0;
```

Step3：用户根据运行模式，运动方式，运动参数对配置参数结构体的各成员赋值

```
cmdpar.cmdgene_in_par.poscfgpar.q0 = 0;
```

```
cmdpar.cmdgene_in_par.poscfgpar.q1 = 20;
```

```
cmdpar.cmdgene_in_par.poscfgpar.v0 = 0;
```

```
cmdpar.cmdgene_in_par.poscfgpar.v1 = 0;
```

```
cmdpar.cmdgene_in_par.poscfgpar.vmax = 10;
```

```
cmdpar.cmdgene_in_par.poscfgpar.amax = 100;
```

```
cmdpar.cmdgene_in_par.poscfgpar.jmax = 1000;
```

```
cmdpar.cmdgene_in_par.poscfgpar.cyclecnt = 1;
```



```
cmdpar.cmdgene_in_par.poscfgpar.cycletype = 0;  
cmdpar.cmdgene_in_par.poscfgpar.dwelltime = 500;  
cmdpar.cmdgene_in_par.poscfgpar.isr_time_s = 1;  
  
cmdpar.cmdgene_in_par.velcfgpar.q0 = 0;  
cmdpar.cmdgene_in_par.velcfgpar.Tv = 5;  
cmdpar.cmdgene_in_par.velcfgpar.v0 = 0;  
cmdpar.cmdgene_in_par.velcfgpar.v1 = 0;  
cmdpar.cmdgene_in_par.velcfgpar.vmax = 10;  
cmdpar.cmdgene_in_par.velcfgpar.amax = 100;  
