

MOTEC 智能驱动器 PVT 控制模式操作手册

Version 2.1

MOTEC(中国) 营业体系
2017-06-01

版本说明:

2017年11月06日发行, Version 2.1;

版权信息:

本手册为MOTEC(中国)营业体系(以下简称“MOTEC(中国)”)版权所有。

MOTEC(中国)对本手册拥有版权, 未经书面授权, 不可将本文的全部或部分内容进行复制、翻印、收录、再加工或任何形式的转让。

本文的编著几经审校。但MOTEC(中国)不对其内容和推论中可能存在的错误担责。因用户原因使用不当而对产品或用户造成的直接或间接损失, MOTEC(中国)同样免责。使用本产品时务必遵照使用说明, 以免造成设备或人身伤害。

本文中的内容的表述力图精确、可靠, 但错误和疏忽之处在所难免, MOTEC(中国)保留随时修改和完善本文档的权利。

最新版本的使用说明书可在www.motec365.com 下载。

联系方式:

MOTEC(中国)营业体系

北京诺信泰伺服科技有限公司

地址: 北京市通州区环科中路17号11B (联东U谷西区)

电话: 010-56298855-666

传真: 010-65546721

邮编: 100027

网址: <http://www.motec365.com>

eMail: motecSupport@sina.com

目 录

| | | |
|--------|---|----|
| 1. | PVT 基础 | 4 |
| 2. | PVT 通讯指令和寄存器 | 5 |
| 2.1 | PVT 通讯指令 | 5 |
| 2.1.1 | 写数据到 PVT 队列 | 5 |
| 2.1.2 | 获取参数表参数 | 6 |
| 2.1.3 | 设置参数表参数 | 6 |
| 2.1.4 | 写 PVT 数据到驱动器 Flash | 7 |
| 2.2 | PVT 相关寄存器 | 8 |
| 3. | PVT 模式使用说明 | 9 |
| 3.1 | PVT 数据存储格式 | 10 |
| 3.1.1 | 位置和速度存储格式 | 10 |
| 3.1.2 | PVT 队列 | 11 |
| 3.1.3 | PVT 数据内置 Flash | 12 |
| 3.2 | PVT 通讯队列模式 | 13 |
| 3.3 | PVT 内部模拟模式 | 14 |
| 4. | 使用 PVT 模式注意事项 | 15 |
| 4.1 | 异常情况发生及处理 | 15 |
| 4.2 | PVT 模式下其他控制功能的使用 | 15 |
| 4.3 | PVT 周期时间的选择 | 15 |
| 4.4 | 速度设定值对 PVT 结果的影响 | 15 |
| 4.5 | 绝对运动和相对运动 | 15 |
| 4.6 | PVT/PT 说明 | 15 |
| 5. | CANopen 模式下使用 PVT 功能 | 16 |
| 5.1 | PVT 功能相关对象字典 | 16 |
| 5.2 | PVT 功能相关对象字典 | 16 |
| 5.2.1 | 对象 0x609C:PVTQUEUE_HEAD (PVT 队列头) | 16 |
| 5.2.2 | 对象 0x609D:PVTQUEUE_TAIL (PVT 队列尾) | 17 |
| 5.2.3 | 对象 0x609E:PVTQUEUE_STATUS (PVT 队列状态) | 17 |
| 5.2.4 | 对象 0x609F:PVTQUEUE_ALARM_SIZE (PVT 队列数据报警阈值) | 17 |
| 5.2.5 | 对象 0x60A0:PVTQUEUE_CLEAR (清空 PVT 队列) | 18 |
| 5.2.6 | 对象 0x60A1:PVT_TIME (PVT 周期) | 18 |
| 5.2.7 | 对象 0x60A2:PVT_RUNMODE (PVT 运行模式) | 19 |
| 5.2.8 | 对象 0x60A3:PVT_RUNSTOP (PVT 启动停止) | 19 |
| 5.2.9 | 对象 0x60A4: CANOPEN_PROFILE_PVTSIMAMPLITUDE (PVT 内部模拟模式幅值) | 19 |
| 5.2.10 | 对象 0x60A5:PVT_SIM_PERIOD(PVT 内部模拟模式周期) | 20 |
| 5.2.11 | 对象 0x60A6: PVT_RUNTIMES (PVT 内部 Flash 模式运行次数) | 20 |
| 5.2.12 | 对象 0x60A7: PVT_FILTER (PVT 滤波器) | 21 |
| 5.2.13 | 对象 0x60A8:CANOPEN_PVT_POSITION (PVT 位置值) | 21 |
| 5.2.14 | 对象 0x60A9:CANOPEN_PVT_VELOCITY (PVT 速度值) | 21 |
| 5.3 | CANopen 通讯下 PVT 功能的实现 | 22 |
| 5.3.1 | 配置步骤 | 22 |

6. 联系方式 22

1. PVT 基础

PVT(Position,Velocity,Time)控制模式即利用一个固定时间间隔的位置和速度序列重建较复杂曲线的运动轨迹的轨迹规划过程。PVT 控制模式使得用户能够利用 MOTEC 智能驱动器实现多轴之间插补和复杂的多轴协调工作,可以利用较低的通讯速率和较少的数据实现较复杂的速度和位置曲线的运动。图 1 显示了 PVT 控制模式的结构简图, MOTEC 智能驱动器提供了三种 PVT 操作模式, 分别是:

1). PVT 实时通讯模式: 通过通讯指令实时发送位置和速度信息至驱动器内部 PVT 队列实现 PVT 功能;

2). PVT 内部 Flash 模式(暂不开放使用): 通过将位置和速度信息存储到驱动器内部 Flash 中, 工作时可以通过加载这些数据实现一次或多次重复运动;

3). PVT 模拟运行模式, 即驱动器内部已经通过内置的余弦位置曲线和正弦速度曲线实现正弦位置曲线和余弦速度曲线的运动。在模拟运行模式下, 正弦曲线和余弦曲线的幅值和周期都可以通过修改参数表实现调整。

