

# NX 11.0.2 Additional Machine Simulation Help

## WORKING WITH MACH SIMULATION EXAMPLES

March 7, 2017

### Version #1.1

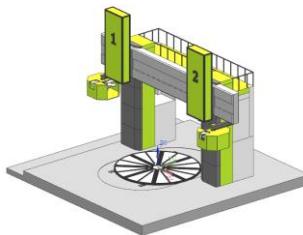
1	What's new with NX11.0.2 .....	3
2	Overview .....	4
3	Load Options .....	4
4	CAM-Setup .....	5
5	List of special machine features – Overview.....	6
6	Best practice rules for MCS using OOTB examples.....	10
6.1	General settings and rules .....	10
6.2	Special rules and settings for Mill-Turn machine tools.....	12
6.3	Library machine tool specific handling .....	12
7	About Post Processors .....	13
7.1	Unit handling OOTB Posts.....	13
7.2	General supported features of the OOTB Posts .....	14
7.3	Creating file with tool and offset data .....	15
7.4	MOM variable for MCS handling .....	17
7.5	Differences between 'pure' template posts and OOTB posts (Sinumerik).....	17
7.6	About UDEs and OOTB.....	18
7.7	Post output for Coolant (Sinumerik, TNC, Fanuc).....	18
7.8	About Special handling for 5 axis motions with Fanuc for table machine tool configuration....	18
7.9	Parameter Setting for new tap cycles.....	18
8	Using the library tools.....	24
9	About CSE Simulation Drivers .....	24
9.1	Handling of Offsets: .....	24
9.2	Handling of the tool change.....	25

9.3	Handling the reference point (Fanuc).....	26
9.4	Handling of tool correction data.....	26
9.5	About Sinumerik Cycles in content .....	27
10	About swiveling cycles .....	28
10.1	Swiveling on Heidenhain – PLANE SPATIAL .....	28
10.2	Swiveling on Sinumerik - CYCLE800 .....	29
10.3	Swiveling in Fanuc - G68.2 .....	31
11	Appendix .....	32
A.	Detailed information about ini files .....	32
B.	Swiveling Cycles .....	32
C.	PLANE SPATIAL.....	33
D.	CYCLE800.....	34
E.	G68.2 .....	39
F.	Default controller settings inside the CCFs.....	40
G.	OOTB MCS Handling output details .....	42
	Global Technical Access Center .....	44
	Installation assistance .....	44

# 1 What's new with NX11.0.2

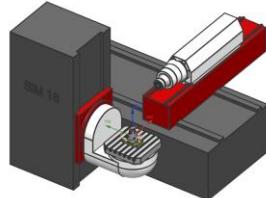
## 1.1 New Machine Kit Sim17

This machine tool is a dual channel vertical Turn-Mill machine tool supporting multiple head devices. Controller support Sinumerik only.



## 1.2 New Machine Kit Sim18

This example is a horizontal 5 axis table table B/C configuration. Supported controllers are Fanuc, Heidenhain, Okuma and Sinumerik.



## 1.3 Support for Post configurator

The support of the OOTB machines will use post processors created with Post Configurator. In this release the following sample machines are added:

- Sim09
- Sim10
- Sim05 Okuma
- Sim17 Sinumerik 840D
- Sim18 horizontal mill

## 1.4 CAM setup with operations showing checking features

Based on the sim08 an additional CAM Setup is added, which includes operations with all possible checking features of ISV. This includes:

- LIMIT CHECK
- COLLISION IPW IN RAPID
- COLLISION TOOL / FIXTURE
- COLLISION BETWEEN MACHINE COMPONENTS
- COLLISION TOOLHOLDER / IPW
- GOUGE

COLLISION IPW / INACTIVE\_TOOL

The CAM setup part file will be located here:

```
$UGII_CAM_BASE_DIR\resource\library\machine\installed_machines\sim08_mill_5ax
\cam_setup
```

## 2 Overview

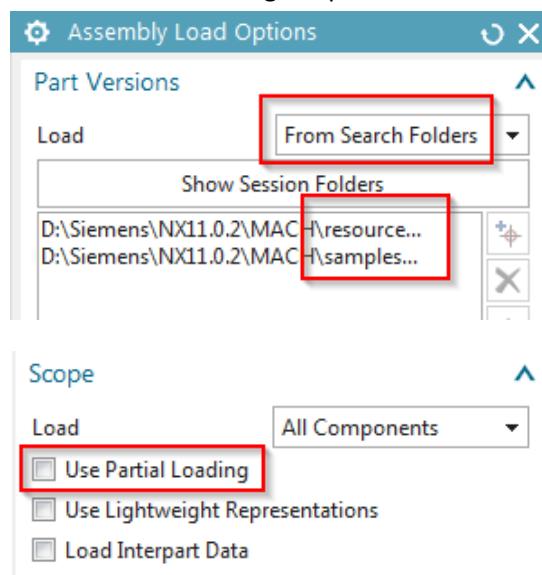
This document explains the handling and usage of the simulation examples provided out-of-the-box (OOTB). The example data is mainly contained in two locations: one is for the library machine tool models in the

`$UGII_CAM_BASE_DIR/resource/library/machine/installed_machines` folder, the other for the CAM examples utilizing the library machine tools. These CAM examples can be found under `$UGII_CAM_BASE_DIR/samples/nc_simulation_samples`. All of the machine tools in the library have preconfigured geometry, assembly and kinematics models as well as post processors and CSE controller models for the major controller types Siemens SINUMERIK 840D, Fanuc family and TNC Heidenhain Conversational; posts support metric and inch. For all machine tools in the library there is at least one CAM setup example available. The intention of these examples is to show best practice and to demonstrate the features of the NX CAM built-in machine simulation. Another intended use of the examples is as seed parts for customer specific simulation.

**NOTE:** *The examples cannot contain or show every possible feature of NX CAM, NX Post and ISV. In certain cases such as complex or multi-function machine tools, and in order to achieve advanced capabilities, specific customization of the provided posts and/or controller models will be necessary. Especially in the case of complex Mill-Turn machine tool like sim15 or head change machine sim16 and sim17 cannot be simply reused for other complex machine tool. Here additional customization in most cases is needed.*

## 3 Load Options

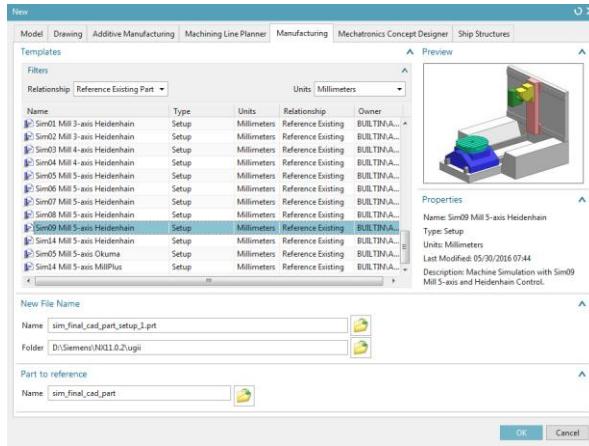
To ensure all related assembly component parts are loaded correctly it is recommended to use the "From Search Folder" load option with the following search paths. Be certain to include the three dots at the end of each path. Leave the "Use Partial Loading" - option unchecked



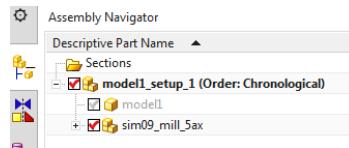
## 4 CAM-Setup

When creating a new CAM setup with the provided set of machine tools, the following best practice is suggested. It is assumed that the CAD geometry of the part to be machined already exists.

- Open the CAD part file in NX
- Select “New” and pick an appropriate entry in the Manufacturing tab.



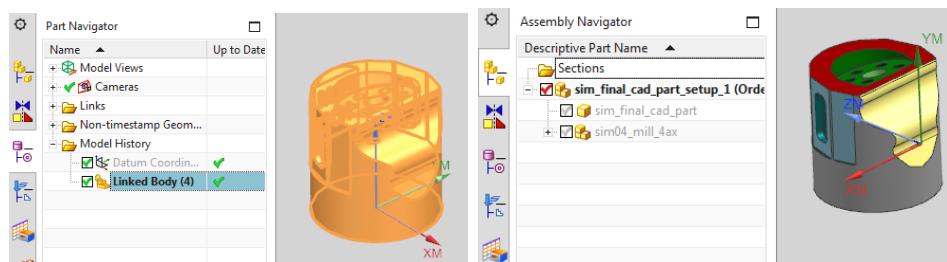
- The system will automatically create a master-model-concept-part-file referencing your CAD geometry.



- Create a Wave link to the CAD geometry (Assembly Ribbon)



- Select the body of the CAD model for the Wave object and hide the CAD component in the assembly navigator



- To prepare for material removal simulation it is recommended to have the blank geometry pre-defined as well. In the example shown it is a simple cylinder added to the assembly.

## 5 List of special machine features – Overview

The following overview shows the available out-of-the-box examples with some key features. A detailed description and how they can be reused is documented separately. This documentation is located in the machines folder under documentation e.g. for sim01\_mill\_3ax

`...\\installed_machines\\sim01_mill_3ax\\documentation`

Name	Picture	Key Features
sim01_mill_3ax		<ul style="list-style-type: none"> <li>- 3-axis milling machine</li> <li>- Controller Heidenhain, Sinumerik, Fanuc</li> <li>- Simulated toolchanger</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
sim02_mill_3ax		<ul style="list-style-type: none"> <li>- 3-axis horizontal milling machine</li> <li>- Controller Heidenhain, Sinumerik, Fanuc</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
sim03_mill_4ax		<ul style="list-style-type: none"> <li>- 4-axis horizontal mill with rotary B-axis table</li> <li>- Controller Heidenhain, Sinumerik, Fanuc</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
sim04_mill_4ax		<ul style="list-style-type: none"> <li>- 4-axis vertical mill with rotary A-axis table</li> <li>- Controller Heidenhain, Sinumerik, Fanuc</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>

<b>sim05_mill_5ax</b>		<ul style="list-style-type: none"> <li>- 5-axis horizontal gantry mill with AC-dual head kinematic</li> <li>- Controller <b>Okuma</b>, Heidenhain, Sinumerik, Fanuc</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
<b>sim06_mill_5ax</b>		<ul style="list-style-type: none"> <li>- 5-axis mill with BC dual-table kinematic</li> <li>- Controller Heidenhain, Sinumerik, Fanuc</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
<b>sim07_mill_5ax</b>		<ul style="list-style-type: none"> <li>- 5-axis mill BC table-head kinematic</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
<b>sim08_mill_5ax</b>  5-axis vertical mill with cradle:		<ul style="list-style-type: none"> <li>- 5-axis mill AC dual-table kinematic</li> <li>- (de-)activate soft limits during tool change</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>
<b>sim09_mill_5ax</b>  5-axis mill with 45 degree head:		<ul style="list-style-type: none"> <li>- 5-axis mill-turn with 45 degree head</li> <li>- Mill and turning capability for Sinumerik controller</li> <li>- Changeable head "Adapter Head" for Sinumerik, Fanuc and Heidenhain controller</li> <li>- Post Configurator postprocessor</li> <li>- Post Builder postprocessor</li> </ul>

sim10_millturn_5ax		<ul style="list-style-type: none"><li>- 5-axis millturn</li><li>- Fanuc controller</li><li>- Turret for tools</li><li>- Post Configurator postprocessor</li><li>- Post Builder postprocessor</li></ul>
sim11_turn_2ax		<ul style="list-style-type: none"><li>- 2axis turning</li><li>- Fanuc, Sinumerik controller</li><li>- Post Configurator postprocessor</li><li>- Post Builder postprocessor</li></ul>
sim12_turn_2ax		<ul style="list-style-type: none"><li>- 2axis turning</li><li>- Fanuc, Sinumerik controller</li><li>- Post Configurator postprocessor</li><li>- Post Builder postprocessor</li></ul>
sim13_turn_4ax		<ul style="list-style-type: none"><li>- 4-axis horizontal lathe with 2 spindles</li><li>- Fanuc Controller</li><li>- Post Builder postprocessor</li></ul>
sim14_mill_5ax		<ul style="list-style-type: none"><li>- Fanuc, Sinumerik, Heidenhain and <b>Millplus</b> controller</li><li>- Post Configurator postprocessor</li><li>- Post Builder postprocessor</li></ul>

<b>sim15_millturn_9ax</b>		<ul style="list-style-type: none"> <li>- 9-axis mill turn</li> <li>- Controller: Sinumerik</li> <li>- Part transfer, part clamping, moving sub spindle</li> <li>- B-axis milling and turning</li> <li>- Axis coupling</li> <li>- TRANSMIT/TRAORI support</li> <li>- Multiple chains</li> <li>- Post Builder postprocessor</li> </ul>
<b>sim16_mill_headchange</b>		<ul style="list-style-type: none"> <li>- 3 axis milling machine (base)           <ul style="list-style-type: none"> <li>+ AC-Head</li> <li>+ Right-Angular Head</li> <li>+ Vertical Head</li> </ul> </li> <li>- Controller: Sinumerik</li> <li>- Multiple chains including heads</li> <li>- TRAORI + CYCLE800 support</li> <li>- Post Builder postprocessor</li> </ul>
<b>sim17_vertical_turnmill_headchange</b>		<ul style="list-style-type: none"> <li>- 9 axis portal mill-turn machine</li> <li>- Controller: Sinumerik</li> <li>- Colinear W-Axis</li> <li>- Changeable heads</li> <li>- TRAORI + CYCLE800 support</li> <li>- Multiple chains including heads</li> <li>- Post Configurator postprocessor</li> </ul>
<b>sim18_mill_5ax</b>		<ul style="list-style-type: none"> <li>- 5 axis horizontal mill</li> <li>- Fanuc, Sinumerik, Heidenhain and Okuma</li> <li>- Post Configurator postprocessor</li> </ul>

