Cheat Sheet

Contacts regarding Code Development

Paul Haworth (Implemented/developed servo code on H16S)

Geoff Vanderbosch (Implemented/developed PMAS servo code on OH20S)

Chris Pauly (Commissioned H16 servo code)

Hack Summer (Moog Animatics Implementation Engineer)

Dave Kennedy (Vendor for Moog animatics SmartMotors)

Geoff Thompson (Elmo Motion Sales/Support)

Note: SMI Code for the H16S servo creel is on Vault (ask Paul H), PMAS OH20 Code is on ei-buildsrv.

Forward

This guide was designed to give the user a basic understanding of the core components of the servo creel system. It is not meant to be an exact comprehensive guide as there are many implementation details that should be considered in a full system. Even after working on both the Moog and Elmo implementations of the servo creel on the H16S and OH20S respectively, there are still gaps in my understanding of the system as there are electrical, mechanical and controls considerations. The following document provides some useful information for the control engineer in particular.

Overview of levels of understanding:

Basic

1. Understanding of the overall control architecture
   1. Where code is uploaded to influence tension control
2. Understanding the basic function of hardware in the system

Intermediate

1. Able to select appropriate gain constants
2. Able to add behavioral functionality to code

Advanced

1. Able to a commission servo system
2. Able to debug malfunctions in hardware/code
3. Able to optimize control loop tradeoffs
4. Able to determine candidate hardware for specification
5. Able to simulate motor torque/dancer response (related to 4)

Note that not all of these topics are explicitly covered in this documentation. This is just a general guideline.

Basic Architecture

H16S:

1. Each motor/drive combo is connected to the PLC over a Profinet chain
2. PLC sends high-level commands over to the SmartMotors over the Profinet chain
3. The tension is controlled directly by the Moog SmartMotors, which is an integrated drive/motor system
4. Encoders monitoring the dancer position are fed directly into the SmartMotors
5. Information regarding spool diameter is passed to the SmartMotors over the Profinet Chain

OH20S:

1. Each drive is connected to a motion-controlled called the Platinum Maestro (PMAS) over an EtherCAT chain
2. All servo motion control is calculated in the PMAS and sent to the individual drives
3. PLC sends a high level command over Modbus using custom conventions developed by EI
4. PLC sends diameter readings over Modbus
5. Encoders monitoring the dancer position are fed directly into the drives

Development

In order for the servo creel to be developed, there are a handful of high-level steps that must be completed. They are listed below and elaborated on for reference.

1. Sizing the motor
2. Setting up a test bench
3. Tuning the system
4. Benchmarking performance
5. Identify mechanical inadequacies
6. Identify electrical inadequacies
7. Further optimization to control scheme
8. Building/Testing a prototype head

Sizing the motor

In order to select the smallest motor possible, one has to have a good idea of what kind of control scheme will be used and what kind of operating bandwidth is required. The main observation after the simulation is that it is adequate to send velocity commands via the feedback of dancer displacement.

If the controller is designed in a specific way such that the velocity command is able to change smoothly, the torque can be optimized as to not over-current the motor. The main takeaway is that the control structure should be at least partially understood so that a better grasp of whether the motor is adequate can be achieved.

In practice, you might estimate the torque requirements of the motor and buy a slightly oversized system. Ensuring that there is a way to measure the torque output from the motor, one can check whether the simulations and experiments agree.

Setting up a test bench

The main thing here is to mimic the towpath and mechanical elements as closely as possible to mitigate the possibility of unexpected issues. For example, here are a few takeaways from the first two test benches:

1. The “pizza cutter” design for fastening the spools to the material spindle is inadequate
   1. Reversing load from the servo motor caused gouging and created FOD
2. If the spring constant is too low, the dancer cannot reset and keep tension on the tow
3. Backer winding issues on the H16 are caused by fast adds which eventually drove the overall dancer tension higher
4. Regeneration of energy onto the voltage bus can cause failures in the motors or in the power supply, the voltage must be clamped by a diode
5. Diameter sensor performance must be strategically filtered to maintain good signal

Tuning the system

The first most obvious job is to get the motor spinning. Afterward, the velocity loop must be tuned to at least 10Hz bandwidth to allow the servo creel equations to work properly. General dancer behavior as a function of servo tuning constants is provided in this document.

