Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE DÉFENSE

Standards for Representation in Autonomous Intelligent Systems

Promoting Interoperability Amongst Autonomous Intelligent Systems

D. Erickson
DRDC Suffield

Technical Report
DRDC Suffield TR 2005-228
December 2005

Canada

Standards for Representation in Autonomous Intelligent Systems

Promoting Interoperability Amongst Autonomous Intelligent Systems

D. Erickson
DRDC Suffield

Defence R&D Canada – Suffield

Technical Report
DRDC Suffield TR 2005-228
December 2005

Author

D. Erickson

Approved by

D. Hanna
Head/Tactical Vehicle Systems Section

Approved for release by

Dr. P. A. D'Agostino
Head/Document Review Panel

Abstract

This memorandum outlines the standards for nomenclature, definitions, units of measure, and representations for use between Autonomous Intelligent Systems (AIS). The motivation for these standards are twofold :

1. to address the needs of software development and experimental data collection,

2. to enable all future systems to be interoperable.

This common standard will make it easier to define contractual requirements and will resolve confusion concerning the data received across an interface from other projects at DRDC as well as allied research. Some of the major conventions adopted here are described in Joint Architecture for Unmanned Systems (JAUS), others are System International (SI) standard (JAUS adopts the SI units as the required units of measure). With these standards, we will be compatible with JAUS research and development ongoing at participating US labs. It is recommended that AIS adopt these standards for all future work.

Résumé

Ce mémorandum souligne les normes de nomenclature, des définitions, des unités de mesures et des représentations devant être utilisées entre les Systèmes intelligents autonomes (SIA). L'objectif de ces normes est à double volet. Il s'agit de :

1. Traiter des besoins en mise au point de logiciels et en collecte de données expérimentales.

2. Permettre à tous les systèmes futurs d'être interexploitables.

Ces normes communes faciliteront la définition des besoins contractuels et éviteront la confusion concernant les données reçues par interface provenant aussi bien d'autres projets ayant lieu à RDDC que de la recherche connexe. Certaines des conventions majeures adoptées ici sont décrites dans Architectures conjointes pour des systèmes sans pilote, d'autres sont les normes adoptées par le Système international (SI) (Architectures conjointes adoptent les unités SI comme unités requises pour les mesures). À l'aide de ces normes, nous serons compatibles avec la recherche et développement continue d'Architectures conjointes pour les systèmes sans pilote, ayant lieu dans les laboratoires américains participants. On recommande que SIA adopte ces normes pour tous ses travaux futurs.

This page intentionally left blank.

Executive summary

This memorandum outlines the standards for nomenclature, definitions, units of measure, and representations for use between systems, subsystems, devices, nodes, and processes. The aim of this document is to reduce ambiguity by establishing common rules for the interpretation of information. The adoption of these standards within the work by DRDC and from contractors who assist the program will increase the efficiency of implementation and increase the interoperability of projects and components. It is recommended that Autonomous Intelligent Systems (AIS) adopt these standards for all future work.

The importance of the recommended standards was reinforced during the Autonomous Land Systems (ALS) program development. Independent software development employed various standards for units of measure, representations, and definition lead to difficulties. Some software components adopted unique representations, or used Imperial units of measure. This lead to data confusion in a number of cases between data sources and data sinks, making the entire effort much more difficult than would have been the case had *a priori* standards been in place.

The reasons for the selection of the standards reported in this publication are manifold: some are due to common convention, some are chosen in order to be compliant with Joint Architecture for Unmanned Systems (JAUS) representations, and a few are arbitrary. However, whatever their origin, it is important to point out that the adoption of comprehensive set of standards is vital in order to reap the benefit of common representations. Data exchange without common semantics leads to confusion and inefficiency, and is unacceptable in a research program that relies on a considerable amount of systems integration and interoperability. All of the conventions outlined are important to provide context to the data that is received from other systems, subsystems, devices, nodes, and processes.

The standards outlined are intended for use at the level of interface between systems, subsystems, devices, nodes, and processes. It is obvious that some devices by their design do not inherently adopt these conventions (e.g. a GPS that represents angle as degrees as opposed to radians) and so the absolute adoption of these standards is not practical. Any system, subsystem, device, node, or process may use its own internal representation. When that system, subsystem, device, node, or process attempts to communicate outside its internal representation, then it is the responsibility of the developer to convert the representation to the proposed standards. As long as all systems, subsystems, devices, nodes, and processes hold to this standard at the interface, then interoperability will be achieved.

D. Erickson. 2005. Standards for Representation in Autonomous Intelligent Systems. DRDC Suffield TR 2005-228. Defence R&D Canada – Suffield.

Sommaire

Ce mémorandum souligne les normes de nomenclature, des définitions, des unités de mesure et des représentations devant être utilisées entre les systèmes, sous-systèmes, appareils, nœuds et processus. Le but de ce document est de réduire l'ambiguïté en établissant des règles communes pour l'interprétation de l'information. L'adoption de ces normes dans les travaux de RDDC et par les entrepreneurs participant au programme augmentera l'efficacité de l'implémentation et de l'interopérabilité des projets et des composants. On recommande que les Systèmes intelligents autonomes (SIA) adoptent ces normes dans leurs travaux futurs.

L'importance des normes qui sont recommandées a été renforcée durant la mise au point du programme des Systèmes terrestres autonomes (STA). Le développement indépendant de logiciels qui employait une variété de normes d'unités de mesures, représentations et définitions a entraîné des difficultés. Certaines composantes de logiciels ont adopté des représentations uniques ou utilisé des unités de mesures anglo-saxonnes. Ceci a abouti à la confusion des données entre les sources de données et les puits de données dans un certain nombre de cas, compliquant beaucoup plus les efforts qu'il en aurait été le cas si les normes avaient été établies *a priori.*

Les raisons soutenant la sélection des normes documentées dans cet article sont à plusieurs volets : certaines sont dues aux conventions communes, certaines sont choisies pour concorder avec les représentations de Architectures conjointes pour les systèmes sans pilote et certaines sont arbitraires. Quelque soit leur origine, il est cependant important de remarquer que l'adoption d'un ensemble compréhensif de normes est vitale pour être en mesure de bénéficier des représentations communes. L'échange de données sans sémantique commune aboutit à la confusion, est inefficient et ne peut être intégré à un programme de recherche qui est basé sur une quantité considérable d'intégration et d'interopérabilité des systèmes. Toutes les conventions soulignées ici sont essentielles à fournir un contexte aux données reçues d'autres systèmes, sous-systèmes, appareils, nœuds et processus.

On prévoit que les normes soulignées seront utilisées au niveau de l'interface entre les systèmes, sous-systèmes, appareils, nœuds et processus. Il est évident que certains appareils, dus à leur concept, n'adoptent pas ces conventions de manière inhérente (par ex : un GPS qui représente un angle en degrés au lieu de radians) et par conséquent, l'adoption absolue de ces normes n'est pas pratique. Chacun de ces systèmes, sous-systèmes, appareils, nœuds ou processus peut utiliser sa propre représentation interne. Quand ce système, sous-système, appareil, nœud ou processus tente de communiquer à l'extérieur de sa représentation interne, le développeur devient responsable de convertir la représentation selon les normes proposées. Il suffit que tous les systèmes, sous-systèmes, appareils, nœuds et processus adhèrent à ces normes au niveau de l'interface pour que se réalise l'interopérabilité.

D. Erickson. 2005. Standards for Representation in Autonomous Intelligent Systems.
DRDC Suffield TR 2005-228. R & D pour la défense Canada – Suffield.

Table of contents

List of figures

List of tables

This page intentionally left blank.

1. Introduction

This report outlines the set of definitions, representations, and units of measure that should be adopted by all components within the AIS program. This format is intended to provide a common interface for all software components and a means of unifying the microcontroller and microprocessor elements within the Robot Nervous Systems of AIS robotics. The vision for communications is to unify the requirements of low-level vehicle control with higher level intelligent software and the operator station(s) so that message formats can be used system wide and be transmitted and received by a variety of means. The general definitions of components were taken from the Joint Architecture for Unmanned Systems (JAUS) Reference Architecture Specification [1].

The importance of the recommended standards was reinforced during the Autonomous Land Systems (ALS) program development. Independent software development employed various standards for units of measure, representations, and definition lead to difficiulties. Some software components adopted unique representations, or used Imperial units of measure. This lead to data confusion in a number of cases between data sources and data sinks, making the entire effort much more difficult than would have been the case had *a priori* standards been in place.

The reasons for the selection of the standards reported in this publication are manifold: some are due to common convention, some are chosen in order to be compliant with Joint Architecture for Unmanned Systems (JAUS) representations, and a few are arbitrary. However, whatever their origin, it is important to point out that the adoption of comprehensive set of standards is vital in order to reap the benefit of common representations. Data exchange without common semantics leads to confusion and inefficiency, and is unacceptable in a research program that relies on a considerable amount of systems integration and interoperability. All of the conventions outlined are important to provide context to the data that is received from other systems, subsystems, devices, nodes, and processes.

2. Outline

Section 1. introduced this report. Section 2. outlines the sections of the report. Section 3. addresses some fundamental naming conventions used throughout this document. The origin of the nomenclature is from JAUS [2] and other sources. Section 4. introduces some definitions for use within the context of robotic representations. Section 5. outlines the standard units of measure to be adopted for data pertaining to physical quantities. Section 6. defines the accepted standards for interpreting data. Section 7. discusses the adoption method for these standards. Section 8. concludes the memorandum.

3. Nomenclature

This section details all data standards that will be used for transferring data between components in the ALS program. These standards restrict the use of units so that there is no confusion about the meaning of any data element. This is important to maintain a common interface between multiple programs sourcing and sinking data concurrently. This standard requires that if another data standard is needed for a particular application/process/thread, then the data is translated into the standard laid out here, transported between elements, and then translated back to the desired form at the receiving application/process/thread etc. In general, there will be little conflict because modules using a specific interface will understand the data constraints of that interface. Some of these standards are taken directly from JAUS [1], [2] in order to remain common to JAUS systems.

This document adopts some JAUS nomenclature. In the language of JAUS, a number of terms are used to delineate position within the overall hierarchy of the system and must be understood. These terms describe the different levels of the architecture and define the required internal hierarchical sub-grouping.

System A system is a logical grouping of subsystems. The system definition provides a functional grouping for the full robotic or unmanned capability. This grouping includes all human interface subsystems and unmanned subsystems common with robotic and unmanned applications.

Subsystem A subsystem performs one or more unmanned system functions as a single localized entity within the framework of the System. A subsystem shall provide one or more communication command and control capabilities. A mobile subsystem shall execute mobility commands as a single unit and retain a defined center of gravity relative to all articulations and payloads.