cmdpar.cmdgene_in_par.velcfgpar.jmax = 1000;  
cmdpar.cmdgene_in_par.velcfgpar.isr_time_s = 1;
```

Step4: 根据运行模式在中断内分别调用对应轨迹规划函数

速度模式下, 调用函数 `vel_cmd_gene`

位置模式下, 调用函数 `pos_cmd_gene`

Step5: 轨迹规划重置

`cmd_gene_disable`

调用该函数重置轨迹规划中间变量, 以便再次对用户指定的 PTP 运动的起点, 终点, 速度规划算法。

具体可参照 apps 下 lib 测试工程: `lib_demo`。

3.1.4 配置示例

为了方便大家更快捷的对此功能了解使用，这里记录了一些基础配置下速度曲线，位置曲线供大家参考，具体如下表所示。

模式	绝对位置 r	速度 r/s	加速度 r/s/s	加加速度 r/s/s/s	匀速时间 s	运动模式	运行次数	中断	示意图
速度	0	10	100	1000	1	/	/	10ms	图 5
	0	10	10	100	1	/	/		图 6
位置	20	10	100	1000	/	0:连续	/		图 7
	20	10	100	1000	/	1:单次	/		图 8
	20	10	100	1000	/	2:多次	2		图 9

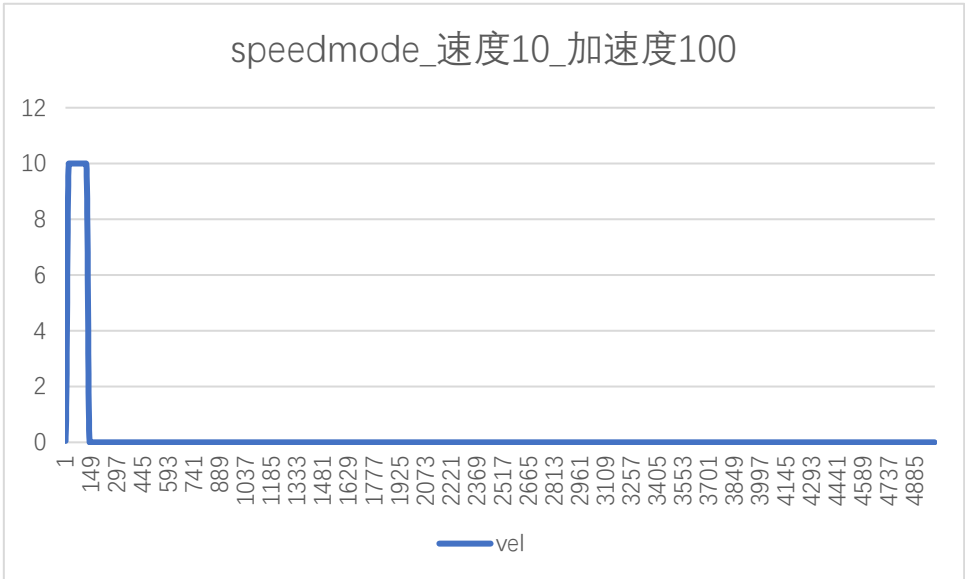


图 5

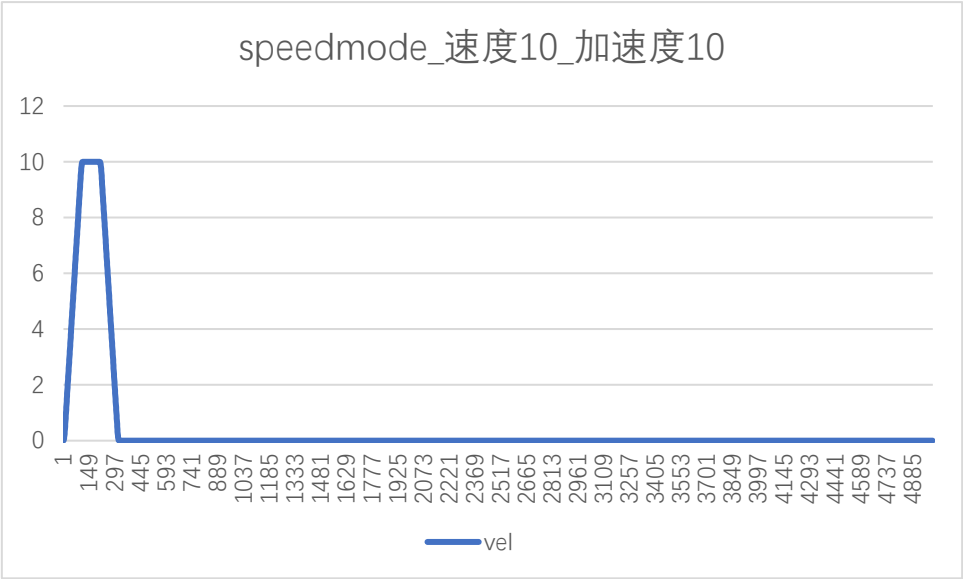


图 6

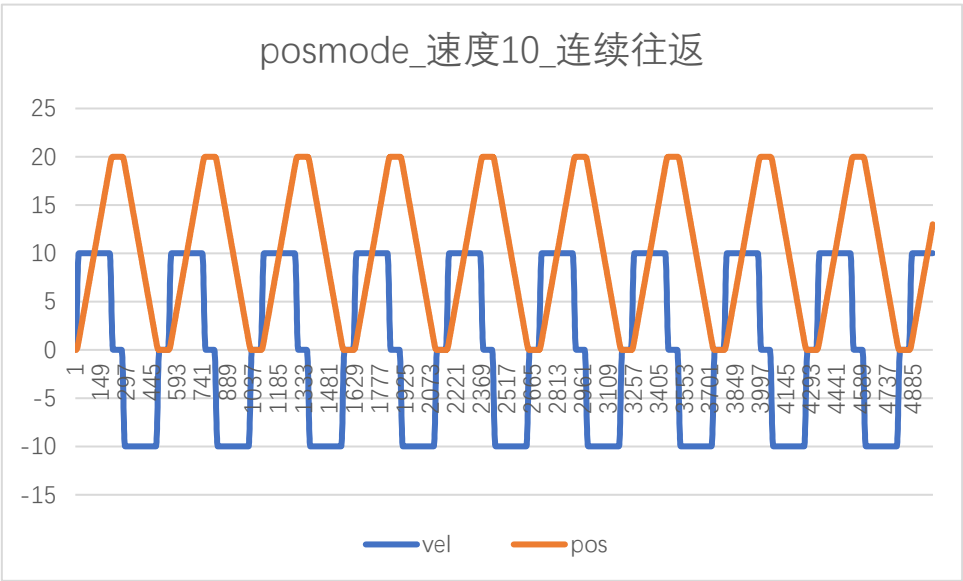


图 7

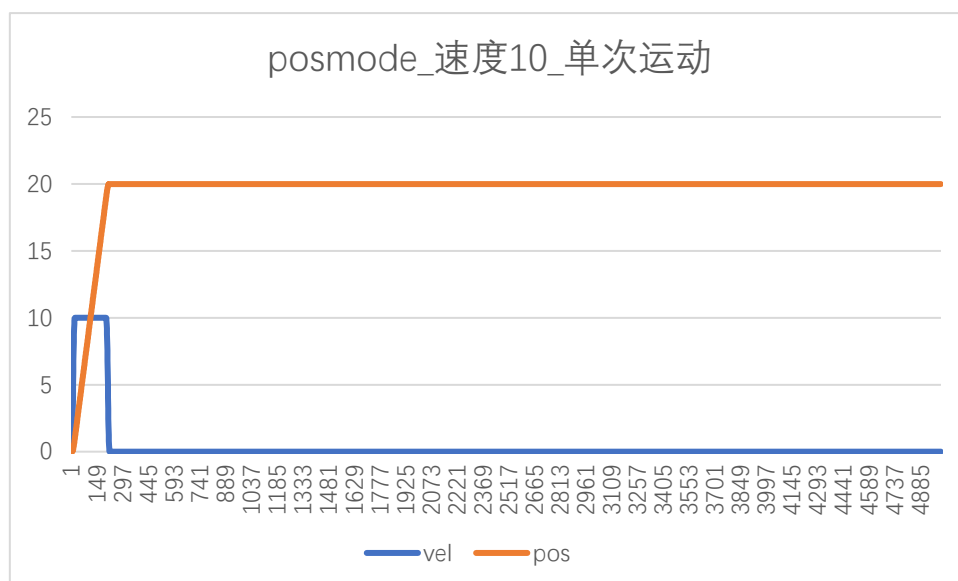


图 8

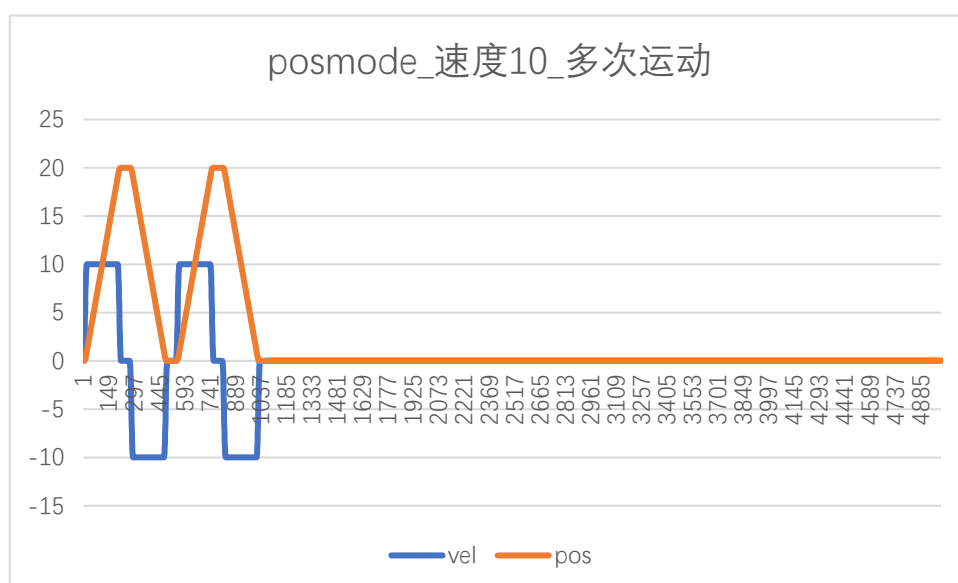


图 9

## 3.2 惯量辨识算法

在额定转速内，电机转矩  $T_m$  恒定。转矩与转动惯量有如下关系：

