P3 JPL Manual

Change History

Revision	Date	Description	Originator
AA	2015-11-13	Original Issue	B. Zenowich
AB	2016-01-26	Added some notes for clarification, updated the parameter tables with scaled position info and new homing variables, edited the homing procedure to reflect new method.	B. Zenowich
AC	2016-03-09	Added information about temperature based parameters and safe position.	C. Woodall
AD	2016-06-28	Added information on over current and over temperature faults.	C. Woodall
AE	2016-07-27	Corrected max temperature from 76 °C to 75 °C	C. Woodall
AF	2016-08-31	Added changes to the document to fit JPL control loop and release.	C. Woodall
AG	2017-01-18	Exposed raw motor gain parameters	I. Gutekunst
AH	2017-03-03	Replaced fixed point PI controller with floating point PID controller. Added functionality to plot step responses. Removed homing section.	I. Gutekunst, C. Woodall, B. Zenowich

Introduction

P3 is the third-generation of miniature motor controllers (Pucks) from Barrett Technology. In cooperation with NASA JPL, we have integrated P3 directly into the Maxon EC-max 30 (272765) and Maxon EC-max 22 (283860) brushless DC motors. P3 transforms the Maxon motors into bus-topology actuators requiring just 5 wires (Motor+, Logic+, GND, CAN_H, CAN_L) and no separate controllers.

How it works: Cascaded Control Loops

P3 provides an industry-standard motion control architecture. The trajectory generator takes a target position and outputs a series of position commands. The position loop uses angular feedback from the integrated encoder to implement a Proportional-Integral-Derivative (PID) controller which outputs a velocity command. The velocity loop implements a second PID controller which outputs a motor effort. The motor effort is transformed by a space vector function to yield the proper PWM commands to the motor. All of the control inputs are user-adjustable to support application-specific motion control characteristics.

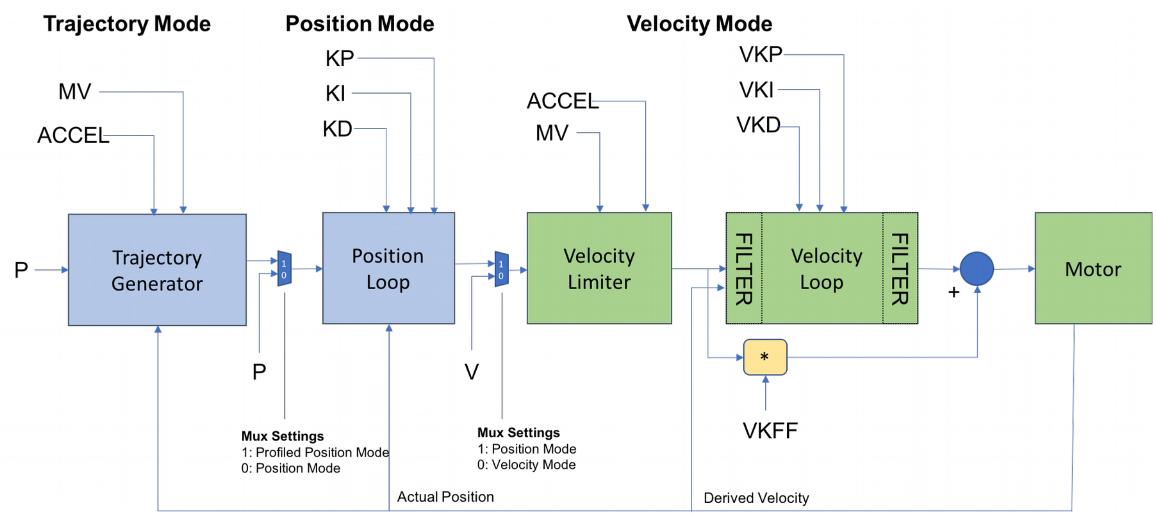

Figure 1: Control Loop Diagram

In Figure 1, the cascaded control loop is represented. Each control loop tuning parameter is represented by its BarrettCAN name. Please see the *Motor Control Constants, Example Values for a supported motor(BarrettCAN) Table* on page 5 and Page 7 for descriptions of these parameters.