注意: 在本手册中介绍的关于内部 FLASH 模式下的所有内容均不开放使用;

在 PVT 实时通讯模式下, MOTEC 智能驱动器提供了 128 个字符长度的队列用于缓冲 PVT 位置和速度信息。在 PVT 内部 Flash 模式下, 驱动器内部预留了 500 个内部数据位用于存储位置和速度信息。而在 PVT 模拟运行模式, PVT 的位置和速度信息由内部程序提供。任何 PVT 模式都可以通过发送通讯命令来实现 PVT 运动的启动和停止。

PVT 模式下, 驱动器的运动可以分为 PVT 模式和 PT 模式(除非特殊说明, PVT 包括 PVT 模式和 PT 模式)。在 PVT 模式下, 每个周期的运算需要有位置和速度信息, 而 PT 模式只需要有位置信息参与运算。对于 PVT 模式来说, 速度信息的正确性非常重要, 如果用户不能给出正确的速度信息, 我们建议使用 PT 模式。

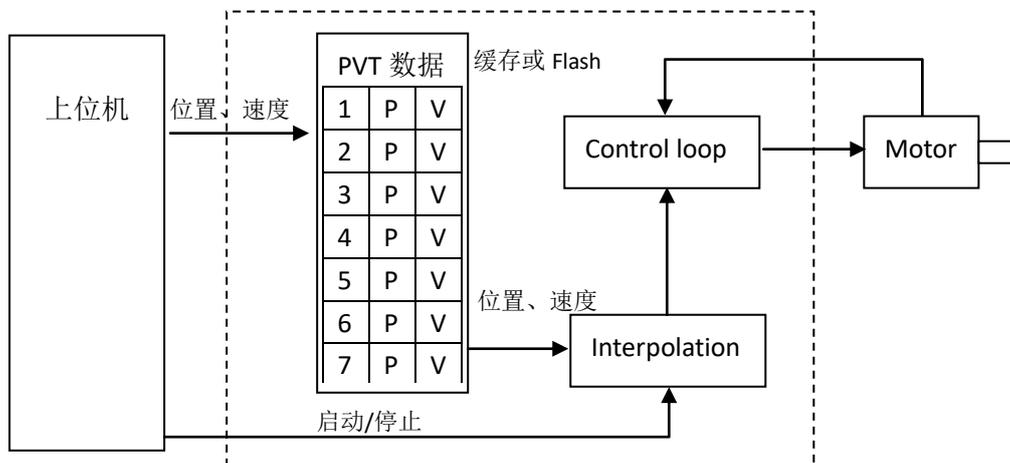


图 1. PVT 控制模式示意图

2. PVT 通讯指令和寄存器

2.1 PVT 通讯指令

关于 MOTEC 驱动器通讯协议以及指令格式的说明，请参考“MOTEC 智能伺服驱动器 MOTECIAN 协议使用手册”。

2.1.1 写数据到 PVT 队列

在 PVT 实时通讯过程中，用 MOTECIAN 协议共三种方式可以写有 PVT 队列数据：

(1) 0x7A 指令用于发送 16bit 位置信息和 16bit 速度信息；

(2) 0x7B 指令用于发送 32bit 位置信息和 16bit 速度信息（只适用于 CAN 通讯 MOTEC 协议）；

(3) 0x7C 指令发送 32bit 位置信息，0x7D 指令发送速度信息。如果是 PT 模式，则可以用 0x7C 指令发送 32bit 位置信息到队列。如果是 PVT 模式，需要在发送 0x7C 指令后紧跟着发送 0x7D 指令，前者发送 32 位位置信息，后者发送 16 位速度信息，必须成对使用。在 PVT 模式，驱动器收到 0x7C 指令后并不往队列中写数据，而是将位置数据存入缓存，等收到 0x7D 指令后一起将位置和速度信息写入队列。

以上单位均为：位置 pulse，速度 pulse/ms。

以下为指令详解：

| | |
|------|---|
| 功能 | 写 PVT 位置和速度数据到驱动器 PVT 队列 |
| 命令号 | 0x7A |
| 助记符 | PRIM_PVTQUEUE_PUTDATA_1 |
| 发送内容 | Ad 7A sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 7A rDATA1 rDATA2 XXXX |
| 备注 | sDATA1 是 PVT 队列位置信息(16 位有符号数，为了节省数据长度位置信息为位置增量信息或 16bit 绝对位置信息)，sDATA2 是 PVT 队列速度信息(16 位有符号数)，rDATA1 和 rDATA2 为返回数据，rDATA1 是当前队列头，rDATA2 是当前队列位尾； |

| | |
|------|--|
| 功能 | 写 PVT 位置和速度数据到驱动器 PVT 队列 |
| 命令号 | 0x7B（此命令只用于 CAN 通讯 MOTEC 协议） |
| 助记符 | PRIM_PVTQUEUE_PUTDATA_2 |
| 发送内容 | Ad 7B sDATA1 sDATA2 sDATA3 |
| 返回信息 | Ad 7B rDATA1 rDATA2 XXXX |
| 备注 | sDATA1，sDATA2 是 PVT 队列绝对位置信息(32 位有符号数)，sDATA3 是 PVT 队列速度信息(16 位有符号数)，rDATA1 和 rDATA2 为返回数据，rDATA1 是当前队列头，rDATA2 是当前队列位尾； |

