

New River Kinematics’

SA Robot Calibration Appliance:

Product Summary

New River Kinematics has a number of product offerings that may be used in robotic applications. The purpose of this document is to introduce the SA Robot Calibration Appliance and discuss its potential applications.

The Appliance gives you the ability to implement a calibrated kinematic model of the robot within your native robot language programs. So, from the robot language, you can trigger metrology measurements at various robot poses, run a calibration on that data, and then use the calibrated model as you continue your program, moving the robot as you normally would.

This is accomplished through a small, stand-alone, computer that the robot controller can communicate with in order to access the calibrated kinematics. This means that you never have to modify the kinematic model that is internal to the robot. Instead, the calibration appliance allows you to work around it and use an external, calibrated, kinematic model. An added feature is that you can easily switch between multiple kinematic models for different areas of focus. This allows you to locally calibrate the robot for a task and gain even higher accuracy.

The metrology measurements used in the calibration process are taken using the SpatialAnalyzer (SA) measurement platform. You simply connect SA to the calibration appliance over the network, and then the appliance will trigger measurements as required by your robot program. Once the calibration is complete, SpatialAnalyzer is no longer required and the Appliance can serve up calibrated kinematics to the robot as a stand-alone device.

# Supported Platforms

The SA Calibration Appliance currently supports:

* Kuka KR C2 controller
* Kuka KR C4 controller
* ABB IRC5 controller
* Fanuc R-30iB controller
* Comau C5G controller

The intention is to continue to add support for other brands and models of robot controllers based on customer demand. The implementation on the controller side will obviously be different with each species of robot, but the overall concept remains the same.

# Requirements

* Controller Option for External Communication: The robot controller must have the required protocol activated. For some robots, this means adding a module, or a different software option. The controller-specific documentation for the Appliance provides more details.
* Familiarity with the Native Robot Language: The user must be comfortable programming the robot in its native language.
* Access to a metrology device and SpatialAnalyzer: During the measurement process, SA and an instrument are required.

# Hardware

The SA Robot Calibration Appliance is a software application that will run on any standard Windows computer. In order to facilitate seamless implementation, the Calibration Appliance software is bundled with a small form factor solid-state computer. This computer is pre-configured with the necessary software and drivers for the specific robot controller model. The device is small, about the size of a 4-port network switch, and could even be mounted inside the robot controller cabinet. The computer requires a power connection and a network connection to the robot and the SA measurement software. The computer is shown below:



# Architecture

The conceptual architecture of the system is presented in the following figure. Note that there are two processes outlined: the production process and the calibration process. The idea is that you run SpatialAnalyzer and the measurement instruments during the calibration process, but then you can remove them for the production process where the calibrated model is put to use. This allows you to have several robots served by common metrology devices.



Note that the Appliance is not driving the process. The native robot language drives the process. The communication between the robot controller native language and the Appliance is accomplished using functions we have created in the native control language to read and write data externally. These functions are wrapped in a way that the robot language programmer has very few changes to make in order to accomplish calibrated motion.

# Implementation

In this section, we will depict portions of the implementation of this system in the Kuka KRL programming language. This is provided to give the reader a reference for how the process can be controlled from the native language side. Note that the calls prefixed with NRK\_ represent functions NRK created and placed on the robot controller as part of the installation. Also note that these calls are specific to the Kuka language, but there are similar calls for the other robot manufacturer’s languages. The goal is to have only minor modifications required to a standard robot program in order to make use of the appliance.

## Running the Calibration Process

This code segment shows how to connect to the calibration appliance and perform a calibration from the augmented KRL language. Note that before (or during) this process, SpatialAnalyzer would need to be connected to the calibration appliance so that the measurement requests can be handled.

NRK\_Connect() ; connect to the Appliance using pre-configured IP and port

NRK\_OpenCalibration(CalId)

NRK\_ClearCalibration(CalId)

…

LOOP ; move through cal poses and take measurements

NRK\_MoveToCalibrationPt(i) ; move to a pre-specified pose

NRK\_AddCalibrationPt(CalId, $ACT\_TOOL, i) ; add to calibration pose list

retVal = NRK\_WaitForMeasurement() ; trigger SA measurement

 ENDLOOP

 NRK\_ConfigureSolver() ; set parameters based on user settings and cal mode.

 NRK\_SolveCalibration(CalId) ; run the optimization based on parameters and data

 NRK\_WaitForSolution()

 ; if the calibration is valid, close it and set base and tool in controller.

 NRK\_CloseCalibration(CalId, 0)

 NRK\_GetToolFrame(CalId)

 NRK\_GetBaseFrame(CalId)

## Using the Calibrated Kinematics to Perform Motion

This code segment shows how to move the robot using the newly calibrated kinematic model. Note that you can have multiple calibrated models stored in the Appliance. If you do this, you can switch between them depending on what you are doing and possibly the zone you are working in. Also, you do not have to run the calibration every time before using the modified model. If it is stored, you can just start things up, load the model and start moving using the calibrated kinematics.

NRK\_Connect() ; connect to the Appliance using pre-configured IP and port

NRK\_OpenCalibration(CalId)

 CalId = NRK\_SetCalibrationId()

 NRK\_GetBaseFrame(CalId)

 NRK\_GetToolFrame(CalId)

 ; these commands could be done in a loop, but are listed for example

 LIN NRK\_RUN\_PT[i].TARGET ; linear move command using controller kinematics

 NRK\_LIN(NRK\_RUN\_PT[i].TARGET) ; same move, but using Appliance kinematics

PTP NRK\_RUN\_PT[i].TARGET ; point to point move using controller kinematics

NRK\_PTP(NRK\_RUN\_PT[i].TARGET) ; same move, but using Appliance kinematics

CIRC AuxPt, EndPt, CA 360 ; circular move using controller kinematics

NRK\_CIRC(AuxPt, EndPt, 360) ; same move, but using Appliance kinematics

…

; you can also take measurements to verify how well the robot moves to a position

 LIN NRK\_RUN\_PT[i].TARGET ; normal move

 NRK\_CheckRawTarget(NRK\_RUN\_PT[i].TARGET) ; measure using SA

 answer = NRK\_PromptOperator("Raw target result", "Continue|Quit")

 NRK\_LIN(NRK\_RUN\_PT[i].TARGET) ; same move, but using Appliance kinematics

 NRK\_CheckRefinedTarget(NRK\_RUN\_PT[i].TARGET)

 answer = NRK\_PromptOperator("Refined target result", "Continue|Quit")

# Additional Information

For more detailed information, please refer to the SA Robot Calibration Appliance User Manual for your specific controller.