NOTE: *We are actively developing the P3 firmware at this time.*

How to communicate with it

- We support the Barrett CAN Protocol for full configuration and all modes of operation: http://web.barrett.com/support/Puck_Documentation/CAN_Message_Format.pdf
- We support a minimal but growing set of CANopen features at this time: http://web.barrett.com/support/Puck_Documentation/P3_CANopen_revAD.doc

How to control it: Control Modes

There are four distinct control modes that you can choose by setting the **MODE** property of P3:

1) **MODE** = 0: Idle. The controller ties the motor phase leads together for a resistive-braking effect. No current can flow through the motor.
2) **MODE** = 3: Position control. Write to the **P** property to set the commanded position.
3) **MODE** = 4: Velocity control. Write to the **V** property to set the commanded velocity. The motion will obey the acceleration (ACCEL) and max velocity (MV) properties.
4) **MODE** = 5: Trapezoidal velocity profile control. Write to the P property to set the target position. The controller will generate a trapezoidal velocity profile using **ACCEL** and **MV** and begin to output a series of commanded positions to the position controller.

About P3 Angular Velocity

You are free to set Max Velocity (MV) to any value you want to achieve your desired motion profile, but take care not to exceed the datasheet's specifications. For example, the max velocity of the Maxon EC-Max 22 is 18000 rpm, but the allowable continuous velocity depends on the torque applied to the motor. Please reference the motor datasheets to set these values properly. The controller will obey MV in Position and Velocity modes.

How to update it: In-System Firmware Updates

Methods for updating firmware are included in Barrett's wxPuck and wxPuckTester applications.

Motor Control Variables (Barrett CAN Protocol)

Property	Key	Frac	Units	Notes
T	42	0	TBD	Open-loop motor effort (read-only)
VERS	0	0	-	Firmware version
SN	2	0	-	Puck serial number
ID	3	0	-	CAN ID
ERROR	4	0	-	Fault status
MODE	8	0	-	Control mode: 0=idle, 3=position, 4=vel, 5=vel profile
V	44	8	rad/s	Gearhead output velocity (write=cmd, read=actual)
P	48	0	Encoder cts	Motor position (write=cmd,

				read=actual). Write in MODE=0 to initialize the position value.
MOFST	61	0	Encoder cts	Raw encoder reading at the start of an electrical cycle
IOFST	62	0	A/D cts	Raw A/D reading of current when MODE = 0
MECH	66	0	Encoder cts	Raw encoder reading (0-4095)
CTS	68	0	Encoder cts	Motor encoder resolution (4096, typ)
IKCOR	93	0	TBD	Motor effort to apply during MOFST calibration
TENST	83	4	Gear Head Ratio * 16	Gear Head Ratio in Q12.4 units. For example, a gearhead ratio of *300* has *TENST = 300*2^4= 4800*

NOTE: All properties listed here are 16-bits wide, except P which is 32-bits wide.

NOTE: Key = Property key value used in the Barrett CAN Protocol message frames, as described in the CAN_Message_Format.pdf document linked above.

NOTE: Frac = Number of bits of fraction defined for each property. For example, property V (velocity) is a 16-bit integer, but the lower 8 bits are interpreted as a fractional value. So an integer value of 64 for V would be interpreted as 64 / 2^8 = 0.25 rad/s.

Motor Control Constants, Example Values for a supported motor(BarrettCAN)

Property	Key	Frac	Units	Value	Conv	Notes
R	109	12	Ω (/φ)	0.82	3359	Per-phase or terminal res/2
L	110	12	mH (/φ)	0.27	1106	Per-phase or terminal ind/2
J	111	15	$mNms^2$	0.7301	23924	Rotor + gearhead inertia at output
KT	112	12	Nm/A_{RMS} (/φ)	1.1/3	1502	Per-phase torque constant
ICONT	113	0	TBD	0.76 A_{RMS}	800	Continuous allowed current
IPEAK	114	0	TBD	1.7 A_{RMS}	1790	Peak allowed current
IPKMS	115	0	ms	1000	1000	Time (in ms) that the current is allowed to exceed ICONT
POLES	90	0	-	8	8	Rotor magnet count or pole pairs/2
MV	45	8	rad/s	100 RPM	2681	Max velocity at gearhead output (RPM*2pi/60*256)
VBUS	21	0	Volts	24	24	Motor bus voltage
MT	43	0	TBD	3.5 A_{RMS}	3685	A_{RMS} * √2 * 8192/11 The Maximum Torque
ACCEL	82	0	rad/s/ (150 us)	10	10	Used to limit acceleration and deceleration in the velocity and trapezoidal control modes.

