Calibration and Serial Robot Drive with SAMachine

Table of Contents

Calibration

1. **Initial Set Up**
   Calibration positions are set in the robot via the joint drive control in the SA Robot Driver interface. The "Joint Drive" control is used to drive to calibration positions, and within it, the "Joint Sets" control is used to save and manage these positions. For naming, keep in mind that the positions will eventually be called by an MP, and it is therefore convenient to name the positions in a numerically sequential order. A good calibration set exercises all degrees of freedom of the machine (joint values for a serial robot). However, note that much higher accuracy can be obtained by performing a "Local Calibration", or one that exercises the joints around a region of interest, but does not introduce backlash effects by changing the robot's closure. Since any number of calibrations can be added, and easily switched between, local calibrations are of great value in SA Machine. Be sure to keep this in mind when naming Calibrations.

2. **Add a Calibration in SA**
   Expand the Robot in the SA Tree View, under "Robots and Machines". Right click on "Calibrations", and select "Add Calibration". Give the Calibration a name, and expand the "Calibrations" to right click on the cal. you just added. Select "Trap Measurements from an Instrument". The Calibration will show an arrow icon to indicate that it is actively trapping measurements.

3. **Gather Calibration Data**
   A calibration data point consists of all joint angles or position data from the machine, and a point measurement at that pose with embedded rotation data optional. If the measurement is purely 3D, then more data points will be needed. The measurement sensor must be mounted after the last joint of the machine, somewhere on or near the end effector in the case of a serial robot. If using a laser tracker 6D target as the sensor, it is convenient to use the offset frame feature in the "Send Frames" Operation to send the actual TCP (Tool Center Point) point (with rotational info embedded) as the measurement. The robot TCP frame (expressed in the default flange TCP frame) can be set in the Robot Driver interface via the "Robot TCP" button. The location of the measured point in the tcp frame is entered in the calibration interface under "Measured Point in Tool Coordinates". This location must match the tcp definition in the robot. So for example, if you have defined the robot TCP as, say, the frame of an end effector tool, and you have your offset frame measurements to match that frame, then the Measured Point in Tool Coordinates in the Cal. interface will be identity (all zeroes). Once the offset frame is defined, and the location of the target in the tcp frame has been determined, the calibration data can be gathered. This is simply a matter of moving the robot to subsequent poses in the joint set, and taking a measurement at each pose with the measurement sensor. This process can easily be automated in an MP.

4. **Perform the Calibration**
   When the calibration data has been gathered, right click on the Calibration, and select "Stop Trapping Measurements". Now double click on the Calibration to open the Machine Calibration interface. Note again that any number of calibrations can exist in the same job for any given Machine. For more details, see the Calibration section in "Machine Model and Control in SA" below.

Move Measure Correct

1. Initial Set Up

True position correction of the robot tcp requires a 6D measurement sensor at the end effector. It is convenient to have defined the 6D sensor to return its measurements at the robot's tcp frame. In the case of a laser tracker with 6D target, use the offset frame feature to define a measurement profile for this purpose.

Each position which is to be corrected should include three or four parts:

1) A joint pose as a starting position, so that motion from the previous position is not performed in Cartesian space. This avoids possible issues with singularities. Note that this could actually be two joint poses, in the case where a via pose is required for e.g. collision avoidance.

2) An Approach frame. This is generally very near the final Goal frame, and in the same orientation. This is a good practice because the move to the final Goal can take the form of a plunge toward the part.

3) The final Goal frame. This is the destination of the tcp, which will be measured and corrected.

Following this three (or four) step criteria, lay out frames and associated joint poses for the MMC operation.

2. Automating the Process

The MP for an MMC will generally take this form (see "mmc with loop1 SA Kin.mp"):

1) Have user confirm that robot is about to move

2) Move Robot/Machine to Named Destination – move the robot to the joint pose start position

3) Move Robot/Machine to Frame – move to the Approach frame

4) Move Robot/Machine to Frame – move to the Goal frame

5) Copy the frame from step 4. This is a placeholder for the modified (corrected) frame.

6) Delay for Specified Time – optional, wait for the robot to settle

7) Configure and Measure – set the measurement instrument's profile, and take the measurement

8) Compare measured frame from 7 to Goal Frame from 4

9) Report delta to user, or check result of 8 against a set threshold.

