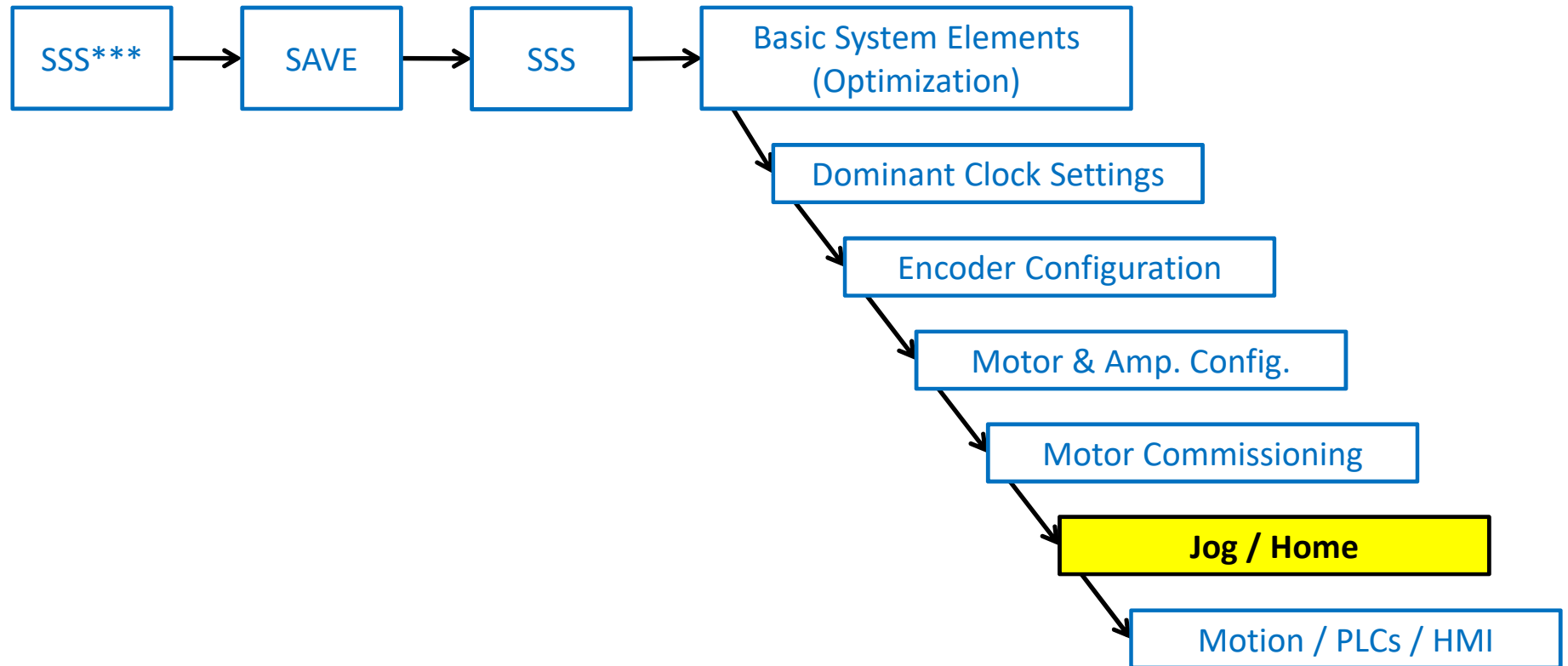




Motor Homing

System Configuration



What is Homing?

➤ Establishing an initial position reference for a motor

- Typically, defining the 0 “home” position
- With user configurable home offset
- Home search (capture) move with incremental encoders
 - Capture on index, flag, or a combination
 - Commonly used w/ quadrature and sinusoidal encoders
 - Capture on hard-stop if no index or flag (slower speed, not as accurate/repeatable)
- Simple “automatic” computation with absolute encoders
 - Absolute serial encoders (single turn, or multi-turn)
 - Resolvers (single turn only)
 - No motion required
 - Motor does not have to be energized

➤ Homing is generally performed once per machine power-up or reset

➤ Why is homing important?