Node A Node defines a distinct processing capability within a subsystem. A node retains a set of coherent functions and shall provide a node manager component to manage the flows and controls of message traffic.

Component A component provides a unique functional capability for the unmanned system. Messages are defined with respect to these capabilities so that context in command and control is provided. A component resides wholly within a Node.

Device For ALS, device shall be taken as synonymous with Component defined above. A Device Interface shall be taken to mean a Component Interface.

Instance Duplication and redundancy of Components are provided by Component Instances. All Components are uniquely addressable using Subsystem, Node, Component and Instance Identifiers.

Process A program operating within a MMU-equipped CPU that has access to resources as if it were the main program execution or the main program execution on a MMU-less processor.

Message A JAUS message is comprised of the message header and associated data fields and are passed between JAUS components.

4. Definitions

The following definitions are included to clarify the terminology used. Some are taken from JAUS [3, 1] in order to remain JAUS compliant.

AttitudeRMS The attitude RMS provides a means of determining the error associated with a reported attitude. This value is measured over time and therefore not representative of the error at any particular attitude value. It provides a statistical measure of the magnitude of the possible error.

DataValidity The Data Validity indicates if the originator of the message had sufficient information to populate the fields indicated accurately. This field is independent of the Presence Vector. This field typically uses an identical mapping as the Presence Vector when a Presence Vector is defined. A set bit ('1') indicates data validity where a clear bit ('0') indicates a possibly invalid field.

Coordinate Frame of Reference - the Coordinate system to which the position and orientation of an object in space is described. Global coordinate reference frames refer to the position of an object with respect to the Earth. Local coordinate reference frames refer to the position of an object with respect to some arbitrary local origin.

Orientation The angle vector which describes the difference in angles from the coordinate frame of reference and the current object's angles. The angle may be a vector of angles in the case of Euler Angles or a set of quaternions.

RMS The root-mean-square (RMS), often used as a synonym for the standard deviation of a variant X, is the square root of the mean squared value of x:

$$\sqrt{\frac{\Sigma_1^N x_i^2}{N}} \tag{1}$$

Eqn1 for a discrete distribution, and

$$\sqrt{\frac{\Sigma P(x) x^2 dx}{\Sigma P(x) dx}} \tag{2}$$

Eqn 2 for a continuous distribution.

Physical scientists often use the term root-mean-square as a synonym for standard deviation when they refer to the square root of the mean squared deviation of a signal from a given baseline or fit. RMS is a mean value and not an instantaneous measurement.

Pose The position and orientation of an object with respect to a local or global coordinate reference frame.

Position The translation in n dimensions of an object with respect to a coordinate reference frame. The translation may be distance in the case of cylindrical or Cartesian coordinate systems or angles in the case of spherical coordinate systems.

PositionRMS The position RMS provides a means of determining the error associated with a reported position. This value is measured over time and therefore not representative of the error for any particular position value. It provides a statistical measure of the magnitude of the possible error in 3 dimensions. This value is reported in meters.

PresenceVector JAUS provides for variable length messages. These messages either have repeating data or have a mixture of required and optional data fields. The Presence Vector is used to indicate which of the optional data fields are included.

VelocityRMS The velocity RMS provides a means of determining the error associated with a reported speed. This value is measured over time and representative of the error associated with the manner in which speed is recorded and reported. It provides a statistical measure of the magnitude of the possible error.

5. Units of Measurement

For this standard, SI units will be the standard units of measurement for data[4] except for temperature, which shall be represented by Celsius. The following base SI (see Table 1) units will be used[1]:

Base Quantity	Name	Symbol
length	metre	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
thermodynamic temperature	Celsius	C
amount of substance	mole	mol
luminous intensity	candela	cd

Table 1: *Base SI Units*

The following derived units in Table 2 will also be used in the representation of data. It is recommended that a universal constants conversion reference [5] is adopted.

---

[1] Kelvin and Celsius are used interchangeably although in truth degrees Kelvin is the SI standard

Derived Quantity	Name	Symbol
area	square meter	m^2
volume	cubic meter	m^3
speed, velocity	meter per second	m/s
acceleration	meter per second squared	m/s^2
wave number	reciprocal meter	m^{-1}
mass density	kilogram per cubic meter	kg/m^3
specific volume	cubic meter per kilogram	m^3/kg
current density	ampere per square meter	A/m^2
magnetic field strength	ampere per meter	A/m
amount-of-substance concentration	mole per cubic meter	mol/m^3
luminance	candela per square meter	cd/m^2
mass fraction	kilogram per kilogram, which may be represented by the number 1	kg/kg = 1
plane angle	radian	rad ($m \cdot m^{-1}$)
solid angle	steradian	sr ($m^2 \cdot m^{-2}$)
frequency	hertz	Hz (s^{-1})
force	newton	N (kg m/s^2)
pressure, stress	pascal	Pa ($m^{-1} \cdot kg \cdot s^{-2}$)
energy, work, quantity of heat	joule	J ($m^2 \cdot kg \cdot s^{-2}$)
power, radiant flux	watt	W ($m^2 \cdot kg \cdot s^{-3}$)
electric charge, quantity of electricity	coulomb	C ($s \cdot A$)
electric potential difference,electromotive force	volt	V ($m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$)
capacitance	farad	F ($m^{-2} \cdot kg^{-1} \cdot s^4 \cdot A^2$)
electric resistance	ohm	$\Omega(m^2 \cdot kg \cdot s^{-3} \cdot A^{-2})$
electric conductance	siemens	S ($m^{-2} \cdot kg^{-1} \cdot s^3 \cdot A^2$)
magnetic flux	weber	Wb ($m^2 \cdot kg \cdot s^{-2} \cdot A^{-1}$)
magnetic flux density	tesla	T ($kg \cdot s^{-2} \cdot A^{-1}$)
inductance	henry	H ($m^2 \cdot kg \cdot s^{-2} \cdot A^{-2}$)
luminous flux	lumen	lm ($m^2 \cdot m^{-2}$)
illuminance	lux	lx ($m^2 \cdot cd$)
activity (of a radionuclide)	becquerel	Bq (s^{-1})
absorbed dose, specific energy (imparted), kerma	gray	Gy ($m^2 \cdot s^{-2}$)
dose equivalent	sievert	Sv ($m^2 \cdot s^{-2}$)
catalytic activity	katal	kat ($s^{-1} \cdot mol$)
engine revolutions	revolutions per minute	rpm($\frac{1}{60} \cdot s^{-1}$)

Table 2: *Derived SI Units*

6. Representations

6.1 Coordinate Frame Representations

This section presents three coordinate frames that should be common to any implementation of autonomous intelligent systems. The global pose, or frame **W**, provides an unambiguous world reference that should allow experimentation and operation over long distances. The local frame provides a common representation of an autonomous intelligent system that should be understood by anyone developing algorithms and using data from components on an autonomous system. With a single conversion function, DRDC can extend the above coordinate frames of reference to work with JAUS systems.

6.1.1 World Coordinate Frame of Reference W or Global Pose

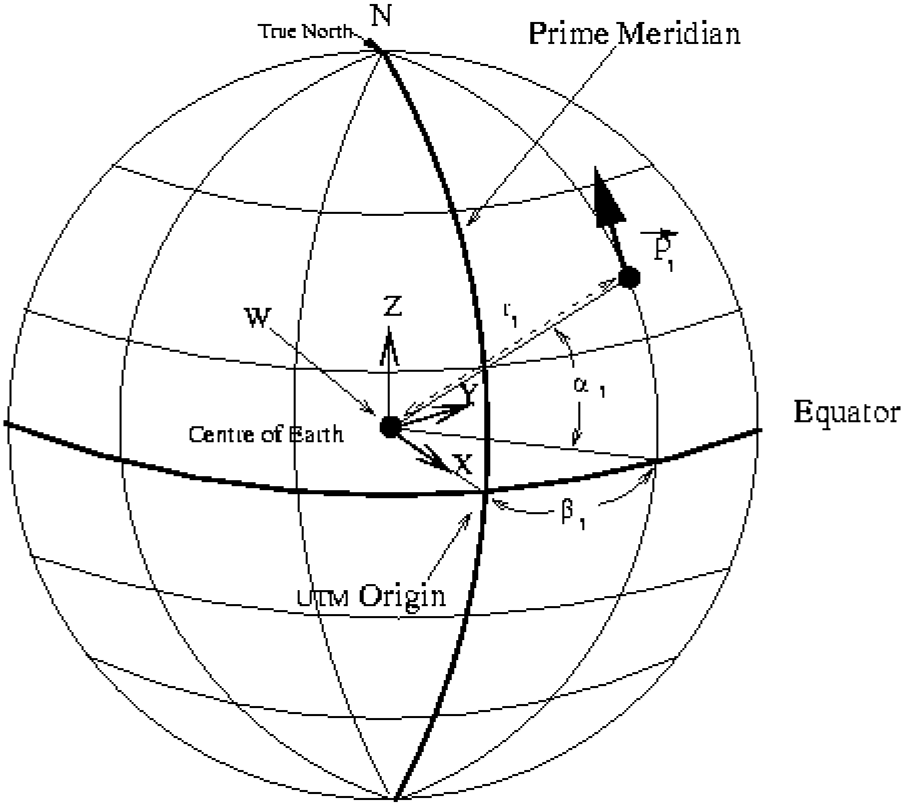


Figure 1: *World Coordinate Frame of Reference W and origin at the vertex of the Prime Meridian and Equator*

This section outlines the adopted convention for global pose with respect to the Earth. The world or global coordinate frame of reference can be referred to as **W** and represents the pose of the object with respect to the global origin. The global position in **W** shall be represented internally by the 2-dimensional spherical coordinates: latitude, longitude, and elevation (height). Refer to

Figure 1. The Cartesian coordinates at the centre of 1 denote the origin, α is the latitude angle, and βis the longitude angle. Longitude angle measurement is measured from X-axis origin, Greenwich Meridian (Prime Meridian), where positive values are East from Greenwich and negative values are West from Greenwich. The Y-axis origin is the Equator for latitude angle measurement, where positive values are northward from the Equator to the North Pole and negative values are southward from the Equator to the South Pole. The radius r_i origin is the centre of the Earth. This standard proposes adding a False Elevation to the radius origin to represent elevation from Mean Sea Level (MSL), also known as orthometric height, where positive values are above MSL (accounting for geoid gravity model variances which may perturb local values) and negative values are below MSL. Elevation in this standard is assumed constant and does not take into account gravity models to adjust MSL regionally. The world coordinate frame of reference will use the geographic coordinates latitude, longitude, and elevation as the position 3-vector.