10) If position is OK, jump to step 2 for the next pose, approach, and goal, or exit MP if this was the last position. If position is NOT OK, jump to step 11

11) Compute Robot/Machine Adjusted Goal Frame – This is the heart of the MMC process. This step has the following arguments:

-**Original Goal Frame**- This is the frame from step 4

-**Last Adjusted Goal Frame**- See step 11.

-**Actual Measured Frame**- This is the measurement from step 7

-**Modified Goal Frame**- This is the modified version of the copied frame from step 5 (the corrected frame that we'll put back into step 5, since we reference it here)

-**Transform Value**- This is the new corrected Goal in the Working Frame (for reporting, etc.)

11) Copy the Modified Goal Frame to the Last Adjusted Goal Frame.

12) Jump to step 6.

Teaching the Robot with a 6D Probe

1. Initial Set Up

Any 6D device can be used to teach individual frame positions, or arrays of frame positions for sequential or splined motion. It is once again convenient to use the offset frame feature in the tracker interface to set up the measurements such that the robot end effector will go where the probe tip goes. That way, the measured frames are immediately ready for the robot to receive commands to move to (or through) them.

For automating these moves, it is critically important to program initial joint space via poses, and (as always) to make sure the first move in the program is a joint space move. In particular, a spline move through a series of frames will be dramatically affected by changing the robot's position before the spline move is initiated. It is therefore imperative to program an initial joint pose which is near in proximity and in the same closure as the first frame in a spline sequence.

2. Implementation

With the set up guidelines above in mind, implementation is simply a matter of laying out via frames with the 6D device. You must keep in mind that subsequent frame positions must be laid out such that no singularities are passed through (otherwise, a via joint pose must be inserted in the sequence), and that no objects are present in the path between the adjacent frame positions. Also, large closure changes are never a good idea between adjacent frame positions.

Machine Model and Control in SA

1. The Machine

Right click on your machine in the SA Tree View under "Robots and Machines". Note the following in the pop-up menu:

- **Show Link Frames** – This displays the kinematics link frames of each Component.
- **Teach Pendant** – This pops the Teach Pendant control, which allows driving the robot via individual actuator motion, or Cartesian motion in the robot's Tool or Base frames, or in SA's World or Working frame. The robot can even be driven by graphically dragging the tcp in the plane of the view, or rotating the tcp about the view normal axis.
- **Command Controller Motion** – This is the place to test frame motion. The machine can be moved to frame destinations, or along a path through frames. The path through frames also provides the option of running through a Joint Space Spline, or through Linear Segments. The former will be a smooth path through the via frames (NOTE: the path is calculated by the machine controller, only the via positions are determined by SA Machine's Calibration), and the latter will be discrete paths between each via frame. Both individual frame moves and multiple frame path moves can be performed using either SA Kinematics or Controller Kinematics. When performing a Spline movement, always remember that the spline is created in the order that the frames are selected. Selecting frames out of sequence can lead to unpredictable, and dramatic robot movement. Also keep in mind that the spline is created from the initial position, so a starting position that does not lie along the same curve as the desired spline will also tend to produce wildly unpredictable motion.
- **Simulate Motion** – These commands will move only the model in SA, and not the actual machine. The next read from the machine (if the driver interface is running) will return the model to its 'real' position. Only SA Kinematics are available for these commands since they do not involve a controller. This is a good way to test whether a frame position is reachable from the current pose, or indeed whether it can be reached at all from the current location.

2. Machine Kinematics and Graphical Components

Expand the machine in the tree view and expand the Components. Right click on one of the components in the machine. Note the following in the pop-up menu:

- **Properties** – This shows the current state, and allows editing, of the component's kinematic properties. Note that the "Segment Transformation" properties are expressed as a transform ("6 DOF") if the component is a base for the machine, or a tool or end effector that has been added to the machine (this transform is the offset frame of the component in its parent frame). The "Segment Transformation" is expressed in D&H ("Denavit-Hartenberg") parameters if the component is a kinematic component (link) of the machine. The same is true for the component's "Joint Variable" which allows the selection of the component's degree(s) of freedom as part of the machine. Note that joint value limits can also be set to match the machine's limits for kinematic calculations.
- **Set Kinematic Control Frame** – This is where the linkages are formed. The component is attached to its parent component here.
- **Insert Before/After** – These allow you to add a component at a specified link in the kinematic chain of the machine. You are allowed to select the kinematic definition of this new component as "6 DOF" or "Denavit-Hartenberg" where appropriate as described above.

