Observer-Based Motion Control

A new control method mathematically models motion control servo loops to correct errors before they occur.

By Steve Reese, Parker Hannifin

For many years, the standard means of controlling servo loops on motion controllers (or other devices) has been the PID loop. Back in the days of analog control, the motion controller was typically a CNC device responsible for closing the proportional gain (P) loop and, if used, the steady state or integral gain (I). A tachometer (tach) or speed loop was used in lieu of a derivative gain (D) function. The closed position and speed loops configured by this type of cascaded control were common.

As processor technology advanced, it became routine for the motion controller to close all servo loops. This was based on the assumption that the fastest and most capable processor was on the controller. Technicians tuning these systems now had loops that were quicker or hotter than anything available previously; however, without a tach loop, the tuning technician had to compensate for stiction and friction. Experienced techs could tune out these impediments to accurate motion using the feed-forward for acceleration (FFACC) and feed forward for velocity (FFV) settings.

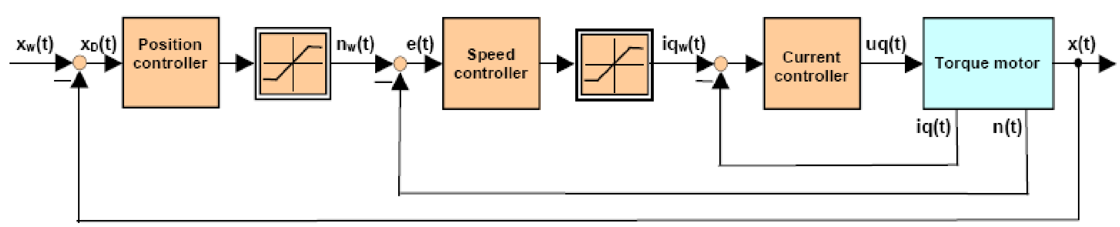

Fig. 30 Cascade control for current, speed and position in a torque motor

$x_w(t)$:	*Setpoint position value*
$x_D(t)$:	*Control difference – position controller (tracking error)*
$n_w(t)$:	*Setpoint speed value*
$e(t)$:	*Control difference – speed controller*
$iq_w(t)$:	*Setpoint torque-forming motor current*
$uq(t)$:	*Voltage input signal*
$iq(t)$:	*Torque-forming motor current*
$n(t)$:	*Motor speed*
$x(t)$:	*Motor position*

The standard PID loop, shown here in block diagram, has been the principle servo loop control method in motion and other systems for many years.

Given the same mechanical system, the usual parameters limiting position loop accuracy were the bandwidth of the servo loop and the PWM frequency of the amplifier. Standard motion-controllers can make changes on a multi-axis trajectory approximately every 500 microsec and on each servo loop in a fraction of that time depending on the number of axes. On the other hand, amplifiers can change their output no faster than one cycle of the PWM. With a 20-kHz amplifier, the fastest change in output current still takes 50 microsec creating a hard limit for any updates.

This means the amount of acceptable position error determines the maximum available speed. Motions that are too fast cannot be corrected quickly enough to avoid exceeding the error limits. The need for ultra accuracy requires slowing the motion so that 50 microsec delay does not produce a gain break or nonlinear situation. Though tricks like an FFACC cutoff started to work their way into modern controllers, in practical terms the controllers have hit a wall for quicker and more accurate control. This is due in large part to the PWM bandwidth limits using a PID with FF control scheme.

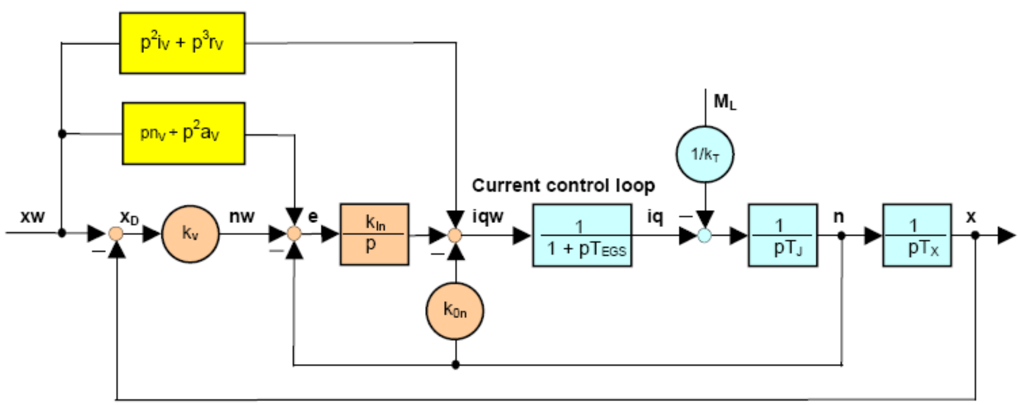

Fig. 32 Servo cascade control with feedforward control in a torque motor. By moving the P component into the feedback path, the command and disturbance behaviour can be optimised separately.

