Magnus Dahrén, SICK IVP, Sweden - March 20, 2018, Updated: December 15, 2020

PLOC2D Integration Tutorial

Background

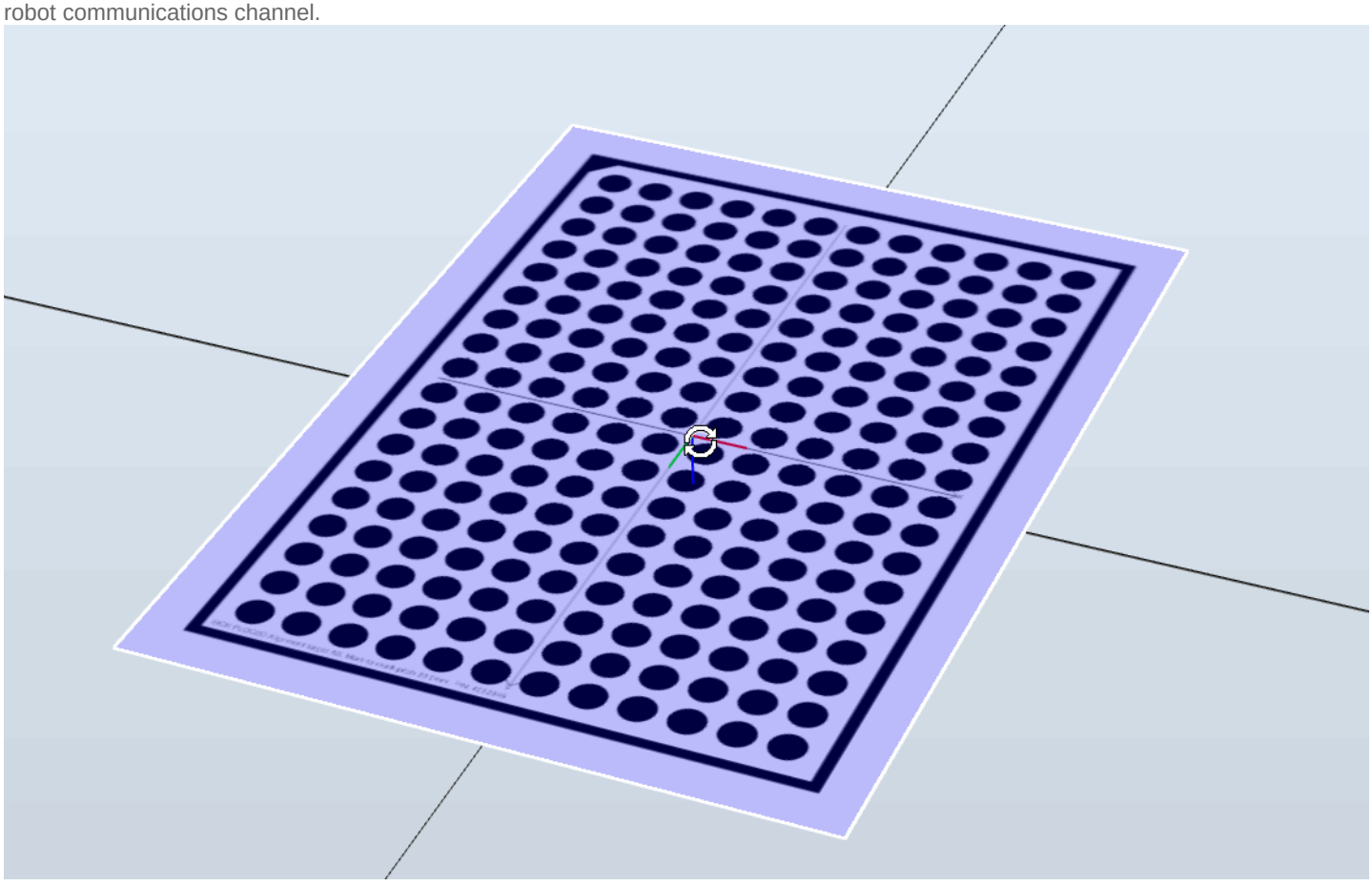
Correction frame concept

The PLOC2D system is a reference based positioning system intended to be used together with industrial robots. The system provides part positioning information as a substitute to complex mechanical fixtures or guides. With the positioning being reference based, the robot needs to know how to pick or manipulate the part in a given reference frame. The measurement result from the PLOC2D system provides a correction frame (may be called an offset frame in some terminology) which when applied to the reference frame will move it in such a way that the robot can use the same picking position as in the initial reference pose. Another way to express this is that the system does not provide the actual position of the part, but instead a correction of the original coordinate system that when applied (to the original coordinate system) makes the part appear at the original position. Applying a correction in this application means performing a pose multiplication.