Benchmarking performance

The two main benchmarks that are run on the test bench are traverse and add/cut speed. In general, the add/cut speed is less than or equal to the traverse speed, but have higher acceleration requirements. Thus, the question is whether the tuning for the system will allow for both velocity regimes to work properly. In general, it is a good idea to find out what gain works with the add/cut parameters such that the motor does not over-torque. Calculate the dancer displacement at the traverse speed in order to check whether overtravel will occur.

Identify Mechanical Inadequacies

There were many small changes to the mechanical system to accommodate the servo creel. A previously used example is that the pizza cutter design created an unexpected FOD failure. The solution was to redesign the material spindle with expanding shoes that hold the inner diameter of the cardboard spools via friction. Many prototypes were made in order to develop the final implementation which was all run on the test bench. Another example is that the dancer stroke was a bit short to allow high-speed traverse payout. Thus, it was optimized to have a longer length.

Identify Electrical Inadequacies

The main consideration in the electrical system is the power supply for the servo amplifiers. Power is a huge concern since during peak load the system cannot have a sharp drop in voltage. The test bench identified that there was a goldilocks zone for the voltage. Too high and the motor electronics would be damaged, while too low would reduce the torque available. Another point is that during braking or regeneration, the voltage on the bus could rise, causing motor failure if it is unchecked. The problem is additive if multiple motors/amplifiers are on the same bus.

Further optimization to controller scheme

The first servo creel control scheme was mathematically solved to be a proportional controller. Given the mathematical model of the plant, and assuming that the dancer response should be 1st order with respect to the payout velocity was how this simple controller was calculated.

The lesson here is not to simply use PID just because it’s normal. The P controller gave a lot of good characteristics such as larger stopping distances as speed increased, bounded torque requirements directly proportional to add/cut speed, easy to tune etc. However, afterward, the torque was optimized by adding the first-order filter to the controller output. This allowed for smoother torque transition and subsequently lower torque.

After the second controller was designed, higher add/cut performance was necessary. Thus, a secondary feedback loop was added to change the behavior of the control loop dependent on payout speed. This was done using saturation and linear gains so switching instability was not an issue. Although simple in concept, a good understanding of loop stability via simulation is recommended before attempting this in practice. It was shown that this system was useful in limiting dancer travel at high traverse speeds and limiting torque requirements at add/cut speeds.

Building/Testing the prototype head

From the test bench, the servo creel hardware stayed almost completely constant as the H16EXP prototype head was built. The power supplies were the main changes and were tested before further assembly was done. After the prototype was built, there were some issues regarding motor reliability and a few dancer spring changes regarding the creel. The most important thing was that the challenges were surmountable and did not require huge redesign processes + mechanical modification.

Main equations

Steady state dancer displacement OH20S

is the steady-state displacement (inches)

is the setpoint (inches)

is the payout velocity in (inches per minute)

is the servo loop effective gain in (rad/s)

Nominal constant ranges for OH20S

Note that these are **not** strict requirements, just ranges that have been used previously. After every set of parameters, it is important to verify the margin for error from each end stop is at least 0.25’’. Usually, this number is around 0.5’’.

Steady-state dancer displacement H16S

The difference between the OH20S and the H16S is that a threshold value exists after which the gain is increased. The reason this control change is implemented is due to the top speed and the cut speed being significantly different. Although there is also code for the OH20S to use the same control scheme, currently in production it has been disabled.

is the steady-state displacement (inches)

is the setpoint (inches)

is the payout velocity in (inches per minute)

is the servo loop effective gain in (rad/s)

is the dancer threshold after which the gain is doubled (inches)

Nominal constants and ranges for H16S

Controller structure

The controller converts dancer error into a tow surface speed command. Surface speed is used to model performance despite rotational speed commands being sent. The reason for this is that the same dancer dynamics are to be maintained during any state of the spool diameter. A division to convert surface speed to the correct rotational speed is calculated afterward.