3. Calibrations

Expand the machine in the tree view and expand the Calibrations. If there are no calibrations defined, right click on Calibrations, and select "Add Calibration". As noted earlier, SA allows multiple calibrations to exist for each machine in the job. One can go back to a given calibration at any time, and activate that calibration. Right click on a Calibration, and note the following in the pop-up menu:

- **Calibration Interface** – This is where you control the calculation and application of each calibration that you make for the machine. Left click on "Calibration Interface" to have a look at it. Note the "Measured Point in Tool Coordinates". This is handy when using 3D measurements for calibration. Since it is impossible to define an offset frame for a 3D measurement, this allows you to measure the location of your target in the robot's tcp frame, and enter it here. Note the "Degrees Of Freedom" group box. This is where you control which variables the calibration is allowed to change. When initially popped, the "Machine Calibration" window is set by default for a machine that has not yet been calibrated. This is why most degrees of freedom are locked, except for the Base (the machine's base coordinate system). Hitting the "Run Calibration" button in this condition simply locates the machine in the measurement device's coordinate system. This is often referred to as an "Extrinsic Cal.". Once that is done, you can open up the remaining degrees of freedom, "Robot" (Intrinsic robot parameters, link lengths and offsets, i.e. D&H) and "Tool" (the end effector). Note that if this calibration has been populated with data (robot poses and measurements), then you can click on a "Pose Name", and the machine model will move to that pose (only the model, not the machine). If this calibration has data, and you have finished running it, you can press "Apply" to apply this cal. to the machine. The machine graphics will be updated accordingly, which provides a nice reality check.
- **Trap Measurements from an Instrument** – This was described in the Calibration section above.
- **Stop Trapping Measurements** – This was described in the Calibration section above.
- **Import Joint Poses and Match to Measurements** – If you have measured points in the SA job (with optional rotation data embedded), and an ASCII joint pose file (.csv) with the format: name, <j1>, <j2>, <j3>, <j4>, <j5>, <j6>, you can import data into a new or existing calibration.
- **Import Frames and Match to Measurements** – If you have measured frames in the SA job, and an ASCII joint pose file (.csv) with the format: name, <j1>, <j2>, <j3>, <j4>, <j5>, <j6>, you can import data into a new or existing calibration.
- **Export Joint Poses** – You can export the joint poses in a Cailbration to an ASCII joint pose file (.csv) with the format: name, <j1>, <j2>, <j3>, <j4>, <j5>, <j6>.

MP Commands for Machine Automation

1. Move Robot/Machine to Named Destination

Arg	Type	Description	Method	Value
0	Machine ID	Machine ID	Enter Value	0
1	String	Destination Name	Enter Value	
2	Boolean	Acknowledge Arrival	Enter Value	FALSE
3	Transform	Actual Transform In Wor	Result Only	Result Only

Machine ID – Same as Instrument ID for measurement devices.
Destination Name – The named joint pose saved in the SA Robot Driver.
Acknowledge Arrival – If TRUE, wait for the motion to complete before moving to the next command.
Actual Transform in Working – The final position of the TCP in Working.

2. Move Robot/Machine to Frame

Arg	Type	Description	Method	Value
0	Machine ID	Machine ID	Enter Value	0
1	Frame Name	Destination Frame	Enter Value	
2	Boolean	Use SA Kinematics	Enter Value	FALSE
3	Boolean	Acknowledge Arrival	Enter Value	FALSE
4	Transform	Actual Transform In Wor	Result Only	Result Only

Machine ID – Same as Instrument ID for measurement devices.
Frame Name – The name of the Frame in the SA job to which the TCP will be moved (Does not matter what the working frame is. The robot is always moved w.r.t. its base.)
Acknowledge Arrival – If TRUE, wait for the motion to complete before moving to the next command.
Actual Transform in Working – The final position of the TCP in Working.