- Setting up machine limits (workspace)
 - Soft limits
 - Offsets
- Establishing reference
 - Compensation tables
 - Kinematics (robot)

Home Search Move Profile

➤ Home search move profile

- Motor[].HomeVel (Positive or Negative)
- Motor[].JogTa
- Motor[].JogTs
- Motor[].HomeOffset

➤ Home search move online commands

- #1HM or #1HOME
- #1..4HM
- #1,2,3,4HM

➤ Home search move program commands

- HOME 1
- HOME 1..4
- HOME 1,2,3,4

➤ Units

- HomeVel [**motor units/msec**]
- JogTa
 - [msec] if > 0 , acceleration time
 - [msec²/motor unit] if < 0 , inverse acceleration rate
- JogTs
 - [msec] if > 0 , s-curve time
 - [msec³/motor unit] if < 0 , inverse jerk rate
- HomeOffset [**motor units**]

➤ Status bits to watch

- Motor[].HomeInProgress
- Motor[].HomeComplete

Home Search Move Control

➤ Motor[].CaptureMode

- = 0 Hardware capture (default)
 - Encoder is processed by the ASIC
 - Input trigger is wired to / processed by the ASIC
 - Position latched on the trigger in the ASIC

- = 1 “Software” capture
 - Encoder not directly processed by the ASIC
 - Input trigger is wired to / processed by the ASIC
 - Position latched from actual position

- = 2 Torque limit capture
 - Hard stop homing or torque limit trigger
 - Triggers when Motor[].WarnFeLimit is exceeded

- = 3 “Timer-assisted software” capture
 - Encoder not directly processed by the ASIC
 - Input trigger not wired to / processed by the ASIC
 - Captured position derived from actual pos. history

Home Search Move Control

➤ Gate3[].Chan[].CaptCtrl

- Range 0 – 15 for combination of inputs (low or high)

Gate3[i].Chan[j].CaptCtrl = 0:	Immediate or Hall capture
Gate3[i].Chan[j].CaptCtrl = 1:	Capture on Index (CHCn) high
Gate3[i].Chan[j].CaptCtrl = 2:	Capture on Flag high
Gate3[i].Chan[j].CaptCtrl = 3:	Capture on (Index low AND Flag low)
Gate3[i].Chan[j].CaptCtrl = 4:	Immediate or Hall capture
Gate3[i].Chan[j].CaptCtrl = 5:	Capture on Index (CHCn) low
Gate3[i].Chan[j].CaptCtrl = 6:	Capture on Flag high
Gate3[i].Chan[j].CaptCtrl = 7:	Capture on (Index high AND Flag low)
Gate3[i].Chan[j].CaptCtrl = 8:	Immediate or Hall capture
Gate3[i].Chan[j].CaptCtrl = 9:	Capture on Index (CHCn) high
Gate3[i].Chan[j].CaptCtrl = 10:	Capture on Flag low
Gate3[i].Chan[j].CaptCtrl = 11:	Capture on (Index low AND Flag high)
Gate3[i].Chan[j].CaptCtrl = 12:	Immediate or Hall capture
Gate3[i].Chan[j].CaptCtrl = 13:	Capture on Index (CHCn) low
Gate3[i].Chan[j].CaptCtrl = 14:	Capture on Flag low
Gate3[i].Chan[j].CaptCtrl = 15:	Capture on (Index high AND Flag high)

➤ Gate3[].Chan[].CaptFlagSel

- = 0 Home input flag
- = 1 Positive limit flag
- = 2 Negative limit flag
- = 3 User Flag

➤ Homing Exercise Examples:

1. Arbitrary homing
2. Homing to Minus Limit AND Encoder Index
3. Homing to Home Flag
4. Homing to Home Flag AND Encoder Index
5. Homing to hard stop

Homing Exercise 1: Arbitrary

➤ Arbitrary home 0 position

- Sets the motor position counter to zero
- Often used when commissioning or troubleshooting

➤ Online command

```
Terminal: Online [192.168.0.201 : SSH]
#1HMZ
#1HOMEZ
#1..4HMZ
#1,2,3,4HMZ
```

```
Position: Online[192....]
#1 0.000000 mm
#2 0.000000 mm
#3 0.000000 mm
#4 0 deg
```

➤ Program command

- HOMEZ 1
- HOMEZ 1..4
- HOMEZ 1,2,3,4

➤ Motor[].pAbsPos must be = 0

- So not reading absolute position
- If pAbsPos \neq 0, PMAC will try to read position from the specified register

➤ Does not care about

- Capture control
- Flag select
- Home offset



Note

The same command is used to read absolute position if pAbsPos is pointing to a register address