Feed forward was introduced to reduce response time of the control. The feed forward component was introduced into the feedback paths to let technicians optimize command and disturbance behavior separately.

Yet the market still needs controllers with quicker response times. In fact, controllers today must often deal with reflected inertia mismatches and lower mechanical and electrical time constants. This is primarily due to the relationship between low inertia and low impedance in today's linear and direct-drive rotary motors.

Observer-based motion control

Because standard servo systems are error-driven and therefore reactive, they suffer lag between the commanded and actual positions. What's needed is a way to anticipate and correct the error before it occurs. For example, by charting the errors that different acceleration commands produce, the controller could use an internal lookup table to anticipate the dynamic error for any given command. Error compensation could then be added directly to the current command of the controller canceling the error before it happens. The PID system would end up having little or no error to compensate.

Today’s modern controllers have the processing power needed to map servo errors at the first (acceleration) and second (jerk) derivatives of speed. So they can solve not just acceleration errors but jerk errors like those created by friction and stiction. By adding offsets to the normal command, servo amplifiers like Parker's Compax3 can correct for many sources of resonance and other repeatable disturbances before the system actually triggers the disturbance.

The base assumption used by this new breed of amplifier is the Luenberg Observer. This type of control gets results that are six times more accurate than those of standard PID controls. Proactively chasing expected errors not only reduces errors in position but also stops resonances before they happen.

For example, a motor coupling winds up as it is accelerating and then, for a short time, overhauls the load as it unwinds. These changes are noted and mapped in the auto-tune sequence. Once mapped by the controller, the low-frequency resonance never happens, nor does the error it would introduce.

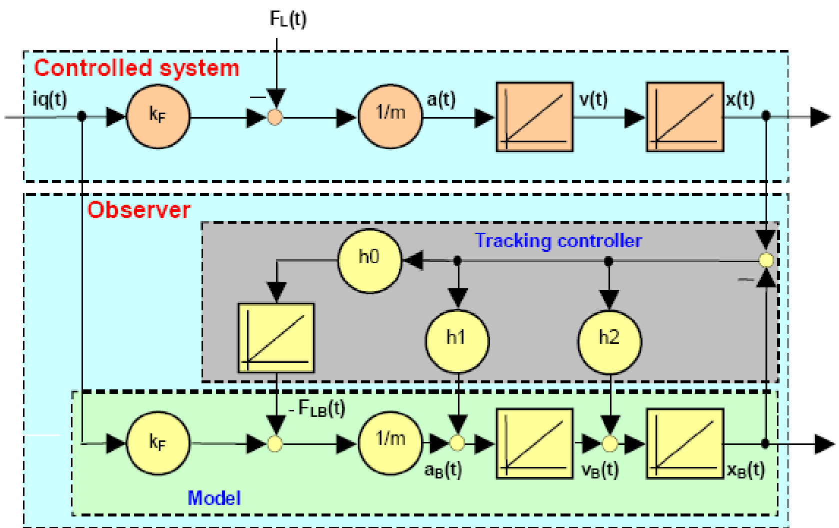

Fig. b1 Signal flow plan – Luenberg observer for linear motor drive

$i_q(t)$:	*Torque-forming motor current*
k_F:	*Force constant*
$F_L(t)$:	*External disturbing force*
m:	*Translatorally moved mass (motor + load)*
a(t):	*Acceleration*
v(t):	*Speed*
x(t):	*Position*
Index b:	*Observed signal quantities*

In motion systems, the Luenberg Observer mathematically models the actions of the position servo loop which is then compared with the actual output. The model anticipates the sources of errors, correcting them before they manifest in the actual output. Any existing errors are used to correct the model, improving overall system operation.

For more information about observer-based motion control, please contact Steve Reese at sreese@parker.com.

This article originally appeared in the February 22, 2007, issue of Machine Design.

www.parkermotion.com