| | |
|------|--|
| 功能 | 写 PVT 32bit 绝对位置数据到驱动器 PVT 队列 |
| 命令号 | 0x7C |
| 助记符 | PRIM_PVTQUEUE_PUTDATA_3_1 |
| 发送内容 | Ad 7C sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 7C rDATA1 rDATA2 XXXX |
| 备注 | sDATA1, sDATA2 是 PVT 队列绝对位置信息(32 位有符号数), rDATA1 和 rDATA2 为返回数据, rDATA1 是当前队列头, rDATA2 是当前队列位尾; 如果是 PT 模式, 则可以只用 0X7C 指令发送 32bit 位置信息到队列。如果是 PVT 模式, 需要在发送 0X7C 指令后紧跟着发送 0x7D 指令, 前者发送 32 位位置信息, 后者发送 16 位速度信息, 必须成对使用。在 PVT 模式, 驱动器收到 0x7C 指令后并不往队列中写数据, 而是将位置数据存入缓存, 等收到 0x7D 指令后一起将位置和速度信息写入队列。 |

| | |
|------|---|
| 功能 | 写 PVT 16bit 速度数据到驱动器 PVT 队列 |
| 命令号 | 0x7D |
| 助记符 | PRIM_PVTQUEUE_PUTDATA_3_2 |
| 发送内容 | Ad 7D sDATA1 0000 XXXX |
| 返回信息 | Ad 7D rDATA1 rDATA2 XXXX |
| 备注 | sDATA1 是 PVT 队列速度信息(16 位有符号数), rDATA1 和 rDATA2 为返回数据, rDATA1 是当前队列头, rDATA2 是当前队列位尾。在 PVT 模式下, 如果发送 32bit 位置信息, 0x7D 指令需要和 0x7C 指令一起使用。 |

2.1.2 获取参数表参数

| | |
|------|--|
| 功能 | 用于获取参数表参数 |
| 命令号 | 0x95 |
| 助记符 | PRIM_GetParam |
| 发送内容 | Ad 95 sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 95 rDATA1 rDATA2 XXXX |
| 备注 | 获取参数表参数, sDATA1 和 sDATA2 分别为参数表中的参数编号, rDATA1 为 sDATA1 中存储的数据, rDATA2 为 sDATA2 中存储的数据, 若 sDATA1, sDATA2 超过参数表长度, 返回信息为 FF; |

2.1.3 设置参数表参数

| | |
|------|---|
| 功能 | 用于设置参数表参数 |
| 命令号 | 0x96 |
| 助记符 | PRIM_SetParam |
| 发送内容 | Ad 96 sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 96 rDATA1 rDATA2 XXXX |
| 备注 | 设置参数表参数, sDATA1 参数编号, sDATA2 为要设置的参数的值, rDATA1=sDATA1, rDATA2=sDATA2; |

2.1.4 写 PVT 数据到驱动器 Flash

PVT 运行于内部 Flash 模式时, 首先需要准备 Flash 中的数据, 然后才能由程序调用 Flash 中的数据开始 PVT 运动。从上位机传送 PVT 位置和速度信息到驱动器 Flash 需要有三个步骤, 分别是 1) 初始化数据传输; 2) 发送数据总长度(最大 500)和 PVT 周期(单位 ms); 3) 发送 PVT 位置和速度信息到驱动器, 数据总长度为最大 500(位置和速度数据各 500); 4) 位置和速度信息发送完毕后发送 CRC 校验信息;

注意: 指令中 Ad 代表 8bit 长度的驱动器地址信息, xxxx 代表 16bit 校验信息;

步骤 1: 初始化

| | |
|------|-----------------------------------|
| 功能 | 写 PVT 位置或速度数据到驱动器 Flash 中, 第一步初始化 |
| 命令号 | 0x77 |
| 助记符 | PRIM_PVT_SAVEDATA_INI |
| 发送内容 | Ad 77 0000 0000 XXXX |
| 返回信息 | Ad 77 0000 0000 XXXX |
| 备注 | 写 PVT 数据到 Flash 的第一步, 初始化。 |

步骤 2: 发送数据长度和 PVT 周期

| | |
|------|---|
| 功能 | 写 PVT 位置和速度数据长度和 PVT 周期到驱动器 Flash 中, 第二步写数据长度和 PVT 周期; |
| 命令号 | 0x78 |
| 助记符 | PRIM_PVT_SAVEDATA |
| 发送内容 | Ad 78 sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 78 rDATA1 rDATA2 XXXX |
| 备注 | sDATA1 是要写入数据的地址 (16 位无符号数), sDATA2 是要写入的数据, rDATA1 和 rDATA2 为返回数据, 分别和 sDATA1, sDATA2 相等; 地址 0 为 16 位 PVT 数据长度, 地址 1 为 16 位 PVT 时间, 单位 ms。 |

步骤 3: 发送位置和速度信息

| | |
|------|--|
| 功能 | 写 PVT 位置或速度数据到驱动器 Flash 中, 第三步写数据 |
| 命令号 | 0x78 |
| 助记符 | PRIM_PVT_SAVEDATA |
| 发送内容 | Ad 78 sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 78 rDATA1 rDATA2 XXXX |
| 备注 | sDATA1 是要写入数据的地址 (16 位无符号数, 地址从 2 开始, 最大地址 501), sDATA2 是要写入的数据 (16 位有符号数), rDATA1 和 rDATA2 为返回数据, 分别和 sDATA1, sDATA2 相等; 为了节省数据长度, 所发送的位置数据为位置增量值。 |

步骤 4: 发送 CRC 校验信息

| | |
|-----|--|
| 功能 | 写 PVT 位置或速度数据到驱动器 Flash 中, 第四步, 写数据完毕并发送 CRC 校验位 |
| 命令号 | 0x79 |

| | |
|------|--|
| 助记符 | PRIM_PVT_SAVEDATA_END |
| 发送内容 | Ad 79 sDATA1 sDATA2 XXXX |
| 返回信息 | Ad 79 rDATA1 rDATA2 XXXX |
| 备注 | sDATA1 和 sDATA2 为 PVT 数据的 CRC 校验高 16 位和低 16 位数据, rDATA1 和 rDATA2 为返回数据. 如果校验正确返回数据 rDATA1 和 rDATA2 为 0x00, 如果校验失败则返回 0xff. |