## 6 Best practice rules for MCS using OOTB examples

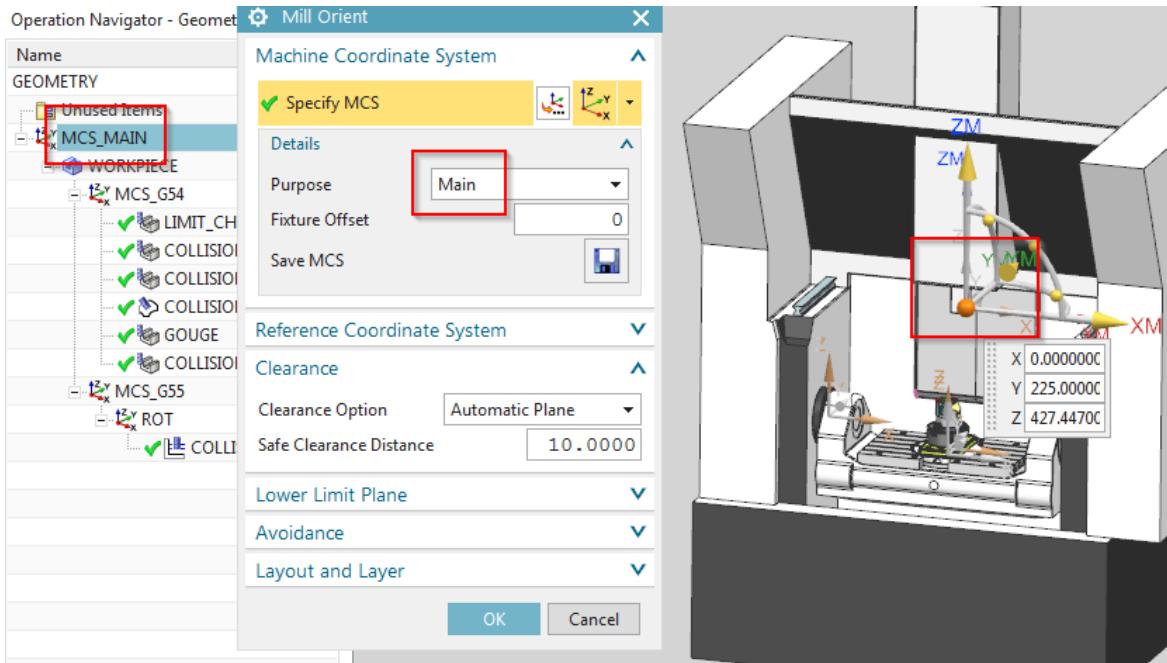
In order to achieve a complete and reliable machine simulation, different modules need to play together. These are:

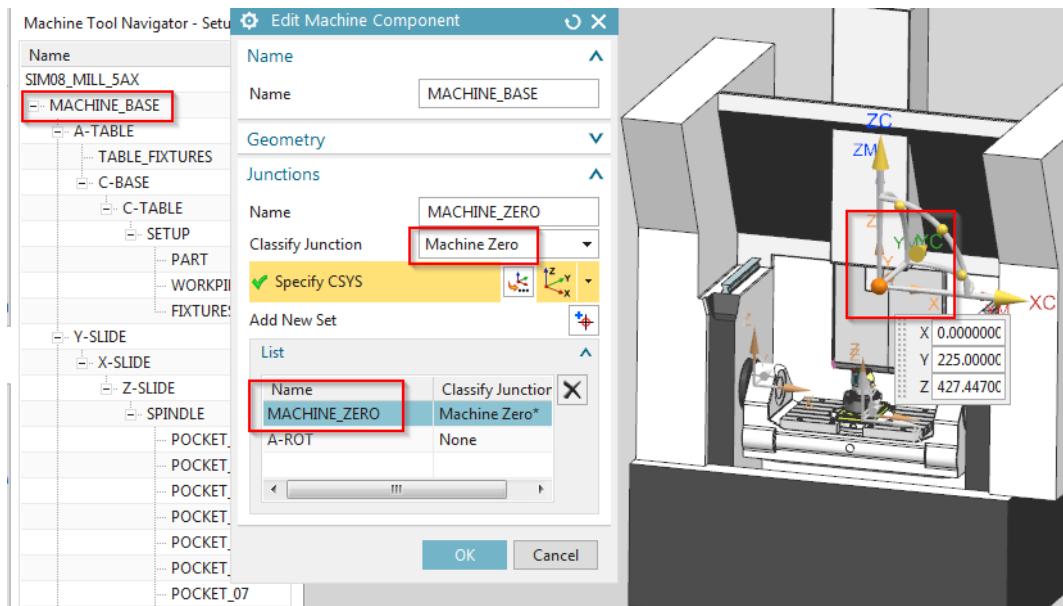
- CAM programming structure in the Operation Navigator (ONT),
- Internal Post Processor (MOM inside NX),
- TCL based post processor
- Simulation controller model.

A few rules need to be considered for NX CAM setup to support the assigned post processor creating a valid NC code which can be used at the physical machine tool as well as for simulation.

### 6.1 General settings and rules

- Each CAM setup example using one of our OOTB MACH machine tools must not include more than one single MCS with the purpose “Main”
- The “Main” MCS needs to be placed at the same location and orientation as the MACHINE\_ZERO coordinate junction of the machine tool.





- All other used MCS's in the ONT need to be of purpose "Local"
- If the "Special Output" of the local MCS is "Fixture Offset", the post processor will output a fixture offset statement based on the number of the "Fixture Offset".  
E.g. if Fixture Offset is 2:  
G55 → SINUMERIK  
G55 → Fanuc  
CYCL DEF 7.0 → TNC (7.1/2/3 will include the offset values e.g. CYCLDEF 7.1 X 100)

The Sinumerik Post will create an to\_ini.ini file including the offset vector from the local MCS with fixture offset number to the Main MCS and output that like \$P\_UIFR[2]=CTRANS(...)

- If the "Special Output" of the local MCS is "CSYS Rotation" the postprocessor will output a special statement to indicate a translation and/or a rotation. The fixture offset number must always inherit its value from the parent MCS object.  
TRANS/ROT or CYCLE800 (based on UDE) → SINUMERIK  
G68 → Fanuc  
PLANE SPATIAL → TNC

With these settings the example will look like:

MCS Name	Purpose	Fixture Offset	Special Output	Post S840D	Post Fanuc	Post TNC
<b>MCS_MAIN</b>	Main	0	-			
→ <b>MCS_G54</b>	Local	1	Fixture Offset	G54	G54	CYCL DEF 7.0
→ <b>MCS_G55</b>	Local	2	Fixture Offset	G55	G55	CYCL DEF 7.0
→ <b>ROT</b>	Local	2 (inherit)	CSYS Rotation	CYCLE800	G68.2	PLANE SPATIAL

More details and example are shown in Appendix: "OOTB MCS Handling output details"

- For the OOTB post, operations must not be placed under MCS objects with the special output "none" or "Use Main MCS". In such cases the post will not identify the correct offset for the operation and the offset creation in the to\_ini.ini files will fail.

## 6.2 Special rules and settings for Mill-Turn machine tools.

Working with Mill-Turn machines the work piece spindle object needs to be represented in the ONT. Therefore a MCS Spindle object is mandatory. That MCS object needs to be aligned with the work piece spindle and the X/Y or Z/X plane represents the lathe work plane. Also the Lathe axis needs to be aligned with the spindle axis. Working with multiple spindles e.g. Main and sub, each spindle needs to be defined by an MCS Spindle object in the ONT.

## 6.3 Library machine tool specific handling

- SIM15 post rule for local MCS object with special output “None”  
A local MCS with special purpose “None” located as child of a local MCS with special purpose “Fixture Offset number” will be used to output G59. See more in the chapter about sim15

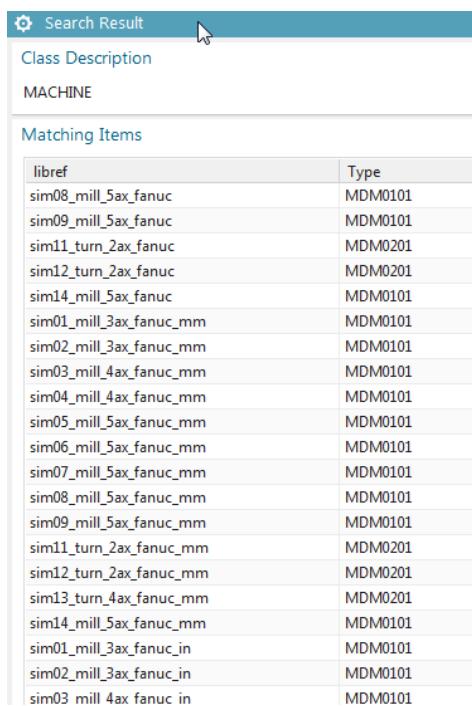
## 7 About Post Processors

All the OOTB delivered post processors are created with Post Configurator library version 4.2. The existing post processors created with Post Builder still remain in the postprocessor folder. Some differences between pure template post and OOTB post are listed in the 5<sup>th</sup> subchapter.

This chapter describes the supported functions of the example posts of the MACH library machine tools. All post processors are available as inch and metric versions.

### 7.1 Unit handling OOTB Posts

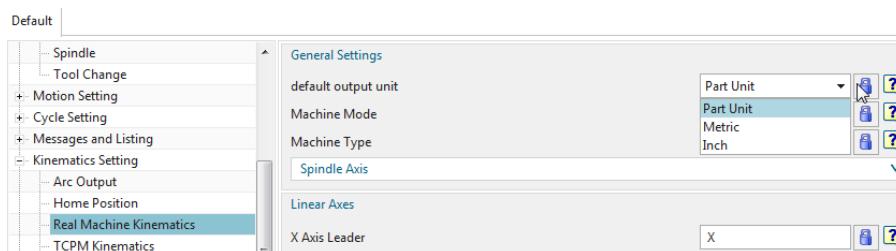
Post Configurator posts are unit independent. Post Builder post exists with inch and metric unit. The name of the library entry in the machine tool lists indicate which post is connected:  
 entries with xxx\_in, xxx\_mm are Post Builder based posts  
 entries without an suffix are Post Configurator based posts.



The screenshot shows a search results interface with a blue header bar containing a gear icon and the text "Search Result". Below the header is a "Class Description" section with the word "MACHINE". Underneath is a "Matching Items" section with a table. The table has two columns: "libref" and "Type". The "libref" column lists various simulation post processor names, and the "Type" column shows they are all "MDM0101".

libref	Type
sim08_mill_5ax_fanuc	MDM0101
sim09_mill_5ax_fanuc	MDM0101
sim11_turn_2ax_fanuc	MDM0201
sim12_turn_2ax_fanuc	MDM0201
sim14_mill_5ax_fanuc	MDM0101
sim01_mill_3ax_fanuc_mm	MDM0101
sim02_mill_3ax_fanuc_mm	MDM0101
sim03_mill_4ax_fanuc_mm	MDM0101
sim04_mill_4ax_fanuc_mm	MDM0101
sim05_mill_5ax_fanuc_mm	MDM0101
sim06_mill_5ax_fanuc_mm	MDM0101
sim07_mill_5ax_fanuc_mm	MDM0101
sim08_mill_5ax_fanuc_mm	MDM0101
sim09_mill_5ax_fanuc_mm	MDM0101
sim11_turn_2ax_fanuc_mm	MDM0201
sim12_turn_2ax_fanuc_mm	MDM0201
sim13_turn_4ax_fanuc_mm	MDM0201
sim14_mill_5ax_fanuc_mm	MDM0101
sim01_mill_3ax_fanuc_in	MDM0101
sim02_mill_3ax_fanuc_in	MDM0101
sim03_mill_4ax_fanuc_in	MDM0101

Default behavior of Post Configurator post processors is “Part Unit”, which means system will chose the unit of the uses CAM part file while posting. To change the unit in a Post Configurator post open the post and switch the property as shown in the figure.



## 7.2 General supported features of the OOTB Posts

All three major controllers support the basic functionality for 2 axis drilling, 3 axis, 3+2 axis and 5 axis milling and also some advanced functionality.

### 7.2.1 Basic functionality

- Addresses (X, Y, Z, A, B, C)
- Program control (M30, M0...)
- Basic motion type: rapid, linear, circular (G0,G1, G2 and G3)
- Metric/Inch units
- Radius/diameter programming
- Fixture offset (G53-G59 /G505.../G54.1 P1... etc )  
Coordinate system rotation(support Local CSYS and Auto3D way) (G68, TRANS/AROT)
- Tool radius compensation (G40, G41 and G42)
- Tool length compensation (G43/G43.1 for Fanuc)
- Feed rate, feedrate unit
- Spindle mode and spindle speed (M03, M04, M05, S)
- Constant surface speed
- Tool change
- TCP (TRAORI, G43.4, M128)
- Standard drilling cycles supported by NX. (G80-G89/CYCLE81-CYCLE89)
- Coolant (M08, M09)

### 7.2.2 Advanced functionality:

- Swiveling function (G68.2, CYCLE800, PLANE SPATIAL)
- Variable axis drilling (fully support cycle plane change)
- Turning cycle (CYCLE95, G71)

### 7.2.3 Sim15 Mill-turn (S840D and Fanuc)

- One mill-turn post
- Part transfer
- Polar mode
- Drilling/ Milling XZC output

### 7.2.4 Sim18 Turn-mill (S840D)

- One turn-mill post
- Multiple chains
- Multiple heads
- Crossrail positioning

### 7.2.5 Specific OOTB post functions

Specific functions supported:

- ini file will be generated by post for SINUMERIK controller including fixture offset values and tool information.
- Remove the tool at end of program for milling machine: example: T0 M06
- For 4 or 5 axis machine, rotary axis limit setting in postprocessor should be same as the machine model.
- For SINUMERIK machine, tool offset value D depends on the adjust register number setting.

