Proceedings of ASME 2017 International Mechanical Engineering Congress & Exposition
IMECE 2017
Nov 3–9, 2017, Tampa, FL, U.S.A.

IMECE2017–70404

TOLERANCES AND UNCERTAINTY IN ROBOTIC SYSTEMS

F. Proctor, Marek Franaszek, and J. Michaloski
National Institute of Standards
and Technology
Gaithersburg, Maryland

ABSTRACT

The ability to be programmed for a wide range of tasks is what differentiates robots from dedicated automation. Consequently, robots can be faced with often-changing requirements and conditions. Conventional application development based on teach programming takes robots out of production and occupies personnel, limiting robots' effectiveness in these environments. Off-line programming solves these problems, but robot inaccuracy must be compensated by a combination of calibration, compliance, and sensing. This complicates up-front systems engineering and application development, but results in systems that can operate in a wider range of requirements and conditions. Performance can be optimized if application tolerances and process uncertainties are known. If they often change, optimization must be done dynamically. Automating this optimization is a goal of smart manufacturing. With its trend of increasing connectivity between the components of robotic systems both within workcells and to the enterprise, exchanging this information has become more important. This includes tolerance information from design through process planning to production and inspection, and measurement uncertainty from sensors into operations. Standards such as ISO 10303 (STEP), the Quality Information Framework (QIF), the Robot Operating System (ROS), and MTConnect support this exchange to varying degrees. Examples include the assignment of assembly tasks based on part tolerances and robot capabilities; the automated generation of robot paths with tolerances arising from sensed obstacles; and the optimization of part placement to minimize the effects of position uncertainty. This paper examines requirements for exchanging tolerance and uncertainty in robotics applications, identifies how these requirements are being met by existing standards, and suggests improvements to enable more automated information exchange.

Keywords

Uncertainty, tolerance, error, robot, computer aided process planning

Nomenclature

ADC	Analog to Digital Conversion
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAPP	Computer Aided Process Planning
D-H	Denavit–Hartenberg
DME	Dimensional Measurement Equipment
DMSC	Dimensional Metrology Standards Consortium
DMIS	Dimensional Measuring Interface Standard
DoF	Degrees of Freedom
GD&T	Geometric dimensioning and tolerancing
HTM	Homogeneous Transform Matrix
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
MBD	Model Based Design
PMI	Product and Manufacturing Information
POI	Point of Interest
QIF	Quality Information Framework
QMR	Quality Measurement Results
ROS	Robot Operating System
STEP	STandard for the Exchange of Product model data
URDF	Unified Robot Description Format

XML	eXtensible Markup Language
XSD	XML Schema Definition

INTRODUCTION

Industrial robots have had much success automating repetitive tasks in structured environments, where their ability to be programmed has enabled their proliferation into a wide variety of applications such as part handling, spray painting, and welding. The predominant method of teach programming is effective but time consuming, and is an impediment to applying robots for short-turnaround jobs. Off-line programming solves this problem, but relies on robot accuracy which is typically much worse than that of machine tools, necessitating the use of sensors to compensate for this inaccuracy. Sensors also enable robots to operate in unstructured dynamic environments, and the continued improvement and cost reductions in sensors, especially vision, have now made it possible to deploy robots into previously hard-to-automate operations and for jobs with small lot sizes and short lead times, and often-changing performance requirements.

These changing requirements and uncertain environments make automated offline programming a necessity, but they also afford an opportunity to optimize manufacturing [1]. Effectively conveying information throughout the design, planning, and production phases is key to achieving this optimization. Until recently, this was a manual process, since product and manufacturing information resided in proprietary systems with no support in standards for preserving the semantic content throughout the export and import steps. The situation has improved, with standards in place for design, process planning, execution, and quality activities that make full semantic exchange possible. This paper will examine the state of these information exchange standards, and identify needs for revisions that can improve the efficiency and effectiveness of the exchange of tolerance and uncertainty information through the manufacturing chain for robots.

USE CASE SCENARIO

In this section, a use case scenario will be presented to provide context for the following sections on information exchange issues. At the outset, systems engineering is done to determine how the overall manufacturing activities will be carried out in the facility, given budgets, the equipment market, and staff capabilities. In this scenario, an assembly workcell has been established that combines industrial robots, dexterous manipulators, auxiliary equipment, and sensors, that together can achieve assembly performance to desired accuracy. This workstation includes two robots, each with dexterous grasping abilities, vision system throughout to determine the actual locations of parts and obstacles, force sensing for insertion of close-fitting parts, and auxiliary equipment for the staging and fixturing of parts as needed.

The process begins with a designer preparing a set of design files for components of an assembly that fulfills customer requirements for form, fit, and function. The design files specify product and manufacturing information, including geometric dimensions and tolerances. Using the design information, an assembly planner determines the order of operations needed to complete the full assembly, including any tooling requirements or other resources necessary to carry out individual steps. For complex assemblies, this may require the assignment of tasks to different resources, including people, depending on their capabilities. In this scenario, a process planner prepares a sequence of jobs for fabricating the assembly components [2, 3]. Robot programming for the assembly tasks is done offline, with no teach programming. Computer-aided design (CAD) files for the assembly components are loaded into an assembly planning system, which generates the sequence of operations, possibly well ahead of the actual assembly event. Because of the potential for robot collisions, the exact sequence of robot motions will be determined during execution, using information from the robot controllers and sensors. This can be done using explicit synchronization elements in the individual robot programs, through the sequential execution of programs through a supervisory controller, or through a single controller that does real-time planning for both robots as if they were a combined unit. In this scenario, a supervisory controller is used for high-level sequencing of programs for each robot.