Homing Exercise 2: Index

- What is an index pulse (or C channel)?
- Zeroes position at the edge of the index pulse
 - Can be low or high
- Basic index capture (default) settings
 - Motor[].CaptureMode = 0
 - Gate3[].Chan[].CaptCtrl = 1
 - Gate3[].Chan[].CaptFlagSel?
 - Does not matter

Gate3[i].Chan[j].CaptCtrl = 0:	Immediate or Hall capture
Gate3[i].Chan[j].CaptCtrl = 1:	Capture on Index (CHCn) high
Gate3[i].Chan[j].CaptCtrl = 2:	Capture on Flag high

a) Without offsets

- For “linear motors” (#1..3)
 - Home velocity 10 mm/s
 - Acceleration rate 0.15 Gs
 - Jerk Limit 0
- For rotary motor (#4)
 - Home velocity -0.5 deg/s
 - Acceleration time 300 msec
 - Jerk Limit 0

b) With offsets

- Find and store (save) home offsets for all motors so that the Delta Tau Logo points upwards after homing

Homing Exercise 3: Minus Limit Flag

- What is a minus limit?
- Zeroes position at the edge of the minus limit
 - Can be low or high
- Basic Minus flag capture settings
 - HomeVel positive or negative?
 - Motor[].CaptureMode = 0
 - Gate3[].Chan[].CaptCtrl = 2 (high) or 10 (low)
 - Gate3[].Chan[].CaptFlagSel = 2

Gate3[i].Chan[j].CaptCtrl = 1:	Capture on Index (CHCn) high
Gate3[i].Chan[j].CaptCtrl = 2:	Capture on Flag high
Gate3[i].Chan[j].CaptCtrl = 3:	Capture on (Index low AND Flag low)

Gate3[i].Chan[j].CaptFlagSel = 0:	HOMEn (Home Flag n)
Gate3[i].Chan[j].CaptFlagSel = 1:	PLIMn (Positive End Limit Flag n)
Gate3[i].Chan[j].CaptFlagSel = 2:	MLIMn (Negative End Limit Flag n)
Gate3[i].Chan[j].CaptFlagSel = 3:	USERn (User Flag n)

a) Perform a home flag search move

- Using the existing profile parameters
 - HomeVel, JogTa, and JogTs
- Set up for high true flag
- Redo for low true flag



Note

Limits must be disabled (Motor[].pLimits = 0) when used for a home search move



Note

Remember to restore the limits pointer once finished with homing

Homing Exercise 4: Home Flag

➤ What is a home flag?

➤ Zeroes position at the edge of the home flag

- Can be low or high

➤ Basic home flag capture settings

- Motor[].CaptureMode = 0
- Gate3[].Chan[].CaptCtrl = 2 (high) or 10 (low)
- Gate3[].Chan[].CaptFlagSel = 0

a) Perform a home flag search move

- Using the existing profile parameters
 - HomeVel, JogTa, and JogTs
- Set up for high true flag
- Redo for low true flag

Gate3[i].Chan[j].CaptCtrl = 1:	Capture on Index (CHCn) high
Gate3[i].Chan[j].CaptCtrl = 2:	Capture on Flag high
Gate3[i].Chan[j].CaptCtrl = 3:	Capture on (Index low AND Flag low)

Gate3[i].Chan[j].CaptFlagSel = 0:	HOMEn (Home Flag n)
Gate3[i].Chan[j].CaptFlagSel = 1:	PLIMn (Positive End Limit Flag n)
Gate3[i].Chan[j].CaptFlagSel = 2:	MLIMn (Negative End Limit Flag n)
Gate3[i].Chan[j].CaptFlagSel = 3:	USERn (User Flag n)

Homing Exercise 5: Index & Home Flag

➤ Basic home index combo capture settings

- Motor[].CaptureMode = 0
- Gate3[].Chan[].CaptCtrl = 15 (high high)
- Gate3[].Chan[].CaptFlagSel = 0

Gate3[i].Chan[j].CaptCtrl = 14: Capture on Flag low

Gate3[i].Chan[j].CaptCtrl = 15: Capture on (Index high AND Flag high)

Gate3[i].Chan[j].CaptFlagSel = 0: HOMEn (Home Flag n)

Gate3[i].Chan[j].CaptFlagSel = 1: PLIMn (Positive End Limit Flag n)

Gate3[i].Chan[j].CaptFlagSel = 2: MLIMn (Negative End Limit Flag n)