NOTE: All properties listed here are 16-bits wide. The current-related properties never exceed 14-bits in practice.

NOTE: Value is an example value for the property. Conv is that example value converted to the fractional fixed-point value used by the controller.

NOTE: For direct PID control see page 7.

Additional Properties

Property	Key	Frac	Units	Value	Conv	Notes
THERM	20	8	°C	40 °C	10240	Thermistor temperature in °C (16-bit number in Q8.8 format. Temperature * 2^8)
PTEMP	10	8	°C	70 °C	17920	Peak temperature in °C (16-bit number in Q8.8 format. Temperature * 2^8)
OTEMP	11	0	Bool	1	Fault	Temperature Fault and Fault Reset property.

Errors and Faults

Under the current system there are 2 faults:

1) Over-Temperature Faults
2) Over-Current Faults

These two faults are handled in different ways and have different parameters associated with them.

Over-Current Faults

The overcurrent faults are run on the commanded currents with a low pass filter, with a cutoff of 20Hz. The commanded current is allowed to run from –IPEAK to IPEAK, currents cannot be commanded which are above IPEAK. If the absolute value of the commanded current, I, is greater than ICONT (the continuous current) a counter starts to increment. If I drops below ICONT the timer restarts. Once the counter has been counting for IPKMS ms the currents are thresholded now to ICONT. If the current I drops below ICONT the limit is released and IPKMS must be exceeded again in order to trigger the overcurrent.

The CANOpen error register flag for current is only held while I is being thresholded to ICONT and is not a permanent state.

Over-Temperature Faults

The over temperature fault is controlled by the following parameters: PTEMP, OTEMP and THERM. PTEMP is the peak temperature, OTEMP is a register which stores the error state, and THERM is the current filtered temperature. The THERM temperatures are filtered with a 20Hz cutoff and are mapped from the raw values to temperature using a multiple region linear interpolation which has a max error of 1 °C in the main operational zones and can peak to 2 °C at temperatures below 20 °C.

The over-temperature fault is triggered when THERM is greater than PTEMP for more than 5 seconds (hard-coded). PTEMP is not allowed to be greater than 75 °C. When the over-temperature fault is generated OTEMP will read back 1. If you write any value to OTEMP it will clear the error. When the fault occurs motor currents are forced to 0.

The CANOpen error register flag is set the whole time the **ERROR** flag is set until it is cleared, which can only be cleared in CANOpen through a reset.

Direct Control of PID Control Parameters

The raw values used by the velocity and position PID controllers have been exposed in the latest firmware revision.

Due to implementation details, the parameters are exposed as a Q8.8 gain with a multiplication factor (Mult).

For example, Actual KP = property[KP] * 10

Raw Controller Inputs

Property	Key	Frac	Mult	Notes
KP	79	8	10	Position KP
KI	80	8	1	Position KI
KD	81	8	1	Position KD
VKP	119	8	10	Velocity KP
VKI	120	8	1	Velocity KI
VKD	124	8	1	Velocity KD
VKFF	126	8	1	Velocity feed forward gain

Plotting Step Responses

The wxPuck Tester Utility has facilities for plotting step responses.

1) Set desired gains
2) Click “Do V Step Response”, or “Do P Step Response”.
3) After the motor has finished moving (< 2 seconds), click **Plot** or **Save CSV** It will take about 30 seconds to download the trace data. During this time, the GUI and motor will not respond to any input.
4) Save the plots from the plot window.

Notes: The position step response will move the motor by 4096 encoder ticks. The velocity step response will set a velocity of 50 rad/s at the output shaft. The motor will attempt to reach TENST * 50 rad/s