The controller is modeled as a first-order system in Laplace domain

Converting to a discrete implementation with being the sample time can be done via Euler approximation. This process is shown below and can be done to new control structures in the future. Currently, the constant is set as in order to critically damp the dancer's response. Thus, it is not a free parameter to be manipulated.

Dancer plant model

is the spool payout speed

is the tow payout speed

is the dancer position

Key observations from H16S development:

1. Since velocity commands are sent from the motor, inertia ratios can be very large with adequate tension control performance (H16S regularly operates with a ratio higher than 50:1)
   1. This is why the inertia ratio was ignored in favor of a torque requirement analysis
2. Controllers that tend to reduce the steady-state dancer error should not be used. The reason is that the dancer must displace more at higher speeds to give reasonable stopping distance so that motors will not over-torque.
   1. In control theory terms, the disturbance should not be rejected in steady-state
3. If the bandwidth of the velocity control loop is around 10Hz, removing the dynamics of velocity transfer functions is a reasonable assumption
4. Generally, the add and cut events create the largest torque requirements from the motor if the acceleration phase is 0.5G
5. The gearbox must have a static moment rating to spool inertia in a cantilevered fashion
6. Peak voltages must be controlled to mitigate the possibility of damaging the motors. Currently, TVS diodes are used to absorb regen onto the bus.
7. Backer winding issues, tend to happen on add/cut events if tension is insufficient or servo creel gain is too high
8. From experimental extrapolation and simulation, it is possible to reach top speeds of 5500ipm if add/cut speeds are unchanged from 2000ipm and the control loop is pushed to its limit. This would require a reduction ratio of 8:1 on the SM23165MT instead of 10:1
9. Using the dancer data over time, the time of a cut event can be estimated and tow end placement, as a result, can also be estimated (RIPIT)
10. Moog Animatics SmartMotors are very easy to code and set up relatively speaking compared to most other servo systems. They are very valuable for temporary experimental setups.

Key observations from OH20S development:

1. Backer winding issues are caused by general payout, not as much by add/cut events
2. PMAS Code support from Elmo is very weak due to primary engineering being done in Israel with applications engineers being understaffed in the US
   1. It is likely that any response will take in excess of one week, one month for a detailed response has been observed in the past. A complaint was sent previously to request better service, hopefully, things get better in the future
3. The kinetic energy within a spool is greatest during payout near 14’’ diameter since the cassette makes up a large portion of the rotating inertia. This is important when testing the consequence of regenerative braking onto the voltage bus.
4. Modbus is the current communication method between the PLC and the PMAS (motion controller). This was very difficult to get working since communications/handshakes must be implemented on the PMAS side and interpreted 1:1 on the PLC side correctly.
5. Due to space constraints, the only motor drive pair that was found to be viable in terms of form factor and power density were the FLA-20 motors by Harmonic Drive and the Gold DC Whistle by Elmo Motion. It is possible that future developments will yield better options.
   1. Ideally, a higher gear ratio (higher than 9:1) on the FLA-20 would be more optimal
   2. Currently, it is likely that gearing ratios of around 15:1 or 20:1 would be preferable due to the high maximum speed at the output of the motor/gearbox
6. Related to point 3, an attempt to use an AnyBus converter on the EtherCAT chain was done and it was a failure. The PMAS could not recognize the device. There might be another way, but currently, it is unknown whether the PMAS supports any bus converter device.
7. Controllers that tend to reduce the steady-state dancer error should not be used. The reason is that the dancer must displace more at higher speeds to give reasonable stopping distance so that motors will not over-torque.
8. If the bandwidth of the velocity control loop is around 10Hz, removing the dynamics of velocity transfer functions is a reasonable assumption
9. The gearbox does not need a large static moment rating as the spool has a center of mass centered on the motor/gearbox combo
10. Generally, the add and cut events create the largest torque requirements from the motor if the acceleration phase is 0.5G
11. Peak voltages must be controlled to mitigate the possibility of triggering a reset on the power supplies. Currently, TVS diodes are used to absorb regen onto the bus.
12. Momentary overcurrent issues cause failures on the OH20 specifically Head 6. There should be a way to recover from this state via software but currently, this causes failures.
13. Although cut speed was reduced in production, the servo creel has the capability of cutting at 4000ipm which was tested in house. 5000ipm was tested and might be possible in production if dancer stroke and cutter design were optimized.