为了检查 Flash 中内置数据的有效和一致性, 设置了一条指令用于检查数据一致有效性, 指令号为 0x76.

| | |
|------|---|
| 功能 | 检查 Flash 中 PVT 数据的一致有效性 |
| 命令号 | 0x76 |
| 助记符 | PRIM_PVT_CHECKCRC |
| 发送内容 | Ad 76 0000 0000 XXXX |
| 返回信息 | Ad 76 rDATA1 rDATA2 XXXX |
| 备注 | rDATA1 和 rDATA2 为返回数据. 如果校验正确返回数据 rDATA1 和 rDATA2 为 0x00, 如果校验失败则返回 0xff. |

需要注意在此模式下, PVT 模式运行于相对运动模式, 即每个位置数据都是每个 PVT 周期位置的增量信息。

具体 PVT 数据的传输请参考 3.1.3 中关于 PVT 在 Flash 中存储格式的说明。无论是 PVT 模式还是 PT 模式, 都需要按顺序存储位置和速度信息。但是对于 PT 模式来说速度信息在运算中将被忽略, 因此 PT 模式的 Flash 中速度信息的内容对系统的运行无关紧要, 但需要有速度信息占据相应的位置而不会导致系统数据格式异常。

2.2 PVT 相关寄存器

表 1 给出了 PVT 控制模式相关寄存器的说明, 各个寄存器的读写可以利用 2.1 节中说明的获取/设置参数表参数指令完成。

表 1. PVT 模式相关寄存器参数

| 参数编号 | 功能说明 | 数据格式 | 属性 | 备注 |
|--------|-----------|----------------|-----|--|
| Pr.87 | 轨迹规划模式 | 16bit unsigned | 读/写 | 0- T 曲线轨迹规划; 1- S 曲线轨迹规划; 2- PVT 控制模式; 3- PT 控制模式 |
| Pr.45 | 运动模式选择 | 16bit unsigned | 读/写 | 0- 相对运动; 以当前位置为起点的运动方式; 1- 绝对运动; 以 0 位置为起点的运动方式; 各种轨迹规划模式下均可以选择; |
| Pr.309 | 清除 PVT 队列 | 16bit unsigned | 读/写 | 清空 PVT 队列内容; 写 1 清空 PVT 队列, 清空后自动恢复为 0; |

| | | | | |
|--------|-----------------------|----------------|-----|--|
| Pr.310 | PVT 队列头 | 16bit unsigned | 只读 | PVT 队列队列头 |
| Pr.311 | PVT 队列尾 | 16bit unsigned | 只读 | PVT 队列队列尾 |
| Pr.312 | PVT 队列状态 | 16bit unsigned | 只读 | 0-PVT 队列满; 1-PVT 队列空; 2-PVT 队列缺少数据; 3-PVT 队列状态正常; |
| Pr.313 | PVT 周期 | 16bit unsigned | 读/写 | PVT 计算周期, 单位: ms, 最小 1ms; |
| Pr.314 | PVT 队列数据报警限值 | 16bit unsigned | 读/写 | PVT 队列中数据个数少于这个数时 PVT 队列状态将改变为缺少数据状态; |
| Pr.315 | PVT 运行模式 | 16bit unsigned | 读/写 | 0- 数据来自网络指令; 1- 数据来自驱动器内部 flash (暂不使用); 2- 驱动器内置正/余弦曲线重复运行; |
| Pr.316 | PVT 模式启动/停止 | 16bit unsigned | 读/写 | 0- PVT 停止; 1- PVT 运动; 2- PVT 暂停 |
| Pr.317 | PVT 模拟运行幅值高 16 位 | 16bit unsigned | 读/写 | 正/余弦曲线的幅值(高 16 位) |
| Pr.318 | PVT 模拟运行幅值低 16 位 | 16bit unsigned | 读/写 | 正/余弦曲线的幅值(低 16 位) |
| Pr.319 | PVT 模拟运行周期 | 16bit unsigned | 读/写 | 正/余弦曲线的周期(单位: ms) |
| Pr.320 | PVT 内部 FLASH 模式运行次数 | 16bit unsigned | 读/写 | PVT 运行模式是内部 Flash 运行模式时, PVT 运行的次数。 |
| Pr.321 | PVT 模式平滑滤波器 | 16bit unsigned | 读/写 | PVT 运行时滤波器系数设定, 用于平滑位置和速度值 |
| Pr.322 | PVT 内部 Flash 模式运行间隔时间 | 16bit unsigned | 读/写 | PVT 内部 Flash 模式多次运行时, 在一次运动完成到下一次启动之间的间隔时间, 该时间为 PVT 周期的整数倍; 即 (间隔时间=Pr.322*Pr.313 ms) |

3. PVT 模式使用说明

MOTEC 智能驱动器的 PVT 功能提供了 3 种操作方式, 分别是:

1). 实时通讯模式: 通过通讯指令实时发送位置和速度信息至驱动器内部 PVT 队列实现 PVT 功能;

2). 内部 Flash 模式: 通过将位置和速度信息存储到驱动器内部 Flash 中, 工作时可以通

过加载这些数据实现一次或多次重复运动；

3). PVT 模拟运行模式，即驱动器内部已经通过内置的余弦位置曲线和正弦速度曲线实现余弦位置曲线和正弦速度曲线的运动，在模拟运行模式下，正弦曲线和余弦曲线的幅值和周期都可以通过修改参数表实现调整。

以下的内容首先说明了 PVT 模式的数据存储方式，包括 PVT 队列和内置 Flash 的 PVT 数据。同时还介绍了 PVT 模式的 3 种不同运动方式，以及其操作步骤。

