3.3 Advanced Shape Tuning [included in 687-1]

3.3.1 About Advanced Shape Tuning

Purpose

The purpose of *Advanced Shape Tuning* is to reduce the path deviation caused by joint friction of the robot.

Advanced Shape Tuning is useful for low speed cutting (10-100 mm/s) of, for example, small circles. Effects of robot joint friction can cause path deviation of typically 0.5 mm in these cases. By tuning parameters of a friction model in the controller, the path deviation can be reduced to the repeatability level of the robot, for example, 0.1 mm for a medium sized robot.

What is included

Advanced Shape Tuning is included in the RobotWare option *Advanced robot motion* and gives you access to:

- Instructions `FricIdInit`, `FricIdEvaluate` and `FricIdSetFricLevels` that automatically optimize the joint friction model parameters for a programmed path.
- The system parameters *Friction FFW On*, *Friction FFW level* and *Friction FFW Ramp* for manual tuning of the joint friction parameters.
- The tune types `tune_fric_lev` and `tune_fric_ramp` that can be used with the instruction `TuneServo`.

Basic approach

This is a brief description of how Advanced Shape Tuning is most commonly used:

1. Set system parameter *Friction FFW On* to TRUE. See *System parameters on page 156*.
2. Perform automatic tuning of the joint friction levels using the instructions `FricIdInit` and `FricIdEvaluate`. See *Automatic friction tuning on page 152*.
3. Compensate for the friction using the instruction `FricIdSetFricLevels`.

3.3.2 Automatic friction tuning

About automatic friction tuning

A robot's joint friction levels are automatically tuned with the instructions `FricIdInit` and `FricIdEvaluate`. These instructions will tune each joint's friction level for a specific sequence of movements.

The automatically tuned levels are applied for friction compensation with the instruction `FricIdSetFricLevels`.

Program execution

To perform automatic tuning for a sequence of movements, the sequence must begin with the instruction `FricIdInit` and end with the instruction `FricIdEvaluate`. When program execution reaches `FricIdEvaluate`, the robot will repeat the movement sequence until the best friction level for each joint axis is found. Each iteration consists of a backward and a forward motion, both following the programmed path. Typically the sequence has to be repeated approximately 20-30 times, in order to iterate to correct joint friction levels.

If the program execution is stopped in any way while the program pointer is on the instruction `FricIdEvaluate` and then restarted, the results will be invalid. After a stop, friction identification must therefore be restarted from the beginning.

Once the correct friction levels are found they have to be set with the instruction `FricIdSetFricLevels`, otherwise they will not be used. Note that the friction levels are tuned for the particular movement between `FricIdInit` and `FricIdEvaluate`. For movements in another region in the robot's working area, a new tuning is needed to obtain the correct friction levels.

For a detailed description of the instructions, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Limitations

There are the following limitations for friction tuning:

- Friction tuning cannot be combined with synchronized movement. That is, `SyncMoveOn` is not allowed between `FricIdInit` and `FricIdEvaluate`.
- The movement sequence for which friction tuning is done must begin and end with a finepoint. If not, finepoints will automatically be inserted during the tuning process.
- Automatic friction tuning works only for TCP robots.
- Automatic joint friction tuning can only be done for one robot at a time.
- Tuning can be made to a maximum of 500%. If that is not enough, set a higher value for the parameter *Friction FFW Level*, see *Starting with an estimated value on page 157*.
- It is not possible to view any test signals with TuneMaster during automatic friction tuning.
- The movement sequence between `FricIdInit` and `FricIdEvaluate` cannot be longer than 10 seconds.

Continues on next page

Note

To use Advanced Shape Tuning, the parameter *Friction FFW On* must be set to TRUE.

Example

This example shows how to program a cutting instruction that encapsulates the friction tuning. When the instruction is run the first time, without calculated friction parameters, the friction tuning is done. During the tuning process, the robot will repeatedly move back and forth along the programmed path. Approximately 25 iterations are needed.

