ORiN2 NetwoRC Provider
" Non-Stop motion calculator " Option

User's Guide

Version 1.0.0.1

August 23, 2011

[Remarks]

【Revision history】

Date	Version	Content
2011-04-28	1.0.0.0	First edition.
2011-08-31	1.0.0.1	Add restriction, Usage

【Hardware】

Date	Versions	Notes
RC7M (6 axis)	V2.330BM ~	"ORiN" option (1214) is required.

【Attention】

Additional license for "Non-Stop motion calculator" is required to use this option.

Contents

1. Introduction

The Non-stop motion calculator option is a function designed for the non-stop inspection applications with DENSO robot. This function calculates motion positions from inspection positions (teaching position), so that the trajectory of DENSO robot PASS motion will pass through the inspection positions. The combination of this motion positions with PASS motion and precise area function (Option) makes it possible to output a trigger signal at the inspection position precisely, without stopping the robot. As an example, the trigger signal is applied to an external trigger signal for a camera.

1.1. Installing license

To use the Non-stop motion calculator option, you need to install ORiN2 SDK, and also need to input "Non-stop motion calculator" license information.

How to add the license is as follows.

1. Run the CaoConfig tool from the [Start] menu, and select the [Cao Provider] tab.
2. Select the [DENSO NetwoRC CAO Provier] item on the provider list.
3. Click the [...] button of the license input box.
4. Click the [Add] button in the "ORiN2 License Manager" window.
5. Input a license key, and click the [OK] button.
6. Click the [Close] button to exit.

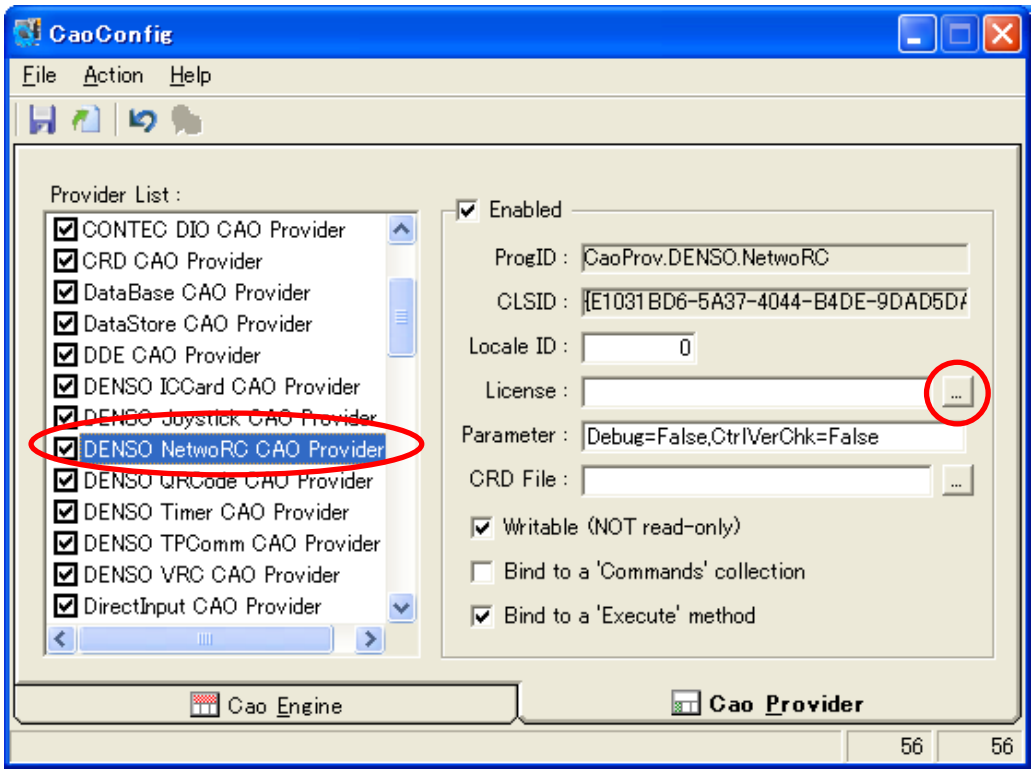

Figure 1-1 Installing 'Non-stop motion calculator' license

2. Outline of function

2.1. Outline

Setting Non-stop motion calculator Option activates related commands in the NetwoRC Provider. These commands generate non-stop motion positions from the inspection positions. By using the non-stop motion positions with “MOVE P, @P” PAC command, DENSO robot passes through the inspection positions.

DENSO Robot controller RC7M is required to calculate the motion positions, since this function uses the processing function in RC7M.

ORiN license is also required.

2.2. Commands

The following two commands in NetwoRC Provider are activated after the license input.