Integration workflow

Alignment of camera and robot frames

To be able to communicate the above mentioned correction frame, the system needs to know about the original reference frame of the robot. This frame is defined by putting the PLOC2D alignment target in the plane that the measurements will take place. Once in position, the alignment target needs to be measured from both the robot and the camera, with the end result being that both the robot and the PLOC2D system knows this common coordinate system. Measurement from the robot typically involves using a well defined TCP pointer and pointing to the origin, x- and y-axes respectively in order to establish the coordinate system. Measurement from the PLOC2D system is performed by the Align command, either performed manually in the web user interface or as a command on the robot communications channel.

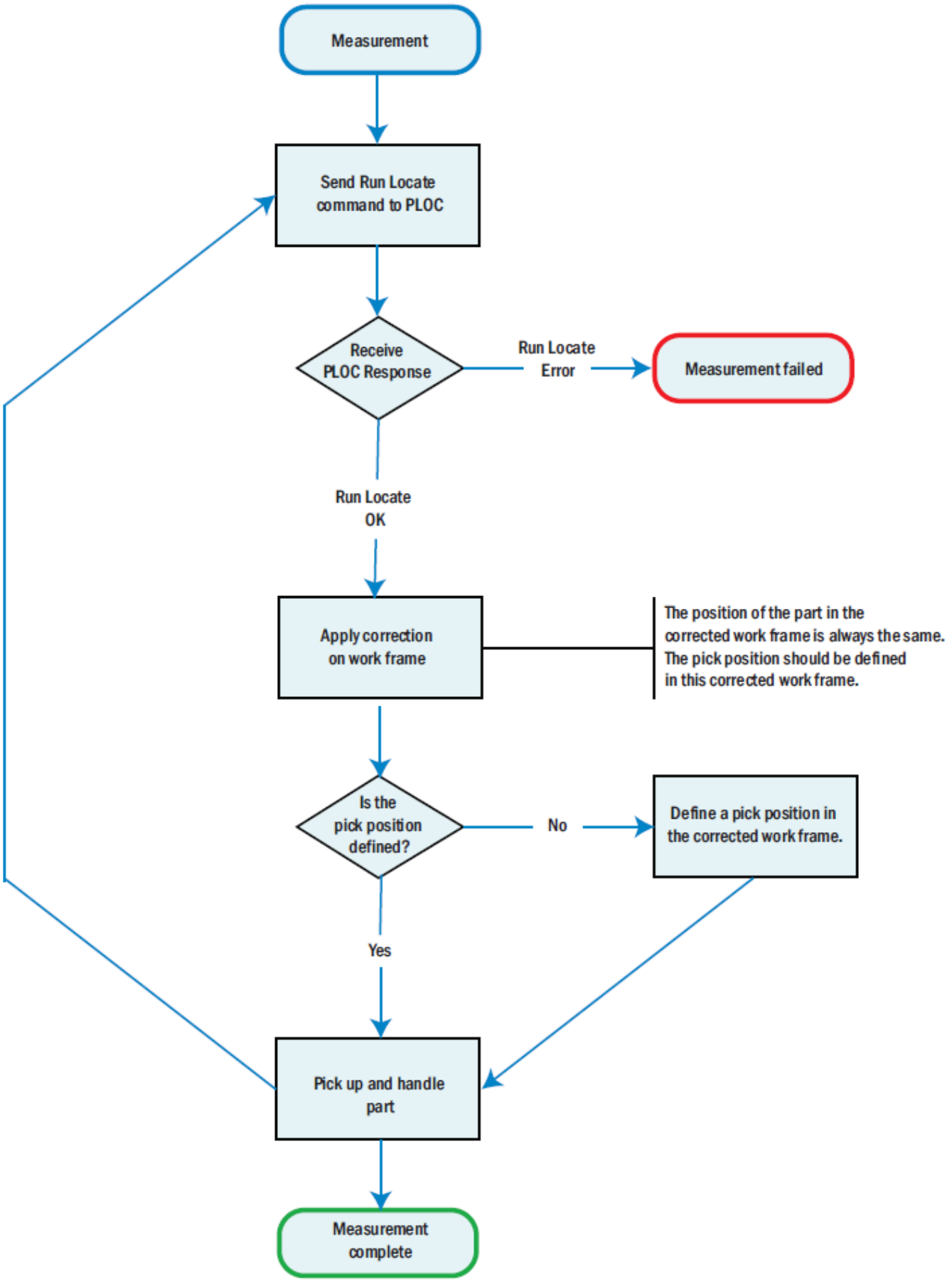


Job configuration on robot

Once the common coordinate system is established and measured from both the robot and the PLOC2D system, the robot needs to define how to act upon a part when it is in the reference position. This is typically done by putting the part in the reference position, moving the robot to a suitable picking position, P, and store this position *as it is defined in relation to the common coordinate system A*.

Runtime communication workflow

With the information on how to pick the part in the (uncorrected) reference frame stored, the robot now ready to let the PLOC2D system perform a measurement. The result from this measurement is the correction that, when applied to (a copy of) the common reference frame (A), allows the robot to pick the part at the original unchanged position, P. In other words, the pick position is always unchanged so the coordinate system needs to be moved accordingly together with the part in order for the robot to pick the part.



The two above steps can be conveniently combined in one workflow, as is pictured in the flowchart above. This makes use of the fact that as long as we have created a corrected coordinate system that follows the part, we can use this corrected coordinate system and (re-)define picking positions as we see fit.

Connecting

The robot connects to the PLOC2D system over a standard TCP/IP socket, where the PLOC2D acts as the server and the robot is a client. The default port for communication is 14158. The workflow is driven by the robot program and the PLOC2D system responds to command requests provided by the robot. Communication commands can be expressed either as CSV or XML formatted strings and for simplicity this tutorial uses CSV.

Sending commands