The components arrive in two sets: a base assembly structure, in this case an engine block, and a kit of parts to be attached to it. The engine block has a single feature with tight assembly tolerances, and must be located so that this feature is accessible to the more accurate robot, and in its zone of highest accuracy. A prior inspection step has measured the actual location of this feature, and the CAD model has been updated. The block is affixed to a pallet that can be shuttled onto a rotary worktable via a mobile cart and conveyor. Once on the rotary table, it can be rotated to its most favorable orientation.

The parts kit is in an area accessible to both robots. The location of parts in the kit is not known a priori, so a vision system is used to determine their identity and location to sufficient accuracy. In general, parts may not be graspable in the orientation in which they are to be inserted, or in the order in which they must be assembled, so robot plans must be generated dynamically to reorient parts or clear out obstructing parts. In this scenario, a shared staging area is provided for these operations.

The location (six-degree-of-freedom (DoF) pose) of parts determined by the vision system will have some uncertainty, as will the location of the robot as it grasps the part, resulting in compounded uncertainty in the part as it is presented to its assembly location. To overcome this, a visual localization step is conducted just prior to assembly, where the robot presents the grasped part to a vision system that determines its actual location. The final robot path for assembly is adjusted by an offset to the nominal pose.

A nominal free space robot path is computed that includes tolerances on position and orientation that ensure an envelope of collision-free motion. The tolerances give flexibility to the real-time path planner to optimize motion using machine-specific cost functions, such as minimizing joint motion or energy consumption [4].

For assembly tasks with tight clearances, position control is not sufficient, and force control must be done to ensure proper fit. The inclusion of force control may reduce the need for high accuracy, necessitating only moderate accuracy to bring the part into the range where it can engage the assembly location within the region required to begin the force-controlled insertion.

The parts kit arrives first, and the vision system shows that the parts are present and accessible, but that several need to be reoriented in order to be grasped properly for insertion. The supervisory controller generates a set of programs for the robots, consisting of a series of pick-and-place moves with path tolerances that ensure collision-free motion over their full duration, eliminating the need to synchronize motion during execution. The robots proceed to reorient the affected parts.

The engine block arrives and is shuttled into its nominal location on the rotary table. The vision system determines its actual pose, and the part is rotated so that the region of minimum uncertainty contains the high-tolerance feature, and this lies within the robot's region of maximum accuracy. This is the first assembly operation, and the robot acquires the part from the kit, takes it to the vision system to determine its actual grasped pose, and the offset insertion path is executed to bring the part into contact for the final force-controlled insertion.

Assembly of the other components continues. In most cases, the robots can operate independently, with their paths enveloped in tolerances that ensure collision-free motion. In some cases, however, there is the potential for collision unless the motion is synchronized. In these cases, the programs are sequenced by the supervisory controller, which suspends the activities of one robot while the other completes its task.

Once the assembly process is done, the completed assembly is transferred to a final quality assurance step, where the location of features and connections is verified according to the design requirements.

In the following sections, this use case scenario will serve as a reference example to illustrate the applicability of standards for exchanging tolerance and uncertainty, and issues that could be addressed through revisions to these standards.

TOLERANCES IN DESIGN

As described in the use case scenario, the process begins with a designer preparing a set of design files that define product geometry including geometric dimensions and tolerances. These tolerances signify how much variation is acceptable for satisfactory form, fit, and function. The assignment of optimal tolerances that keep manufacturing costs low while still meeting requirements can be difficult, and in many cases, they are informed by practical experience of the manufacturer. Standard practice is to represent this information in geometric dimensioning and tolerancing (GD&T) callouts, which could be interpreted visually but do not support automated querying, as shown in Figure 1.

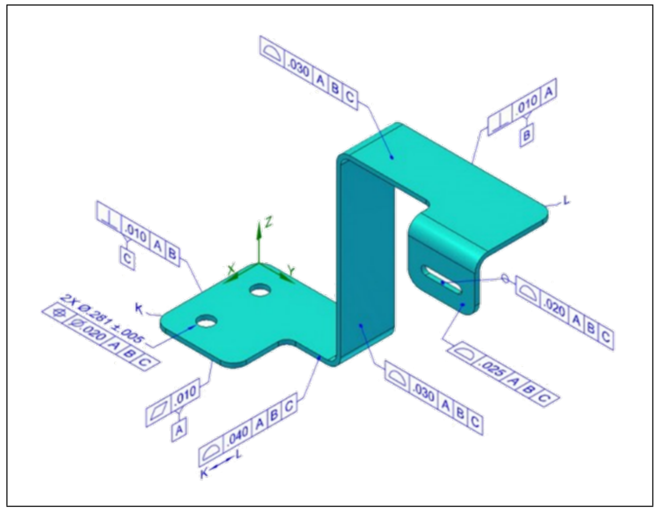

FIGURE 1: A part with typical GD&T annotations

Standards for visually representing GD&T are ASME Y14.5-2008 [5] and ISO 1101:2012 [6]. These standards have been used for decades and cover a wide range of variations in product features, such as flatness, perpendicularity, location, circularity, and straightness. It is important to distinguish between the presentation of GD&T information, and the representation. Representation includes the information necessary to fully define the meaning of the tolerances, without resorting to visual depiction.