Gate3[i].Chan[j].CaptFlagSel = 3: USERn (User Flag n)

a) Perform a home & index flag search move

- Using the existing profile parameters
 - HomeVel, JogTa, and JogTs
- Set up for high true index and low true flag
- Redo for high true index and high true flag

Homing Exercise 6: Hard Stop

➤ Basic hard stop capture settings

- Motor[].CaptureMode = 2
- Gate3[].Chan[].CaptCtrl?
- Gate3[].Chan[].CaptFlagSel?
- Recommended motor settings
 - Disable the position error integrator
 - Motor[].Servo.SwZvInt
 - Short time or slow accel rate
 - Slower home velocity
 - Opposite direction home offset (recommended)
 - No stopping
- Optimize with trial and error (system dependent)
 - Start slow and increase gradually
 - Plot following error during move
 - High acceleration could trigger immediately
 - High speed can create a clunk

a) Example

```
Motor[1].CaptureMode = 2
Motor[1].WarnFeLimit = 0.050
Motor[1].Servo.SwZvInt = 1
Motor[1].JogTa = 18
Motor[1].JogTs = 2
Motor[1].HomeVel = 0.005
Motor[1].HomeOffset = -1
```

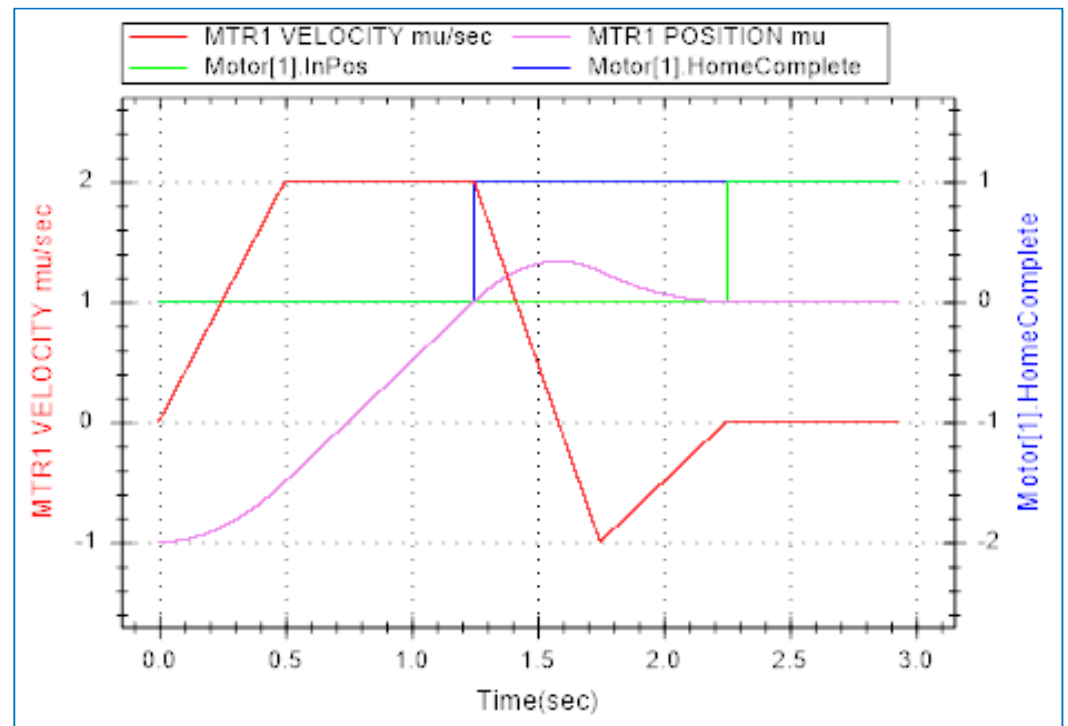


Note

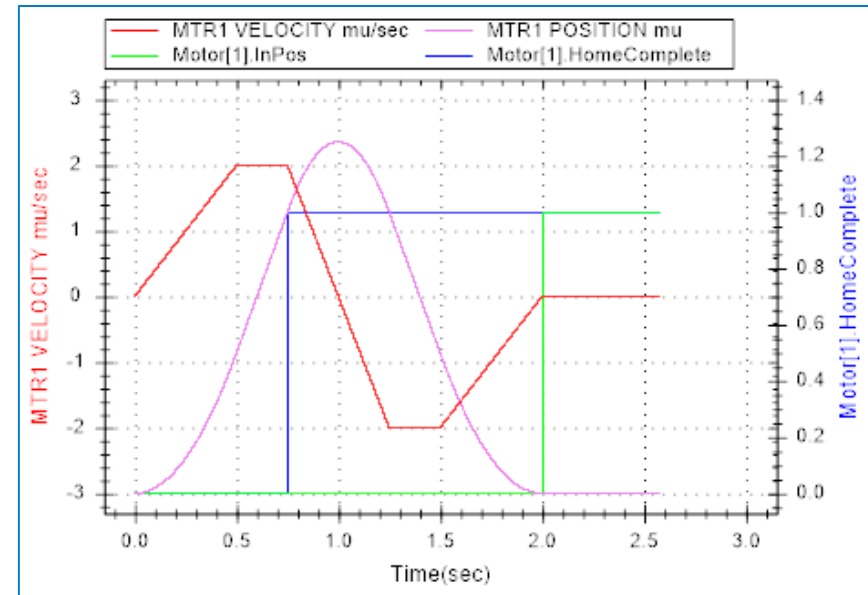
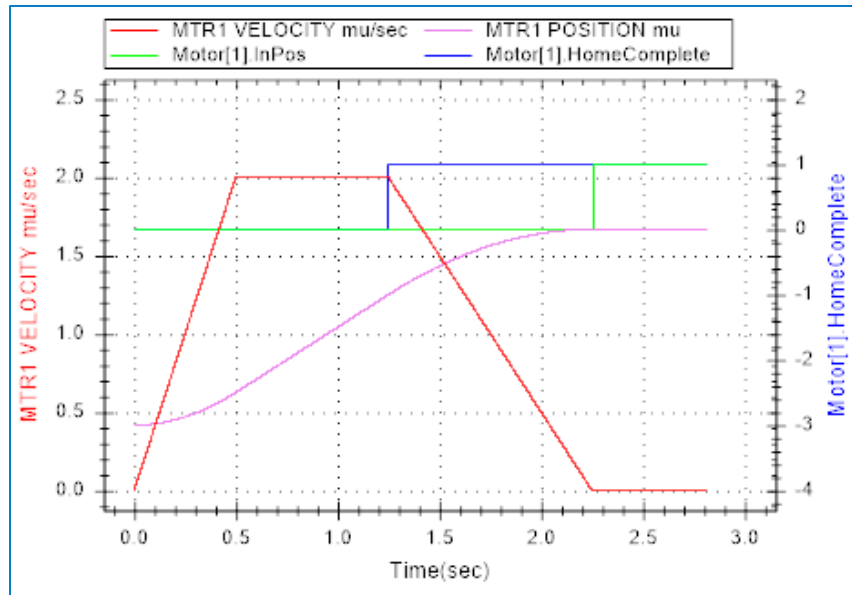
When finished with this exercise, set and save homing to only index (with original offset), restore integration mode (=0), and jog acceleration settings

Typical Home Search Move Profile

- For a Homing search move at 2 mu/s, and acceleration time of 500 msec (JogTa=500, JogTs=0)
 - No home offset, HomeOffset = 0