- Fixture offset registers range  
SINUMERIK G54-G57 G505-G599  
Fanuc G54-G59 G54.1 P1 G54.1 P2....
- Fixture offset number in CAM setup will decide fixture offset output.
- For SINUMERIK machine, if number is between 1 and 4, corresponding output is G54-G57, if number is 5 output will be G505 and so on.
- For Fanuc machines, if number is between 1 and 6, corresponding output is G54-G59, if number is greater than 6, G54.1 Px will be output. X = number -6.
- Tool tip control
- TRAORI and M128 are similar function in Sinumerik and Heidenhain T530. They are both kinematics independent, means mom\_mcs\_goto should be output instead of mom\_pos for X Y Z position.  
But for OOTB Fanuc examples, G43.4 only has capability to compensate tool axis length in variable axes milling operations. G43.1 should be used in fixed axes milling with head rotation. G43 is used in all operations with Z orientation spindle.

### 7.3 Creating file with tool and offset data

The SINUMERIK 840D post processor creates a CAM setup specific initialization data file (to\_ini.ini or to\_ini\_Channel.ini) for work offset values and the tool data. This file will be located in the *cse\_files/subprog* subfolder of the CAM setup part:

```
...MACH/samples/nc_simulation_samples/cse_files/subprog
```

This to\_ini.ini file is handled as a subprogram and in the ini files of the machine tool in the library this file gets called (executed and loaded). When working with CSE controller models, the different ini files are loaded and executed before the simulation starts. The format and syntax of the files is controller specific and their purpose is to initialize certain settings upfront. For more details about ini files please see [Appendix](#).

Before the post creates a new to\_ini.ini file it will back up the existing ini file by renaming it to “\*.bck”.

With NX11.0.2 the array index number of the tool data written into the Sinumerik to\_ini file use an ascending numbering to improve supporting the use case of NX CAM tool numbers higher than 32000. So the new format will look like:

\$TC_TP1[1]=32001	Array index always starts at 1
\$TC_TP2[1]="UGT0201_088"	

#### 7.3.1 Machine zero junction

The post query the kinematics model using the TCL query function **MOM\_ask\_init\_junction\_xform** and store the results in the variable **mom\_sim\_result[9]**, **mom\_sim\_result1[3]** to get the data of the machine zero junction needed to calculate offset information for e.g. G54.

#### 7.3.2 Call the post created TO\_INI

The OOTB example machine tool ini files for Sinumerik (e.g. sim07\_mill\_5ax-Main.ini) have the entry TO\_INI to call the post created file as a subprogram. If the creation of the file not work properly or the file gets deleted or the post does not have write access to create this file and the CSE Simulation starts it

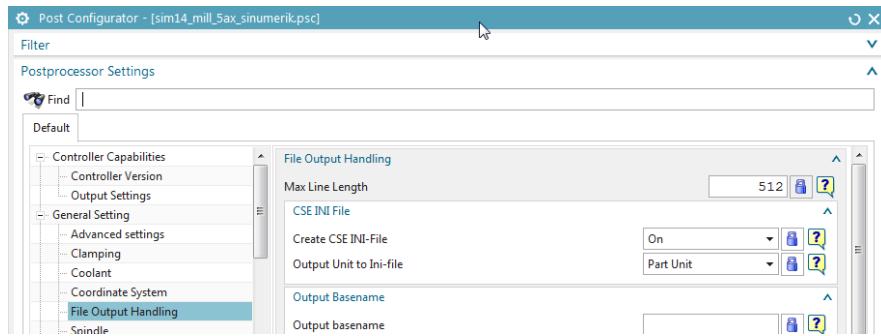
will give you a warning in the details window, that the related TO\_INI file could not be found. The simulation will not stop in that case.

### 7.3.3 To\_ini file location

The location of ini file will be reset to directory which defined by UGII\_CAM\_CSE\_USER\_DIR, if this environment variable is already set up in system.

The values inside to\_ini.ini are in the same unit as the ones inside the main program. Make sure Post processor and CSE driver are from the same NX version.

### 7.3.4 Create to\_ini with Post Configurator



Created posts with Post Configurator have the option to switch on/off easily the creation of to\_ini file.

### 7.3.5 Create to\_ini with Postbuilder

```
Custom Command
proc PB_CMD_int_ini_files {} {
    # This procedure is used with simulation_ini.tcl file which should be sourced
    # in postprocessor as a user tcl file for generating ini files.
    # This procedure will be excuted when simulation_ini.tcl is sourcing.
    #
    # Following mom variables are used to set the options for ini files.
    # mom_sinumerik_ini_create
    #   "Yes"      ini files will be created
    #   "No"       ini files will not be created
    #
    # mom_sinumerik_ini_location
    #   "Part"    ini files will be output to a subfolder \cse_files
    #             of cam part file location. If \cse_files is not exists
    #             post will create this folder.
    #   "CSE"     ini files will be output to a \cse_driver folder which
    #             located in same directory of \postprocessor as machine kits.
    #   "ENV"     If it is not exists, ini files will be output to "Part Setup".
    #             ini files will be output to a enviornment variable defined
    #             folder.
    #
    # mom_sinumerik_ini_existing
    #   "Rename"  Rename existing ini files in place where the one is created.
    #   "Keep"    Keep ini files in place where the one is created.
    #   "Delete"  Delete ini files in place where the one is created.
    #
    # mom_sinumerik_ini_end_status
    #   "Rename"  Rename created ini file after post run.
    #   "Keep"    Keep created ini files after post run.
    #   "Delete"  Delete created ini files after post run.

    global mom_sinumerik_ini_create
    global mom_sinumerik_ini_location
    global mom_sinumerik_ini_keep_existing
    global mom_sinumerik_ini_end_status

    set mom_sinumerik_ini_create      "Yes"
    set mom_sinumerik_ini_location   "Part"
    set mom_sinumerik_ini_existing   "Rename"
    set mom_sinumerik_ini_end_status "Keep"
}
```

## 7.4 MOM variable for MCS handling

For coordinate rotation NC codes output inside the post, the related coordinate matrix is changed. Reason for this is because the Main MCS is representing Machine zero and Local fixture offset representing machining coordinate G54, G55..., but mom\_csys\_origin and mom\_csys\_matrix still map current local MCS to Main MCS. For CSYS rotation coordinate output, the value of linear offset (G68, CYCLE800 ...) should be the offset between current local MCS to parent local MCS(e.g. G54,G55). Therefore, mom\_parent\_csys\_matrix replaced mom\_csys\_matrix in postprocessor.

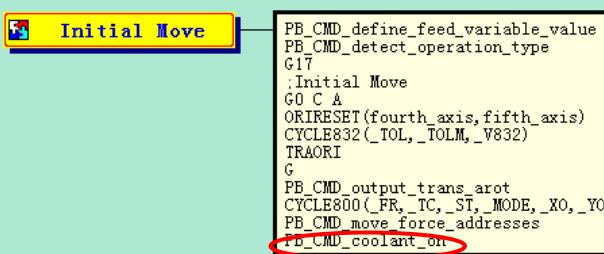
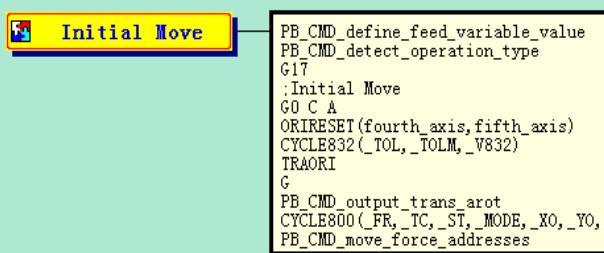


## 7.5 Differences between 'pure' template posts and OOTB posts (Sinumerik)

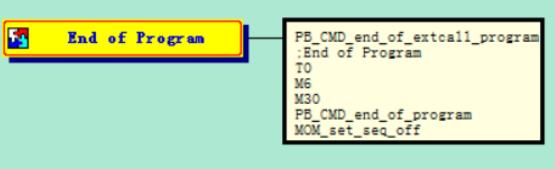
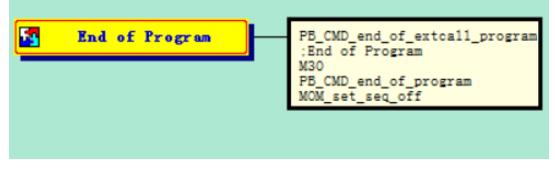
The OOTB posts are created based on the latest Sinumerik S840D basic template posts. Some settings and or features are slightly different, which are listed here.

### 7.5.1 Coolant

Coolant output is different.

Posts of OOTB Examples	Post from template post
Coolant code always be M8 , except "Deep Hole Drilling" operation.  	Coolant code depends on coolant status  

### 7.5.2 Output of "T0 M6" at end of program

Posts of OOTB Examples	Post from template post
Output T0 M6 at end of program  	No T0 M6 at end of program  

## 7.6 About UDEs and OOTB

The post shipped with the OOTB machine tools in the library are coming with their own UDEs (CDL file), which provides a series of UDE which all are supported.

## 7.7 Post output for Coolant (Sinumerik, TNC, Fanuc)

OOTB post processors only support M8 (coolant on) and M9 (coolant off), to output other coolant codes, post processors need to be customized.

## 7.8 About Special handling for 5 axis motions with Fanuc for table machine tool configuration

The OOTB setting for Fanuc controller will not move the programming system together with the table when doing 5 axis motions like G43.X. The post takes care about that when outputting the X/Y/Z data. The CSE controller takes care about that by using a specific mode when activating the 5 axis transformation. This mode can be changed:

In Post do the following:

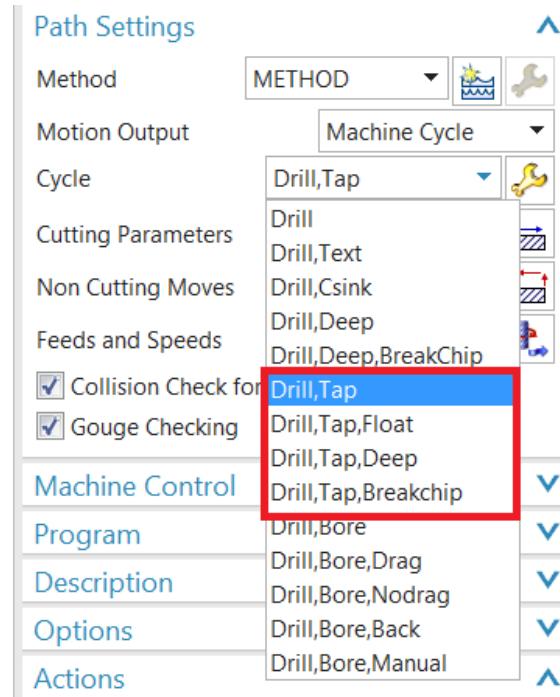
Change variable dpp\_ge(sys\_output\_coord\_mode) value to “TCP\_FIX\_MACHINE” in command PB\_CMD\_customize\_output\_mode

In Simulation do the following:

Change the setting of the variable #19696 in the INI file to =0 (default is 32 for non-rotating workpiece coordinate system)

## 7.9 Parameter Setting for new tap cycles

In the NX version, cycle type “Drill, Tap, Float”, “Drill, Tap, Deep”, and “Drill, Tap, Breakchip” are introduced. Please see the picture below. So accordingly, some enhancements are implemented in OOTB post processor. The detail will be showed in the content below.