Standards for exchanging semantic information on nominal product information have been available for many years through the ISO 10303 "STEP" family [7]. STEP is a methodology for describing product information throughout its life cycle. CAD systems exchange STEP information through their import and export facilities, converting between native formats and STEP so that partners in the supply chain can each use software that best fits their needs. The advent of ISO 10303 Part 242 in 2014 standardized an information model for the meaning of GD&T. This improvement allows for the more full automation of information exchange further downstream, into production and quality assurance. An example from this information model is depicted in Figure 2.

```
ENTITY flatness_tolerance;
name  : label;
description  : OPTIONAL text;
magnitude  : OPTIONAL length_measure_with_unit;
toleranced_shape_aspect : geometric_tolerance_target;
END_ENTITY;
```

FIGURE 2: EXPRESS entity showing an example of semantic GD&T for flatness

PROCESS PLANNING

The use of semantic GD&T greatly simplifies the task of exchanging models between CAD systems from different vendors, a problem faced by systems integrators and suppliers who must support different formats from different customers. It also opens the possibility of more fully automating the downstream processes that are influenced by tolerance requirements. Following the use case scenario, from the assembly design, a planner determines the order of operations needed to complete the full assembly, including any tooling requirements or other resources necessary to carry out individual steps depending on their capabilities. This process is variously known as computer-aided process planning (CAPP) or computer-aided manufacturing (CAM) depending on the domain. These systems automate much of the tedious calculating of end-effector motions or tool paths, and provide graphical aids for visualizing results, but rely on interaction from a human expert to guide the process and make determinations on suitable alternatives due to the absence of tolerances and other performance requirements. Even when these CAPP and CAM systems can use this information to automate the planning process, it must be manually entered. Here we see the full value of the exchange of semantic GD&T both in preserving information exchanged between design systems, and streamlining its use in downstream planning processes.

To determine if a robot can achieve the tolerance requirements, and to compute optimal robot motion, robot performance metrics are needed. The most common performance metric for industrial robots is repeatability, or the consistency with which the robot returns to a given point. This is due to the prevalence of the method of teach programming, where the robot is physically brought to a series of poses that are recorded for later playback in a programmed sequence. This method of programming can be time consuming, and requires a person to do the teaching while the robot is taken out of production. While it is cost effective for high-volume applications, it is often faster to do small jobs manually. In contrast, off-line programming uses models of the parts, robot, and work volume to generate sequences of nominal robot poses, relying on the accuracy of the robot to achieve the computed points. This is analogous to the primary method of programming machine tools, using CAD data and a CAM system. However, due to their construction, robots are typically much less accurate than machine tools, and off-line programming must be supplemented with other techniques to increase the accuracy to acceptable levels. Calibration can be done, but errors vary considerably throughout the work volume due to flexing of robot links, so calibration tables must be generated at many locations and orientations. This process is known as *error mapping*. Because robots typically have low stiffness and will deflect appreciably under loads, error maps are only effective when developed under loaded conditions. If these vary during an application, error mapping may be ineffective. Process sensing is effective under varying loading conditions, because the actual location of the robot is measured and deviations can be adjusted in real time. This requires an increased investment in sensing technology, and possible changes in the process to reduce noise or occlusions. With a combination of calibration and sensing, robot accuracy can be increased to approach its repeatability.

Supplemental sensing technology is often brought to bear in robot applications driven by offline programs, after calibration and error mapping have reduced systematic errors. A hybrid technique is to use a set of taught points at key areas in the work volume, benefitting from high repeatability, and calculating offsets from these taught points based in sensor data from cameras or other vision systems. Using this technique, accuracy errors are reduced to the much smaller neighborhood around the offset. Robots have been successfully used in this way for semiconductor chip placement, with placement accuracy requirements well below the 1-mm level. ISO 9283 [8] specifies methods for determining overall values of repeatability and accuracy that are guaranteed throughout the work volume for a given load. These include accuracy and repeatability for a target point under conditions of varying approach, accuracy and repeatability for motion along paths, position settling time and overshoot, cornering deviations, static compliance, and other metrics. These values are worst-case values, and in many cases the accuracy of a robot is much greater. On the other hand, these metrics are valid only for the test conditions, such as at 20 $^\circ$C, and performance can be worse under other conditions. It is desirable to know where these regions of higher or lower accuracy lie, so that application programs can be placed accordingly and benefit from higher performance. To do this, a more sophisticated model of robot accuracy is needed. The Open Source Robotics Foundation's Robot Operating System (ROS) provides an information model for robots that helps achieve this, called the Unified Robot Description Language (URDF). URDF allows the description of geometric, kinematic, and dynamic information about the links and joints of a robot. Figure 3 shows a sample representation of a robot description.

The xyz and rpy attributes are the Cartesian and orientation (roll, pitch, and yaw) of the transforms between the links and joints. Other attributes define dynamic properties for mass, inertia, and friction. Note that there is no uncertainty associated with any of these values. Automated process planning could be im-

```xml
<link name="link1">
  <inertial>
    <origin xyz="0 0 0.5" rpy="0 0 0"/>
    <mass value="1"/>
    <inertia ixx="100" ixy="0" ixz="0"
          iyy="100" iyz="0" izz="100"/>
  </inertial>

  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="1" length="0.5"/>
    </geometry>
  </collision>
</link>

<joint name="joint1">
   <origin xyz="0 0 1" rpy="0 0 3.1416"/>
   <parent link="link1"/>
   <child link="link2"/>
   <dynamics damping="0.0" friction="0.0"/>
   <limit effort="30" velocity="1.0"
        lower="-2.2" upper="0.7"/>
</joint>
```

FIGURE 3: Sample Unified Robot Description Format (URDF) showing how links and joints are represented

proved if this information were available when selecting robots for assembly tasks.

Run-time tolerances are supported by ROS in the form of poses with covariance, where the pose representation is Cartesian location and quaternion orientation. The tolerances (or uncertainty, depending on the context) are expressed using a covariance matrix on the Cartesian location, and the roll, pitch, and yaw equivalents of the quaternion orientation. This covariance approach to orientation is not well suited for uncertainty analysis, a point described in detail in the section on the use of orientation uncertainty in assembly tasks.