At all subsequent runs the friction levels are set to the tuned values identified in the first run. By using the instruction `CutHole`, the friction can be tuned individually for each hole.

```
PERS num friction_levels1{6} := [9E9,9E9,9E9,9E9,9E9,9E9];
PERS num friction_levels2{6} := [9E9,9E9,9E9,9E9,9E9,9E9];

CutHole p1,20,v50,tool1,friction_levels1;
CutHole p2,15,v50,tool1,friction_levels2;

PROC CutHole(robtarget Center, num Radius, speeddata Speed, PERS
     tooldata Tool, PERS num FricLevels{*})
  VAR bool DoTuning := FALSE;

  IF (FricLevels{1} >= 9E9) THEN
    ! Variable is uninitialized, do tuning
    DoTuning := TRUE;
    FricIdInit;
  ELSE
    FricIdSetFricLevels FricLevels;
  ENDIF

  ! Execute the move sequence
  MoveC p10, p20, Speed, z0, Tool;
  MoveC p30, p40, Speed, z0, Tool;

  IF DoTuning THEN
    FricIdEvaluate FricLevels;
  ENDIF
ENDPROC
```

Note

A real program would include deactivating the cutting equipment before the tuning phase.

3.3.3 Manual friction tuning

Overview

It is possible to make a manual tuning of a robot's joint friction (instead of automatic friction tuning). The friction level for each joint can be tuned using the instruction `TuneServo`. How to do this is described in this section.

There is usually no need to make changes to the friction ramp.

> **Note**
>
> To use Advanced Shape Tuning, the parameter *Friction FFW On* must be set to TRUE.

Tune types

A tune type is used as an argument to the instruction `TuneServo`. For more information, see *tunetype* in *Technical reference manual - RAPID Instructions, Functions and Data types*.

There are two tune types that are used expressly for Advanced Shape Tuning:

Tune type	Description
TUNE_FRIC_LEV	By calling the instruction `TuneServo` with the argument `TUNE_FRIC_LEV` the friction level for a robot joint can be adjusted during program execution. A value is given in percent (between 1 and 500) of the friction level defined by the parameter *Friction FFW Level*.
TUNE_FRIC_RAMP	By calling the instruction `TuneServo` with the argument `TUNE_FRIC_RAMP` the motor shaft speed at which full friction compensation is reached can be adjusted during program execution. A value is given in percent (between 1 and 500) of the friction ramp defined by the parameter *Friction FFW Ramp*. There is normally no need to tune the friction ramp.

Configure friction level

The friction level is set for each robot joint. Perform the following steps for one joint at a time:

	Action
1	Test the robot by running it through the most demanding parts of its tasks (the most advanced shapes). If the robot shall be used for cutting, then test it by cutting with the same tool as at manufacturing. Observe the path deviations and test if the joint friction levels need to be increased or decreased.
2	Tune the friction level with the RAPID instruction `TuneServo` and the tune type `TUNE_FRIC_LEV`. The level is given in percent of the *Friction FFW Level* value. Example: The instruction for increasing the friction level with 20% looks like this: `TuneServo MHA160R1, 1, 120 \Type:= TUNE_FRIC_LEV;`
3	Repeat step 1 and 2 until you are satisfied with the path deviation.

Continues on next page

	Action
4	The final tuning values can be transferred to the system parameters. Example: The *Friction FFW Level* is 0.5 and the final tune value (`TUNE_FRIC_LEV`) is 120%. Set *Friction FFW Level* to 0.6 and tune value to 100% (default value), which is equivalent.

Tip

Tuning can be made to a maximum of 500%. If that is not enough, set a higher value for the parameter *Friction FFW Level*, see *Setting tuning system parameters on page 157*.