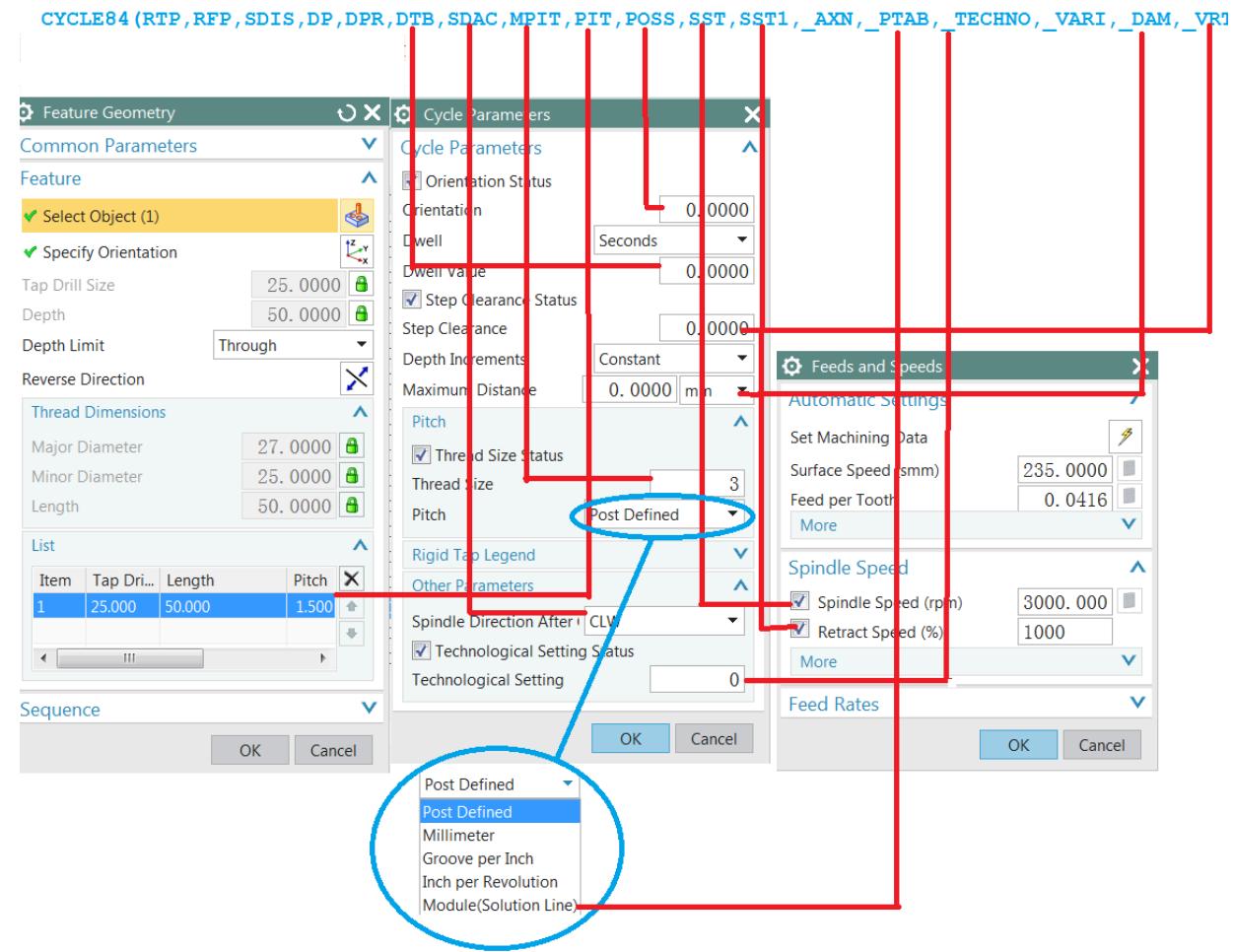
The table below shows the tap cycle of three kinds of controller.

	Tap cycle	Tap	Tap Float	Tap Deep	Tap Breakchip
Controller					
S840D	CYCLE84(PTP,...VARI,...) <b>NOTE:VARI=0</b>	CYCLE840(xx,...)	CYCLE84(PTP,...VARI,...) <b>NOTE:VARI=2</b>	CYCLE84(PTP,...VARI,...) <b>NOTE:VARI=1</b>	
Fanuc	M29 S G84.2/G84.3X Y Z R P F	G84/G74X Y Z R P F	M29 S G84.2/G84.3 X Y Z R P Q F <b>NOTE: PCP(No.5200)=0</b>	M29 S G84.2/G84.3 X Y Z R P Q F <b>NOTE: PCP(No.5200)=1</b>	
ITNC	CYCL DEF 207 Q200 Q201 Q239 Q203 Q204	CYCL DEF 206 Q200 Q201 Q206 Q211 Q203 Q204	CYCL DEF 209 Q200 Q201 Q239 Q203 Q204 Q257 Q256 Q336 <b>NOTE:Q256=0</b>	CYCL DEF 209 Q200 Q201 Q239 Q203 Q204 Q257 Q256 Q336 <b>NOTE:Q256 !=0</b>	

**NOTE:** In OOTB Fanuc post, it uses tap cycle of series 15 format.

### 7.9.1 Set CYCLE84 parameter for S840D in UI

Take tap break chip UI as an example for Powerline CYCLE84. Please see the picture below.

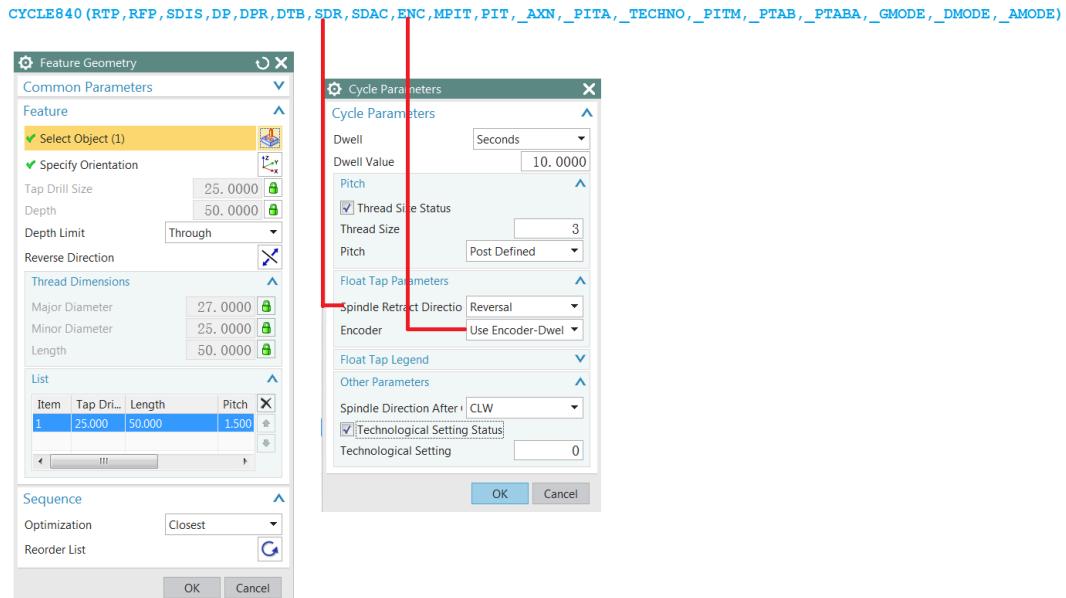


**NOTE:** Except parameters mentioned above, other parameters information is showed in the table below.

CYCLE84 Parameter	Comment
<b>RTP</b>	Retract plane
<b>RFP</b>	Reference plane
<b>SDIS</b>	Safety clearance
<b>DP</b>	Final drilling depth
<b>DPR</b>	Final drilling depth relative to the reference plane. It is always set to none in the OOTB.
<b>_AXN</b>	Tool axis. It is always set to none in the OOTB.
<b>MPIT</b>	If it is set, then PIT will not output.
<b>_PTAB(solutionline)</b>	Its setting is the same as <b>_PTAB</b> .
<b>PITM(solutionline)</b>	It is set to 0 in the OOTB.
<b>_PTABA(solutionline)</b>	It is set to 0 in the OOTB.
<b>_GMODE(solutionline)</b>	It is set to none in the OOTB.
<b>_DMODE(solutionline)</b>	It is relative with spindle axis, the value may be 1, 2 or 3.
<b>_AMODE(solutionline)</b>	It is set to 0 in the OOTB.

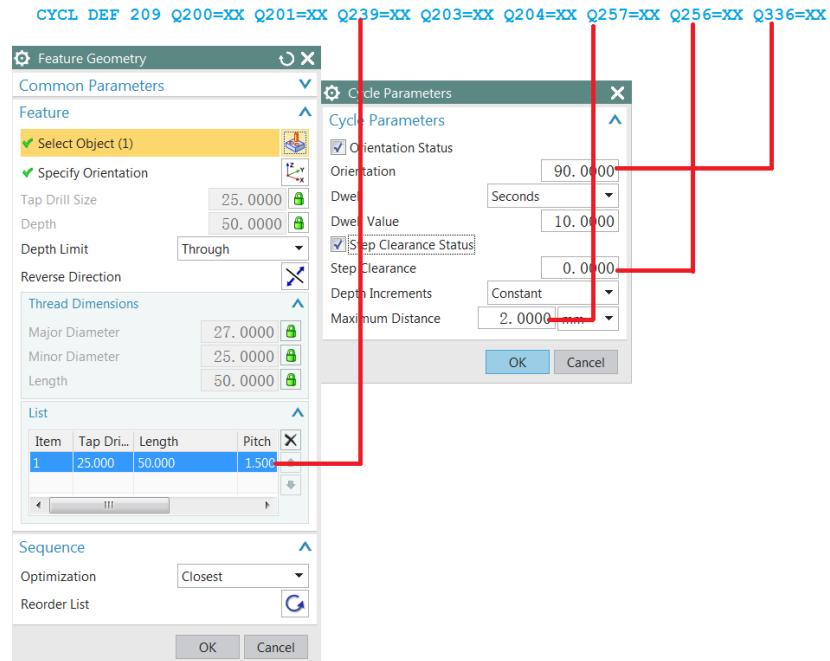
### 7.9.2 Set CYCLE840 parameter for S840D in UI

Except parameter SDR and ENC, other parameter settings are the same as CYCLE84. Please see the SDR and ENC setting in the picture below.



### 7.9.3 Set CYCLE209 parameter for ITNC in UI

Take tap breakchip UI as an example for ITNC. Please see the picture below.



**NOTE:** Except parameters mentioned above, other parameters information is showed in the table below.

CYCLE 209 Parameter	Comment
<b>Q200</b>	Distance between tool tip (at starting position) and workpiece surface.
<b>Q201</b>	Distance between workpiece surface and end of thread.
<b>Q203</b>	Coordinate of the workpiece surface.
<b>Q204</b>	Coordinate in the tool axis at which no collision between tool and workpiece (clamping devices) can occur.

#### 7.9.4 Set CYCLE207 parameter for ITNC in UI

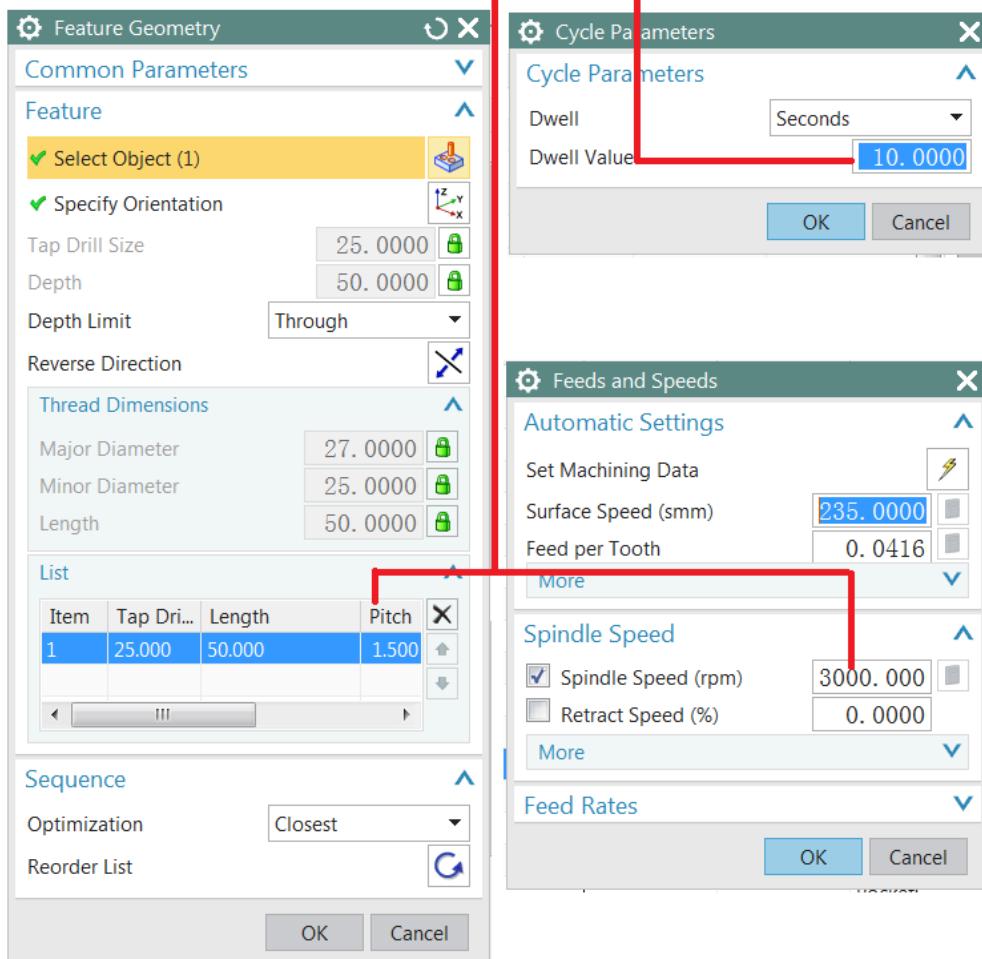
CYCL DEF207 Q200=XX Q201=XX Q239=XX Q203=XX Q204=XX

CYCLE207 parameters are similar to CYCLE209, so the parameters setting are the same as CYCLE209.

#### 7.9.5 Set CYCLE206 parameter for ITNC in UI

Except parameter Q206 and Q211 of CYCLE206, other parameters setting are the same as the CYCLE209. Please see the picture below.