(1) InitNonStopPathLib property : Initialization of the non-stop motion calculator
(2) GenerateNonStopPath property : Calculation of the non-stop motion position

The details of the commands are shown at “Appendix F. Non-Stop Motion Calculator - Trajectory Generator Command for Non Stop Inspection” in “NetwoRC Provider User’s Guide”

2.3. Restrictions

Restrictions of GenerateNonStopPath Command are as follows:

- Max. Number of Inspection Points = 200
- Available for 6-axis robot only
- Area size for additional axis should be assigned in [degree] (for rotational axis) or [mm] (for linear axis) in accordance with the axis setting.
- Unavailable for Unlimited rotation of the extra-joint
- Unavailable when Auto Speed and Auto Acceleration mode is used
- Payload setting is restricted to the unit of 1,000g.

3. Sample Program

The program with non-stop motion calculation option can be developed by commercially available programming languages (C/C++, VB, etc.) as same as ORiN programming. The easiest method is to use CaoScript including in ORiN SDK, which is a script language based on VBScript. Section 3.1 shows a sample program based on CaoScript. Other samples are shown in section 3.2.

3.1. CaoScript sample program

In this sample, the robot type is assumed as VS-6577G-BA, and the IP address of the target RC7M is 192.168.0.1.

```
' Create CAS Object
Set rc = cao.AddController("RC", "CaoProv.DENSO.NetwoRC", "", " Conn=eth:192.168.0.1" )
' Initialization
call rc.Execute("InitNonStopPathLib")
' Set inspection positions Data
redim vntTeachPos(7)
vntTeachPos(0) = Array(300.0, 100.0, 600.0, 180.0, 0.0, 180.0, 5, 0.0, 0.0, 100 * 0.01, 1, 0)
vntTeachPos(1) = Array(300.0, 91.0, 600.0, 180.0, 0.0, -180.0, 5, 0.0, 0.0, 100 * 0.01, 0, 0)
vntTeachPos(2) = Array(310.0, 30.0, 600.0, 180.0, 0.0, -180.0, 5, 0.0, 0.0, 100 * 0.01, 1, 0)
vntTeachPos(3) = Array(315.5, 24.5, 600.0, 180.0, 0.0, -180.0, 5, 0.0, 0.0, 100 * 0.01, 0, 0)
vntTeachPos(4) = Array(300.0, 10.0, 600.0, 180.0, 0.0, 173.0, 5, 0.0, 0.0, 100 * 0.01, 1, 0)
vntTeachPos(5) = Array(300.0, 10.0, 600.0, 180.0, 0.0, 176.0, 5, 0.0, 0.0, 100 * 0.01, 0, 0)
vntTeachPos(6) = Array(300.0, 10.0, 600.0, 180.0, 0.0, 171.0, 5, 0.0, 0.0, 100 * 0.01, 0, 0)
vntTeachPos(7) = Array(300.0, 10.0, 600.0, 180.0, 0.0, -180.0, 5, 0.0, 0.0, 100 * 0.01, 1, 0)
' Area Data for Trigger Signal
redim vntAreaInfo(7)
vntAreaInfo(0) = Array(4,4,4,4,0,0)
vntAreaInfo(1) = Array(4,4,4,4,0,0)
vntAreaInfo(2) = Array(4,4,4,4,0,0)
vntAreaInfo(3) = Array(4,4,4,4,0,0)
vntAreaInfo(4) = Array(4,4,4,4,0,0)
vntAreaInfo(5) = Array(4,4,4,4,0,0)
vntAreaInfo(6) = Array(4,4,4,4,0,0)
vntAreaInfo(7) = Array(4,4,4,4,0,0)
' Calculate Non-stop motion positions data
vntMovePos = rc.Execute("GenerateNonStopPath", Array(vntTeachPos, vntAreaInfo, Ubound(vntTeachPos) +1,
100.0 * 0.01, 0.7, 1))
```

3.2. Other Samples

Please refer sample programs stored in the following folder.

<ORiN2>¥ CAO¥ProviderLib¥DENSO¥NetwoRC¥Samples¥NonstopPath

Table 3-1 List of Sample program

Sample	Contents	Language
NonStopPath.vbs	Sample for VS-6577G-BA	CaoScript
OrbitGen_Sample.exe	Sample application for the generation of the non-stop motion positions from inspection positions with user interface.	Visual Basic 2005

3.3. How to use for inspection vision application

In order to constitute vision inspection robot application, the following two functions are also required.

・Precise Area Function

This function monitors the position and posture of robot, and outputs a trigger signal. The signal is applied as a camera trigger signal.

The related PAC command is “POSTUREAREA.”

This function requires inputting license information in RC7M.

・High Accuracy Trajectory Control Function

This function improves the accuracy of the robot motion.

In a PAC programs which include robot motion commands, call the library “SetHighPathAccuracy.pac” before motion commands.