$$T_m = (J_m + J_l)p\omega_r$$

其中， $J_m$ ：电机转子的转动惯量， $J_l$ ：负载的转动惯量， $p$ ：微分算子， $\omega_r$ ：电机旋转角

速度。

由方程（1），可知当电机所带的负载转动惯量变化时，会对系统的伺服特性造成明显的影响：

- （1）负载转动惯量变大，角加速度变慢，系统响应变慢，易造成系统不稳定；
- （2）负载转动惯量变小，角加速度变快，系统响应变快，易使速度出现超调，甚至震荡现象。

因此，为达到伺服控制系统的良好动静态特性，需要辨识出转动惯量，对系统控制参数进行调整。

### 3.1.1 功能简介

惯量辨识包括几下步骤：

Step1: 电机以加速度 $a_1$ ，从速度 $w_0$ 匀加速至 $w_1$ ，匀加速时间为 $T_1$ ；

第一时刻采样得到的电磁转矩，负载转矩，电机机械角速度满足如下关系：

$$T_{e1} * T_s - T_{L1} * T_s = J(w_{t1} - w_0) \quad (2)$$

第二时刻：

$$T_{e2} * T_s - T_{L2} * T_s = J(w_{t2} - w_{t1}) \quad (3)$$

...

第 $n-1$ 时刻：

$$T_{e(n-1)} * T_s - T_{L(n-1)} * T_s = J(w_{n-1} - w_{n-2}) \quad (4)$$

第 $n$ 时刻：

$$T_{en} * T_s - T_{Ln} * T_s = J(w_n - w_{n-1}) \quad (5)$$

将 $n$ 个时刻的电磁转矩，负载转矩，电机机械角速度累加积分得到 $T_1$ 时间内的电机运动方程为：

$$\int_0^{T_1} T_e(t) * T_s dt - \int_0^{T_1} T_L(t) * T_s dt = J(w_{T_1} - w_0) \quad (6)$$

Step2:  $T_1$ - $T_2$ 时间内, 电机以 $n$ 倍的 $a_1$ 加速度, 从速度 $w_1$ 匀加速至 $w_2$ ;