3. Set Robot/Machine Parameter

Arg	Type	Description	Method	Value
0	Machine ID	Machine ID	Enter Value	0
1	String	Parameter Name	Enter Value	
2	Double	Parameter Value	Enter Value	0.000000

Machine ID – Same as Instrument ID for measurement devices.
Paramter Name – The name of the Parameter to Set.
Paramter Value – The value to set the parameter.

Paramter Examples (Parameter Names in Bold):
"Velocity" – For Cartesian (e.g. tool frame) velocity. Parameter Value converted to an int, and is the index of the available options in the interface's combo box. For Kuka, this is percentage of max: 0 is 1%, 1 is 5%, 2 is 10%, 3 is 20%, 4 is 40%, 5 is 60%, 6 is 80%, and 7 is 100%. Any number outside the 0 through 7 range will result in no change, and a return of false.
"Acceleration" – For Cartesian (e.g. tool frame) acceleration. Parameter Value converted to an int, and is the index of the available options in the interface's combo box. For Kuka, this is percentage of max: same as for "Velocity"

"Joint Velocity" – For robot arm joint space velocity. Parameter Value converted to an int, and is the index of the available options in the interface's combo box. For Kuka, this is percentage of max: same as for "Velocity"
"Joint Acceleration" – For robot arm joint space acceleration. Parameter Value converted to an int, and is the index of the available options in the interface's combo box. For Kuka, this is percentage of max: same as for "Velocity"
"Ext Velocity" – For external axis (e.g. rail) velocity. Parameter Value converted to an int, and is the index of the available options in the interface's combo box. For Kuka, this is percentage of max: same as for "Velocity"
"Ext Acceleration" – For external axis (e.g. rail) acceleration. Parameter Value converted to an int, and is the index of the available options in the interface's combo box. For Kuka, this is percentage of max: same as for "Velocity"

"Joints to x" – Where x is the path to an ascii file. This reads the joint values from the robot, and writes them to a file with the format, **<j1>, <j2>, <j3>, <j4>, <j5>, <j6>. (**Parameter value is ignored.)

"E1 to mm" – For external axis (e.g. rail). Moves external axis 1 to Parameter Value expressed in mm.
"E1 to inches" – For external axis (e.g. rail). Moves external axis 1 to Parameter Value expressed in inches.

4. Move Robot/Machine through Path

Arg	Type	Description	Method	Value
0	Machine ID	Machine ID	Enter Value	0
1	Collection Object Name Ref List	Path Frames	Enter	Empty
2	Boolean	Use SA Kinematics	Enter Value	TRUE
3	Boolean	Linear Segments	Enter Value	FALSE
4	Boolean	Acknowledge Arrival	Enter Value	TRUE

Machine ID – Same as Instrument ID for measurement devices.
Path Frames – The set of Frames through which to move. The motion will be IN THE ORDER ENTERED.
Use SA Kinematics – Use the calibrated model from SA. If FALSE, use the robot's internal model (rarely used).
Linear Segments – If TRUE, move linearly between frames, and stop at each one before moving to the next. If FALSE, perform a smooth, splined move.
Acknowledge Arrival – If TRUE, wait for the motion to complete before moving to the next MP command (almost always want to leave this TRUE.)

5. Compute Robot/Machine Adjusted Goal Frame

Arg	Type	Description	Method	Value
0	Collection Object Name	Original Goal Frame	Enter Value	::::Frame
1	Collection Object Name	Last Adjusted Goal Frame	Enter Value	::::Frame
2	Collection Object Name	Actual Measured Frame	Enter Value	::::Frame
3	Collection Object Name	Modified Goal Frame	Enter Value	::::Frame
4	Transform	Transform Value	Result Only	Result Only

NOTE: This command requires that a 6D measurment device is attached after the last joint.
Original Goal Frame – The frame to which the robot has been moved, and is to be corrected.
Last Adjusted Goal Frame – When running in a Move-Measure-Correct loop, this is the Modified Goal Frame from the previous iteration. For the first iteration, this is generally just a copy of the Original Goal Frame.
Actual Measured Frame – The measurement from the 6D device of the Original Goal Frame. (The frame to which the robot has been moved, and is to be corrected.)
Modified Goal Frame – The corrected frame. (This is generally used as the answer. This is the frame to feed back into the robot for it to move to the corrected frame.)
Transform Value – The transform of the corrected frame in Working coordinates. (This is the output to be used for reporting, etc. since it is expressed in the current Working frame.)