3.3.4 System parameters

3.3.4.1 System parameters

About the system parameters

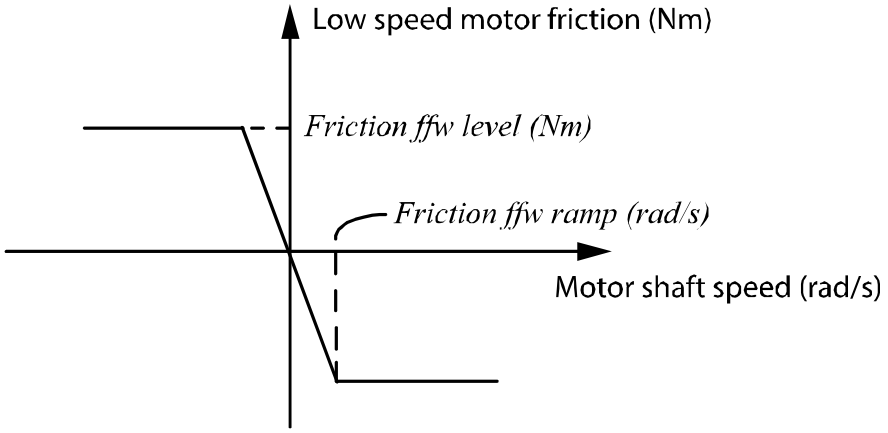
This is a brief description of each parameter in the option *Advanced Shape Tuning*. For more information, see the respective parameter in *Technical reference manual - System parameters*.

Friction Compensation / Control Parameters

These parameters belong to the type *Friction Compensation* in the topic *Motion*, except for the robots IRB 1400 and IRB 1410 where they belong to the type *Control Parameters* in the topic *Motion*.

Parameter	Description
Friction FFW On	Advanced Shape Tuning is active when *Friction FFW On* is set to TRUE.
Friction FFW Level	*Friction FFW Level* is the friction level for the robot joint. See illustration below.
Friction FFW Ramp	*Friction FFW Ramp* is the speed of the robot motor shaft, at which the friction has reached the friction level defined by *Friction FFW Level*. See illustration below. There is normally no need to make changes to *Friction FFW Ramp*.

Illustration



en0900000117

3.3.4.2 Setting tuning system parameters

Automatic tuning rarely requires changes in system parameters

For automatic tuning, if the friction levels are saved in a persistent array, the tuning is maintained after a power failure. The automatic tuning can also be used to set different tuning levels for different robot movement sequences, which cannot be achieved with system parameters. When using automatic tuning, there is no need to change the system parameters unless the default values are very much off, see *Starting with an estimated value on page 157*.

Transfer tuning to system parameters

When using manual tuning, the tuning values are reset to default (100%) at power failure. System parameter settings are, however, permanent.

If a temporary tuning is made, that is only valid for a part of the program execution, it should not be transferred.

To transfer the friction level tuning value (`TUNE_FRIC_LEV`) to the parameter *Friction FFW Level* follow these steps:

	Action
1	In RobotStudio, open the **Configuration Editor**, **Motion** topic, and select the type **Friction comp** (except for the robots IRB 1400 and IRB 1410 where they belong to the type **Control parameters**).
2	Multiply *Friction FFW Level* with the tuning value. Set this value as the new *Friction FFW Level* and set the tuning value (`TUNE_FRIC_LEV`) to 100%. Example: The *Friction FFW Level* is 0.5 and the final tune value (`TUNE_FRIC_LEV`) is 120%. Set *Friction FFW Level* to 0.6 (1.20x0.5) and the tuning value to 100% (default value), which is equivalent.
3	Restart the controller for the changes to take effect.

Starting with an estimated value

The parameter *Friction FFW Level* will be the starting value for the tuning. If this value is very far from the correct value, tuning to the correct value might be impossible. This is unlikely to happen, since *Friction FFW Level* is by default set to a value approximately correct for most situations.

If the *Friction FFW Level* value, for some reason, is too far from the correct value, it can be changed to an new estimated value.

	Action
1	In RobotStudio, open the **Configuration Editor**, **Motion** topic, and select the type **Friction comp** (except for the robots IRB 1400 and IRB 1410 where they belong to the type **Control parameters**).
2	Set the parameter *Friction FFW Level* to an estimated value. Do not set the value 0 (zero), because that will make tuning impossible.
3	Restart the controller for the changes to take effect.

3.3.5 RAPID components

About the RAPID components

This is an overview of all instructions, functions, and data types in *Advanced Shape Tuning*.

For more information, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Instructions

Instructions	Description
`FricIdInit`	Initiate friction identification
`FricIdEvaluate`	Evaluate friction identification
`FricIdSetFricLevels`	Set friction levels after friction identification

Functions

Advanced Shape Tuning includes no functions.

Data types

Advanced Shape Tuning includes no data types.