Preventing Runaway Conditions on PMAC

A runaway condition is potentially one of the most dangerous failures of a motion control system. Runaway is the uncontrolled motion of the system at full force and velocity. In putting together a motion control system, it is vital that all reasonable steps be taken to both prevent possible runaway conditions, and to deal safely with a runaway condition should one occur.

There are three lines of defense against runaway conditions and the damage that they could cause:

1. Proper hardware and software setup to prevent runaway.
2. Shutdown mechanisms in the event of runaway.
3. Physical protection of people and equipment in the event of runaway.

All three of these lines of defense should be planned carefully to keep an application safe against runaway.

Proper Setup to Prevent Runaway

In order to prevent runaway, it is important to understand the potential mechanisms that can cause runaway. These mechanisms differ depending on whether PMAC is performing the motor commutation or not. Each case will be examined separately.

Motors not Commutated by PMAC

If PMAC is not commutating the motor or other actuator (commutation is performed in the drive, commutation is performed in the motor, or the motor or other actuator does not require commutation), the following conditions can cause runaway:

- Loss of position feedback
- Polarity mismatch between position feedback and command output
- Partial loss of command output power supply

In addition, with a DC brush motor, it is possible for the amplifier to fail full on in one direction, which creates a runaway condition.

Loss of Position feedback: When the position loop loses its feedback, it believes the system to be stuck and will steadily increase its output to the drive, often resulting in runaway. Loss of feedback can result from a failure, full or partial, of the position sensor, of the wiring of the sensor to PMAC, or of the PMAC receiver circuits. It can also possibly result from a wrong setting of the PMAC software registers processing the feedback: the encoder conversion table, Ix03, or Ix04.

Polarity Mismatch: For a stable feedback loop, it is required that a positive output-command result in a movement in the positive direction, and a negative output command result in movement in the negative direction. If the system has the opposite polarity, the system will create a runaway condition when the position loop is closed. At rest, friction may hold the motor in place, but runaway will occur on the first disturbance larger than the friction force, or the first move command.

The first time a system is put together, never assume that the polarity match is correct. Test the polarity commands with open-loop commands and do not connect the motor to the load until a stable position loop has been obtained with a bare motor.

Most runaways due to this failure mechanism result from improper rewiring of the command outputs (e.g. exchanging DAC and DAC/) or position inputs (e.g. exchanging A and B) after the position match has first been established, or changing the software decode direction with Encoder I-variable 0.

Partial Loss of Command Output Power Supply: If one half of the +15V supply for the analog command outputs is lost, the analog outputs will saturate at the other supply, resulting in a full command to the drive, and runaway. This can result from a failure in the supply, or in the wiring. To prevent this failure mode, it is necessary to have an interlock circuit that shuts down one half of the supply in the event of the failure of the other. PMAC2’s Acc-8E analog interface board has this interlock.
Motors Commutated by PMAC

If PMAC is performing the commutation of the motor, and using the same position sensor for commutation as for the position servo loop, the conditions that can cause runaway on a motor not commutated by PMAC will not cause runaway here, although they will cause a loss of motion. On synchronous motors commutated by PMAC, commutation position reading approximately opposite in the commutation cycle from the true commutation position can cause runaway.

How this failure could occur depends on how the commutation phase position reference is established:

- Reading an absolute position sensor at power-on
- Performing a power-on phasing search move

**Absolute Sensor**: Reading an absolute motor position sensor at power-on to tell the controller is probably the most common method of establishing a phase reference. There are several possibilities for failure here:

- The sensor is not powered up when the controller reads it, resulting in an erroneous reading, but does have power later.
- Noise, particularly on the most significant bit, causes an erroneous reading.
- The sensor was mounted, or especially, re-mounted on the motor improperly, particularly reversed for a 180° error.
- The sensor was wired, or especially re-wired, to cause an effective reversal of the position reading.
- The software offset parameter Ix75 holding the difference between the sensor zero position and the commutation cycle zero position has been set, or especially, changed to an improper value.

**Phasing Search**: When a power-on phasing search is performed, attempting to move the motor to establish a phase reference, there are several possibilities for failure:

- The drive does not have power when the phasing search is attempted, but does get power later, without re-phasing.
- The drive is in fault condition when the phasing search is attempted, and is brought out of fault later, without re-phasing.
- The motor is physically prevented from moving, such as by a brake, a hard stop, or a physical obstruction.
- Loads on the motor are too great for the phasing search technique to create predictable movement, and therefore a predictable phase reference.

For a robust phasing search, the algorithm must check the following:

- Do not perform the phasing search if the drive is in fault condition
- Do not perform the phasing search if the drive is not powered (ideally the controller will see a fault if this is the case).
- Do not perform the phasing search if the axis is into the overtravel limits.
- Consider the phasing search to have failed unless the expected pattern of movement has occurred.

To use a phasing search in the application, ensure that there is a robust phasing search algorithm that either succeeds or properly reports failure.