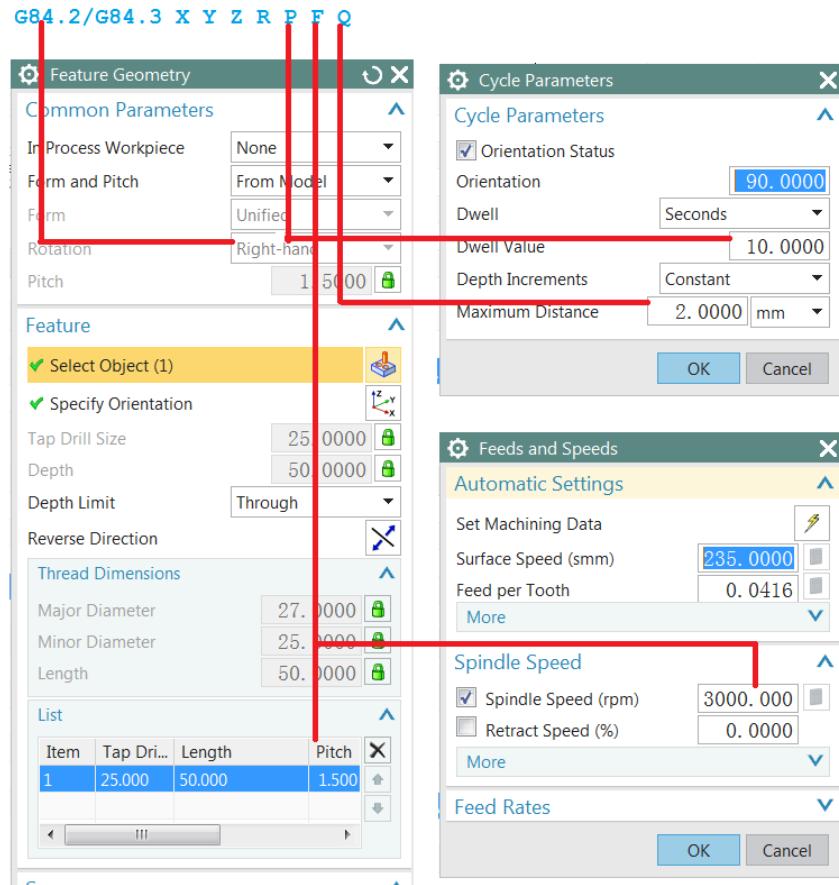
CYCL DEF 206 Q200=XX Q201=XX Q206=XX Q211=XX Q203=XX Q204=XX



**NOTE:** Q206 is decided by pitch and spindle speed.

### 7.9.6 Set G84.2/84.3 parameter for Fanuc in UI

Take tap breakchip UI as an example for Fanuc. Please see the picture below.



**NOTE:** G84.2 and G84.3 are decided by “Rotation” value, if it is “Right-hand”, then it outputs G84.2, or it outputs G84.3. And parameter F is decided by pitch and spindle speed. Except parameters mentioned above, other parameters information is showed in the table below.

G84.2/G84.3 Parameter	Comment
X Y	Hole position data.
Z	The distance from point R to the bottom of the hole and the position of the bottom of the hole.
R	The distance from the initial level to point R level.

**NOTE:** G84/G74 parameters are similar to G84.2/G84.3, so G84/74 parameters setting are the same as G84.2/84.3.

## 8 Using the library tools

The CAM setup simulation examples use as many as possible tools available from the OOTB tool library (ASCII in NX native) as delivered with the actual NX version. It is not suggested to adding part files with the existing tool library entry names. (e.g. ugt010101\_003.prt). It may happen that the added 3D tool does not match the geometry created based on the parameters.

## 9 About CSE Simulation Drivers

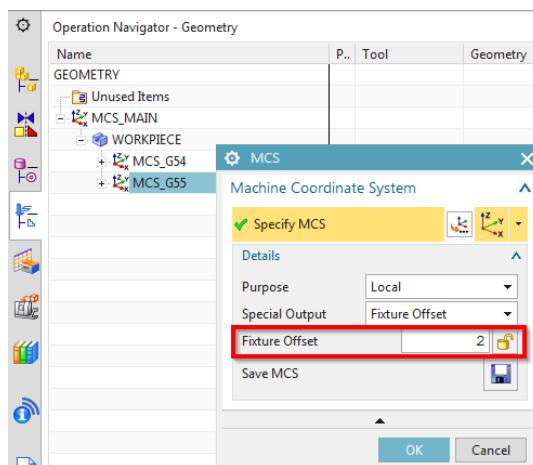
### 9.1 Handling of Offsets:

Like any physical NC controller, CSE drivers can process offset information (activated e.g. through G55) only if the required data is provided, i.e. coordinate values between the actual chosen offset e.g. G55 and the machine tool zero position. There are two ways to achieve this with CSE and both are used in the OOTB examples.

#### 9.1.1 Alternative 1 (used by Fanuc CSE controller)

In the case of Fanuc, the controller queries the information during simulation from the application (ISV NX). This is done internally by a command called “*LoadOffset*” during interpretation of an offset statement in the NC code like G55. The system searched in the ONT for an MCS object with the related fixture offset number.

Example: In the NC code a G55 (2<sup>nd</sup> offset) is used. In this case the system cycles through all MCS objects and compares the value use in “Fixture Offset” with the given offset in the NC code.



#### 9.1.2 Alternative 2 (used by Sinumerik and TNC CSE controller)

The SINUMERIK 840D CSE and the TNC controller doesn't use the “*LoadOffset*” mechanism as for the FANUC controller model, but follows an alternative implementation. On the physical controller in the shop floor, the offset values are typically measured by probing operations or are manually set in the controller. The controller stores these values in an offset table and in controller variables.

The SINUMERIK OOTB post processors creates an initialization file for the actual CAM setup including information about the offsets and tool data. This initialization file is loaded before the simulation starts. So the offset definition is achieved by the definition of the variables in the initialization file.

Snapshot of the SINUMERIK ini file:

```

#H:\kit\mach\samples\nc_simulation_samples\cse_files\subprog\to_ini.ini - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
H:\kit\mach\samples\nc_simulation_samples\cse_files\subprog\to_ini.ini
1
2 $P_UIFR[0]=CTRANS(X,0.0,Y,0.0,Z,0.0)
3 $P_UIFR[1]=CTRANS(X,0.000000,Y,-1994.602958,Z,-846.713943)

```

Later, when one of the offsets is activated in the NC code, the CSE controller uses these variables to define the offset transformation.

The TNC OOTB post processors outputs the offset data into the main program based on the actual CAM setup. The offset gets activated right away by parsing these NC code lines.

Example for TNC offset data:

```

...
8 CYCL DEF 7.0
9 CYCL DEF 7.1 X -0.0000
10 CYCL DEF 7.2 Y -225.0000
11 CYCL DEF 7.3 Z -327.4470
...

```

**Note:** Working with CYCLE247 and/or is possible with CSE, but not part of the OOTB examples.

## 9.2 Handling of the tool change

Tool changes with CSE controller models in the existing examples are achieved by calling a tool change subprogram. This subprogram is located under the “subprog” folder for each CSE driver; it’s kept in the corresponding NC code syntax and basically positions the tool to the tool change location. Using one of these subprograms for a different machine tool will typically require an adjustment of the tool change position. Another section in the tool change subprogram is the Anycontroller (AC) part, which mimics the PLC portion of the tool change. This mainly takes care of the mounting and un-mounting the tool itself. Refer to NX Help for more details about the AC language. The OOTB examples demonstrate different kinds of tool change methods.

- The “standard” way is that one spindle is defined on the kinematics model of the milling machine tool and the tools are mounted ‘on the fly’ during simulation. This can be seen in sim02 to sim09 as well as sim14.
- sim08 is similar, but shows how an advanced tool change mechanism can be animated using the AC language to open and close doors.

- In sim01, tools are already pre-mounted and visible in the CAM scenario on an eight pocket tool changer. During simulation, the system moves the spindle and the tool changer to mount and un-mount the tool.
- The turning examples sim10 to sim13 don't mount tools during simulation. Here, all tools are already pre-mounted on the turret in the CAM scenario. The tool change subprogram takes care of the rotation of the turret and the activation of the selected tool.

**Note:** *The machine tool view in the ONT reflects the definition of turrets and pockets in the kinematics model. Each time the machine tool is retrieved from library, the ONT is updated based on the kinematics model. It is strongly recommended not to add or remove turrets and pockets in the ONT machine tool view when working with machine tools.*

### 9.3 Handling the reference point (Fanuc)

In Fanuc NC code, the G command G28 is often used to move the reference point. The OOTB examples include this in the post and in the CSE controller models. This section describes how the position of the reference point is defined and stored to get correct simulation results.

On physical machine tools, the reference point is stored inside of the controller. To mimic this in the CSE simulation, the position of the reference point is defined in the \*.ini file along with the machine tool.

The NC code to define reference points in Fanuc syntax is shown in the sim01 example (sim01\_mill\_3ax\_fanuc-Main.ini) as:

```
(This part sets the position of the reference point G28)
G54
G17
G90
G10I52
N1240P1R0000 -> X Position of the reference point related to machine zero
N1240P2R225.425 -> Y Position of the reference point related to machine zero
N1240P3R406.425 -> Z Position of the reference point related to machine zero
G11
```

If needed, the values of the reference point can easily be changed in the ini file. All coordinates are assumed to be metric.

### 9.4 Handling of tool correction data

Handling tool data CSE is able to get the data on different ways. One way is a direct connection to the application ISV and query the data through an API function during simulation e.g. when activating the tool correction e.g. Dxx or G43 Hxx).

An alternative way is to rely on data predefined in controller variables e.g. in an ini file. The behavior can be triggered in the OOTB CCF by setting the global variable GV\_bUseSetToolCorrection in the CSInitializeChannel. Most controller use the first described way and get the tool correction data via the API during simulation form ISV.

## 9.5 About Sinumerik Cycles in content

The OOTB library for Sinumerik supports list of cycles for simulation. The cycles are covered by an encrypted archive file (\*.cyc) and placed in the subprog folder of the Sinumerik CSE simulation. The cycles of Operate Version 4.7 SP2 HF1 are in the file: *SinumerikSL\_Cycles\_47\_SP2\_HF1.cyc*

- The cycles of the Solutionline Version 4.4 are in the file: *SinumerikSL\_Cycles.cyc*  
*shipped since Version NX85*
- The cycles of the Solutionline Version 2.6 are in the file: *SinumerikSL\_Cycles.cycV26*  
*shipped until Version NX85*
- The cycles of the Powerline are in the file *SinumerikPL\_Cycles.cyc\_powerline*  
shipped until Version NX85

To use the cycle other than form the default rename the files so that the desired one has the extension \*.cyc. The cyc archive file includes the following cycles:

Solutionline/Operate

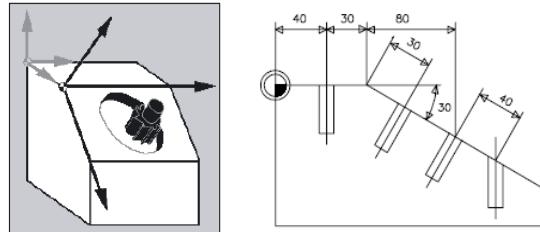
- CYCLE71
- CYCLE81 – CYCLE90
- CYCLE801
- HOLES1, HOLES2
- LONGHOLE
- SLOT1, SLOT2
- POCKET3, POCKET4
- CYCLE95
- CYCLE97
- CYCLE800

As an example the file *Demo\_sinumerik\_Cycles\_Setup.prt* can be simulated with the external NC code *Demo\_sinumerik\_Cycles.MPF*. These files are located at:

*\$UGII\_CAM\_BASE\_DIR\mach\samples\nc\_simulation\_samples\Demo\_sinumerik\_Cycles*

## 10 About swiveling cycles

This will give an overview about what are swiveling cycles and how they are used. The main target is to machine on a plane not perpendicular to an existing linear axis.



Working with swiveling cycles will make the life easier and let the controller do the work. In the following subchapters it is explained in detail how the swiveling Cycle PLANE SPATIAL/AXIAL and CYCLE19 on Heidenhain, CYCLE800 on S840D and G68.2 on Fanuc controllers work and how this is implemented in the CSE and the OOTB examples. In addition you will see a section how this is configured and can be reused for a different machine tool.

### 10.1 Swiveling on Heidenhain – PLANE SPATIAL

#### 10.1.1 Settings in Operation Navigator (ONT) to achieve a proper Post output

To achieve an output of PLANE SPATIAL by the Post the following prerequisites needs to be fulfilled. (PLANE SPATIAL is supported by OOTB post; PLANE AXIAL and CYCLE19 are supported in the CCF for simulation only)

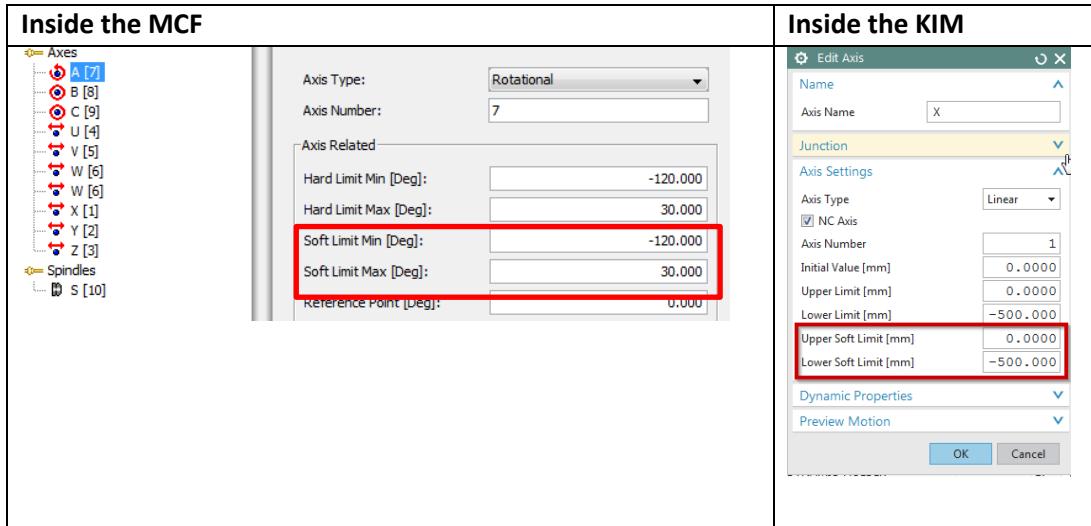
- milling operations and drilling operations needs to be 3+2 axis operation
- The MCS need to be set to “CSYS Rotation”

#### 10.1.2 Settings in CSE driver to achieve correct simulation

The settings are valid for all supported modes which are PLANE SPATIAL, PLANE AXIAL and CYCLE19. Inside the MCF global variable in Internal variables tab must be set related to the type of the machine tool and the axis vector directions. Name of the variables are:

- GV\_strMachineType (“T”, “M”, “P”)
  - „T“ for Tool → Head/Head
  - “P” for Part → Table/Table
  - “M” for Mixed → Head/Table
- GV\_strSwivelingChainName
  - The default name for the chain will be “default”
- GV\_strFourthAxisName
- GV\_strFifthAxisName
- GV\_dFourthAxisX (x component of fourth axis)
- GV\_dFourthAxisY (y component of fourth axis)
- GV\_dFourthAxisZ (z component of fourth axis)
- GV\_dFifthAxisX (x component of fifth axis)
- GV\_dFifthAxisY (y component of fifth axis)
- GV\_dFifthAxisZ (z component of fifth axis)

And the soft min/max limit of the rotary axes should be set if you wish rotary axis solution can be selected based on axes limit.