UTM and MGRS produce ambiguous data if the subregion and map letters are not included/missing in the data. Latitude, in radians, longitude, in radians, and elevation (or height), in metres, represent a unique position with respect to the Earth (ellipsoid) as three unique real numbers in a spherical (geographical) coordinate system. Spherical coordinates are favoured over UTM and MGRS; they are topographical projections onto Earth's surface. For a complete treatment of the topic of map projections, refer to [6]. Spherical coordinates remain continuously differentiable and therefore position error can be estimated without approximation. Global position should be kept in spherical coordinates and converted to MGRS or UTM when necessary. The standard for UTM should be to convert the global angles into distances from the local UTM region (6 ° intervals) adding a False Easting of 500,000m. Current UTM conversion functions employed by AIS calculate this automatically. Position data storage should be in spherical coordinates for data accuracy. MGRS would require map labels in addition to the position values.

Latitude and longitude represent angles measured from an arbitrary global origin. Elevation represents distance from an arbitrary origin. The World coordinate frame W will use the World Geodetic System 1984 (WGS84) geodetic datum. The WGS84 ellipsoid parameters are listed in Table 3. Please refer to Figure 1 for a representation of the **W** frame.

Semi-major axis (a):	6378137.00 metres
Inverse Flattening (1/f):	298.257223563

Table 3: *WGS84 ellipsoid parameters*

This global coordinate reference frame is indicated in Figure 1.

Orientation for global and local coordinates shall be represented by quaternions. Quaternions are hypercomplex numbers discovered by W.R. Hamilton in 1833. Unit quaternion representations, also known as Euler parameters, of orientation are unique and exhibit the property that their inverse is equal to their conjugate. Quaternions do not suffer from gimbal lock because quaternion rotation operators are singularity-free unlike Euler Angles and every other rotation operator in $\mathbf{SO}(3)$[2]. The transmission of unique pose requires only 4 real numbers in comparison with 9 for a rotation matrix. Also, quaternion multiplication is faster than rotation matrix computation. The rotation of a vector by a unit quaternion q can be accomplished by using the formula:

$$Rot_q(v) = qvq^* \tag{3}$$

$$Rot_q(v) = \begin{bmatrix} q_s^2+q_x^2-q_y^2-q_z^2 & -2q_sq_z+2q_xq_y & 2q_sq_y+2q_xq_z \\ 2q_sq_z+2q_xq_y & q_s^2-q_x^2+q_y^2-q_z^2 & -2q_sq_x+2q_yq_z \\ -2q_sq_y+2q_xq_z & 2q_sq_x+2q_yq_z & q_s^2-q_x^2-q_y^2+q_z^2 \end{bmatrix} \tag{4}$$

where $q^\star$is the complex conjugate of q and v is the pure quaternion representation of the vector v. Annex A proves the relationship between rotation matrices and unit quaternions. It can also be shown that quaternions can be transformed into Euler Angles [7]. For more detail about quaternions, please refer to [7], [8], [9], [10]and [11] or other references. [7] and [10] details the properties and identities of quaternion mathematics. Orientation data should be stored in quaternions.

The quaternion vector is defined in Equation 5:

$$q \equiv q_x i + q_y j + q_z k + q_s \tag{5}$$

where q_x is the imaginary projection along the i axis, q_y is the imaginary projection along the j axis, q_z is the imaginary projection along the k axis, and q_s is the real scalar. Unit quaternions are represented as: $q \in \mathbb{R}, q \in [-1,1]$. Unit quaternions are governed by Equation 6:

$$q_s^2 + q_x^2 + q_y^2 + q_z^2 = 1 \tag{6}$$

---

[2] Euler angles are differentiated using the small angle assumption, i.e. for a small rotation in Euler RPY, $\Delta\Theta \approx 0, \Delta\Psi \approx 0, \Delta\Phi \approx 0$ at each time step. This is only an approximation.

A position 3-vector for position and a 4-vector quaternion for orientation represent the global and local pose respectively. The global pose is thus a 7-vector :

$$\mathbf{p}_W \equiv [\varphi_W, \lambda_W, z_W, q_s, \mathbf{q}]^T \tag{7}$$

where W represents the global frame of reference, φ_W is the longitude displacement from the Prime Meridian in *radians*, λ_W represents the latitude displacement from the Equator in *radians,* and z_W represents the translation above/below Median Sea Level (MSL) in *metres*. The quaternion q defined by $(q_s, \mathbf{q}) \equiv [q_s, q_x, q_y, q_z]^T$ is dimensionless. UTM provided in addition to latitude and longitude, is represented as x_W and y_W are measured in *metres* and substituted for φ_W and λ_W respectively. The UTM version is represented in equation 8:

$$\mathbf{p}_{W-UTM} \equiv [x_W, y_W, z_W, q_s, \mathbf{q}]^T \tag{8}$$

The quaternion $q = [1.0, 0.0, 0.0, 0.0]$ is defined as the orientation origin and represents the System pointing True North, in the sense of Euler yaw, while having no pitch and no roll. Figure 1, illustrates point p_1 is facing North along the global orientation origin towards the True North pole. Imagine a pencil facing True North and lying on a perfectly flat desk as an analogy of the orientation origin in our representation. This orientation remains the origin no matter where on the Earth the system is. This orientation definition is JAUS-compliant [12] to Section 2.4.

6.1.2 Robot Frame of Reference ℓ or Local Pose

The definition applies to the local robot coordinate frame of reference ℓ. The robot origin/frame of reference shall attach itself to the centre of the front axle/rotation axis, or single axis of wheels. If the robot has legs, then ℓ shall attach itself to the bottom centre of the front hip joint. The exact placement must be visible from outside of the vehicle so as to allow for calibration through survey. Figure 2 demonstrates the local coordinate frame of reference ℓ affixed to the front axle of a multi-axle UGV. If it is an aerial vehicle, the reference frame ℓ shall attach to the bottom centre of the fuselage for fixed wing or rotary aircraft. Note in Figure 2 that the z-axis points downward from ℓ. This z-axis direction means that clockwise rotations are positive. This selection coincides with the *de facto* aerospace standard as well as the coordinate frame of reference for most commercial inertial measurement unit (IMU) sensors. The x-axis extends forward from the vehicle in the along-axis direction. This means that forward motion is positive when measured relative

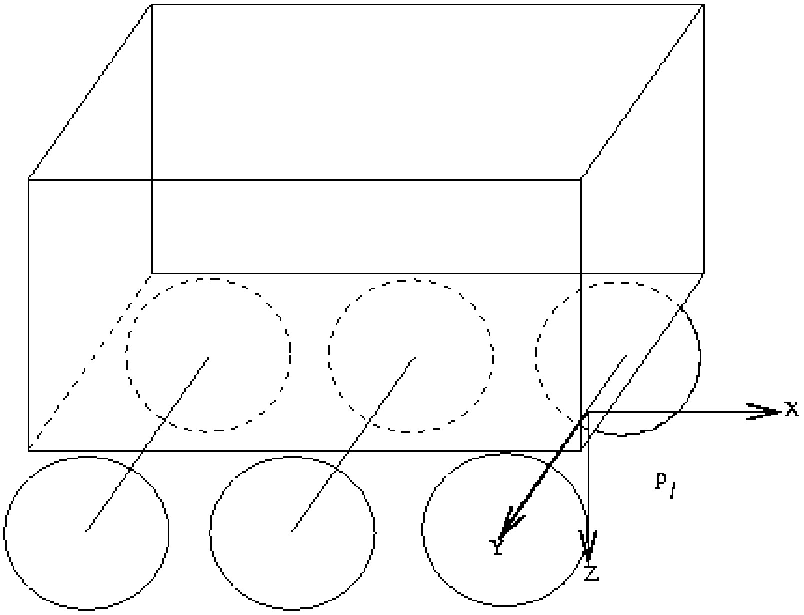


Figure 2: *Local pose affixed to vehicle*

to a previous pose. The y-axis, or across-axis, extends out from the local pose origin in the right-handed sense.

Local pose is defined as:

$$\mathbf{p}_\ell \equiv [x_\ell, y_\ell, z_\ell, q_s, \mathbf{q}]^T \tag{9}$$

where ℓ identifies the local ego-centric frame fixed to the System's front axle and x_ℓ, y_ℓ and z_ℓ represent the translation along axes of ℓ in *metres*. The use of distances instead of angles in local frames of reference is useful for distance-based methods of localization, modeling, and most importantly time-of-flight sensors. The local quaternion should use a local origin relative to the absolute global orientation vector. The quaternion $q = [1.0, 0.0, 0.0, 0.0]^T$ is arbitrarily used as the orientation origin and represents pointing True North, in the sense of Euler yaw, while having no pitch and no roll.

Figure 3 details the relationship between two local poses on the Earth's surface and the difference in orientations while remaining at the local orientation of $q = [1.0, 0.0, 0.0, 0.0]^T$. Over small ranges, perhaps less than 40 km, it is reasonable to ignore the curvature between two local poses as the transformation will be small. Small ranges should assume a rectangular approximation to the curvature of the Earth and therefore the orientations are equivalent. For larger distances, operations on the scale of a hemisphere as in

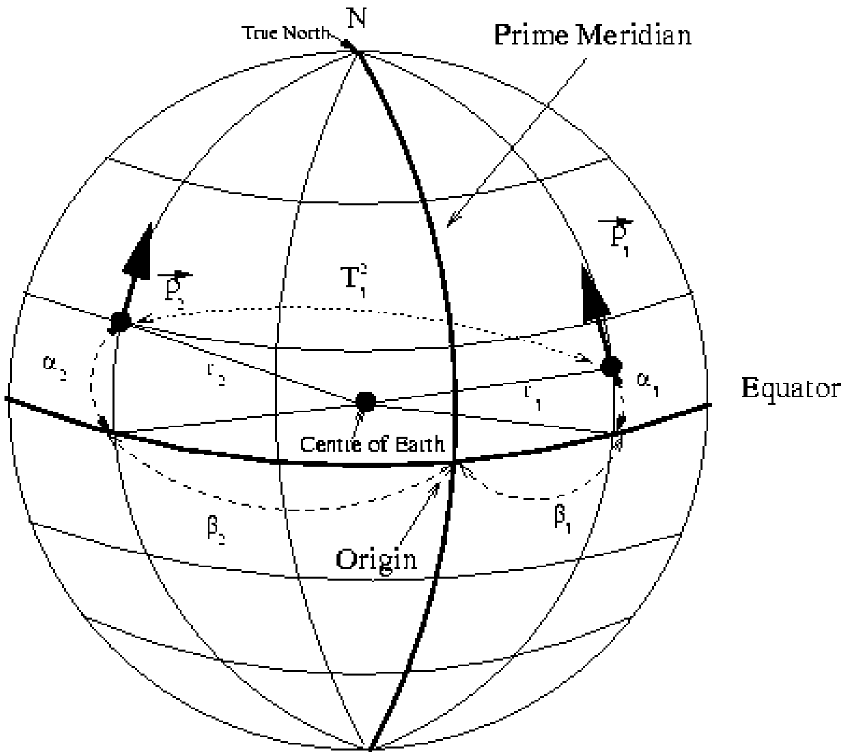


Figure 3: *Transformation between two local poses on the Earth oriented to* $q = [1.0, 0.0, 0.0, 0.0]^T$

Figure 3, it is necessary to transform the local pose of one object into the pose of another. The transformation between local poses is governed by the following equations [13] [14]:

(10) $$T_0^1 = R_y(\alpha_1)R_z(\beta_1)T(r_1, 0, 0)$$

(11) $$T_0^2 = R_y(\alpha_2)R_z(\beta_2)T(r_2, 0, 0)$$

where T_0^1 is the homogeneous transform from the **W** coordinate frame of pose p_1, where p_1 and $p_2$are the local poses shown in Figure 3, α_1 and $\alpha_2$are the latitudes for p_1 and p_2 respectively, β_1 and $\beta_2$are the longitudes for p_1 and p_2 respectively, and r_1 and $r_2$are the radii above the centre of the Earth for p_1 and p_2 respectively.