After the phasing search has been completed, if position information is lost (for example, if encoder power is turned off during an emergency stop) for any reason, a new phasing search must be performed.

**Another Source of Runaway**

Whether or not PMAC is performing the commutation, a runaway can occur due to a request for a move that is too fast and too long for the system. Usually, this results from confusion in the programming units, such as assuming the machine is programmed in millimeters when it is expecting inches. This would result in moves 25 times too fast and 25 times too long. In many cases, this would cause an effective runaway.
**Shutdown Mechanisms in the Event of Runaway**

**Safety Shutdowns**
If a runaway does occur, there are two important safety mechanisms, either one of which should shut down the system quickly, to prevent damage or injury. These are:

- PMAC following error limit
- Drive overcurrent limit

**Following Error Limit:** If there is a runaway condition, there will be a very quick divergence between commanded and actual positions, which means a rapidly growing following error. PMAC has a fatal, or shutdown, following error limit parameter Ix11 for each motor that it controls. If the error exceeds this limit, PMAC kills this motor, disabling the amplifier and forcing the command outputs to zero.

It is essential that as soon as the motor is connected to a load and has any potential of causing damage or injury that the following error limit be enabled and set to a value small enough to cause a quick shutdown. The smaller the following error limit is, the more quickly it can stop a runaway condition. Proper tuning of the servo loop, especially the feedforward, can greatly reduce the maximum valid following error, permitting a much tighter limit without nuisance trips.

**Overcurrent Limit:** In a runaway condition, PMAC will quickly request the maximum current from the drive. This should cause the drive to fault for excessive current, which forces it to shut down. Make sure a drive is selected that can shut down quickly and reliably on an overcurrent fault.

**Shutdown Action**
When either of these limits causes a shutdown of the motor, the motor and its load will coast to a stop in the absence of external loads, unless some other action is taken. If the coasting stop is not acceptable, one of two mechanisms to force a quicker stop must be employed. If the result of shutdown would be continued movement due to an external load, only the second solution would be valid:

- Shorting motor leads together
- Engaging a brake

Either of these methods should be fail-safe: motor leads should be shorted together unless actively forced apart; the brake should be engaged unless actively forced open.

**Shorting Motor Leads Together:** This method turns the motor into a self-generator, forcing a much faster stop than simply withdrawing power with the motor leads open-circuited. On a permanent magnet brush motor, the leads should only be shorted together through an external resistor to prevent excessive current flow from demagnetizing the motor. On a permanent magnet brushless motor, the AC nature of the current flows and the motor inductance will limit current enough to prevent demagnetization, even if the leads are directly shorted together. It is important to realize that this does not hold the motor once it is stopped. On an AC induction motor, shorting the leads together does not aid in bringing the motor to a stop.

Some drives, when disabled or in the event of loss of DC bus supply, force all of the bottom transistors on using the remaining bias supply for as long as possible. This shorts the motor leads together through the transistors, fly-back diodes, and the common of the DC supply.

**Engaging a Brake:** A mechanical brake that automatically engages except when the drive is enabled and powered is probably the most robust method of forcing the motor to a stop, and holding it at a stop. Such a brake may also be necessary for reasons other than stopping runaway, such as holding a vertical axis in the absence of power.

It is important to test the effectiveness of these shutdown mechanisms in your system. First using an unloaded motor, deliberately create a runaway condition and see how quickly these two mechanisms can stop the runaway. Remember that an unloaded motor will coast far further than a loaded motor after shutdown.
Re-Enabling after Shutdown
It can improve the safety of the system to make it more difficult to re-enable after a shutdown that it is to do a normal enabling. If it is too easy to re-enable after a shutdown, the runaway condition will probably just occur again. In the worst case, repetitive automatic software re-enabling of the system may effectively completely defeat the shutdown mechanisms.

A shutdown mechanism that requires a special manual step as part of the re-enabling process is a good idea. Some drives require a special fault-clear input after a fault before the drive can be re-enabled. Others require a full cycling of power before they can be re-enabled.

Physical Protection in the Event of Runaway
Prudent design practices require that none of the above protections be relied on absolutely for the protection of equipment, or especially, people. The moving mechanisms should be encased if possible so that even the most violent runaway condition is contained. If this is not possible, there must be barriers and sensors surrounding the mechanisms to keep people out, and to shut down the motion if people enter the zone of motion of the mechanism.

These barriers and sensors should be in place and active even when the system is under development. It is at this time that mistakes are most likely to occur which can cause a runaway. It is also at this time that people working on the system are most likely to be in a potentially harmful position. Most accidents and narrow escapes on motion control systems occur on systems under development, not on finished systems.

Exactly what physical protections are used is system specific, but these can include:

- Hard stops at the end of a linear travel range
- Physical encasing of the motion axes
- Door access with automatic interlock shutting down motion when opened
- Light curtains that shut down motion when breached

These protections are important for reasons other than runaway as well, such as to prevent injury during normal operation. In deciding which protections to use, the designer must consider all safety aspects of the machine.