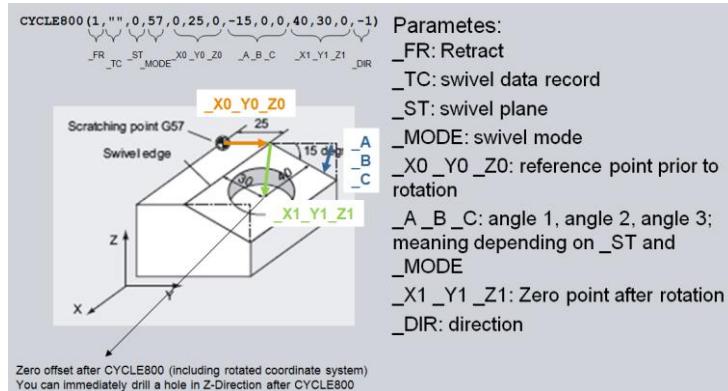


More details about the cycle and the way how it is implemented in CSE can be found in the [Appendix](#)

## 10.2 Swiveling on Sinumerik - CYCLE800

The Sinumerik Solutionline and Operate version is implemented in this approach. Much more details about the cycle itself and the way how it is implemented in CSE can be found in the [Appendix](#)

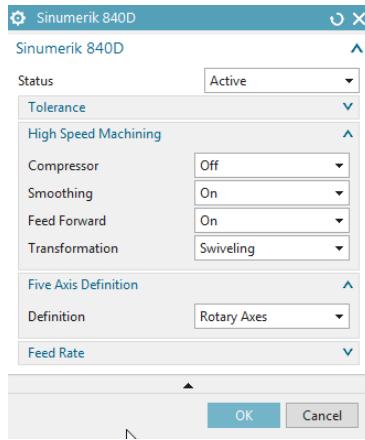
### 10.2.1 Example and parameter



### 10.2.2 Settings in Operation Navigator (ONT) to achieve a proper Post output

To achieve an output of CYCLE800 by the Post the following prerequisites needs to be fulfilled.

- milling operations and drilling operations needs to be 3+2 axis operation
- Sinumerik 840 UDE should be added on the operations, Transformation option should choose "Swiveling" as shown below:



### 10.2.3 Settings in CSE driver to achieve correct simulation

This is the list of variables defined for each machine tool in the machines ini file and others files to define data for the CYCLE800. These are the data, which need to match the related machine tool.

- Content of the OOTBINI file for NCK4.4. cycles

```
SMAC; adopt the original DEF file from MC
PGUD; need remove the line with REDEF
TC_CARR; define tool holder data
G40 D0
$MN_MM_NUM_TOOL_CARRIER=1
M17
```

- Content of the OOTBINI file for NCK4.7. cycles

```
PMAC; adopt the original DEF file from MC
PGUD; need remove the line with REDEF
TC_CARR; define tool holder data
CHAN_DATA
G40 D0
M17
```

- Definition data of the TC variables:

The following \$tc\_carr variables are used from CSE for cycle800 calculation

```
$TC_CARR7      ;x component of rotary axis v1
$TC_CARR8      ;y component of rotary axis v1
$TC_CARR9      ;z component of rotary axis v1
$TC_CARR10     ;x component of rotary axis v2
$TC_CARR11     ;y component of rotary axis v2
$TC_CARR12     ;z component of rotary axis v2
$TC_CARR23*    ;kinematic type
$TC_CARR24     ;Offset of rotary axis v1
$TC_CARR25     ;Offset of rotary axis v2
$TC_CARR30*    ;software minimum limit of rotary axes v1
$TC_CARR31*    ;software minimum limit of rotary axes v2
$TC_CARR32*    ;software maximum limit of rotary axes v1
$TC_CARR33*    ;software minimum limit of rotary axes v2
$TC_CARR34     ;tool holder name
$TC_CARR35*    ;Axis name 1
$TC_CARR36*    ;Axis name 2
$TC_CARR37     ;Identifier
$TC_CARR40     ;Z axis retract value
```

\*With global variable "GV\_bSetToolCarrierDataFromKim" the user can decide whether the local toolcarrier dataset should be used or the information should be retrieved from the current active kinematic chain and axis information from kinematic model. Default value in CCF file is TRUE.

## 10.3 Swiveling in Fanuc - G68.2

### 10.3.1 Settings in Operation Navigator (ONT) to achieve a proper Post output

To achieve an output of G68.2 by the Post the following prerequisites needs to be fulfilled.

- Milling operations and drilling operations needs to be 3+2 axis operation
- The MCS need to be set to "CSYS Rotation"

### 10.3.2 Settings in CSE driver to achieve correct simulation

The requirement for G68.2 is the same to the requirement for PLANE SPATIAL. Please refer to previous chapter. More details about the cycle and the way how it is implemented in CSE can be found in the [Appendix](#)

## 11 Appendix

### A. Detailed information about ini files

A setup-specificINI file could contain, for example

- a list of setup-specific tools and tool count
- initial machine positions
- default fixture offset register
- inch or metric unit selection

The ISV CSE system will follow a strict search order to load ini files. Three different locations are supported. Here an overview of that:

	<b>Machine Tool Library</b> Folder with the MCF	Folder with <b>CAM Part</b> file	ENV <b>UGII_CAM_CSE_USER_DIR</b>
Search/Load Order	1.	2.	3.
Look for ini files *-channelname.ini	../	../CSE_files	../
Example	sim07_mill_5ax-Main.ini	to_ini-1.ini	
Look for subprograms	../subprog	../CSE_files/subprog	../subprog

### B. Swiveling Cycles

All three CCFs for TNC, Sinumerik and Fanuc include one swivel cycle implementation for the related NC code PLANE SPATIAL, TCARR/PAROT and G68.2. The way how that is implemented is consolidate over the different controllers and follow one general workflow. Here the details of that workflow:

- Initialize the system/global variables to define chain kinematics.
- Apply work piece coordinate rotation (and translation) to transformation (WPFRAME for Sinumerik, ROTATIONAL for Fanuc, PLANE for TNC).
- Calculate the tool vector after work piece coordinate rotation.
- Calculate tool angle and part rotating angle with rotated tool vector.
- Select the tool and part rotating angle based on the restriction (SEQ, limitation).
- Compensate the linear offset caused by tool and part rotating.
- Compensate the rotating offset caused by table rotating if applicable (for HeadTable and TableTable machine).
- Move and rotate head and/or table.

## C. PLANE SPATIAL

Inside the used MCF global variable are set related to the type of the machine tool and the axis vector directions. Name of the variables are:

- GV\_strMachineType ("T", "M", "P")
- GV\_strFourthAxisName (fourth axis name)
- GV\_strFifthAxisName  
(fifth axis name – is dependent on forth axis for T and P; for M it is the one near to the part)
- GV\_dFourthAxisX (x component of forth axis)
- GV\_dFourthAxisY (y component of forth axis)
- GV\_dFourthAxisZ (z component of forth axis)
- GV\_dFifthAxisX (x component of fifth axis)
- GV\_dFifthAxisY (y component of fifth axis)
- GV\_dFifthAxisZ (z component of fifth axis)
- GV\_dFourthAxisLimitMin (fourth axis limit min)
- GV\_dFourthAxisLimitMax (fourth axis limit max)
- GV\_dFifthAxisLimitMin (fifth axis limit min)
- GV\_dFifthAxisLimitMax (fifth axis limit max)

Here an example how that will look like in the case of the complex DMU EVO machine tool

- GV\_strMachineType: "P"
- GV\_strFourthAxisName: "B"
- GV\_strFifthAxisName: "C"
- GV\_dFourthAxisX 0.584825
- GV\_dFourthAxisY 0.573576
- GV\_dFourthAxisZ 0.573576
- GV\_dFifthAxisX 0.0
- GV\_dFifthAxisY 0.0
- GV\_dFifthAxisZ 1.0
- GV\_dFourthAxisLimitMin 9999
- GV\_dFourthAxisLimitMax 9999
- GV\_dFifthAxisLimitMin 9999
- GV\_dFifthAxisLimitMax 9999

### About SEQ+/- parameter in PLANE SPATIAL

Choose the shortest way as the preferred solutions. Implement this in the CCF file. First check if both solutions are achievable – due to limits of the axis. If neither solution is within traverse range, error message will be generated. This mechanism uses the axis limits set in the MCF file.

### About TABLE ROT/COORD ROT in PLANE SPATIAL

Transformation mode is an optional parameter for Plane function and TABLE ROT. This takes effect only if the configuration is TABLE ROT, machine type head-head and SPC>0 (Q122=0)

- Workflow of internal PLANE SPATIAL implementation
- Apply SPA SPB SPC rotation to PLANE
- Calculate the tool vector with SPA SPB SPC rotation

- Generate possible tool and part angle solution using calculateIKSAngles
- Get valid tool and part angle solution via GMe\_SwivelingGetRotateSolution
- Set Q120 Q121 Q122 accordingly
- Set linear compensation due to tool holder rotation via GMe\_SwivelingCalculateLinears
- Align WCS on table rotation via GMe\_SwivelingCompTableRotation
- Handle TABLE ROT exception
- Move rotary axis with meta code MOVE/TURN/STAY and ABST.

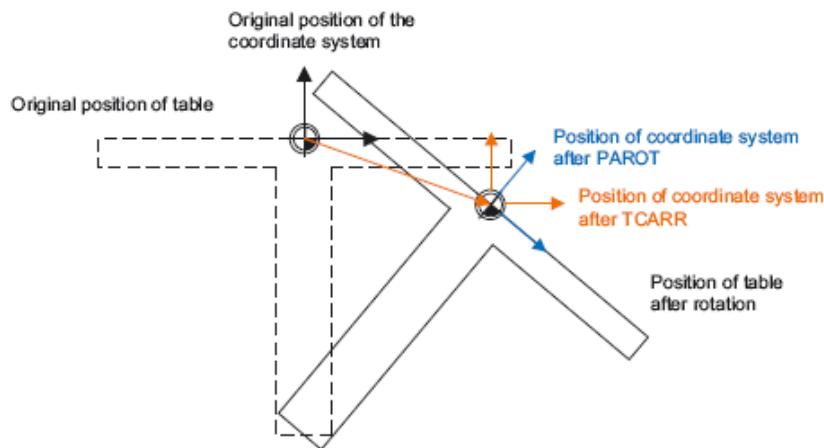
## D. CYCLE800

### Basic about TCARR and PAROT

Cycle 800 using commands TCARR and PAROT. TCARR calculates rotational axis angle and calculates the translation compensation based on current kinematics, and apply it to the according transformation. The command PAROT rotates the transformation “PARTFRAME” for the table.

Depending on the mode of \$P\_GG[42] (TCOABS, TCOFR) the compensation is calculated in TCARR.

