Low level controller design and trajectory generation

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*—The previous high level control design document describes how the torque requirements can be limited by selection of controller constants that smooth the velocity control signal. This paper documents how the lower level position, which outputs a torque command signal can be tuned to generate the desired results.**

I. Introduction

Of the three motors thoroughly investigated so far, the Moog Animatics motor has the best form factor, and best power density with all required I/O for the process. It is also unlikely that other companies have the same or similar functionality and form factor since Animatics has spent years on litigation to reduce competition. However, the software's lack of identification methods is a major weak point in the system and a theoretical basis is needed to verify the performance of the system as the dynamics evolve in order to develop robust tuning rules. Testing is thus done in order to verify the theoretical prediction of performance and the validity of using pre-tuned parameters in practice.

II. System Identification

In order to approximate the system, an assumption is made that the plant is second order. The reasoning is that the motor is sized to respond in the low bandwidth region where the current control loop is approximated as a constant gain. That is torque is assumed to be delivered instantly.

The plant then takes on the assumed form. The units of the constants J_e, B_e are scaled versions of inertia and damping. However, as far as identification, the exact units are not necessary.

$$P\ s = \frac{1}{J_e s^2 + B_e s}$$

The SmartMotor always operates in closed loop and thus, setting all constants to zero, other than proportional gain is represented in the control loop following.

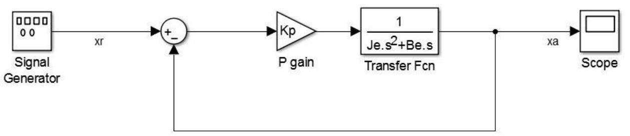

Figure 1 - Servo control loop model

The overall transfer function describing x_a/x_r is then a second order system. K_p in theory then can be adjusted until oscillation is observed in the signal.

This was verified using the SMITuner tool in the software. $K_p = 2000$ was found to be sufficient in creating approximately 15% overshoot which is used in order to identify the system which is shown in Figure 2.

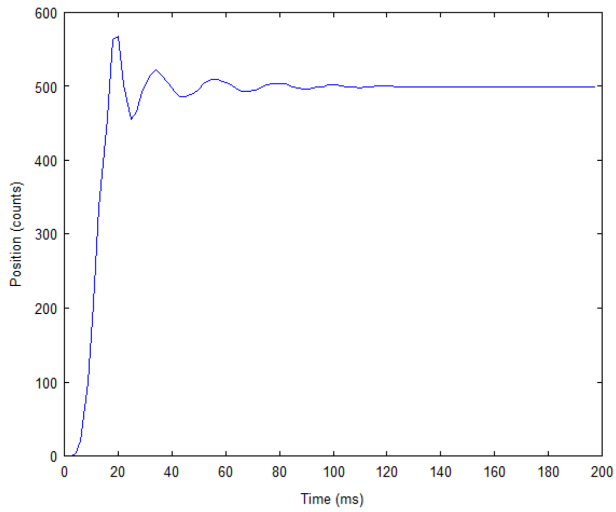

Figure 2 - Step response from SMITuner

From Figure 2 the graph of the step response is jagged due to the low data transfer rate settings of the SMITuner. Thus, a custom script was used to generate a trapezoidal waveform in order to identify the system. Using a least squares approximation, the system's constants were identified, the graphical result of the predicted response is shown in Figure 3.

