

Technical Brief

Brief Number: TB-016-12-06
Reference: MAX3000, microMAX
Application: Position Compensator
Revision: 1.0
Date: December 13 2006

Introduction

The digital position compensation algorithm implemented on the MAX family of motion controllers goes beyond traditional PID compensation. The on board trajectory generation software calculates not only position profiles, but also profiles for velocity, acceleration, and jerk. These additional profiles can be used as state variables in the position compensator, each with its own compensation loop.

The compensator also implements a velocity observer to help smooth velocity feedback at low speed when little encoder information is available. The velocity observer will calculate an improved estimate of velocity to be used as feedback to the velocity state variable compensation loop.

Three configurable third order IIR filters are also available within the state variable compensators. The filters are optionally located within the position, velocity and acceleration state compensators or can be located cascading at the compensators output.

Features for current and force offset, various feed forward terms and anti-hunt algorithms are also implemented in the position compensator.

The overall equation for the compensator is given by:

$$I_{setpoint} = (I_J + I_A + I_V + I_P + I_O + I_{DT}) * K_{BaseFactor} * K_{AntiHunt}$$

Where:

$$K_{BaseFactor} = \text{Power Base Current Rating (in Amps)} / 16383$$

I_J is the component of the output attributable to jerk compensation

I_A is the component attributed to acceleration compensation

I_V is the component attributed to velocity compensation

I_P is the component attributed to position compensation (sum of PI and D terms)

I_O is the component attributed to fixed current and force offsets

I_{DT} is the component attributed to detent torque compensation

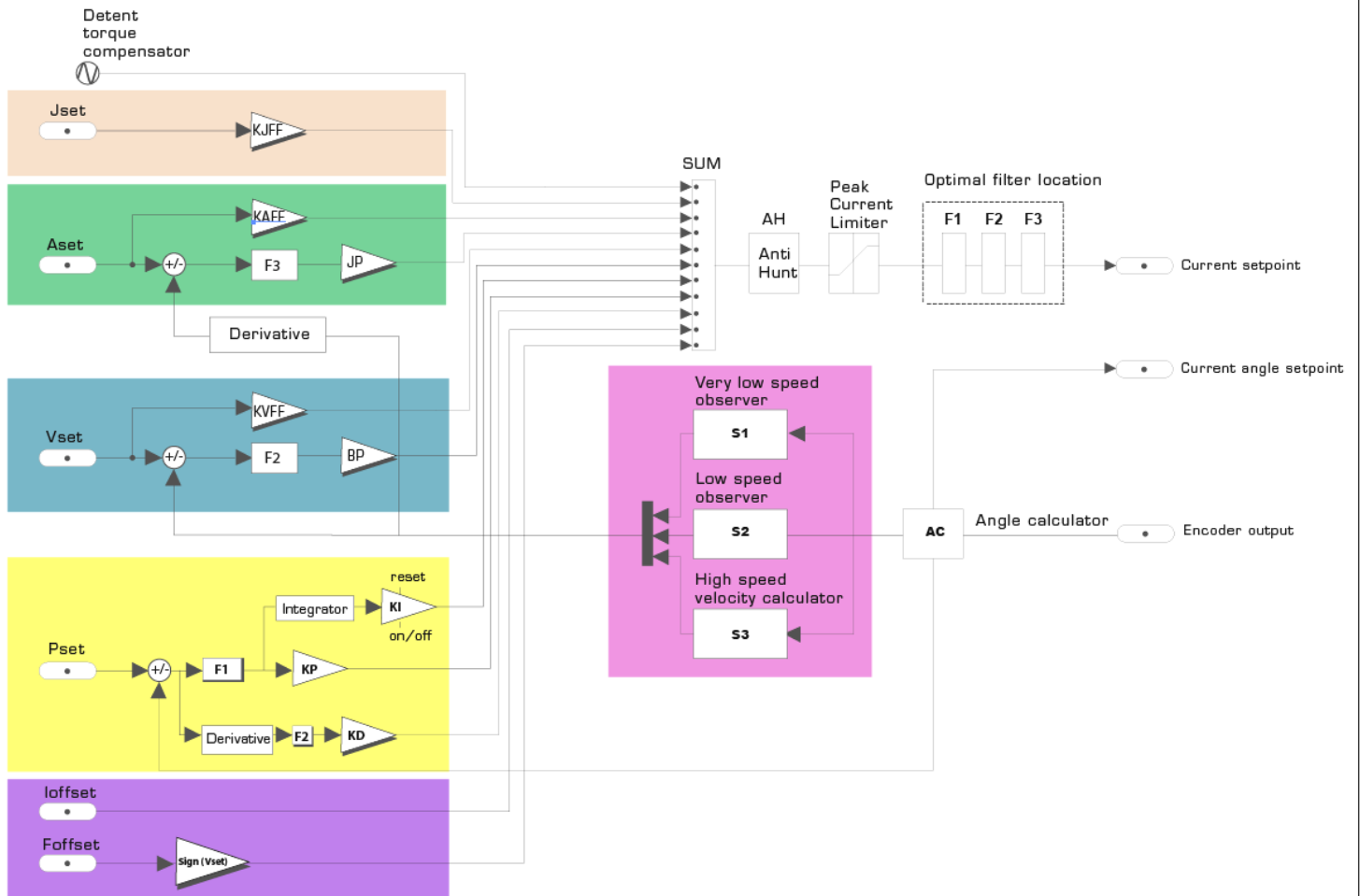
$K_{AntiHunt}$ is a scalar applied to the entire compensation output under certain conditions