3.1 PVT 数据存储格式

PVT 控制模式中涉及数据存储的有两方面内容，分别是 PVT 队列数据的存储和 PVT 数据内置 Flash 时位置和速度信息的存储。下面分别介绍 PVT 队列和内置 Flash 中的 PVT 数据的格式。

3.1.1 位置和速度存储格式

PVT 队列数据的存储分为绝对位置数据和相对位置数据（实时通讯模式）；

PVT 内置 FLASH 数据存储位置信息为相对位置信息（内置 FLASH 模式）；

位置数据，单位为脉冲。速度数据单位为脉冲/毫秒。如表 3 所示。

表 3. PVT 位置数据格式

| 序号 | 原始位置数据 (绝对坐标) | 原始速度数据 | PVT 绝对位置数据 | PVT 相对位置数据 | PVT 速度数据 | 说明 |
|----------------------------------|------------------|--------|------------|------------|----------|--|
| 0 | 10000 (P0) | 0 | 10000 (P0) | 0 | 0 | PVT 模式运动之前的电机位置，PVT 位置数据为 0,速度数据为 0; |
| 1 | 11000 (P1) | V1 | 11000 (P1) | 1000 | V1 | 第 1 个 PVT 数据点 绝对运动: (P1, V1); 相对运动: (P1-P0, V1); |
| 2 | 13000 (P2) | V2 | 13000 (P2) | 2000 | V2 | 第 2 个 PVT 数据点 绝对运动: (P2, V2); 相对运动: (P2-P1, V2); |
| 3 | P3 | V3 | P3 | P3-P2 | V3 | 第 3 个 PVT 数据点 绝对运动: (P3, V3); 相对运动: (P3-P2, V3); |
| 4 | P4 | V4 | P4 | P4-P3 | V4 | 第 4 个 PVT 数据点 绝对运动: (P4, V4); 相对运动: (P4-P3, V4); |
| 5 | P5 | V5 | P5 | P5-P4 | V5 | 第 5 个 PVT 数据点 绝对运动: (P5, V5); 相对运动: (P5-P4, V5); |
| 注意：关于绝对运动和相对运动的说明，请参考 4.5 节中的说明。 | | | | | | |

3.1.2 PVT 队列

当 PVT 模式运行于实时通讯模式时，其 PVT 数据来自网络数据。而网络数据首先会插入 PVT 队列，当驱动器启动运行的时候会从队列中取得数据进行 PVT 运算。图 2 说明了 PVT 队列的存取数据和读取数据的操作过程。MOTEC 智能驱动器的 PVT 队列一共有 128 个数据位，分别用来存储 PVT 位置和速度信息。在发送 PVT 数据之前可以先查询 PVT 队列状态，确保队列中数据的正确性，或者通过控制参数表（Pr.309）清空 PVT 队列。

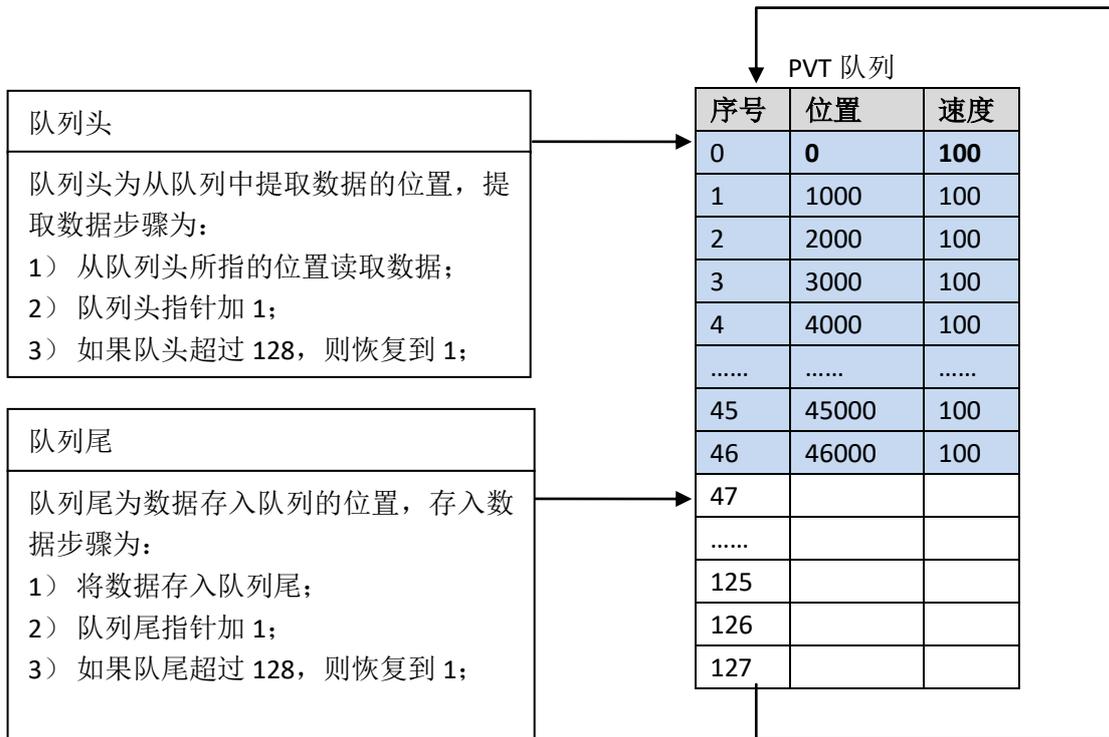


图 2. PVT 队列

初始化后队列的状态：

系统初始化后的队列头和队列尾指针都指向 0，队头和队尾指针相等，此时队列为空：

判断队列数据已满：

当队列尾指针加 1 等于队列头指针时，表明队列已经满。

判断队列数据已空：

当队列头指针加 1 等于队列尾指针时，表面队列已经空了。

计算队列中的数据个数：

- 如果队列尾指针大于队列头，则队列中有的数据个数为队列尾指针减去队列头指针；

- 如果队列尾指针小于队列头，则队列中有的数据个数为 128 加上队列尾指针减去队列头指针；
- 当队列中的数据小于所设定的 PVT 队列数据报警限值时，队列状态更新为缺少数据状态；