Robot path tolerances are also available in ROS, using tolerances on end points.

These tolerances apply to the joints values, not the Cartesian values, and so are dependent on the robot selected. ROS does provide various Cartesian motion planners, such as Descartes [9], but tolerance information is experimental. For example, orientation tolerances using a cone about the tool's directional axis have been used for insertion tasks.

QUALITY ASSURANCE

The Quality Information Framework (QIF) [10] is an ANSI standard sponsored by the Dimensional Metrology Standards Consortium (DMSC) that defines an integrated set of Extensible Markup Language (XML) information models to enable the

```
FollowJointTrajectoryActionGoal
    trajectory_msgs/JointTrajectoryPoint[] points
      float64[] positions
      float64[] velocities
      float64[] accelerations
      float64[] effort
      duration time_from_start
   control_msgs/JointTolerance[] path_tolerance
     float64 position
     float64 velocity
     float64 acceleration
   control_msgs/JointTolerance[] goal_tolerance
     float64 position
     float64 velocity
     float64 acceleration
   duration goal_time_tolerance
```

FIGURE 4: Tolerances on paths and goals for robot trajectories in ROS

effective exchange of metrology data throughout the entire manufacturing quality measurement process from product definition to inspection planning to execution to analysis and reporting. QIF handles feature-based dimensional metrology, quality measurement planning, first article inspection, and discrete quality measurement. QIF is gaining attention as an important quality technology [11–16].

QIF is based on XML, and uses terminology and semantics from the inspection world to represent the various elements in the QIF specification. The QIF information models are contained in files written in the XML Schema Definitions (XSD). The QIF XSD Version 2.0 models consists of six application schema files QIFRules, QIFResults, QIFPlans, QIFProduct, QIFStatistics, and QIFMeasurementResources bundled into a QIF Document. QIF also includes a library of XSD schema files containing information items used by all QIF applications (Auxiliary, Characteristics, Expressions, Features, GenericExpressions, Geometry, IntermediatesPMI, Primitives, PrimitivesPD, PrimitivesPMI, Statistics, Topology, Traceability, Units, and Visualization).

The flow of QIF data starts with generation of CAD + PMI data exported as QIF Model Based Design (MBD) application data. Quality planning systems import the MBD and generate Plans (whats), then import Resources and Rules information and export Plans (whats and hows). Programming systems import Plans to generate Dimensional Measurement Equipment (DME) specific programs, or general instructions to guide inspection. Dimensional measurement equipment executes programs and evaluates characteristics of a single manufactured part or assembly and exports the measurements as Results. Analysis systems, typically performing statistical process control, import single parts Results and generate analysis of multiple part batches as QIF Statistics data.

The role of uncertainty in QIF would be to characterize the

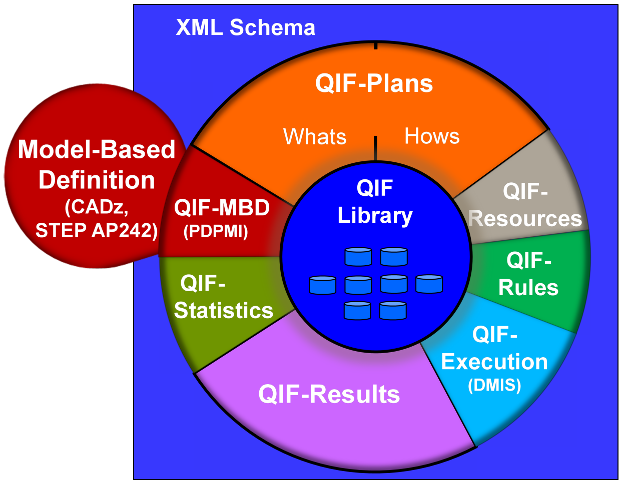

FIGURE 5: QIF 2 Architecture

statistical distribution of the error. Underlying all measurement data in QIF is the assumption that the QIF Dimensional Measurement Equipment is of order of magnitude ten times more accurate than the inspected feature. Thus, a feature characteristic of 1 mm would require a DME that measures to .1 mm accuracy. This inspection rule-of-thumb allays many concerns, but is not absolute. In fact, QIF has a separate section on detailing the environment and other inspection factors, (e.g., ambient temperature), that could contribute to quality inspection errors.