H16 Motor Settings

Below is the code that specifies the tuning constants for the SM23165MT. These were tested to be working on the H16S Gen 1 although it is possible changes were made in production machines.

Notes:

* 10:1 reducer, the system should be returned if the reduction is changed
* Since dancer tow tension acts as a disturbance torque, the integral gain should be limited to prevent steady-state oscillation

Clicking the code will open a document and code can be copied



OH20 Motor Settings

Currently, the FLA-20 motors have been properly configured and tuned in the OH20 cassette and drive parameters can simply be uploaded to all drives. Different parameters must be uploaded to the odd/even motors.

If you want to start from scratch (for new motors):

Since the OH20 motors run on the Elmo Motion platform, motor settings are configured through the Elmo Application studio. Unlike the Moog Animatics system, the Elmo drive is compatible with almost any commercial motor. However, this means that if new motors are to be used, a configuration process must be done since the drive will not have prior knowledge of what motor is electrically connected.

If a configuration is to be done from scratch, settings must be manually entered in the setup process with parameters that correspond to the physical motor parameters and limits. Using the motor spec sheet and talking to Elmo/suppliers is the best way to get the proper parameters.

Detailed Example Calculations (H16S):

Rules of thumb:

1. The higher the gain, the less variation in dancer stroke
2. The higher the gain, the more torque required
3. The higher the speed, the more dancer stroke is required
4. The more dancer stroke used at a set velocity, the less torque required

During the development of the H16S servo creel, the dancer stroke was extended from roughly 3.5’’ to 4.25’’. Using the fourth rule of thumb, this means that the torque requirements on the system can actually be reduced.

Consider the following design scenario:

We wish to have the set point at 0.5’’ and the maximum dancer displacement to be 3.25’’ during in-process payout. Suppose that the maximum speed of the payout is 3000ipm. Suppose we do not add the threshold calculation.

What is the minimum gain required to satisfy these requirements?

From the equation for steady-state displacement, it is evident that increasing reduces the steady-state dancer displacement hence the minimum gain is the solution to the equation.

As long as the motor has sufficient torque, the gain can be raised. However, this will result in higher spool accelerations, which can cause over-torque or poor film spindle winding.

Setting the threshold value

Relevant equation

Assuming that the maximum payout speed is significantly larger than the add/cut speed, this is an effective control scheme.

To verify that the system will not fail in steady-state at high payout speed we simply check the steady-state displacement

Concrete example:

So the maximum we expect the stroke to be at 3.41’’ maximum during payout. Consider the case where we did not have the threshold addition to the control scheme.

So by implementing this regime, effectively, the stroke variation off setpoint was reduced by 12% with no functional disadvantage. In general, the larger the difference between add/cut and traverse, the more effective this method is.

The main observation that was made in simulation and in implementation was that 0.5G acceleration on part does not require as much torque as the add/cut. Typically, on the H16S, the add/cut speed is much lower than the maximum payout speed.

This means that during acceleration on a part, the gain can be raised to reduce the dancer stroke. However, during the add/cut the gain must be reduced to ensure that the motor does not over-torque.

In short:

1. During material add/cut we want a low gain since rapid changes in tow velocity to require large amounts of torque
2. During acceleration/max payout speed we want to raise the gain to be efficient with dancer stroke as we know the torque requirements are less

How this is achieved in practice is not to adjust the control constants in real-time. Instead, dual feedback is added to the system in the event the threshold is passed, resulting in a simple LTI analysis.

First, we require that the add/cut speed must be different than the maximum speed, otherwise, this control scheme is not useful.

Suppose that the add/cut speed is 2500ipm. If the original control scheme was used then below calculation would yield the dancer displacement in steady-state