- Where did the trigger occur?
- Is the home complete bit alone sufficient to consider that homing is finished?
 - No, must also look at the in-position bit
- Are all accels and decels equal?
 - Yes, as programmed by JogTa and JogTs



Typical Home Search Move Profile



- For a Homing search move at 2 mu/s, and acceleration time of 500 msec ($JogTa=500$, $JogTs=0$)
 - With home offset
- Which plot corresponds to a positive / negative home offset?
- How many motor units is the home offset (roughly) in both cases?
 - Position is on the right hand side vertical scale
- Are all accels and decels equal?
 - Yes, as programmed by $JogTa$ and $JogTs$

Triggered Moves

- Triggered moves are Jog or rapid mode moves
- The occurrence of a trigger (specified by the capture control elements) brings the motor to the trigger location or a user configurable location relative to the trigger position
 - The trigger capture control is set up similarly to homing search moves
 - Triggered moves are particularly useful for probing or measurement type applications
- Triggered online Jog move examples
 - #1J=100^0
 - Jog #1 to 100 mu
 - If trigger occurs before, stop and move 0 mu
 - #2,3J:15^-2
 - Jog #1, and #2 15 mu incrementally
 - If trigger occurs before, stop and move -2 mu
- Triggered program move examples
 - JOG1=100^0
 - JOG2,3:15^-2
 - INC LINEAR RAPID X15^0



Note

Triggered moves do not work with indefinite jog commands J+, or J-