QIF provides for inspection measurements to have an attached uncertainty associated with the result. In QIF, the measurement actual values (e.g., the diameter of an instance of CircleFeatureActualType) are elements that can have the optional attributes “meanError” and “combinedUncertainty”. Those are attributes of the ActualDecimalType in Units.xsd shown below. Measurement data which correspond to actual value types in QIF are derived from ActualDecimalType.

```
<xs:complexType name="ActualDecimalType">
  <xs:annotation>
   <xs:documentation>
    An ActualDecimalType defines a SpecifiedDecimalType
   with two additional optional attributes: meanError
   and combinedUncertainty. These attributes should
   either both be used or both be omitted.
   </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
   <xs:extension base="SpecifiedDecimalType">
    <xs:attribute name="combinedUncertainty"
     type="NonNegativeDecimalType">
     <xs:annotation>
      <xs:documentation>
       The optional combinedUncertainty attribute
       is a value expressing the combined
       uncertainty assigned to the
       SpecifiedDecimalType.
      </xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="meanError"
     type="NonNegativeDecimalType">
     <xs:annotation>
      <xs:documentation>
       The optional meanError attribute is a
       value expressing the mean error
       assigned to the SpecifiedDecimalType.
      </xs:documentation>
     </xs:annotation>
    </xs:attribute>
   </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Determining the uncertainty for a QIF actual measurement is an optional reporting attribute often viewed as unnecessary. It would be preferable if the vendor of the inspection software could have access to the numerous factors that play into determining the uncertainty, for example, ambient temperature, last calibration, and the model of DME reported benchmarks. If this were the case, taking inspection measurements in an intemperate 30 °C ambient temperature would result in a large uncertainty.

SENSOR UNCERTAINTY

MTConnect is an integration standard to solve the “Island of Automation” problem in the discrete manufacturing industry. MTConnect is an open, royalty-free standard that uses prevalent commercial off-the-shelf technology - XML and HTTP. The MTConnect intent is to foster greater interoperability between controls, devices, and software applications by publishing data over networks using the Internet Protocol [17]. Over the course of the MTConnect standards development, sensors have gone from an implicit modeling role within devices to an explicit information model. This is especially important as sensors can provide real-time production information to better understand and optimize manufacturing activities in a factory.

A sensor may measure in one dimension, such as temperature or acceleration whose fluctuations are a function of time, or a sensor can be multidimensional, such as an image which can be a function of two or three-dimensional space and time. MTConnect Part 2 V1.2 [18] presents a sensor model that defines sensor data formats and communication interfaces. MTConnect includes sensor information models for sensors of one dimension: acceleration, angular acceleration, angular velocity, amperage, angle, concentration, conductivity, direction, displacement, electrical energy, flow, frequency, fill level, linear force, load, mass, pH, pressure, position, power factor, resistance, rotary velocity, sound level, strain, temperature, time, tilt, torque, volt ampere, volt ampere reactive, velocity, viscosity, voltage, and wattage. At present, MTConnect only provides information models for one-dimensional sensor values.

In MTConnect, a sensor is comprised of two major components - a sensing element and a sensor interface. A sensing element provides a signal or measured value. It is modeled as an MTConnect DataItem. Each sensor model includes a sensing element, calibration, signal conditioning, and analog-to-digital conversion (ADC) information [19]. A sensor interface has capabilities, such as signal processing, conversion, and communications, and it is modeled as an MTConnect Component called Sensor. Each sensor interface may have multiple sensing elements, which represent the data for a variety of measured values. Further, when an MTConnect sensor represents multiple sensing element(s), each sensing element is represented by a Channel. A Channel represents one sensing element and can have its own attributes and Configuration data.

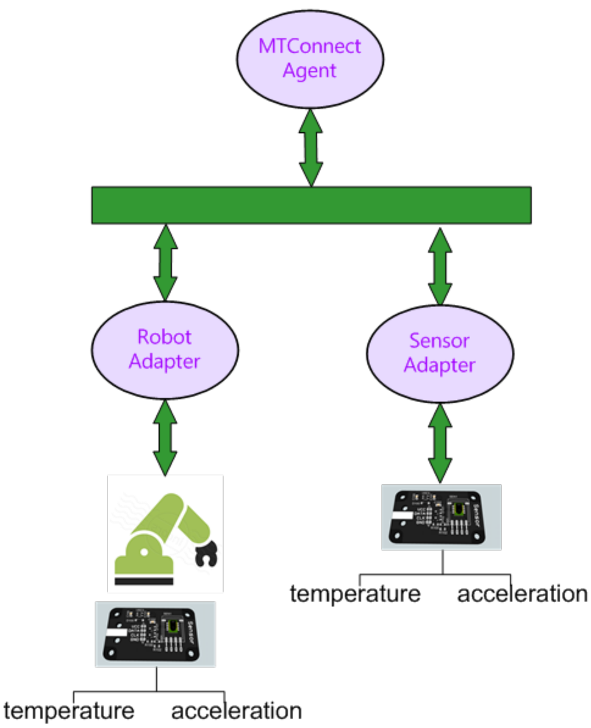

FIGURE 6: Sensor Architecture of MTConnect

The MTConnect architecture has an "Agent" that is a web service and acts as a "bridge" between an MTConnect "Device" and a Client Application. An MTConnect Device is a piece of equipment, like a robot, organized as a set of components that provide data. An MTConnect "Adapter" is a process that provides a data stream from a device to the agent. MTConnect defines XML information models in order to exchange standard data items. MTConnect has a so-called "Streams" information model that defines data reporting of Events, Samples, Conditions, and Asset data items through continually-updated channels. MTConnect has a standard configuration information model that provides data as a "probe". Figure 6 shows the duality of MTConnect sensor configuration as two types: A Sensor built into an MTConnect Device (i.e., Robot) and an independent Sensor as a standalone MTConnect Device. In Figure 6, the robot "contains" the sensors and will report sensed values as part of its data reporting.

The MTConnect standard provides an XML configuration report colloquially known as a "probe". The MTConnect probe enumerates the DataItems that will be reported in a "stream". The MTConnect probe represents sensor values as a DataItem sample, which includes XML attributes for sensor type, units, name, and XML id. Also included in the DataItem XML for acceleration is an attribute called statistic which is calculated specific to the sensor DataItem. The statistic attribute indicates that the data has been processed using a statistical operation like average, mean, or root square. Examples of MTConnect statistic attributes are AVERAGE, MINIMUM, MAXIMUM, ROOT_MEAN_SQUARE, RANGE, MEDIAN, MODE, and STANDARD_DEVIATION. Below is the XML reported from the temperature (thermistor) sensor when querying the MTConnect Agent for a system configuration (i.e., probe) [20].