Each of these terms is explained in more detail throughout this note.










Note: The output of the compensation algorithm is a current set-point.

This current set-point is then subjected to a current limiter to prevent damaging current levels from being injected to the motor.

Position Compensator Diagram.



State Variable Compensator Diagram

Key	Loop description	Term
	Jerk Feed Forward	I_J
	Acceleration Compensator	I_A
	Velocity Compensator	I_V
	PID Compensator	I_P
	Current and Force Offset	I_O
	Velocity Conditioner/Observer	
	Detent Torque Compensator	I_{DT}
	Anti-hunt logic	$K_{AntiHunt}$
	Third order IIR filters	

PID Compensator

The foundation of the overall state variable compensator is a PID control loop. In this portion of the compensator, position error is measured and a proportional, integral and derivative gain is applied to produce a current set-point. The input to this compensator is a position set-point and the feedback is direct from the encoder (called Actual Position). The proportional and integral terms in this section of the compensator are always used, but the derivative term is optional. Direct velocity compensation can be done instead using the velocity state variable compensator. Using the state variable compensator for velocity allows the system to take advantage of some features that are not available using derivative control.

The filters F1 and F2 can be optionally located in this section of the compensator. Filter F2 will be located in this portion of the compensator if the B_p term (located in the velocity compensator) is equal to zero. Both F1 and F2 can be moved to cascade at the output of the compensator output.

This area of the compensator is a classic PID controller with the following enhancements:

1. Integrate on hold only – This feature is designed to turn the integrator off when the velocity state variable (velocity set-point) is non-zero.
2. Integral loading control – This feature will keep the integral term from clearing to zero when the position loop is turned off.
- 3.

The equation for this portion of the position compensator is given by:

$$I_P = K_p * P_{error} + K_i * Sum_{(P_{error})} + K_d * Delta_{(P_{error})}$$

Current and Force Offset

These two simple terms add fixed current offsets to the rest of the position compensator output.

The current offset term adds a fixed amount of current with a constant direction (or sign). This current offset can be used to counter gravitational or other fixed direction forces.

When the position compensator is enabled, a fixed amount of current will be injected into the motor producing a force countering the fixed direction force. This current is added to the position compensator whenever it is enabled

The force-offset term adds a fixed amount of current with variable direction. The direction is determined by the sign of the velocity state variable (velocity set-point).