(12) $$T_0^1 = \begin{bmatrix} & & & D_x{}_0^1 \\ & R_0^1 & & D_y{}_0^1 \\ & & & D_z{}_0^1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where R_0^1 is the rotation matrix from the **W** coordinate frame of pose p_1, and $D_x{}_0^1$, $D_y{}_0^1$, and $D_z{}_0^1$ are the displacements along the axes from the centre of the Earth to p_1.

(13) $$T_0^1 = \begin{bmatrix} \cos\alpha_1\cos\beta_1 & -\cos\alpha_1\sin\beta_1 & \sin\alpha_1 & \cos\alpha_1\cos\beta_1 r_1 \\ \sin\beta_1 & \cos\beta_1 & 0 & \sin\beta_1 r_1 \\ -\sin\alpha_1\cos\beta_1 & \sin\alpha_1\sin\beta_1 & \cos\alpha_1 & -\sin\alpha_1\cos\beta_1 r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(14) $$T_0^2 = \begin{bmatrix} & & & D_x{}_0^2 \\ & R_0^2 & & D_y{}_0^2 \\ & & & D_z{}_0^2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T_0^2 is the homogeneous transform from the **W** coordinate frame of pose p_2, and $D_x{}_0^2$,$D_y{}_0^2$, and $D_z{}_0^2$ are the displacements along the axes from the centre of the Earth to p_2.

$$(15)\; T_0^2 = \begin{bmatrix} \cos\alpha_2 \cos\beta_2 & -\cos\alpha_2 \sin\beta_2 & \sin\alpha_2 & \cos\alpha_2 \cos\beta_2 r_2 \\ \sin\beta_2 & \cos\beta_2 & 0 & \sin\beta_2 r_2 \\ -\sin\alpha_2 \cos\beta_2 & \sin\alpha_2 \sin\beta_2 & \cos\alpha_2 & -\sin\alpha_2 \cos\beta_2 r_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = T_0^2 T_1^0 \tag{16}$$

$$T_1^0 = (T_0^1)^{-1} = \begin{bmatrix} & (R_0^1)^T & & -(R_0^1)^T D_0^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$T_1^0 = \begin{bmatrix} \cos\alpha_1 \cos\beta_1 & \sin\beta_1 & -\sin\alpha_1 \cos\beta_1 & -r_1 \\ -\cos\alpha_1 \sin\beta_1 & \cos\beta_1 & \sin\alpha_1 \sin\beta_1 & 0 \\ \sin\alpha_1 & 0 & \cos\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$T_1^2 = \begin{bmatrix} & R_1^2 & & D_1^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

$$R_1^2 =$$

$$\begin{bmatrix} c\alpha_2 c\beta_2 c\alpha_1 (c\beta_1 + s\beta_1) + s\alpha_2 s\alpha_1 & c\alpha_2 c\beta_2 (s\beta_1 - c\beta_1) & -c\alpha_2 c\beta_2 s\alpha_1 (c\beta_1 + s\beta_1) + s\alpha_2 c\alpha_1 \\ c\alpha_1 (s\beta_2 c\beta_1 - c\beta_2 s\beta_1) & c\beta_1 (s\beta_2 c\alpha_1 + c\beta_2) & s\alpha_1 (c\beta_2 s\beta_1 - s\beta_2 c\beta_1) \\ -s\alpha_2 c\alpha_1 (c\beta_2 c\beta_1 + s\beta_2 s\beta_1) + c\alpha_2 s\alpha_1 & s\alpha_2 (s\beta_2 c\beta_1 - c\beta_2 s\beta_1) & s\alpha_2 (c\beta_2 s\alpha_1 c\beta_1 + s\beta_2) + c\alpha_2 c\alpha_1 \end{bmatrix}$$
(20)

$$c = cos, s = sin$$

$$D_1^2 = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = \begin{bmatrix} c\alpha_2 c\beta_2 (r_2 - r_1) \\ s\beta_2 (r_2 - r_1) \\ s\alpha_2 c\beta_2 (r_1 - r_2) \end{bmatrix} \tag{21}$$

$$p_1^2 = T_1^2 p_1 \tag{22}$$

where R_1^2 is the rotation matrix from the $p_1$coordinate frame to p_2, and D_x D_y D_z are the displacements of D_1^2 along the axes of $p_1$to p_2. T_1^2 can be used to transform an arbitrary global into the frame of another when considering large distance operations.

6.1.3 JAUS-compliant Pose

It is proposed that our systems should also generate a JAUS-compliant[1] pose to interoperate with JAUS systems. JAUS specifies a 6-vector using Euler Angle-axis sequence Roll Pitch Yaw (Euler ZYX) as the representation of pose. JAUS-compliant pose can be achieved from the above coordinate frames by converting the quaternions into Euler angles.

The JAUS-compliant ([1] [15] [3] [12]) Euler angles Ψ,Θ, and Φ define the platform orientation of the vehicle. This definition is illustrated in Figure 4

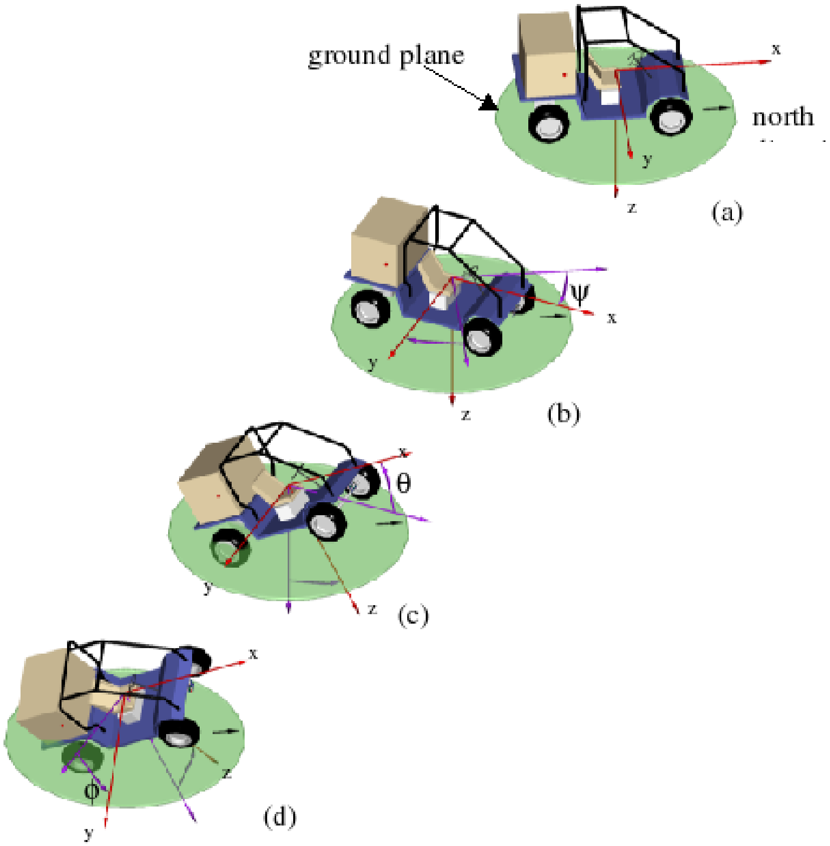


Figure 4: *JAUS-compliant definition of platform orientation [taken from JAUS [1] Figure 2.2.1]*

where a Cartesian XYZ coordinate system has been attached to the vehicle with its X axis pointed forward and its Z axis downward. The vehicle is then rotated by an angle Ψ in a right-handed sense about the Z axis (YAW) as shown in Figure 4 (b). Subsequently, the vehicle is rotated by an angle Θ about the modified Y axis (PITCH) as shown in Figure 4 (c) followed by a rotation of Φ about the modified X axis (ROLL) as shown in Figure 4(d). Local platform orientation is defined in a similar manner by initially aligning the platform coordinate system with the local coordinate system and then rotating it in the same sequence as defined for the global platform orientation. The pose p_{rpy} of an object is a 6-vector as described by JAUS and defined as:

$$p_{rpy} \equiv [x, y, z, \boldsymbol{\Psi}, \Theta, \boldsymbol{\Phi}]^T \tag{23}$$

where x represents the translation along the x-axis in metres, y represents the translation along the y-axis in metres, z represents the translation along the z-axis in metres, $\boldsymbol{\Psi}$ is the rotation about initial z-axis w.r.t initial orientation in radians, Θ is the rotation about the modified y-axis in radians, and $\boldsymbol{\Phi}$ is the rotation about the modified x-axis in radians.