```
<DataItem type="TEMPERATURE" category="SAMPLE"
           name="Rtemp" id="temp" units="CELSIUS" >
   <Source componentId="s1">channel:1</Source>
</DataItem>
```

The above XML uses the MTConnect Devices XSD as the Information Model schema to describe each device and its data items available. Below, the XML shows the data reporting as configured using the MTConnect Streams XSD as the Information Model. MTConnect streams describe a time series of data items, including samples, events, and conditions. Below the Streams snippet describes the sensor readings for sensor one, the "temp" temperature sensor:

```
<Temperature dataItemId="temp"
  timestamp="2017-03-07T21:17:42.814257"
  name="Robotemp"
  sequence="839777883"> 19.9
</Temperature >
```

A sensor measured value is rarely observed in isolation from a combination of noise and distortion [21]. In fact, noise and distortion are the fundamental source of the limitations in the accuracy of sensor measurements. For example, the sources of accelerometer noise can be broken down into the electronic noise from the circuitry that is converting the motion into a voltage signal or mechanical noise from the sensor itself. If an MTConnect sensor sample is returned as a data item, especially a sensor that may be noisy and prone to providing outlier values, a quantification of the error would be desirable. For example, temperature sensors typically provide a statement that the thermistor is accurate to $\pm$ 3 degrees. To translate this into an uncertainty value (or statistical distribution of the error), we assume that the error mean is zero and that three standard deviations from this mean provides approximately 99.7 % of the error statistical population assuming a normal error distribution. Of note, it is assumed that the vendor in providing a bounds on the sensor accuracy, has used a sufficiently large number of observations to provide a reliable estimate of the accuracy. For the thermistor example, the three-degree bounded limit translates into a standard deviation uncertainty of one degree.

It would be desirable for MTConnect to report the uncertainty associated with any measurement, especially sensors, since noisy measurements or outliers could pass through the data reporting system as ground truth, when in fact the numbers are abnormal and should be discarded or should be filtered. Below is an MTConnect sensor that incorporates uncertainty as an associated MTConnect data item:

```
<DataItem type="TEMPERATURE"
 category="SAMPLE"
 name="Robotemp"
 id="temp" units="CELSIUS" >
   <Source componentId="s1">channel:1</Source>
</DataItem>
<DataItem type="TEMPERATURE"
 statistics="STANDARD_DEVIATION"
 category="SAMPLE" name=" RobotempUnc"
 id="tempuncertainty" units="CELSIUS" >
</DataItem>
```

Now, every sensor temperature sensor measurement will have an associated uncertainty value associated with the reading. Below is MTConnect data stream XML snippet reporting the temperature sensor measurement, but now containing an associated data item to explicitly state the uncertainty of the temperature sensor measurement. For example, in the temperature sensor uncertainty reading, we assume measurement units correspond to those described in the probe XML given above, so the temperature uncertainty is given as a standard deviation of one degree

```
<Temperature dataItemId="temp"
 timestamp="2017-03-07T21:17:42.814257"
 name="Robotemp"
 sequence="839777883"> 19.9
</Temperature >
<Temperature dataItemId="tempuncertainty"
 timestamp="2017-03-07T21:17:42.814257"
 name="RobotempUnc" sequence="839777883"> 1.0
</Temperature >
```

Now, the uncertainty of the MTConnect data item can also be used to register abnormal changes detected internally by the sensor that may affect its measurements. For example, suppose an acceleration sensor contains an internal temperature monitor and detects that its board's internal temperature is exceeding 40 °C, which adversely effects the sensor operation and its acceleration measurements. In this case, the uncertainty value could be negative, indicating the measures can never be equal to the mean. Although there is no explicit MTConnect facility for expressing multidimensional sensor data such as images, MTConnect has the ability to incorporate and transport XML data independent of the core MTConnect information models. Using the MTConnect "asset" model, MTConnect agents can pass sensor data as embedded "asset" data. This facility along with asset notification and the "statistic" attribute can form the basis for reporting 2D and 3D sensor data. A brief overview will show the deployment of the MTConnect "asset" mechanism.

MTConnect defines "assets," which use an associative array of key/value stores to store the XML. This allows the ability to collect and report entire XML documents as they change within applications. Below, the XML shows how an AssetChanged tag with an asset type Sensor and READING value that would be updated within the MTConnect XML query to indicate new quality results from an inspection.

```
<AssetChanged dataItemId="multidimsensor_asset_chg"
timestamp="2016-09-08T19:42:16.855924Z" sequence="46"
assetType="Sensor"> Reading
</AssetChanged>
```

For our implementation, the Quality Measurement Results (QMR) XML Schema was used to develop the XML that is then assessable via the Internet with the following query to an MTConnect agent –

```
http://xxx.xxx.xxx.xxx/asset/INSPECTION?type=Part
```

where xxx.xxx.xxx.xxx is the ip address of the MTConnect server, and which returns a so-called "blob" of otherwise unstructured multidimensional sensor data, outlined by the following XML snippet:

```
<MultiDimSensor timestamp="2011-07-25T13:55:22"
 assetId="Reading">
<Reading>
<!-- this is the start of the sensor blob data -->
. . .
</Reading>
</MultiDimSensor>
```

Multidimensional sensors are affected by noise and distortion, so the uncertainty should reflect this. Although it is possible to embed another sensor data item in the MTConnect system configuration to reflect the uncertainty, it would be easier to incorporate an existing standard such as the Metadata Working Group Standard [22] to handle image data.

In theory, using the MTConnect sequence number to package data, a client could query an MTConnect agent to stream multidimensional data. No tests were done to verify this capability.

USE OF ORIENTATION UNCERTAINTY IN ASSEMBLY TASK

If a CAD model of a part is available, any Point of Interest (POI) associated with that part can be determined using six-degree-of-freedom data acquired by a pose measuring system. Uncertainty of a selected POI is derived from uncertainty of pose measurement. Uncertainty of a part's location propagates homogeneously to its all POIs but propagation of orientation uncertainty may be more complicated and may have directional dependence.

FIGURE 7: Two vector bars mounted rigidly to a part form a local frame which is tracked by a pose measuring system. The location of any POI on a part (marked by an arrow) can be determined from dynamically tracked pose and fixed location of the POI in a local frame. Uncertainty in pose measurement must be propagated to the POI.

Let us assume the j-th noisy pose measurement yields $\boldsymbol{R}_j$ rotation and $\boldsymbol{t}_j$ translation. If a location of a POI in the CAD coordinate frame is $\boldsymbol{U} = U\ \boldsymbol{u}(\theta,\varphi)$, where $U = \|\boldsymbol{U}\|$ and a unit vector u points in the direction of azimuth φ and elevation θ, then the location of a POI on the rotated object in the coordinate frame of the pose measuring system is

$$\boldsymbol{U}_i = U\boldsymbol{w}_j + \boldsymbol{t}_j \quad , \tag{1}$$

where a unit vector $\boldsymbol{w}_j$ points to a rotated POI

$$\boldsymbol{w}_i = \boldsymbol{R}\boldsymbol{u}(\theta,\varphi) \quad . \tag{2}$$

Uncertainty in the orientation $\boldsymbol{R}_j$ is propagated to $\boldsymbol{w}_j$ and the noisy orientation can be represented as

$$\boldsymbol{R}_j = \bar{\boldsymbol{R}}\,\Delta\,\boldsymbol{R}_j \quad , \tag{3}$$

where the mean orientation $\bar{\boldsymbol{R}}$ is used as the approximation of the unknown true orientation and $\Delta\boldsymbol{R}_j$ is a small random rotation which can be expressed in axis-angle representation (a_j, ρ_j) as

$$\Delta\boldsymbol{R}_j(\boldsymbol{a}_j,\rho_j) \approx \boldsymbol{I} + \begin{bmatrix} 0 & -q_j^z & q_j^y \\ q_j^z & 0 & -q_j^x \\ -q_j^y & q_j^x & 0 \end{bmatrix} \quad , \tag{4}$$

where I is the identity matrix and

$$\boldsymbol{q}_j = \rho_j\ \boldsymbol{a}_j \quad . \tag{5}$$

From N repeated measurements of noisy orientation $\boldsymbol{R}_j,\ j = 1,\ldots,N$, a covariance matrix of orientation data $cov(q)$ can be calculated and its three eigenvalues $\{\wedge_1,\wedge_2,\wedge_3\}(\wedge_1 < \wedge_2 < \wedge_3)$ and associated eigenvectors $\{\boldsymbol{e}_1,\boldsymbol{e}_2,\boldsymbol{e}_3\}$ can be calculated. Noisy rotations $\boldsymbol{R}_j$ can also be used to investigate distribution of transformed unit vectors w_j in (2). One way of doing this is to characterize a distribution of μ_j defined as the angle between $\boldsymbol{w}_j$ and the mean unit vector $\bar{\boldsymbol{w}}$. The spread of this distribution may be characterized by the angular uncertainty σ: smaller σ correspond to a tighter concentration of noisy $\boldsymbol{w}_j$ around the mean direction $\bar{\boldsymbol{w}}$ and larger σ correspond to wider spread of $\boldsymbol{w}_j$ around $\bar{\boldsymbol{w}}$.

For a large class of pose measuring systems, angular uncertainty depends on direction, i.e., $\sigma = \sigma(\theta,\varphi)$ where azimuth and elevation angles determine direction of unit vector $u(\theta,\varphi)$ in (2). Directional distribution of $\sigma(\theta,\varphi)$ is closely correlated with the distribution of axes a_j of small noisy rotations $\Delta\boldsymbol{R}_j$ in (4,5), as can be seen in Figure 8 and Figure 9.

Thus, depending on the directions of eigenvectors of the covariance matrix of the orientation data, different POIs of the measured object will be affected differently. Such situation is shown in Figure 10.

In summary, the covariance matrix of the orientation data needs to be carefully analyzed, as not only the values of its diagonal elements (variances of orientation data) and off-diagonal elements (correlation coefficients) are important.

CONCLUSION

Improvements in sensing and control have enabled robots to be deployed in difficult-to-automate applications characterized

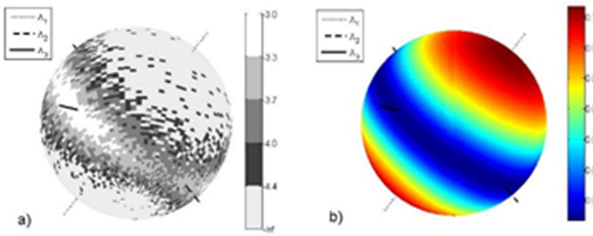