- With TCOABS  
Based on the angles defined in TC\_CARR13 and TC\_CARR14
- With TCOFR  
Reads the current orientation from WPFRAME and calculates (with IKS) 2solutions per rotary axis

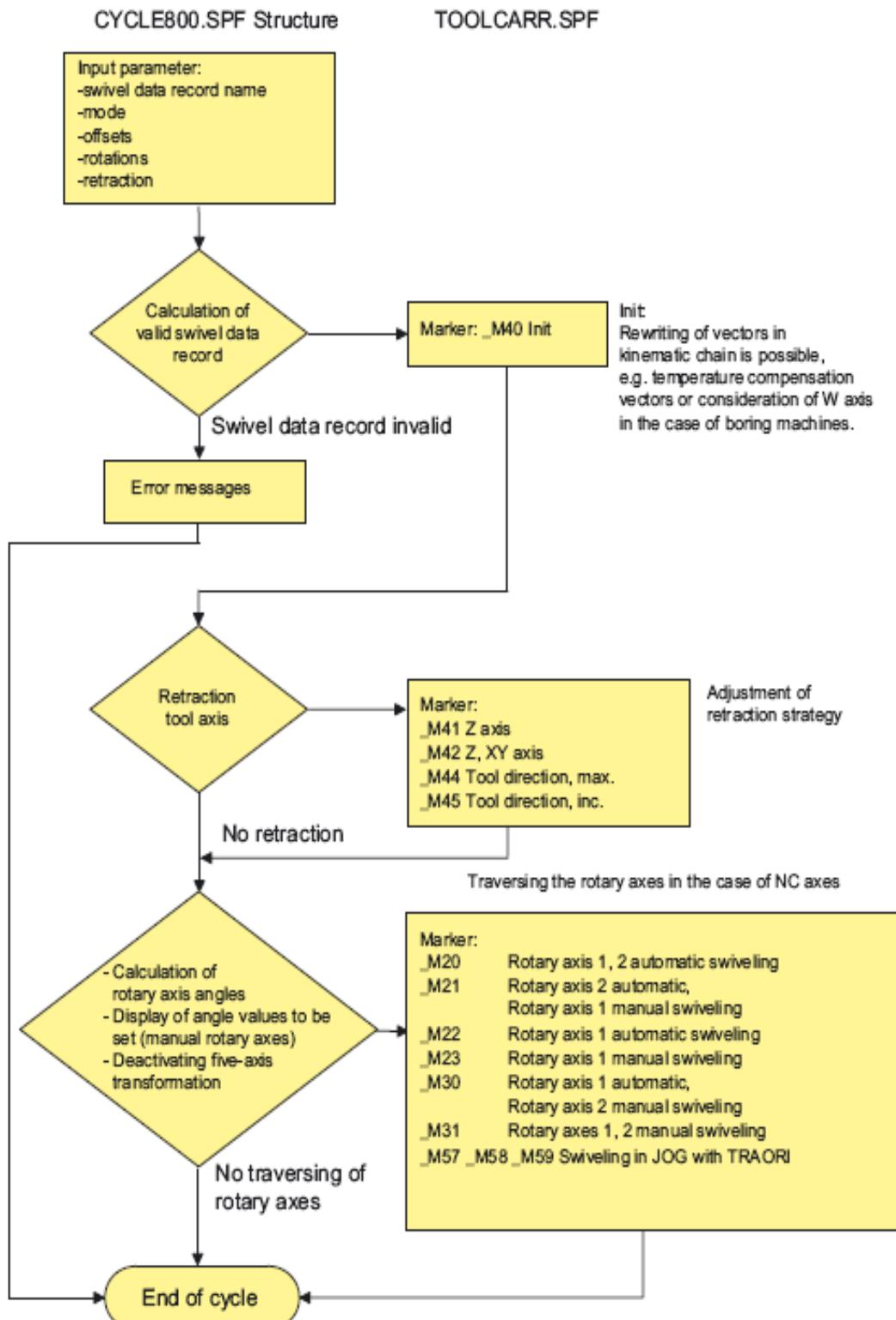


### Basic Workflow

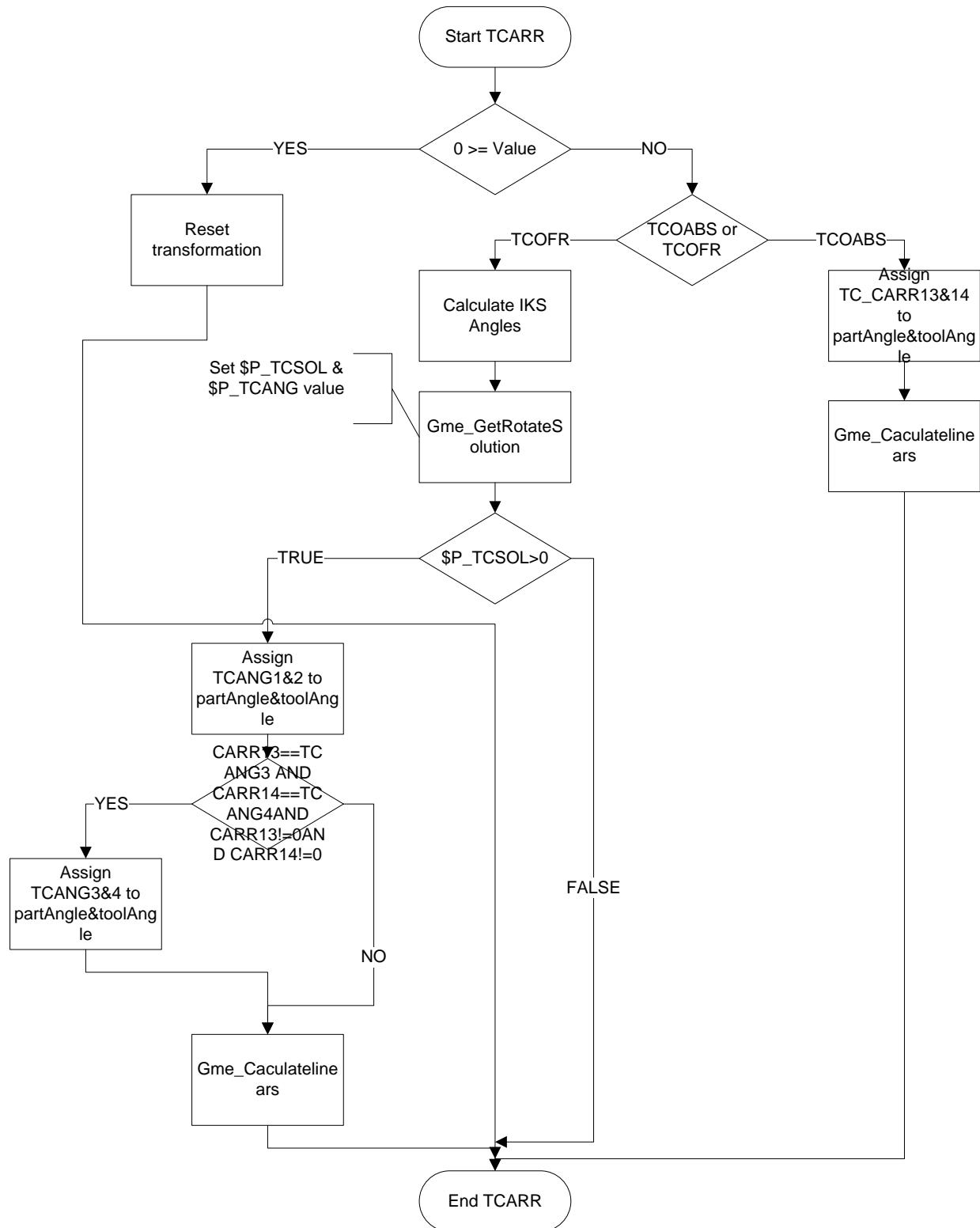
The CYCLE800.SPF consists of two subprograms, which are:

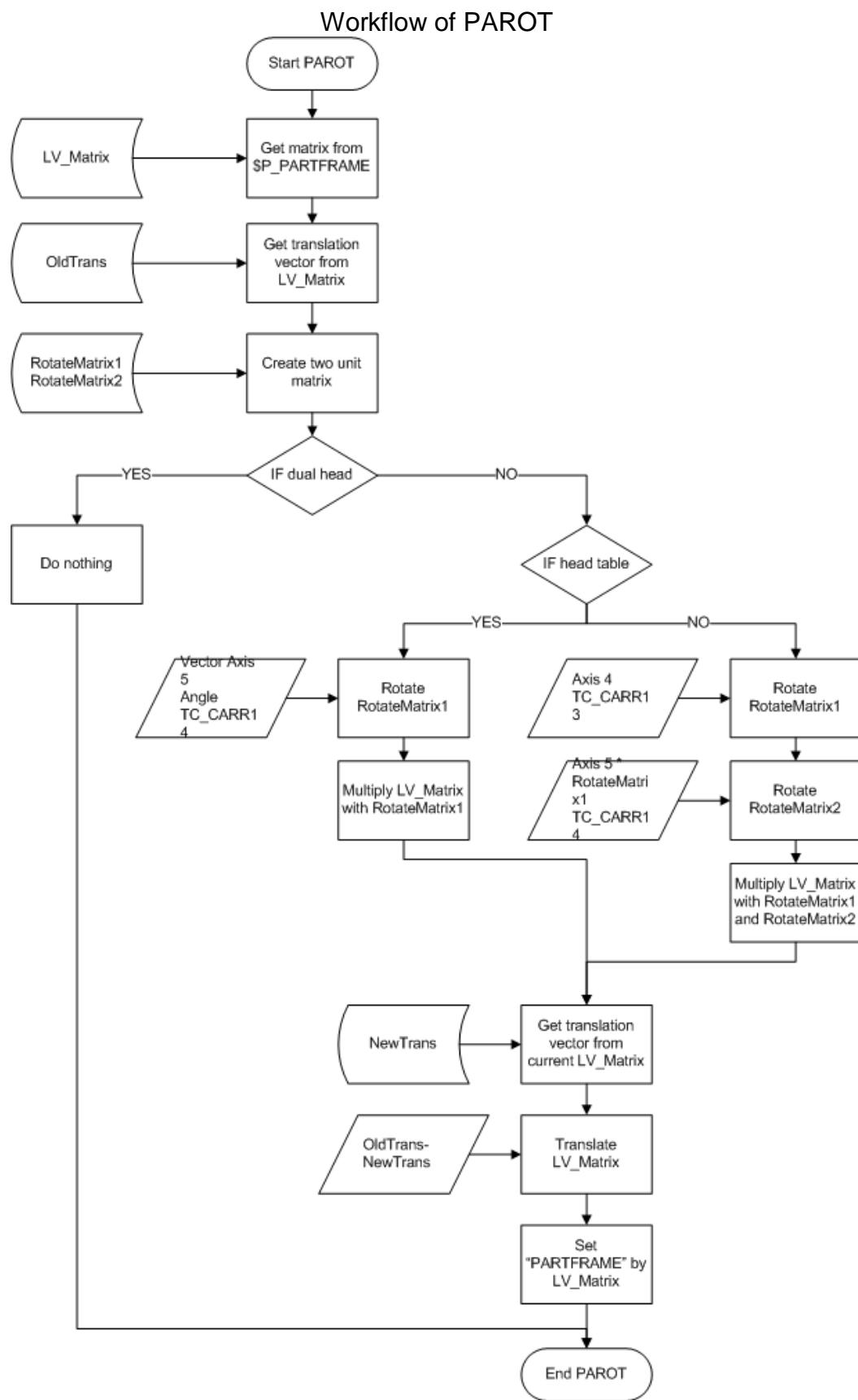
- On Powerline: TOOLCARR.SPF
- On Solutionline/Operate: CUST\_800.SPF

In addition CYCLE800 makes use of lots of system-variables and machine data, e.g. TC\_CARRxx[x] that holds the kinematic configuration and also needs some definition files like PGUD.DEF and SMAC.DEF so on Powerline even more.



## Workflow of TCARR





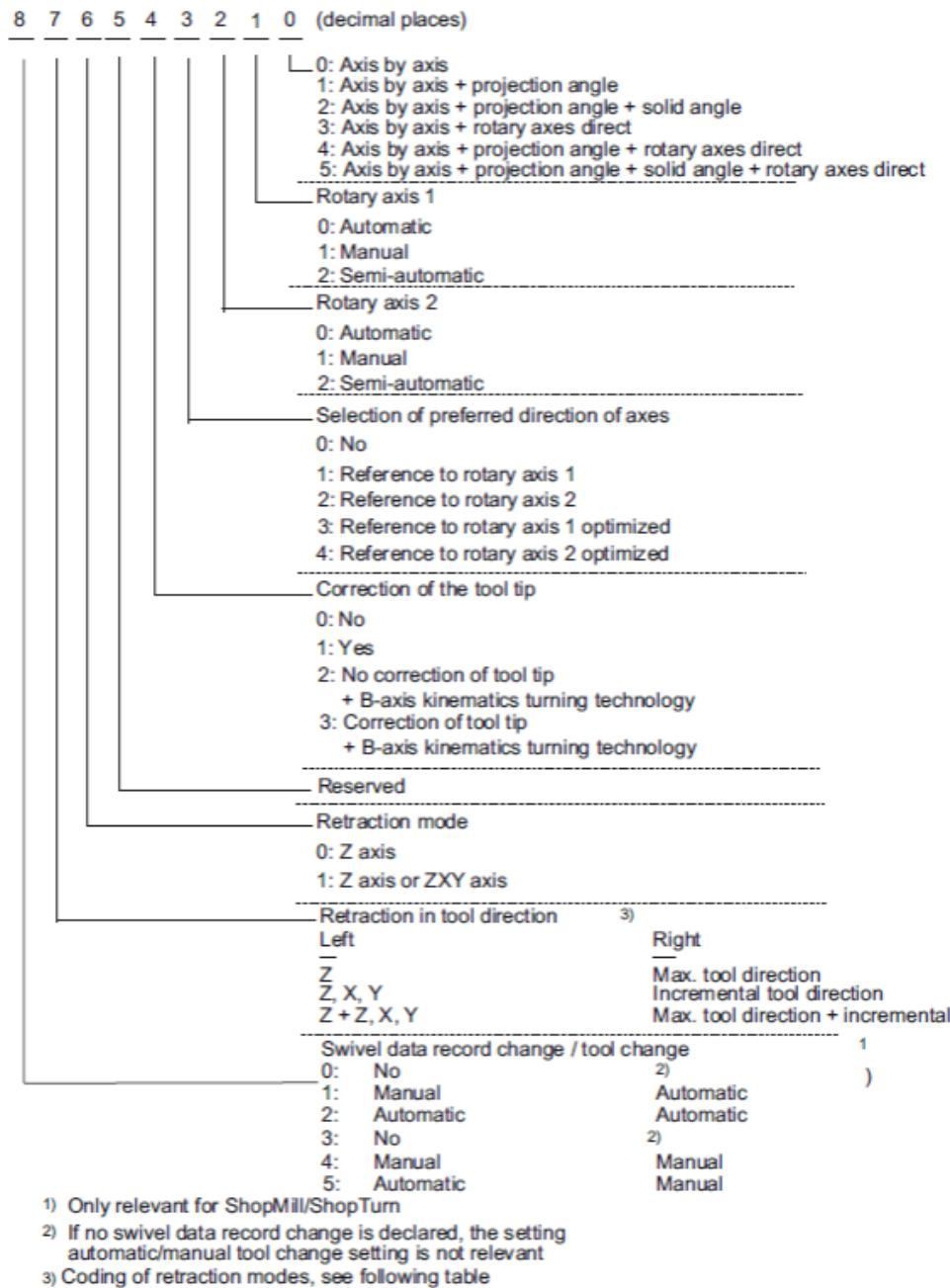
## How to set \$TC\_CARR37

The OOTB examples uses the Solutionline version 4.4/7 example: \$TC\_CARR37[1]=201003003

Display variants of input screen forms CYCLE800  
\$TC\_CARR37[n]

( $n \Rightarrow$  swivel data record)

If the following display variants are not set, the value will not be displayed in the input screen form (see Section "Programming via Screen Form").