It is recommended that systems use the global or local pose 7-vector representing position and orientation using quaternions instead of Euler Angles. The orientation quaternions can be converted to Euler RPY for use by JAUS systems. The equation governing conversion from quaternions to RPY Euler Angles are:

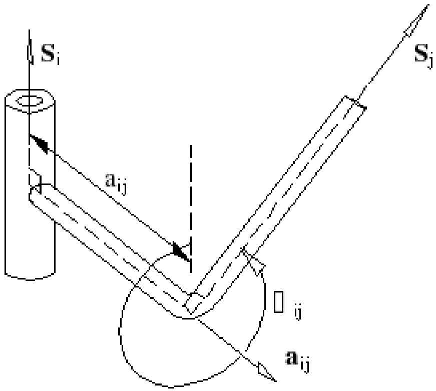
$$\psi = \arctan\left(\frac{2q_xq_y + 2q_sq_z}{2q_s^2 + 2q_x^2 - 1}\right) \tag{24}$$

$$\theta = \arcsin(2q_sq_y - 2q_xq_z); \cos(\theta) \geq 0 \tag{25}$$

$$\phi = \arctan\left(\frac{2q_yq_z + 2q_sq_x}{2q_s^2 + 2q_z^2 - 1}\right) \tag{26}$$

6.2 Manipulator Linkage Representations

For future manipulator links, or indeed legged robotic locomotion systems, a standard method for referencing link variables has been established. This standard is JAUS-compliant[1] and were taken from same. There are many representations of link parameters, particularly Denavit-Hartenburg, that have been developed for pose representation. However, even Denavit-Hartenburg has two interpretations (the original Denavit-Hartenburg and the Craig version) which differ. The focus in the short term is on mobile robotic vehicles, but as payloads become more demanding, manipulators will return to the UGV's. Manipulators will become a key aspect of the future systems here and therefore we need a standard representation. Adopting standards now will make this transition to standard representation effortless.

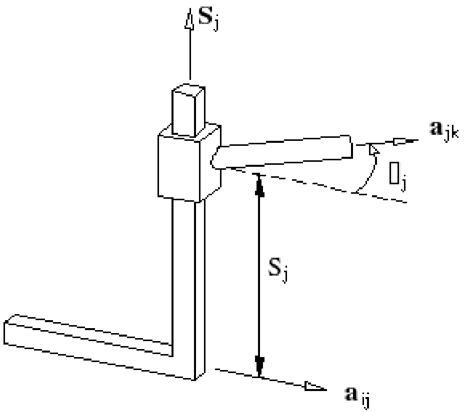


Symbol	Definition
a_{ij}	link length
α_{ij}	twist angle
$\mathbf{a}_{ij}$	unit vector along link ij
$\mathbf{S}_i$	unit vector along joint axis i
$\mathbf{S}_j$	unit vector along joint axis j
Notes	Link length a_{ij} is measured as the perpendicular distance between joint axis i and joint axis j along the vector $\mathbf{a}_{ij}$
	Twist angle α_{ij} is the angle between $\mathbf{S}_i$ and $\mathbf{S}_j$ measured in a right hand sense about $\mathbf{a}_{ij}$

Figure 5: *JAUS-compliant Link Definition [taken from JAUS [1]]*

6.2.1 Link Representation

To standardize the method of referencing manipulator links (or joints), the following figures illustrate the notation that will be used later in the document. Figure 5 illustrates the parameters used for JAUS-compliant links. Figure 6 and Figure 7 illustrates additional parameters especially used for prismatic/translational joints and revolute/rotational joints respectively.

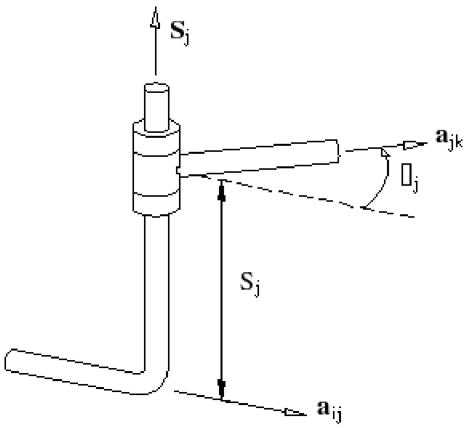


Symbol	Definition
S_i	variable joint offset distance
Θ_j	constant joint angle
$\mathbf{a}_{\mathrm{ij}}$	unit vector along link ij
$\mathbf{a}_{\mathbf{jk}}$	unit vector along link jk
$\mathbf{S}_{\mathrm{j}}$	unit vector along joint axis j
Notes	Joint offset $S_{\mathrm{j}}$is measured as the perpendicular distance between link $\mathbf{a}_{\mathrm{ij}}$ and the link $\mathbf{a}_{\mathrm{jk}}$along the vector $\mathbf{S}_{\mathrm{j}}$. Note that it can have a negative value.
	Joint angle $\Theta_j$is the angle between $\mathbf{a}_{\mathrm{ij}}$ and $\mathbf{a}_{\mathrm{jk}}$measured in a right hand sense about $\mathbf{S}_{\mathrm{j}}$.

Figure 6: *JAUS-compliant Prismatic Joint Definition [taken from JAUS [1]]*

6.2.2 Prismatic Joint Representation

A pure prismatic/translational joint represents a joint between links with one linear degree of freedom that allows motion in only constrained range. Table 6 defines the prismatic joint parameters that should be used.



Symbol	Definition
S_j	constant joint offset distance
Θ_j	variable joint angle
$\mathbf{a}_{ij}$	unit vector along link ij
$\mathbf{a}_{jk}$	unit vector along link jk
$\mathbf{S}_j$	unit vector along joint axis j
Notes	Joint offset S_j is measured as the perpendicular distance between link $\mathbf{a}_{ij}$ and the link $\mathbf{a}_{jk}$ along the vector $\mathbf{S}_j$. Note that it can have a negative value.
	Joint angle Θ_j is the angle between $\mathbf{a}_{ij}$ and $\mathbf{a}_{jk}$ measured in a right hand sense about $\mathbf{S}_j$.

Figure 7: *JAUS-compliant Revolute Joint Definition [taken from JAUS [1]]*

6.2.3 Revolute Link Representation

A pure revolute/rotational joint represents a joint with one radial degree of freedom that allows motion in rectangular coordinates within the constrained range.

6.3 Data Representation

6.3.1 Time Representation

The standard time scale representation adopted is UTC- Coordinated Universal Time. UTC is equivalent to mean solar time at the prime meridian (0° longitude), formerly expressed as UT1 or Greenwich Mean Time (GMT). UT1 is also known as world time, Z time, and Zulu time. UTC is defined by the CCIR Recommendation 460-4 and is maintained by various military and governmental organizations. UTC is chosen because it more accurately portrays the day/night conditions locally. This will manifest itself when attempting to correlate visual imagery at dawn and dusk.

Time for the purposes of time stamping shall be represented from the Unix-chosen epoch (00:00:00 January 1, 1970) for all data.Each time value is represented as two integers; one gives the number of seconds elapsed since the epoch and the other gives the number of microseconds since the last second elapsed. There will be a conversion needed for GPS data to the above standard. GPS time is measured from the TAI.

$$GPS = TAI - 19s \tag{27}$$

where TAI is International Atomic Time. TAI is defined by about a dozen atomic clocks distributed worldwide. GPS readings were started from a different epoch. The GPS epoch is 00:00 (midnight) UTC on 1980-01-06. At the beginning the time origin of GPS and UTC time were identical but no leap seconds are inserted into GPS time. Therefore, the relationship between GPS and UTC time is governed by the equation:

$$UTC = GPS - L \tag{28}$$

where L is the number of leap seconds added since 1981-07-01. Currently, GPS time is 13 seconds ahead of UTC as of 2005-01-01. The GPS time will vary relative to UTC as more leap seconds are added onto UTC. According to International Earth Rotation Service (IERS), no positive leap second will be introduced at the end of June 2005 so this difference of 13 seconds will hold until December 31 2005. After this point, the time difference for GPS readings will need to be adjusted as IERS adjusts UTC. Since GPS time is universal and not subject to the latencies of Network Time Protocol (NTP) using UTC, it can be used to correct errant onboard clocks.

6.3.2 Time Correction

Internal clocks within ALS experimental vehicles will require synchronization periodically. Onboard computer clocks drift so it is impractical to allow them to operate without correction for more than a few hours. In addition, the collation of data taken from separate machines requires that the time of each vehicle be as close to identical as possible. The impact of clock drift will be felt significantly when these machines are operating for long durations, predictably over weeks at a time. There are two sources of absolute time available to us: GPS time and UTC time via Network Time Protocol (NTP). NTP time suffers from latency when requesting time from a time server located at a distance. This is compensated for to some degree in the NTP service, but will not be nil. GPS, on the other hand, has a direct link to atomic clock time and where reception is possible will give a more accurate absolute time. The following procedure should be followed to ensure that all data collected is timestamped accurately:

1. For each experiment, a GPS-equipped NTP server shall be setup to correct all vehicles involved. The GPS time shall be used to correct the local vehicle network NTP server;

2. Where the vehicle does not have a GPS present, update the onboard clock using NTP and the network time server;

3. Where the vehicle has a GPS, update the onboard clock time using the GPS time and convert that time to UTC;

4. If the vehicle has both GPS and NTP, then use the GPS to correct the onboard clock first then NTP second ;

5. If the vehicle has both GPS and NTP and if the vehicle loses GPS access, then use NTP and the network time server;

6. If the vehicle has neither a GPS nor access to the NTP, correct this clock manually whenever possible.

It is the responsibility of the system, or subsystem as the case may be, to correct the onboard clock and not individual nodes, devices, or processes. Individual nodes, devices, or processes will request a timestamp through the local computer and will be unaware of the proposed time correction scheme. Future systems should incorporate, where practical, a time service that will combine the local time correction task with global time updating so that when the onboard vehicle clock diverges from the absolute time then it can adjust the current time.

Language	OS	Compliant	Comment
C	Unix/GNU gcc	YES	
C++	Unix/GNU gcc	YES	
Java	Unix/GNU gcc	YES	
Assembler	Unix/GNU gcc	YES	
C	WindowsCE/ME/NT/XP	YES	
C++	WindowsCE/ME/NT/XP	YES	
Java	WindowsCE/ME/NT/XP	YES	
Assembler	WindowsCE/ME/NT/XP	YES	

Table 4: *Data Type Compliance Table*

6.3.3 Data Types

Data types should conform to the ISO/IEC Working Group (WG) 14 C data types (Standard C) Section 6.2.5 [16]. Refer to ISO/IEC (http://www.open-std.org/jtc1/sc22/wg14/) for the current version of the data types. Hardware, firmware, or software that does not conform to the basic data types outlined in Standard C should be converted to compliant types before communication between nodes, components, and processes. Table 4 indicates some commonly used languages, operating systems, and compilers that conform to the latest ISO/IEC WG 14 standard.

6.3.4 Floating Point Data

All floating point variables shall adhere to the ANSI/IEEE Standard 754-1986 (Standard for Binary Floating Point Arithmetic) until such time as IEEE Standard 754R is adopted. According to ANSI/IEEE Standard 754-1986, there are two precision levels for fixed floating point variables: single precision and double precision. For single precision, the floating point value is represented by a 32 bit word with a format as per Table 5. The bits are numbered from most significant bit to least significant bit and designated with either an S for sign, E for exponent, or F for fraction (decimal number). The first bit is always the sign, the following 8 bits are the exponent and the remaining 23 bits represent the fraction.