FIGURE 8: Histograms of axes a_j of small random rotations $\Delta\boldsymbol{R}_j$ on log scale in (a), $-inf$ indicates empty bins; (b) directional distribution of angular uncertainty σ in $[mrad]$. Based on orientation data obtained with an Optitrack Duo, the length of data $N > 50,000$. Plotted directions of eigenvectors correspond to eigenvalues $\wedge_1, \wedge_2, \wedge_3$ of the covariance matrix of the orientation data q.

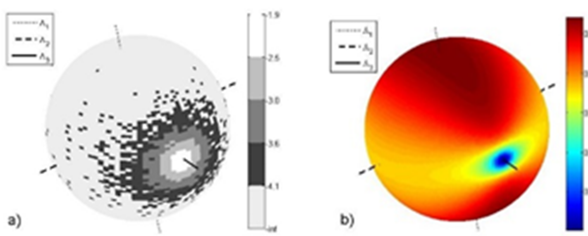

FIGURE 9: The same as in Figure 8 but based on orientation data obtained with another pose measuring system (iGPS).

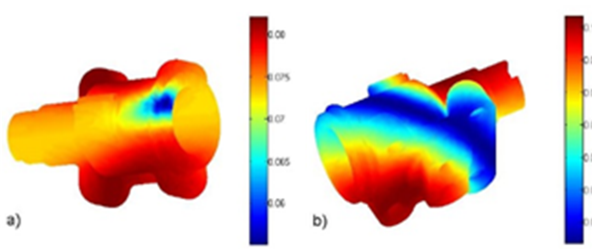

FIGURE 10: Angular uncertainty mapped onto a CAD model based on the data acquired with: a) iGPS (same as in Figure 9b) and b) OptiTrack Duo (same as in Figure 8b).

by often-changing requirements for operation in uncertain environments. To more fully automate the optimization of these activities, it is important to be able to exchange information about the tolerance on required performance, and the uncertainty in measured performance. Standards for the exchange of this information have been revised with this objective, to varying success. This paper examined the support of these standards using a use case scenario, and showed the results of a case study on the use of orientation uncertainty to optimally place objects in a robot workcell. Future work by the authors will examine the performance improvements achievable in robot path control.

DISCLAIMER

Commercial equipment and software, many of which are either registered or trademarked, are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] Proctor, F. M., Hoorn, G. v. d., and Lipman, R., 2016. “Automating robot planning using product and manufacturing information”. *Procedia CIRP,* ***43***, pp. 208 – 213.
[2] Huang, Y. F., and Lee, C. S. G. “A framework of knowledge-based assembly planning”. In Proceedings. 1991 IEEE International Conference on Robotics and Automation, pp. 599–604 vol.1.
[3] Dini, G., and Santochi, M., 1992. “Automated sequencing and subassembly detection in assembly planning”. *CIRP Annals - Manufacturing Technology,* ***41***(1), pp. 1–4.
[4] Proctor, F., Balakirsky, S., Kootbally, Z., Kramer, T., Schlenoff, C., and Shackleford, W., 2016. “The canonical robot command language (crcl)”. *Industrial Robot: An International Journal,* ***43***(5), pp. 495–502.
[5] American Society of Mechanical Engineers, 2008. ASME Y14.5-2008, Dimensioning and Tolerancing.
[6] International Organization for Standardization, 2004. ISO 1101:2004, Geometrical Product Specifications (GPS) - Geometrical tolerancing – Tolerances of form, orientation, location and run-out.
[7] International Organization for Standardization, 1994. ISO 1101:2004, ISO 10303-1:1994 Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles.
[8] International Organization for Standardization, 1998. ISO 9283, Manipulating industrial robots - Performance criteria and related test methods.
[9] ROS Industrial. Descartes. http://wiki.ros.org/descartes. [Online: accessed 8-March-2017].
[10] ANSI, 2014. ANSI/QIF Part 1-2014: Quality Information Framework (QIF) an integrated model for manufacturing quality information part 1: Overview and fundamental principles version 2.0. Report, Dimensional Metrology Standards Consortium, Inc. (DMSC).
[11] Kramer, T., Stone, R., Hoffman, M., Hoffman, S., and Rippey, W., 2012. “Design and usage guide for version 0.92 of the Quality Information Framework data model and

XML schemas". *US Department of Commerce, National Institute of Standards and Technology Technical Note 1777*.

[12] Morey, B., 2014. "Metrology takes flight". *Manufacturing Engineering,* ***152***(2).

[13] Pippenger, B. "Three-dimensional model for manufacturing and inspection". In Proceedings ASME Turbo Expo 2013: Turbine Technical Conference and Exposition.

[14] Zhao, Y., Kramer, T., Brown, R., and Xu, X., 2011. *Information modeling for interoperable dimensional metrology*. Springer Science & Business Media.

[15] Zhao, Y. F., Horst, J. A., Kramer, T. R., Rippey, W., and Brown, R. J., 2012. "Quality information framework - integrating metrology processes". *IFAC Proceedings Volumes (IFAC-PapersOnline),* ***14***(Part 1), pp. 1301 – 1308.

[16] Zhao, Y., Kramer, T., Rippey, W., Horst, J., and Proctor, F. "An integrated data model for quality information exchange in manufacturing systems". In Proceedings of the 37th International MATADOR 2012 Conference, pp. 239 – 242.

[17] MTConnect FAQ. www.mtconnect.org/institute/mtconnect-overview/faqs.aspx. [Online: accessed 8-March-2017].

[18] MTConnect. MTConnect Standard Version 1.3.1, Part 2: Components and data items. www.mtconnect.org/standard-documents. [Online: accessed 8-March-2017].

[19] Lee, K. B., Song, E. Y., and Gu, P. S. "A sensor model for enhancement of manufacturing equipment data interoperability". In ASME 2012 International Mechanical Engineering Congress and Exposition, pp. 705–712.

[20] MTConnect Sensors. http://www.mtcup.org/wiki/Sensors. [Online: accessed 8-March-2017].

[21] Vaseghi, S. V., 2008. *Advanced digital signal processing and noise reduction*. John Wiley & Sons.

[22] Metadata Working Group . Specifications. www.metadataworkinggroup.com. [Online: accessed 8-March-2017].