In a standard application there is only one command that the robot need to be able to send, the Run.Locate-command. This command requires at a minimum one argument, which is the requested job number (or a list of job numbers if more than one localization is to be executed). So, sending the string Run.Locate,1 will make the system perform a localization of job number 1. (Sending Run.Locate,2 3 6 will similarly run localizations of job 2, 3 and 6.)

Command	Description
Run.Locate,[Job numbers]	Acquire a new image and attempt to locate the parts from the specified job. More than one job can be specified. If no job is specified, the command locates all configured jobs and returns the first result.
Run.Locate,[Job numbers], [Match]	Return a result from the previous Run.Locate command. The result to return is specified by the Match parameter. No new image is acquired when using the Run.Locate command with the Match parameter.
System.Restart.Software, [String]	Initiates a software restart with a text message string for logging.

If multiple results are expected, the Run.Locate command can be used with an additional parameter, added after the job(s) parameter. This parameter can be used to request a specific result out of the available results. The parameter can have the following values:

- 'acquire' - forces the PLOC2D system to acquire a new image
- 'next' or 'previous' - selects the next or previous result available
- any number N - selects the Nth result of the available results

Sending the command Run.Locate,1,acquire will acquire an image and report the first available result for job 1. If more than one result is available, sending the command Run.Locate,1,next will provide the next available result in the same image and sending the command Run.Locate,1,7 will get the 7th result from the current image.

Receiving and parsing messages

Once the PLOC2D system has performed the localization the result will be communicated through a response message on the same communication channel. The response message for a successful localization will be Run.Locate.Ok followed by a comma separated list of information containing

- the job number used for the localization
- the match number and total number of matching shapes
- the correction frame in terms of (x,y,z,rx,ry,rz)
- the scale, score and time required for the match
- the exposure settings used for the match
- the identified job (if running multiple jobs)

The robot as a minimum needs to be able to parse the correction frame so that can be used to correct the common coordinate system before picking the part.

Command	Response
Aligment.Align,1	Aligment.Align.Ok alignment x y z rx ry rz Alignment.Align.Error alignment error
Run.Locate,[Job numbers]	Run.Locate.Ok job match matches x y z rx ry rz scale score time exposure identified Run.Locate.Error job score time error
System.Restart.Software, [String]	System.Restart.Software.Ok System.Restart.Software.Error error