3.1.3 PVT 数据内置 Flash

PVT 内部 Flash 模式只能运行于相对运动模式，即位置数据为每个 PVT 周期的增量信息。

当 PVT 模式运行于内部模式时，其 PVT 数据来自驱动器内部 Flash。驱动器 Flash 中一共能存储最多 500 个 PVT 数据位(500 个位置，500 个速度；对于 PT 模式也需要按顺序存位置和速度信息，只是速度信息可以是任意数字，因为 PT 模式运算中不需要有速度信息)，其数据存储格式如下表所示。

其存储格式如下：

- 1) 每个位的数据长度为 16bit；
- 2) 地址 1 为 PVT 数据长度，最大 500；
- 3) 地址 2 为 PVT 控制周期，单位 ms；
- 4) 以下数据分别为 16 位位置增量信息和 16 位速度信息，速度单位为脉冲/ms；
- 5) 最后两个 16 位数据分别为 CRC 校验高 16 位和低 16 位；
- 6) CRC 校验从地址 1(即数据长度)开始，到最后一个速度数据位置，其 CRC 校验算法如下所示。

当 PVT 运行于内部数据模式时，驱动器首先会对 Flash 中存储的数据进行校验，如果数据有误，将会报警并停止运行。

表 4. Flash 中 PVT 数据存储格式

| 地址 | 内容 |
|-------|-------------|
| 1 | PVT 数据总长度 N |
| 2 | PVT 周期 |
| 3 | POS |
| 4 | VEL |
| 5 | POS |
| 6 | VEL |
| | |
| 497 | POS |
| 498 | VEL |
| 499 | POS |
| 500 | VEL |
| 501 | POS |

| | |
|-------|--------|
| 502 | VEL |
| | |
| N-5 | POS |
| N-4 | VEL |
| N-3 | POS |
| N-2 | VEL |
| N-1 | CRC HB |
| N | CRC LB |

3.2 PVT 通讯队列模式

当 PVT 运行于通讯队列模式时，需要进行如下的操作：

- 1) 将驱动器操作模式和控制模式设置为网络操作模式(Pr.36 = 0)和位置控制模式(Pr.37 = 2)；
- 2) 设置位置模式下运动方式（Pr.45=0 相对运动/Pr.45=1 绝对运动）；
- 3) 将 PVT 模式设置为数据来自网络命令模式(即 PVT 队列，Pr.315 = 0)；
- 4) 设置 PVT 控制周期(Pr.313, 单位 ms; 时间的选择需要根据通讯波特率和数据源确定)；
- 5) 将轨迹规划模式设置为 PVT 模式(Pr.87 = 2)；
- 6) 发送 PVT 数据到队列准备 PVT 队列中的数据(利用 0x7a/0x7b(CAN)/0x7c+0x7d 指令)；
- 6) 使能电机(Pr40 = 1)；
- 7) 启动 PVT(Pr.316 = 1)，只有在使能的情况下才能启动 PVT 模式；
- 8) 继续发送 PVT 数据到队列，并监控 PVT 队列状态；

注意：以上步骤 1、2、3、4 无需严格遵守所描述的步骤。如需停止 PVT 运行，只需发送指令将 Pr.316 置为 0.

PVT 内置数据模式

当 PVT 运行于内部数据模式时，需要进行如下的操作：

- 1) 在网络模式下准备好内置 Flash 中的 PVT 数据；
- 2) 将驱动器操作模式和控制模式设置为网络操作模式(Pr.36 = 0)和位置控制模式(Pr.37 = 2)；
- 3) 设置位置模式下运动方式（只能设置为相对运动模式）；
- 4) 将 PVT 模式设置为数据来自内部 Flash 的模式(即 PVT 队列，Pr.315 = 1)；
- 5) 设置 PVT 运行次数(寄存器 Pr.320)，如果设置运行 1 次，则运行完毕后自动停止 PVT

运行。如果设置运行多次，则 PVT 控制模块将循环访问 Flash 中的数据，直至运行次数到了所设定的次数为止；

- 6) 设置 PVT 内置 FLASH 模式运行间隔时间 (Pr.322 时间为 PVT 周期的倍数)，PT 模式时至少设置为 1,即一倍的 PVT 周期时间；
- 7) 将轨迹规划模式设置为 PVT 模式(Pr.87 = 2)；
- 8) 使能电机(Pr40 = 1)；
- 9) 启动 PVT(Pr.316 = 1)；
- 10) 在 PVT 内部 Flash 模式第一次运行之前，驱动器会自动检查数据的一致性，如果校验正确将会继续执行，如果校验不通过，PVT 将会停止执行并报警；
- 11) 监控 PVT 运行；

注意：

- (1) 以上步骤 1、2、3、4、5、6、7 无需严格遵守所描述的步骤。如需停止 PVT 运行，只需发送指令将 Pr.316 置为 0.
- (2) 当 PVT 滤波系数设置比较大时，建议设置内置 FLASH 模式运行间隔时间为 1~2 个周期时间

3.3 PVT 内部模拟模式

当 PVT 运行于内部仿真模式时，驱动器会控制电机根据所设定的幅值和周期循环运动。其位置为 cos 曲线，速度为 sin 曲线。需要进行如下的操作使得 PVT 模式正常运行：

- 1) 设置正弦曲线的幅值(Pr.317、Pr.318，单位脉冲)和周期(Pr.319，单位 ms)；
- 2) 将驱动器操作模式和控制模式设置为网络操作模式(Pr.36 = 0)和位置控制模式(Pr.37 = 2)；
- 3) 将 PVT 模式设置为模拟运行模式(即 PVT 队列，Pr.315 = 2)；
- 4) 将轨迹规划模式设置为 PVT 模式(Pr.87 = 2)；
- 5) 使能电机(Pr40 = 1)；
- 6) 启动 PVT(Pr.316 = 1)；
- 7) 监控 PVT 运行；

注意：以上步骤 1、2、3、4 无需严格遵守所描述的步骤。如需停止 PVT 运行，只需发送指令将 Pr.316 置为 0.