The variable V may be found from a IEEE single precision fixed floating point data word by the following rules:

1. If E=255 and F is nonzero, then V=NaN ("Not a number")

2. If E=255 and F is zero and S is 1, then V=-Infinity

3. If E=255 and F is zero and S is 0, then V=Infinity

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5
S	E	E	E	E	E	E	E	E	F	F	F	F	F	F	F

1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

S = Sign E = Exponent F = Fraction

Table 5: *IEEE 754 Single Precision Floating Point*

4. If 0<E<255 then V=(-1)**S * 2 ** (E-127) * (1.F) where "1.F" is intended to represent the binary number created by prefixing F with an implicit leading 1 and a binary point.
5. If E=0 and F is nonzero, then V=(-1)**S * 2 ** (-126) * (0.F). These are "unnormalized" values.
6. If E=0 and F is zero and S is 1, then V=-0
7. If E=0 and F is zero and S is 0, then V=0

For double precision, the floating point value is represented by a 64 bit word with a format as per Table 6. The bits are numbered from most significant bit to least significant bit and designated with either an S for sign, E for exponent, or F for fraction (decimal number). The first bit is always the sign, the following 11 bits are the exponent and the remaining 52 bits represent the fraction.

The variable V may be found from a IEEE single precision fixed floating point data word by the following rules:

1. If E=2047 and F is nonzero, then V=NaN ("Not a number")
2. If E=2047 and F is zero and S is 1, then V=-Infinity
3. If E=2047 and F is zero and S is 0, then V=Infinity
4. If 0<E<2047 then V=(-1)**S * 2 ** (E-1023) * (1.F) where "1.F" is intended to represent the binary number created by prefixing F with an implicit leading 1 and a binary point.
5. If E=0 and F is nonzero, then V=(-1)**S * 2 ** (-1022) * (0.F). These are "unnormalized" values.
6. If E=0 and F is zero and S is 1, then V=-0

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5
S	E	E	E	E	E	E	E	E	E	E	E	F	F	F	F

1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4
2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

4	4	5	5	5	5	5	5	5	5	5	5	6	6	6	6
8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

S = Sign E = Exponent F = Fraction

Table 6: *IEEE 754 Double Precision Floating Point*

7. If E=0 and F is zero and S is 0, then V=0

Figure 7 details operating systems, languages, and compilers that are known to comply with the IEEE floating point standards.

6.3.5 Engineering Notation Data

Engineering notation is scientific notation in which the powers of ten are limited to an exponent power that is a multiple of three, (i.e, the powers are

Language	OS	Compliant	Comment
C	Unix/GNU gcc	YES	
C++	Unix/GNU gcc	YES	
Java	Unix/GNU gcc	YES	
Assembler	Unix/GNU gcc	YES	
C	WindowsCE/ME/NT/XP	YES	
C++	WindowsCE/ME/NT/XP	YES	
Java	WindowsCE/ME/NT/XP	YES	
Assembler	WindowsCE/ME/NT/XP	YES	

Table 7: *Floating Point Compliance Table*

powers of 1000 but written as 10^6, not 1000^2). It is another means of transmitting a decimal real number without the need for floating point representation. This is important for standardizing data received from transducers with and without a great deal of computational power and floating point units (FPU). These numbers can be sent across low bandwidth communications media using integers for the whole and fraction elements and characters for the sign and the prefix. Where floating point data is not practical, then use this engineering notation for all real variables except time (see Section 6.3.1 for a treatment of time). Formats A and B are listed in Table 8 and Format C is listed in Table 9. Each format requires four separate elements: a sign (S), a whole number (W), and decimal (D) and a prefix (P). The size of the whole number and fraction varies for the unsigned 8 bit integer, 16 bit integer, and 32 bit integer format but the sign and the prefix remain the same size as one 8 bit ASCII character. The prefix represents the power of three exponent to be applied to the whole number and fraction. The prefixes are the exponent, as a power of 10, that the whole number and the fraction must be multiplied by. These prefixes are found in appendix C of [17].

The value of the engineering notation data can be found with the following rules:

1. If S = 0 then S = +1;
2. If S = 1 then S = -1
3. Prefix P is an 8 bit ASCII character. The prefix values are found by consulting Table 10;
4. $V \doteq S * (W.F) * 10^P$

Format A - 8 bit engineering notation

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2
										0	1	2	3	4	5	6	7	8	9	0	1	2	3
S	S	S	S	S	S	S	S	W	W	W	W	W	W	W	W	D	D	D	D	D	D	D	D

0	1	2	3	4	5	6	7
P	P	P	P	P	P	P	P

Format B - 16 bit engineering notation

0	1	2	3	4	5	6	7
S	S	S	S	S	S	S	S

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5
W	W	W	W	W	W	W	W	F	F	F	F	F	F	F	F

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

0	1	2	3	4	5	6	7
P	P	P	P	P	P	P	P

S = Sign W = Whole Number D = Decimal P = Prefix

Table 8: *Engineering Notation Data Formats A and B*

Format C - 32 bit engineering notation

0	1	2	3	4	5	6	7
S	S	S	S	S	S	S	S

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5
W	W	W	W	W	W	W	W	D	D	D	D	D	D	D	D

1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

0	1	2	3	4	5	6	7
P	P	P	P	P	P	P	P

S = Sign W = Whole Number D = Decimal P = Prefix

Table 9: *Engineering Notation Data Format C*

10^n	P	Prefix	Symbol	Short scale	Long scale Name	Decimal Equivalent
10^{24}	24	yotta	Y	Septillion	Quadrillion	1 000 000 000 000 000 000 000 000
10^{21}	21	zetta	Z	Sextillion	Trilliard (thousand trillion)	1 000 000 000 000 000 000 000
10^{18}	18	exa	E	Quintillion	Trillion	1 000 000 000 000 000 000
10^{15}	15	peta	P	Quadrillion	Billiard (thousand billion)	1 000 000 000 000 000
10^{12}	12	tera	T	Trillion	Billion	1 000 000 000 000
10^{9}	9	giga	G	Billion	Milliard (thousand million)	1 000 000 000
10^{6}	6	mega	M	Million		1 000 000
10^{3}	3	kilo	k	Thousand		1 000
10^{2}	2	hecto	h	Hundred		100
10^{1}	1	deca, deka	t	Ten		10
10^{-1}	-1	deci	d	Tenth		0.1
10^{-2}	-2	centi	c	Hundredth		0.01
10^{-3}	-3	milli	m	Thousandth		0.001
10^{-6}	-6	micro	u	Millionth		0.000 001
10^{-9}	-9	nano	n	Billionth	Milliardth	0.000 000 001
10^{-12}	-12	pico	p	Trillionth	Billionth	0.000 000 000 001
10^{-15}	-15	femto	f	Quadrillionth	Billiardth	0.000 000 000 000 001
10^{-18}	-18	atto	a	Quintillionth	Trillionth	0.000 000 000 000 000 001
10^{-21}	-21	zepto	z	Sextillionth	Trilliardth	0.000 000 000 000 000 000 001
10^{-24}	-24	yocto	y	Septillionth	Quadrillionth	0.000 000 000 000 000 000 000 001

Table 10: *Engineering Notation Prefixes from NIST[17]*

7. Interface Requirements

Given the definitions, representations, and units of measure defined above, it is clear that there will be noncompliant components within AIS. Within AIS, any system, subsystem, device, node, or process can use its own internal representation. When that system, subsystem, device, node, or process attempts to communicate outside its internal representation, then it is the responsibility of the developer to convert the representation to the herementioned AIS representations. As long as all systems, subsystems, devices, nodes and processes hold to this standard at the interface, then interoperability will be achieved.

For full interoperability, conversion to the recommended AIS standards should be delegated down to the individual components so that interoperability is assured. If this were not the case, noncompliant components from one system would not interoperate when inserted into another system. All COTS components, whether they are OEM equipment like a GPS receiver or previous software elements, should be converted to the recommended standards. This conversion can be as simple as converting internal units to the prescribed SI units before communication, or in some cases the introduction of a wrapper layer the hides the internal functionality of the non-compliant unit. It is recommended that future contracts for system modules should include these standards to ensure all future components are compliant with these standards.

7.1 JAUS System Topology and Interoperation

Figure 8, taken from JAUS Reference Architecture Specification: Architecture Framework[15], details the composition of a JAUS proposed system. According to section 3.6 from [15]:

> To achieve the desired level of interoperability between intelligent computing entities, all messages that pass between JAUS defined components (over networks or via airwaves) shall be JAUS compatible messages.

The goal of the JAUS Working Group is laudable, to promote interoperability within the various UGV research and fielded systems. However, examination of the Subsystem, Node and Component topology of the JAUS system reveals that it prescribes unique naming conventions and message passing regime on any architecture. It forces another protocol, which is less advanced and at this time incomplete, upon systems that can use CORBA or other proven middleware on top of a very mature ethernet. This message protocol is inflexible and should not be adopted within the internals of AIS systems. Since this is the case, we need another method to achieve the same aim.

JAUS interoperability with our AIS systems is possible if we consider JAUS message passing from the internals of our systems through a JAUS widget and name our AIS system a System in the JAUS context. This convention would allow our systems to

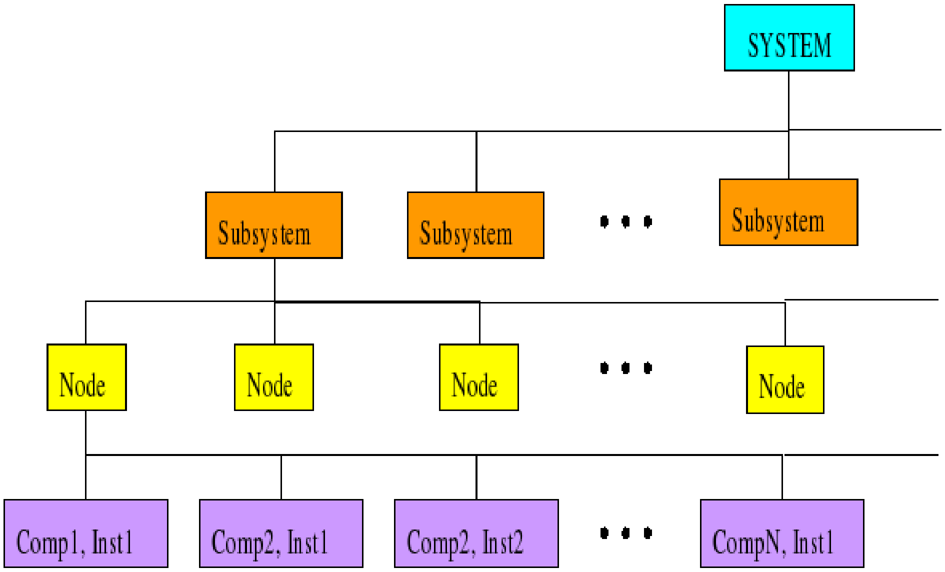


Figure 8: *JAUS System Topology (taken from [15] Section 2.3)*

communicate with JAUS as if it were a complete JAUS system without unnecessary restrictions at the lower level. This would require a MIRO to JAUS widget (JAUSService) to handle communication on behalf of our system's components. This JAUS service would follow the message protocol rules, but wouldn't address anything beneath the System level. Instead, it would act as a Name Service router and pass information from AIS components as if they were coming from a single JAUS system. It would simply deny all requests for direct access but honour requests for information from the "subsystems".