When the velocity set-point is zero, the force offset is not used.

$$I_O = I_{offset} + sign(velocity) * F_{offset}$$

Jerk Feed Forward

The jerk (third derivative of position) is used to inject current into the motor when the jerk state variable (jerk setpoint) becomes non-zero. The feed forward term means a difference between jerk setpoint and actual jerk does not have to exist before current will be added to the position compensator output.

$$I_J = K_{jff} * J_{setpoint}$$

Velocity and Acceleration Compensation

The velocity state variable is compensated using a proportional gain applied to the velocity error and a feed forward term. There are some additional features available when compensating velocity directly as opposed to using the derivative term in the PID compensator. The main difference is the availability of useful velocity conditioners and observers for velocity feedback.

Filter F2 can be optionally located here. Setting the Bp term to non-zero causes the filter to receive its input from this loop instead of from the derivative term in PID section of the compensation loop.

The proportional term of the acceleration loop is best used on very high-resolution systems. This is because encoder feedback is plentiful even at low speed.

Both the velocity and acceleration loops have a feed forward term. Use this term to inject a proportional amount of current in the motor when the respective state variables become non-zero. Using the feed forward variables reduces the loop dependency on first observing an error, and then compensating.

$$I_V = K_{vff} * V_{setpoint} + B_p * V_{error}$$

$$I_A = K_{aff} * A_{setpoint} + J_p * A_{error}$$

Anti-Hunt Zone

A scalar can be applied to the gains of the position loop when the velocity set-point is zero and the encoder is within a configurable number of counts from the desired set-point. This is useful in systems where load may change at rest. For example, in a belt driven system, the belt may slack when the motor is at rest. The apparent load on the motor is significantly reduced because the real load is no longer tightly coupled. In this case, small amounts of current can cause large changes in position. The system may even be unstable at rest but perform very well when in motion.

The anti-hunt zone is a symmetrical zone around the position set point where the output of the compensator will be scaled. If the actual position is outside the symmetrical zone, no anti-hunt scaling will be performed.

Detent Torque Compensation

All motors have some inherent torque that adds and subtracts from current induced torque as the motor turns. This torque disturbance is called detent torque. It is most readily observed in stepper motors (turn the shaft and you'll feel the discrete steps). If the magnitude, phase and frequency of detent torque are known (or measured), it can be compensated by applying a sinusoid 180 degrees out of phase with the detent torque.

Velocity Observer

Velocity can be measured in three ways by the MAX motion controller.

The first way is during high-speed motion (> 20kHz encoder count rate). In this mode, real velocity can be directly measured from the rate of change of the encoder. At this counting rate, there is lots of encoder information between position loop calculations (every 250us). Velocity feedback is calculated by multiplying the number of encoder counts observed by 4 giving units of encoder counts per millisecond.

The second method is used when the encoder pulse rate drops below 20kHz but above 4kHz. Now, velocity feedback is calculated by measuring the time between pulses (with 50us resolution). The velocity feedback is the reciprocal of time between pulses.

When encoder counts slow down to below 4kHz, a third observer mode is used to calculate velocity. In this mode, the time between encoder transitions is measured. If the times measured between pulses are becoming longer, then an eventual decay to 0 velocity is assumed. If the time between pulses is not becoming longer, then the velocity is given by the reciprocal of time and no decay is assumed.

Filters

Three third order IIR filters are included for each axis. These filters allow a user to attenuate unwanted system reactions. A common use for the filter is to attenuate system bandwidth below a mechanical resonance point.

The location of the filters is optional. They can be located at the output of the current limiter, or can be used to perform filtering in separate state variable compensators. One is dedicated to each of position, velocity and acceleration filtering. Further, the velocity filter can be placed in the velocity state variable compensation loop or in the derivative in the PID compensator.