4. 使用 PVT 模式注意事项

4.1 异常情况发生及处理

- 电机 Disable 时会停止 PVT 模式运行(包括故障报警时)(Pr.316 自动修改为 0)。也就是电机在 Disable 的情况下不能修改 Pr.316 号参数;
- PVT 模式运动时, 需要先 Enable 电机, 然后启动运行 PVT;
- 可以使用暂停命令, 此时电机会立即停止运行(没有设定加减速, 速度直接变为 0), 应该考虑此时电机及机械受到的冲击。对于通讯队列模式, 暂停命令会暂停 PVT 的执行, 但在一次启动时, 系统会从暂停之前队列的位置继续执行, 而停止运动命令将会清除所有队列里面的数据信息和指针。对于内部 Flash 模式和模拟模式, 无论暂停和停止运动都会将 PVT 模式停止而不保留运行断点。
- PVT 队列模式如果队列中没有数据了, 会进入执行暂停模式;
- PVT 模式运行时, 可以发送停止运动指令和急停指令, 此时 PVT 停止运行, 直到重新启动 PVT;

4.2 PVT 模式下其他控制功能的使用

PVT 模式是网络操作模式及位置控制模式下的一种轨迹规划方式, 它与 S 曲线轨迹规划, T 曲线轨迹规划相并列。PVT 模式可以与 T 曲线和 S 曲线轨迹规划切换使用, 这种切换只需通过切换 Pr.87 参数进行。

4.3 PVT 周期时间的选择

原则上 PVT 周期时间越短越好(最短时间为 1ms), 但具体 PVT 周期时间的选择还要考虑驱动器的通讯速率和数据源的数据, 从而选择一个合理的 PVT 周期时间。

4.4 速度设定值对 PVT 结果的影响

PVT 控制方式是利用一定时间间隔的位置和速度序列重建位置、速度和加速度信息。如果速度信息不准确, 依据不准确的速度信息重建的加速度值将是不连续的, 这就会影响到 PVT 运动过程中位置和速度的连续性和运动的平稳性。

因此在速度信息不准确的情况下, 可以尝试使用 PT 模式。

4.5 绝对运动和相对运动

PVT 控制模式下, 通讯队列模式可以运行于绝对运动或相对运动模式, 而内部 Flash 和模拟运行模式下只能运行于相对运动模式。

4.6 PVT/PT 说明

上面已经说明 PVT 模式依赖于速度信息的完整和正确性, PT 模式无需速度信息即可完成系统的轨迹规划。对于通讯队列模式, 上位机无需传递速度信息。而对于内部 Flash 模式, 其数据的存储格式和 PVT 模式是一致的, 只是速度信息只需存储任意的数据即可。

5. CANopen 模式下使用 PVT 功能

5.1 PVT 功能相关对象字典

| 索引 | 名称 | 描述 |
|--------|---------------------------------|---------------|
| 0x609C | PVTQUEUE_HEAD | PVT 队列头； |
| 0x609D | PVTQUEUE_TAIL | PVT 队列尾； |
| 0x609E | PVTQUEUE_STATUS | PVT 队列状态； |
| 0x609F | PVTQUEUE_ALARM_SIZE | PVT 队列数据报警阈值； |
| 0x60A0 | PVTQUEUE_CLEAR | 清空 PVT 队列数据； |
| 0x60A1 | PVT_TIME | PVT 计算周期； |
| 0x60A2 | PVT_RUNMODE | PVT 运行模式； |
| 0x60A3 | PVT_RUNSTOP | PVT 启动 停止； |
| 0x60A4 | CANOPEN_PROFILE_PVTSIMAMPLITUDE | 内部模式模式幅值； |
| 0x60A5 | PVT_SIM_PERIOD | 内部模拟模式周期； |
| 0x60A6 | PVT_RUNTIMES | PVT 运行次数； |
| 0x60A7 | PVT_FILTER | PVT 滤波器； |
| 0x60A8 | CANOPEN_PVT_POSITION | PVT 位置数据 |
| 0x60A9 | CANOPEN_PVT_VELOCITY | PVT 速度数据 |

5.2 PVT 功能相关对象字典

5.2.1 对象 0x609C:PVTQUEUE_HEAD (PVT 队列头)

0x609C 对象定义了 PVT 队列的当前队列头。

对象描述：

| | |
|---------------|----------|
| Index | 0x609C |
| Name | PVT 队列头 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RO |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~65535 |
| Default Value | 0 |

5.2.2 对象 0x609D:PVTQUEUE_TAIL (PVT 队列尾)

0x609D 对象定义了 PVT 队列的当前队列尾。

对象描述:

| | |
|---------------|----------|
| Index | 0x609D |
| Name | PVT 队列尾 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RO |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~65535 |
| Default Value | 0 |

5.2.3 对象 0x609E:PVTQUEUE_STATUS (PVT 队列状态)

0x609E 对象定义了 PVT 队列的当前队列状态。

对象描述:

| | |
|---------------|----------|
| Index | 0x609E |
| Name | PVT 队列状态 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RO |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~65535 |
| Default Value | 0 |

5.2.4 对象 0x609F:PVTQUEUE_ALARM_SIZE (PVT 队列数据报警阈值)

0x609F 对象定义了 PVT 队列的数据报警阈值。

对象描述:

| | |
|-------------|--------------|
| Index | 0x609F |
| Name | PVT 队列数据报警阈值 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |

| | |
|---------------|---------|
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~65535 |
| Default Value | 0 |