以此类推, 此段时间内电机的运动方程可以表示为:

$$\int_{T_1}^{T_2} T_e(t) * T_s dt - \int_{T_1}^{T_2} T_L(t) * T_s dt = J(w_{T_2} - w_{T_1}) \quad (7)$$

Step3: 负载转矩恒定,  $T_2=T_2-T_1$ 的情况下, 可以得到:

$$\int_0^{T_1} T_L(t) * T_s dt = \int_{T_1}^{T_2} T_L(t) * T_s dt \quad (8)$$

Step4: 将第一阶段内运动方程与第二阶段内运动方程作差得到转动惯量的表达式为:

$$J = \frac{\int_{T_1}^{T_2} T_e(t) * T_s dt - \int_0^{T_1} T_e(t) * T_s dt}{w_{T_2} - 2w_{T_1}} \quad (9)$$

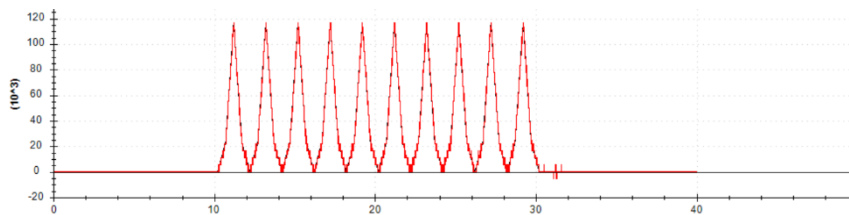
Step5: 实验结果

为了提高辨识精度, 将辨识周期分为加速阶段和减速阶段。

加速阶段又分为第一加速阶段和第二加速阶段, 第一阶段加速最大速度我们用第一阶段速度参数表示, 设置为  $350r/min$ 。第二加速阶段加速度与第一阶段加速度成  $k$  倍的比例关系, 这个关系我们用第二阶段加速度增益参数来表示, 这里我们将增益设置为  $4$ , 每段加速时间  $500ms$ , 由这几个参数我们就可以规划出辨识阶段的指令速度。

减速阶段指令速度与加速阶段指令速度对称。

本次实验将辨识周期设置为  $10$  次, 取加速阶段与减速阶段辨识值的平均值为最后的辨识结果。辨识参数设置, 辨识过程中速度指令、反馈速度以及辨识结果如下图所示。



实测出辨识结果为  $7.0299e-06 (kg \cdot m^2)$ , 与电机实际转动惯量的误差为  $6.2\%$ , 辨识

算法准确性基本得到验证。