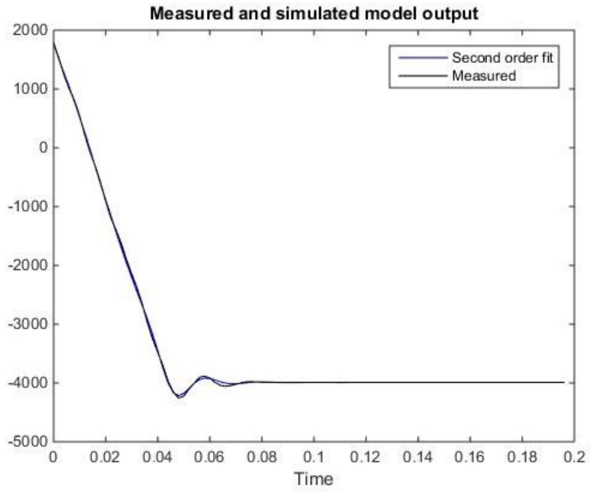

Figure 3 - Second order transfer function fit result

III. Simulation

The model of $P(s)$ was generated in MATLAB in order to use the “pidtool” function. This allowed the specification of bandwidth and phase margin. The key while using this tool was to keep the gain of the system near $K_p = 2000$ to insure the stability of the system. Otherwise, large undamped oscillations might be present. In this case, the phase margin was selected to be 70 degrees, and bandwidth to be 326 rad/s. This is the bandwidth of the system without a spool, which will undoubtedly become much lower as more inertia is added.

At this point, all three values of K_p, K_i, K_d can all be tuned easily using the tuner based on a model of the plant. Only K_a and K_v the acceleration and velocity feed-forward constants must then be specified to complete the controller design.

Fortunately, in the ideal case, selection of $K_a = J_e$ and $K_v = B_e$ is a simple rule that allows excellent system performance if the inertia is constant. Theoretically, this would create a unity gain transfer function, although in practice this is impossible.

As the inertia decreases over time, the performance decays, although not to an unacceptable level. This control scheme is generally known as a 2 DOF controller, note that feeding the velocity and acceleration directly to the control output is only possible because the trajectory generator ensures smoothness of v since acceleration a is specified. If higher performance is necessary, the K_a constant can be modified periodically with the quartic model of inertia to interpolate using radial measurements.

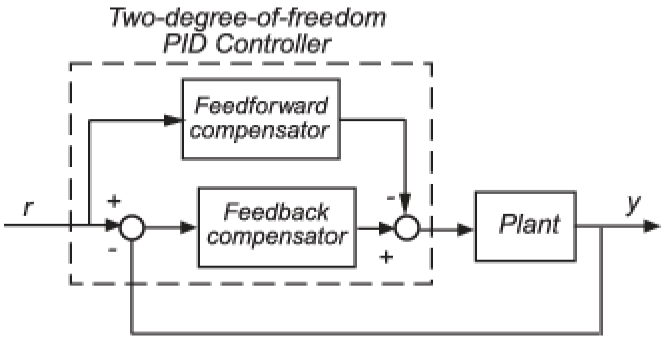

Figure 4 - Schematic of the 2 DOF control scheme from MathWorks

IV. Selection of Trajectory Parameters

The Animatics motor uses a simple trajectory generator that requires the specification of parameters ADT, and VT in order to run in velocity mode. ADT specifies the acceleration and deceleration, while VT sets the cruising velocity. These parameters are easily read off the torque curve of the motor in Figure 5. The red region is where maximum speed operation takes place, notably the reducer was designed to keep the motor operating near the peak power point. This red region has a minimum torque of approximately 0.4Nm at 4000RPM, which means 4Nm at the output of the reducer (which has a predicted 98% of the inertia). Since the velocity is never required to go over 3600RPM, this builds a safety factor into the system in terms of torque and ensures that the velocity cannot go far beyond what is necessary.

$$ADT = \frac{4Nm}{I_{system}}$$
$$\max VT = 4000RPM$$

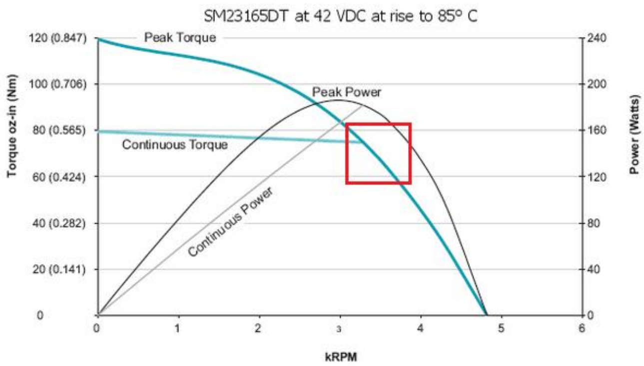

Figure 5 – SM23165DT Torque-speed curve at 42VDC