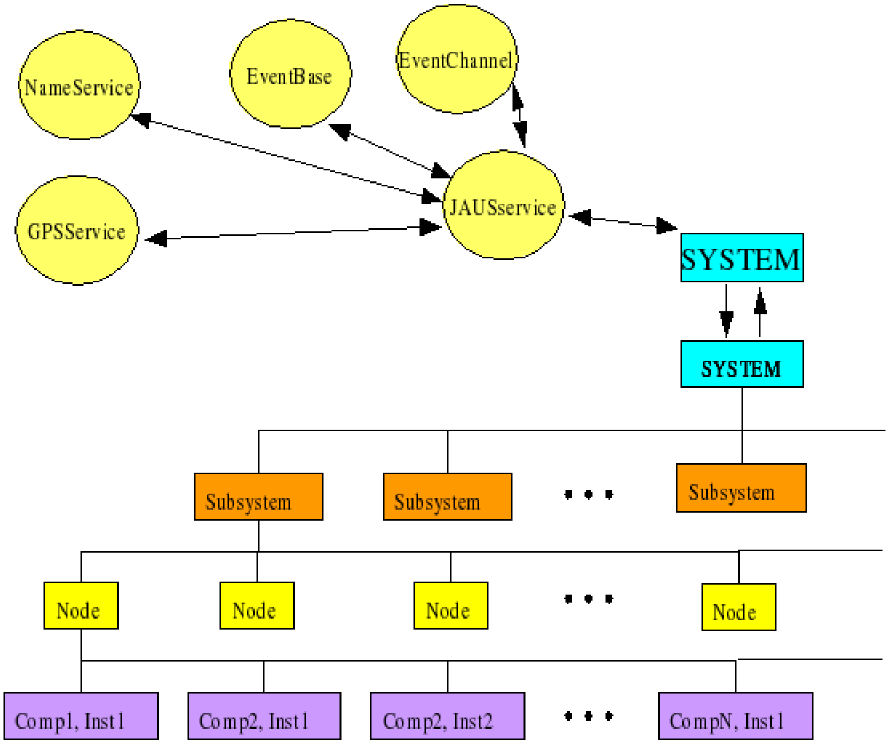


***Figure 9:** AIS interface to JAUS Systems*

8. Conclusion

This memorandum presented a set of representations, units of measure, and definitions for the AIS program. The aim of this document is to reduce ambiguity by establishing common rules for the interpretation of information. The representations have been selected to reduce ambiguity in data transferred amongst components and between systems in future AIS projects. It is recommended that these standards are adopted and steps be taken to convert noncompliant software, hardware, and firmware. The adoption of these standards within the work by DRDC and from contractors who assist the program will increase the efficiency of implementation and increase the interoperability of projects and components. It is recommended that AIS adopt these standards for all future work.

The importance of the recommended standards was reinforced during the ALS program development. Independent software development employed various standards for units of measure, representations, and definition lead to difficiulties. Some software components adopted unique representations, or used Imperial units of measure. This lead to data confusion in a number of cases between data sources and data sinks, making the entire effort much more difficult than would have been the case had *a priori* standards been in place.

The reasons for the selection of the standards reported in this publication are manifold:

some are due to common convention, some are chosen in order to be compliant with JAUS representations, and a few are arbitrary. However, whatever their origin, it is important to point out that the adoption of comprehensive set of standards is vital in order to reap the benefit of common representations. Data exchange without common semantics leads to confusion and inefficiency, and is unacceptable in a research program that relies on a considerable amount of systems integration and interoperability. All of the conventions outlined are important to provide context to the data that is received from other systems, subsystems, devices, nodes, and processes. Compliance with the JAUS standards for representation and units of measure means that DRDC can interoperate with US research and fielded systems with the implementation of a single service, JAUSService, to act as the router between the independent systems.

The standards outlined are intended for use at the interface between systems, subsystems, devices, nodes, and processes. It is obvious that some devices by their design do not inherently adopt these conventions (e.g. a GPS that represents angle as degrees as opposed to radians) and so the absolute adoption of these standards is not practical. Any system, subsystem, device, node, or process may use its own internal representation. When that system, subsystem, device, node, or process attempts to communicate outside its internal representation, then it is the responsibility of the developer to convert the representation to the herementioned representations, origins, and units of measure. As long as all systems, subsystems, devices, nodes, and processes hold to this standard at the interface, then interoperability will be achieved.

References

1. JAUS Working Group (2004). JAUS Vol II- Reference Architecture Specification, Vol. II. Washington DC: JAUS - AS-4 Working Group. Department of Defense.
2. JAUS Working Group (2003). JAUS Volume I Domain Model, Number V3.0. Washington DC: JAUS-AS-4 Working Group. Department of Defense.
3. JAUS Working Group (2004). Reference Architecture Specification: Message Set, Number V3.2. Washington DC: JAUS-AS-4 Working Group. Department of Defense.
4. Taylor, Barry, (Ed.) (2001). The International Standard of Units, 7th ed. Vol. Special Publication 330. Gaithersburg MD: National Institute for Standards and Technology NIST.
5. Tennent, R.M. (1971). Science Data Book, 9th. ed. Hong Kong: Oliver and Boyd.
6. Bugayevskiy, Lev M. and Snyder, John P. (1995). Map Projections: A Reference Manual, London UK: Taylor and Francis.
7. Kuipers, Jack (1998). Quaternions and Rotation Sequences, 5th ed. Princeton NJ: Princeton University Press.
8. Schwab, Arend L. (2002). Quaternions, Finite Rotation, and Euler Parameters. Cornell University Notes, Ithaca NY.
9. Burbanks, Andrew (1996). Quaternions in C++. University of Bristol Notes, Bristol UK.
10. Shoemake, Ken (1994). Quaternions. In *ACM SIGGRAPH 94*, Association for Computing Machinery ACM. New York NY: Association for Computing Machinery.
11. A., Kyrala (1967). Theoretical Physics: Applications of Vectors, Matrices, Tensors, and Quaternions, 1st ed. Number 5 in Studies in Physics and Chemistry. Philadelphia PA: W.B. Saunders Company.
12. JAUS Working Group (2005). Reference Architecture Specification: Message Definition, Number V3.2. Washington DC: Department of Defense.
13. Paul, R.P. (1981). Robot Manipulators-Mathematics, Programming and Control, 7th ed. MIT Press series in artificial intelligence. Cambridge MS: MIT Press.
14. Wolovich, William A. (1986). Robotics: Basic Analysis and Design, 1st. ed. Holt, Rinehart, and Winston series in electrical and computer engineering. New York NY: CBS College Publishing.
15. JAUS Working Group (2005). Reference Architecture Specification: Architecture Framework, Number V3.2. Washington DC: Department of Defense.

16. ISO/IEC Working Group 14 (1999). ISO/IEC 9899 - Programming languages - C, Vol. WG N1124. Geneva Switzerland.

17. Taylor, Barry (1993). Federal Standard 376B Preferred Metric Units for use by the Federal Government, B ed. Washington DC: General Services Administration. supercedes 376A 1983.

Annex A
Derivation of Rotation Matrix from Quaternions

This annex works through the derivation of a Rotation Matrix from a representation with unit quaternions. Further information on quaternions may be found at [7]. This derivation uses a modified notation for quaternion components from the current document. These notations are equivalent.

(A.1) $$q = (q_0, < q_1 q_2 q_3 >)$$

(A.2) $$q = (q_0, \mathbf{q})$$

(A.3) $$q = q_0 + q_1 i + q_2 j + q_3 k$$

which is the same as

(A.4) $$q = q_s + q_x i + q_y j + q_z k$$

A unit quaternion is defined such that the quaternions must lie on a unitary 4-dimensional hypersphere:

(A.5) $$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

Given a rotation representation in unit quaternions, it is possible to convert this

(A.6) $$x = Rx'$$

(A.7) $$x' = (x'_0, < x'_1 x'_2 x'_3 >)$$

(A.8) $$x' = (0, \mathbf{x}')$$

(A.9) $$\overline{q} = (q_0, < -q_1 - q_2 - q_3 >)$$

$$\overline{q} = (q_0, -\mathbf{q}) \tag{A.10}$$

$$Rx' = q \circ x' \circ q^{-1} = q \circ x' \circ \overline{q} \tag{A.11}$$

$$q \circ x' = (q_0 x'_0 - \mathbf{q} \cdot \mathbf{x}', q_0 \mathbf{x}' + x'_0 \mathbf{q} + \mathbf{q} \times \mathbf{x}') \tag{A.12}$$

We can substitute a new quaternion p for the product of the quaternion q and the vector $\mathbf{x}'$. This simplifies the computation of the second quaternion multiplication.