## E. G68.2

### Workflow of internal G68.2 implementation

- Apply XYZ translation and IJK rotation to ROTATIONAL
- Calculate the tool vector with IJK rotation
- Generate possible tool and part angle solution using calculateIKSAngles
- Select the tool and part rotating angle based on the axis limitation by GMe\_SwivelingGetRotateSolution.
- Set GV\_dFourthAxisAngle and GV\_dFifthAxisAngle accordingly
- Set linear compensation due to tool holder rotation using GMe\_SwivelingCalculateLinears
- Align WCS on table rotation via GMe\_SwivelingCompTableRotation

### About G53.1

If G68.2 specifies the relationship between the feature coordinate system and the work piece coordinate system, G53.1 will automatically specifies the +Z direction of the feature coordinate system as the tool axis direction even if no angle is specified for the rotary axis.

G53.1 must be specified in a block after the block that contains G68.2

G53.1 must be specified in a block in which there is no other command.

## F. Default controller settings inside the CCFs

This section lists the settings, which are taken place when a CSE simulation is initialized based on the OOTB CCF. Thus the method `CSEInitializeChannel` is called, which sets these defaults. Afterwards possible ini files get registered, which add additional defaults or can overwrite the ones already set.

### Sinumerik:

Controller default set in the CCF:

- G0 G17 G54 G60 G601 G710 G90 G94 CFC NORM G450 BRISK ORIMKS DIAMOF TCOABS
- GV\_bUseLoadOffset=FALSE to deactivate the LoadOffset functionality.
- GV\_bUseSetToolCorrection=FALSE to not use SetToolCorrection in D-Metacode and use data from ini file instead.
- \$MC\_FRAME\_ADD\_COMPONENTS=1. This is used by G58 and G59. Also means that TRANS stores in \$P\_PFRAME[.., TR] and ATRANS stores in \$P\_PFRAME[.., FI]
- \$MN\_SCALING\_SYSTEM\_IS\_METRIC=25.4
- \$MC\_DIAMETER\_AX\_DEF="X"
- \$MN\_INT\_INCR\_PER\_MM=1000
- \$AN\_NCK\_VERSION=75000
- \$P\_PROG[0]="\_N\_MAIN\_MPFI"
- \$SCS\_DRILL\_TAPPING\_SET\_GG12[0]=1
- \$SCS\_DRILL\_TAPPING\_SET\_GG12[1]=1
- \$SCS\_DRILL\_TAPPING\_SET\_GG21[0]=1
- \$SCS\_DRILL\_TAPPING\_SET\_GG12[1]=1
- \$SCS\_DRILL\_TAPPING\_SET\_GG24[0]=1
- \$SCS\_DRILL\_TAPPING\_SET\_GG24[1]=1
- \$SCS\_DRILL\_TAPPING\_SET\_MC[0]=1
- \$SCS\_DRILL\_TAPPING\_SET\_MC[1]=1
- \$MC\_MM\_SYSTEM\_FRAME\_MASK=62

Controller defaults from the ini File:

- TO\_INI ; load a to\_ini file including tool and offset data produced by the OOTB post (the file TO\_INI is handled like a subprogram. If this is not found the system will NOT raise an error message as usual)
- PMAC ; definition subprogram from Siemens SL
- PGUD ;definition subprogram from Siemens SL
- CHAN\_DATA; Controller Variable Initialization
- TC\_CARR ; Sets data for the CYCLE800 see TC\_CARR.DEF

## Fanuc:

Defaults set in the CCF:

- G21 G0 G17 G90 G94
- Default-Configuration for the controller:#983=1 : Serie M - machining centers.
- (Change in INI program to 0 for Serie T
- #3401=1: calculator notation, G-Code System
- #983=1
- ABit 0: 1 = calculator notation; 0 = smallest increment notationBit 6+7: 0 = G-Code System A;  
64 = G-Code
- System B; 128 = G-Code System CThe smallest increment multiplier can be specified in  
GMe\_SetUnit (Default:
- 1000.0).#19696=0 : rotating table coordinate system for G43.4/5. Change in INI program to 32  
for non-rotating
- workpiece coordinate system
- GV\_bUseLoadOffset=TRUE to activate the LoadOffset functionality

Defaults from the ini File:

- (#19696=0 : rotating table coordinate system for G43.4/5. Change in INI program to 32 for  
non-rotating workpiece coordinate system)
- #19696=32
- G54
- (Implement the setting of the reference point G28 unit dependent)
- G10L52
- N1240P1R0
- N1240P2R0
- N1240P3R-200.000
- G11

## TNC Heidenhain:

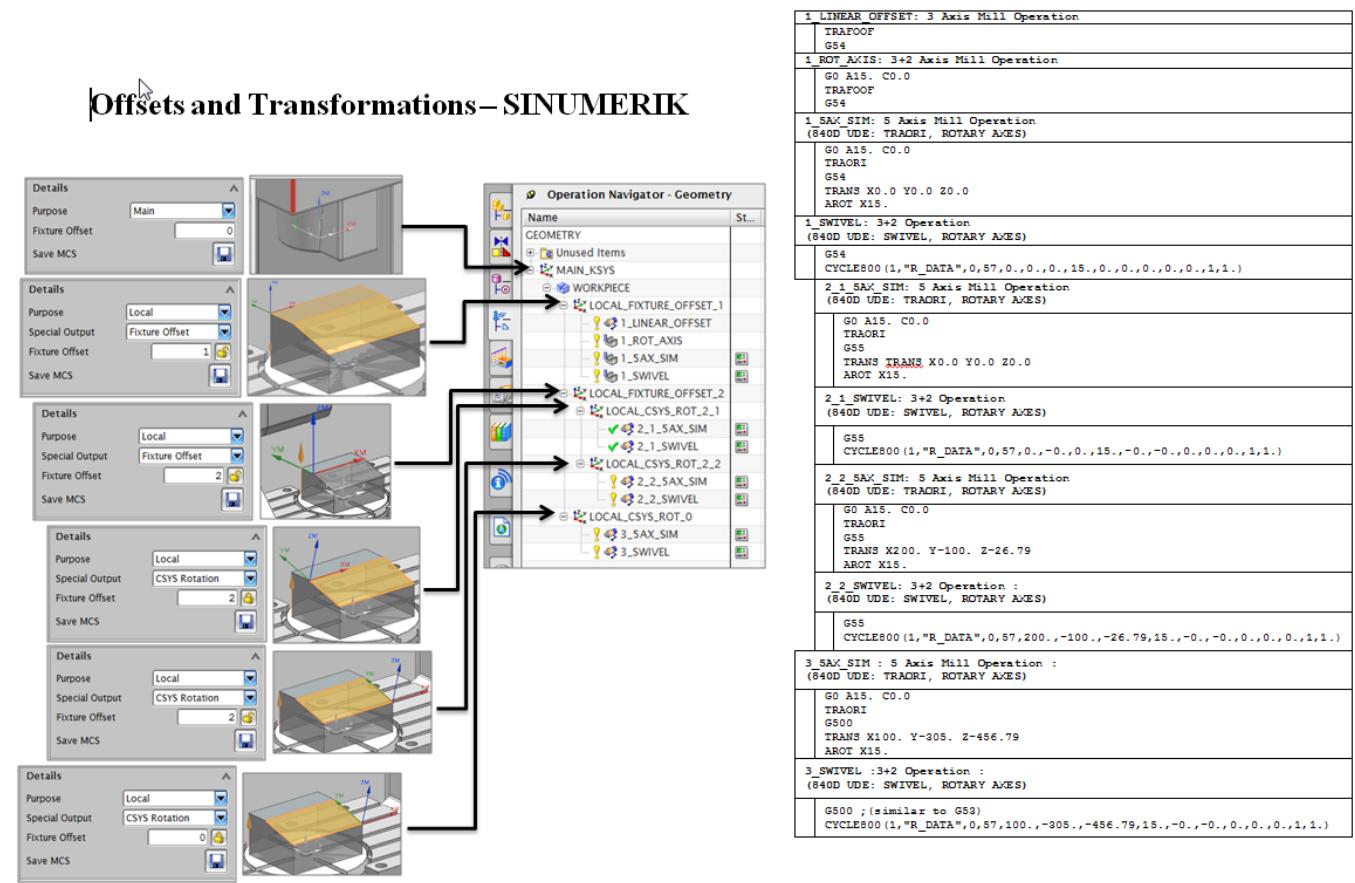
Defaults set in the CCF:

- GV\_bUseLoadOffset=TRUE (used for CYCLE\_DEF\_7)
- GV\_bUseLoadOffset247=FALSE (used for CYCLE\_DEF\_247)
- Systemdata: 7680[6]=0
- Systemdata: 7682[4]=1 ;1 when GV\_strFourthAxisName is available, 0 if not
- Systemdata: 7682[5]=1 ;1 when GV\_strFifthAxisName is available, 0 if not

Defaults from the ini File:

- No ini File

## G. OOTB MCS Handling output details



## Offsets and Transformations – Fanuc

**Note:**  
With that operation examples and the default settings in NX CAM it's not possible to obtain similar 5 axis simultaneous operation like in Sinumerik.

Further information you'll find in the **BASIC VERIFICATION TESTS for FANUC controller**

<b>1_LINEAR_OFFSET:</b> 3 Axis Mill Operation
G54 G0 A15. C0.0
<b>1_ROT_AXIS:</b> 3+2 Axis Mill Operation
G54 G64.2 X0 Y0 Z0 I1 J0 K0 R15.
<b>1_SWIVEL:</b> 3+2 Operation
<b>2_1_SAK_SIM:</b> 5 Axis Mill Operation
<b>2_1_SWIVEL:</b> 3+2 Operation
G55 A=15 C90 G68.2 X0 Y0 Z0 I1 J0 K0 R15.
<b>2_2_SWIVEL:</b> 3+2 Operation
<b>3_SAK_SIM:</b> 5 Axis Mill Operation
<b>3_SWIVEL:</b> 3+2 Operation
G60.2 X0.0 Y0.0 Z0.0 I1 J0 K0 R15.

## Offsets and Transformations – Heidenhain

**Note:**  
With that operation examples and the default settings in NX CAM it's not possible to obtain similar 5 axis simultaneous operation like in Sinumerik.

Further information you'll find in the **BASIC VERIFICATION TESTS for Heidenhain controller**

<b>1_LINEAR_OFFSET:</b> 3 Axis Mill Operation
CYCL DEF 7.0 CYCL DEF 7.1 X-100 CYCL DEF 7.2 Y-145 CYCL DEF 7.3 Z-400
<b>1_ROT_AXIS:</b> 3+2 Axis Mill Operation
L A15. C0.0 M128
<b>1_SWIVEL:</b> 3+2 Operation
<b>2_1_SAK_SIM:</b> 5 Axis Mill Operation
<b>2_1_SWIVEL:</b> 3+2 Operation
CYCL DEF 7.0 CYCL DEF 7.1 X-100 CYCL DEF 7.2 Y-205 CYCL DEF 7.3 Z-400 PLANE SPATIAL SPA15. SPC0.0 SPC0.0 TURN F+500 TABLE ROT
<b>2_2_SAK_SIM:</b> 5 Axis Mill Operation
<b>2_2_SWIVEL:</b> 3+2 Operation
CYCL DEF 7.0 CYCL DEF 7.1 IX 200 CYCL DEF 7.2 IY -100 CYCL DEF 7.3 IZ -26.7949 PLANE SPATIAL SPA15. SPC0.0 SPC0.0 TURN F+500 TABLE ROT
<b>3_SAK_SIM:</b> 5 Axis Mill Operation
<b>3_SWIVEL:</b> 3+2 Operation :
CYCL DEF 7.0 CYCL DEF 7.1 X 0 CYCL DEF 7.2 Y 0 CYCL DEF 7.3 Z 0  CYCL DEF 7.1 IX 100 CYCL DEF 7.2 IY -305 CYCL DEF 7.3 IZ -99  PLANE SPATIAL SPA15. SPC0.0 SPC0.0 TURN F+500 TABLE ROT

## Global Technical Access Center

### Installation assistance

For additional installation assistance, or to report any problems, contact the Global Technical Access Center (GTAC).

**Website:**

<http://support.industrysoftware.automation.siemens.com/gtac.shtml>

**Phone:**

United States and Canada: 800-955-0000 or 714-952-5444

Outside the United States and Canada: Contact your local support office.

This software and related documentation are proprietary to Siemens Product Lifecycle Management Software Inc .

©2017 Siemens Product Lifecycle Management Software Inc.

Siemens and the Siemens logo are registered trademarks of Siemens AG. NX, Solid Edge, and Teamcenter are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or their subsidiaries in the United States and in other countries. All other trademarks, registered trademarks, or service marks belong to their respective holders.