5.2.5 对象 0x60A0:PVTQUEUE_CLEAR (清空 PVT 队列)

0x60A0 对象定义了清空 PVT 队列内容功能。

对象描述:

| | |
|---------------|-----------|
| Index | 0x60A0 |
| Name | 清空 PVT 队列 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~1 |
| Default Value | 0 |

5.2.6 对象 0x60A1:PVT_TIME (PVT 周期)

0x60A1 对象定义了 PVT 计算周期, 单位: ms, 最小 1ms。

对象描述:

| | |
|---------------|----------|
| Index | 0x60A1 |
| Name | PVT 周期 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~65535 |
| Default Value | 0 |

5.2.7 对象 0x60A2:PVT_RUNMODE (PVT 运行模式)

0x60A2 对象定义了 PVT 运行模式；0-实时数据模式；1-内部 Flash 程序（保留）；2-内部模拟模式。

对象描述：

| | |
|---------------|----------|
| Index | 0x60A2 |
| Name | PVT 运行模式 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~2 |
| Default Value | 0 |

5.2.8 对象 0x60A3:PVT_RUNSTOP (PVT 启动停止)

0x60A3 对象定义了 PVT 启动停止命令；1-PVT 启动；0-PVT 停止；2-PVT 暂停。

对象描述：

| | |
|---------------|----------|
| Index | 0x60A3 |
| Name | PVT 启动停止 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~2 |
| Default Value | 0 |

5.2.9 对象 0x60A4: CANOPEN_PROFILE_PVTSIMAPLLITUDE (PVT 内部模拟模式幅值)

0x60A4 对象定义了 PVT 内部模拟模式幅值；

对象描述：

| | |
|-------------|--------------|
| Index | 0x60A4 |
| Name | PVT 内部模拟模式幅值 |
| Object Code | VAR |

| | |
|---------------|----------|
| Data Type | Unit32 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | Unit32 |
| Default Value | 0 |

5.2.10 对象 0x60A5:PVT_SIM_PERIOD(PVT 内部模拟模式周期)

0x60A5 对象定义了 PVT 内部模拟模式周期，单位 ms；

对象描述：

| | |
|---------------|--------------|
| Index | 0x60A5 |
| Name | PVT 内部模拟模式周期 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | Unit16 |
| Default Value | 0 |

5.2.11 对象 0x60A6: PVT_RUNTIMES(PVT 内部 Flash 模式运行次数)

0x60A6 对象定义了 PVT 内部 Flash 模式运行次数；

对象描述：

| | |
|---------------|---------------------|
| Index | 0x60A6 |
| Name | PVT 内部 Flash 模式运行次数 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | Unit16 |
| Default Value | 0 |

5.2.12 对象 0x60A7: PVT_FILTER (PVT 滤波器)

0x60A7 对象定义了 PVT 滤波器;

对象描述:

| | |
|---------------|----------|
| Index | 0x60A7 |
| Name | PVT 滤波器 |
| Object Code | VAR |
| Data Type | Unit16 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | 0~999 |
| Default Value | 0 |

5.2.13 对象 0x60A8: CANOPEN_PVT_POSITION (PVT 位置值)

0x60A8 对象定义了写 PVT 队列时写的 32bit 位置信息值 (只用来配置 RPDO3);

(注: 对象 0x60A8 和 0x60A9, 只能用 RPDO3 写入 PVT 队列中)

对象描述:

| | |
|---------------|----------|
| Index | 0x60A8 |
| Name | PVT 位置值 |
| Object Code | VAR |
| Data Type | Int32 |
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | Int32 |
| Default Value | 0 |

5.2.14 对象 0x60A9: CANOPEN_PVT_VELOCITY (PVT 速度值)

0x60A9 对象定义了写入 PVT 队列的 16bit 速度信息值 (只用来配置 RPDO3);

对象描述:

| | |
|-------------|---------|
| Index | 0x60A9 |
| Name | PVT 速度值 |
| Object Code | VAR |
| Data Type | Int32 |

| | |
|---------------|----------|
| Category | Optional |
| Access | RW |
| PDO Mapping | No |
| Units | NA |
| Value Range | Int32 |
| Default Value | 0 |

5.3 CANopen 通讯下 PVT 功能的实现

5.3.1 配置步骤

通讯协议：CANopen 协议；

操作模式：网络位置模式；

控制模式：位置模式；

PVT 运行模式：PVT 模式或 PT 模式；

在以上条件基础之上，举例说明实现 PVT 功能（以驱动器地址是 1 为例）；

（1）配置 RPDO3 映射：

RPDO3 通讯参数：

601 2f 02 14 02 fe 00 00 00 ;/*配置 RPDO3 的传输方式为 254，异步，事件触发；

RPDO3 映射参数：

601 23 02 16 01 20 00 a8 60 ;/*配置 RPDO3 的第一映射为 PVT 位置信息；

601 23 02 16 02 10 00 a9 60 ;/*配置 RPDO3 的第二映射为 PVT 速度信息；

601 2f 02 16 00 02 00 00 00 ;/*配置 RPDO3 的总映射个数；

（2）写 PVT 队列：

CANopen 模式下写入 PVT 队列只能用 RPDO3 写入；

401 Sdata1 Sdata2 Sdata3 ;/*6 位数据，其中 Sdata1 和 Sdata2 为 32bit 位置信息；Sdata3 为 16bit 速度信息；

（3）使能电机；

使能电机方法任意；

（4）PVT 启动：

601 2b a3 60 00 01 00 00 00 ;/*PVT 启动，使用 PDO 方法；

6. 联系方式

MOTEC(中国) 营业体系

北京诺信泰伺服科技有限公司

Website: <http://www.motec365.com.cn>

地址：北京市通州区环科中路 17 号 11B

服务热线：010-56298855-666

Email: motecSupport@sina.com