$$p = q \circ x' = (q_0 x'_0 - \mathbf{q} \cdot \mathbf{x}', q_0 \mathbf{x}' + x_0 \mathbf{q} + \mathbf{q} \times \mathbf{x}') \tag{A.13}$$

$$p = (p_0, < p_1 p_2 p_3 >) = (p_0, \mathbf{p}) \tag{A.14}$$

$$p \circ \overline{q} = (p_0 q_0 - \mathbf{p} \cdot (-\mathbf{q}), q_0 \mathbf{p} + p_0(-\mathbf{q}) + \mathbf{p} \times (-\mathbf{q})) \tag{A.15}$$

$$p \circ \overline{q} = (p_0 q_0 + \mathbf{p} \cdot \mathbf{q}, q_0 \mathbf{p} + p_0(-\mathbf{q}) + \mathbf{p} \times (-\mathbf{q})) \tag{A.16}$$

$$p_0 = (q_0 x'_0 - \mathbf{q} \cdot \mathbf{x}') = (0 - \mathbf{q} \cdot \mathbf{x}') = -\mathbf{q} \cdot \mathbf{x}' \tag{A.17}$$

$$p_0 q_0 = -q_0(\mathbf{q} \cdot \mathbf{x}') \tag{A.18}$$

$$\mathbf{p} = q_0 \mathbf{x}' + x_0 \mathbf{q} + \mathbf{q} \times \mathbf{x}' = q_0 \mathbf{x}' + \mathbf{q} \times \mathbf{x}' \tag{A.19}$$

$$-\mathbf{p} \cdot (-\mathbf{q}) = (q_0 \mathbf{x}' + \mathbf{q} \times \mathbf{x}') \cdot (\mathbf{q}) = q_0 \mathbf{x}' \cdot \mathbf{q} + \mathbf{q} \times \mathbf{x}' \cdot \mathbf{q} \tag{A.20}$$

$$q_0 \mathbf{p} = q_0(q_0 \mathbf{x}' + \mathbf{q} \times \mathbf{x}') \tag{A.21}$$

$$x_0(-\mathbf{q}) = 0 \tag{A.22}$$

$$\mathbf{p} \times (-\mathbf{q}) = (q_0\mathbf{x}' + \mathbf{q} \times \mathbf{x}') \times (-\mathbf{q}) \tag{A.23}$$

$$(q_0\mathbf{x}' + \mathbf{q} \times \mathbf{x}') \times (-\mathbf{q}) = (q_0\mathbf{x}' \times -\mathbf{q}) + \mathbf{q} \times \mathbf{x}' \times -\mathbf{q}) \tag{A.24}$$

$$p \circ \overline{q} = (-q_0(\mathbf{q}\cdot\mathbf{x}') + q_0\mathbf{x}'\cdot\mathbf{q} + \mathbf{q}\times\mathbf{x}'\cdot\mathbf{q}, q_0(q_0\mathbf{x}' + \mathbf{q}\times\mathbf{x}') + (q_0\mathbf{x}'\times -\mathbf{q}) + (\mathbf{q}\times\mathbf{x}'\times -\mathbf{q}) + \mathbf{q}(\mathbf{q}\cdot\mathbf{x}')) \tag{A.25}$$

$$\mathbf{x} = R\mathbf{x}' = q_0(q_0\mathbf{x}' + \mathbf{q} \times \mathbf{x}') + (q_0\mathbf{x}' \times -\mathbf{q}) + (\mathbf{q} \times \mathbf{x}' \times -\mathbf{q}) + \mathbf{q}(\mathbf{q} \cdot \mathbf{x}') \tag{A.26}$$

$$\mathbf{x} = R\mathbf{x}' = q_0^2\mathbf{x}' + q_0\mathbf{q} \times \mathbf{x}' + q_0\mathbf{x}' \times -\mathbf{q} + (\mathbf{q} \times \mathbf{x}' \times -\mathbf{q}) + \mathbf{q}(\mathbf{q} \cdot \mathbf{x}') \tag{A.27}$$

Useful cross product identities are applied to the solution at this point to put it in a more convenient form.

$$q_0\mathbf{x}' \times -\mathbf{q} = -q_0\mathbf{x}' \times \mathbf{q} \tag{A.28}$$

$$\mathbf{q} \times \mathbf{x}' \times -\mathbf{q} = -\mathbf{q} \times \mathbf{x}' \times \mathbf{q} \tag{A.29}$$

$$-q_0(\mathbf{x}' \times \mathbf{q}) + q_0(\mathbf{q} \times \mathbf{x}') = 2q_0(\mathbf{q} \times \mathbf{x}') \tag{A.30}$$

$$-\mathbf{q} \times \mathbf{x}' \times \mathbf{q} = -((\mathbf{q} \cdot \mathbf{q})\mathbf{x}' - (\mathbf{q} \cdot \mathbf{x}')\mathbf{q}) = (\mathbf{q} \cdot \mathbf{x}')\mathbf{q} - (\mathbf{q} \cdot \mathbf{q})\mathbf{x}' \tag{A.31}$$

The above identities are substituted back into the equation to reduce the complexity of the solution:

$$\mathbf{x} = q_0^2\mathbf{x}' + 2q_0(\mathbf{q} \times \mathbf{x}') + (\mathbf{q} \cdot \mathbf{x}')\mathbf{q} - (\mathbf{q} \cdot \mathbf{q})\mathbf{x}' + \mathbf{q}(\mathbf{q} \cdot \mathbf{x}') \tag{A.32}$$

$$\mathbf{x} = (q_0^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{x}' + 2q_0(\mathbf{q} \times \mathbf{x}') + 2(\mathbf{q} \cdot \mathbf{x}')\mathbf{q} \tag{A.33}$$

The dot and cross products are expanded into equivalent matrix form:

(A.34)
$$(q_0^2 - \mathbf{q}\cdot\mathbf{q})\mathbf{x}' = (q_0^2 - q_1^2 - q_2^2 - q_3^2)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{x}'$$

(A.35)
$$2q_0(\mathbf{q}\times\mathbf{x}') = 2q_0\begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}\mathbf{x}'$$

(A.36)
$$2(\mathbf{q}\cdot\mathbf{x}')\mathbf{q} = 2\begin{bmatrix} q_1^2 & q_1q_2 & q_1q_3 \\ q_1q_2 & q_2^2 & q_2q_3 \\ q_1q_3 & q_2q_3 & q_3^2 \end{bmatrix}\mathbf{x}'$$

$$\mathbf{x} = \left((q_0^2 - q_1^2 - q_2^2 - q_3^2)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 2q_0\begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} + 2\begin{bmatrix} q_1^2 & q_1q_2 & q_1q_3 \\ q_1q_2 & q_2^2 & q_2q_3 \\ q_1q_3 & q_2q_3 & q_3^2 \end{bmatrix}\right)\mathbf{x}'$$
(A.37)

(A.38)
$$\mathbf{x} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & -2q_0q_3+2q_1q_2 & 2q_0q_2+2q_1q_3 \\ 2q_0q_3+2q_1q_2 & q_0^2-q_1^2+q_2^2-q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}\mathbf{x}'$$

Therefore, the rotation matrix can be derived from a unit quaternion representation.

(A.39)
$$\mathbf{R} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & -2q_0q_3+2q_1q_2 & 2q_0q_2+2q_1q_3 \\ 2q_0q_3+2q_1q_2 & q_0^2-q_1^2+q_2^2-q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

Annex B
Acronyms

AO Area of Operations

AOR Area of Responsibility

ANSI American National Standards Institute

BIT Built-In Test

C/A Course Acquisition GPS

C4ISR Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance

CPU Central Processing Unit

DM Domain Model

DMU Dynamic Measurement Unit

DGPS Differential GPS

ECEF Earth-Centred, Earth-Fixed

FOG Fibre Optic Gyroscope

GPS Global Positioning System

HAAW Heavy Anti-Armour Weapon

IEC International Electrotechnical Commission

IEEE Institute of Electrical and Electronics Engineers

IMU Inertial Measurement Unit

ISO International Standards Organization

JAUS Joint Architecture for Unmanned Systems

JTA Joint Technical Architecture

LAAW Light Anti-Armour Weapon

LAN Local Area Network

MGRS Military Grid Reference System

MMU Memory Management Unit

MSL Mean Sea Level

NBC Nuclear, Biological, Chemical

NIST National Institute of Standards and Technology

NSU Navigational Sensor Unit

NTP Network Time Protocol

OCU Operator Control Unit

OEM Original Equipment Manufacturer

OPI Office of Primary Interest

PID Proportional Integral Differential

POST Power-On Self Test

RA Reference Architecture

RGA Rate Gyro Accelerometer

RMS Root Mean Square

RPG Rocket Propelled Grenade

RPY Roll, Pitch, Yaw

RTK Real Time Kinematic

SAE Society of Automotive Engineers

SI System International

SMA Senior Military Advisor

TNA Thermal Neutron Activation

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

US United States

USA United States of America

USV Unmanned Space Vehicle

UTC Universal Time Coordinated

UTM Universal Trans Mercator

UUV Unmanned Underwater Vehicle

UxV Unmanned (Aerial, Ground, Underwater, Space) Vehicle

WG Working Group

WGS World Geodetic System

Annex C
Notation

α latitude angle

β longitude angle

σ^2 Variance of a variable population

σ Standard deviation of a variable population

θ rotation about the modified y-axis in radians for Euler RPY

ϕ rotation about the modified x-axis in radians for Euler RPY

ψ rotation about initial z-axis in radians for Euler RPY

ℓ Local Coordinate Frame of Reference (Robot ego-centric)

$\mathbf{p}_\ell$ Local Coordinate Frame pose (Robot ego-centric)

p_{rpy} JAUS-compliant

$\mathbf{p}_W$ World Coordinate Frame pose

$\mathbf{p}_{W-UTM}$ World Coordinate Frame pose with UTM

$\mathbf{q}$ quaternion vector

$\mathbf{q}^\star$ quaternion conjugate

q_s quaternion scalar component

q_x quaternion imaginary projection along the i axis

q_y quaternion imaginary projection along the j axis

q_z quaternion imaginary projection along the k axis

s^2 Variance of a variable sample

s Standard deviation of a variable sample

x_ℓ x displacement in local pose

x_W x displacement in global pose

y_ℓ y displacement in local pose

y_W y displacement in local pose

z_ℓ z displacement in global pose

z_W z displacement in global pose

D Displacement vector

G Units of gravity ($9.81\frac{m}{s^2}$)

R Rotation Matrix

T Transformation Matrix

W World Coordinate Frame of Reference

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada - Suffield
PO Box 4000, Medicine Hat, AB, Canada T1A 8K6

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Standards for Representation in Autonomous Intelligent Systems

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Erickson, D.

5. DATE OF PUBLICATION (month and year of publication of document)

December 2005

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc).

56

6b. NO. OF REFS (total cited in document)

17

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Technical Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R&D Canada - Suffield
PO Box 4000, Medicine Hat, AB, Canada T1A 8K6

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant).

12ph01

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

n/a

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.)

DRDC Suffield TR 2005-228

10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report outlines the standards for nomenclature, definitions, units of measure, and representations for use between systems, subsystems, devices, nodes, and processes. The aim of this document is to reduce ambiguity by establishing common rules for the interpretation of information. Ths adoption of these standards within the work by DRDC and from contractors who assist the program will increase the efficiency of implementation and increase the interoperability of projects and components.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

coordinate reference frames, SI units, pose representations, ISO/IEC WG14, IEEE 754 fixed floating point notation, engineering notation, UT1, UT, UTC, GPS, Z Zulu Time, NTP network time protocol, message formats, communication, standards, robots, autonomous robotics, AIS program