Abstract—We propose a fuzzy qualitative (FQ) version of robot kinematics with the goal of bridging the gap between symbolic or qualitative functions and numerical sensing and control tasks for intelligent robotics. First, we revisit FQ trigonometry, and then derive its derivative extension. Next, we replace the trigonometry role in robot kinematics using FQ trigonometry and the proposed derivative extension, which leads to a FQ version of robot kinematics. FQ transformation, position and velocity of a serial kinematics robot are derived and discussed. Finally, we propose an aggregation operator to extract robot behaviours with the highlight of the impact of the proposed methods to intelligent robotics. The proposed methods have been integrated into XTRIG MATLAB toolbox and a case study on a PUMA robot has been implemented to demonstrate their effectiveness.

I. INTRODUCTION

Future robotics is a multidisciplinary research area. Its central aim is to integrate traditional robotics, artificial intelligence, cognitive science and neuroscience etc. Research into robotics has traditionally emphasized low-level sensing and control tasks including sensory processing, path planning, and control. Future robotics is concerned with endowing robots and software agents with higher-level cognitive functions that enable them to reason, act and perceive in dynamic, incompletely known and unpredictable environments. The methods used should be flexible enough to combine the strengths of conventional programming with those of machine learning methods, e.g., the existing combination methods of fuzzy inference and neural networks by Kiguchi and Fukuda [1] and Ge et al [2]. On the other hand, cognitive-function descriptions for robot behaviour and behaviour-based robotics are urgently required to confront challenges raised by future robotics. Robots can be generally categorised as manipulators, mobile robots and mobile manipulators. Due to the nature of sensors (e.g, sonar) that a mobile robot has, methods for its behaviour description have been developed in the past three decades [3–5]. However, there have been very few behaviour methods for both robotic arms and mobile manipulators reported so far. For instance, Jenkins and Mataric [6] presented performance-derived behaviour vocabularies as a methodology for automatically deriving behaviour vocabularies to serve as skill-level interfaces for autonomous humanoids. Although Khatib et al [7] presented a whole-body control framework that decouples the interaction between the task and postural objectives and compensates for the dynamics in their respective space, it is difficult to use the same framework to further describe the behaviours of a body.

The state-of-the-art fuzzy qualitative reasoning convinces that it could provide a solution for closing the gap between low-level sensory processing and control tasks, and high-level symbolic functions. Fuzzy reasoning has been significantly developed and has attracted much attention and exploitation from industry and research communities in the past four decades [8]. Fuzzy reasoning is good at communicating with sensing and control level subsystems by means of fuzzification and defuzzification methods. It has powerful reasoning strategies utilising compiled knowledge through conditional statements so as to easily handle mathematical and engineering systems in model free manner. Fuzzy reasoning also provides a means of handling uncertainty in a natural way making it robust in significantly noisy environments. However, the fact that its knowledge is primarily shallow, and the questions over the computational overhead associated with handling grades of membership of discrete fuzzy sets must be taken into account if multi-step reasoning is to be carried out. On the other hand, qualitative and model-based reasoning has been successfully deployed in many applications such as autonomous spacecraft support [9], Systems Biology [10] and qualitative systems identification [11]. It has the advantage of operating at the conceptual modelling level, reasoning symbolically with models which retain the mathematical structure of the problem rather than the input/output representation of rule bases. These models are incomplete in the sense that, being symbolic, they do not contain, or require, exact parameter information in order to operate. Qualitative reasoning can make use of multiple ontologies, can explicitly represent causality, enable the construction of more sophisticated models from simpler constituents by means of compositional modelling, and infer the global behaviour of a system from a description of its structure [12], [13]. These features can, when combined with fuzzy values and operators, compensate for the lack of ability in fuzzy reasoning alone to deal with qualitative inference about complex systems. The computational cause-effect relations contained in qualitative models facilitates analysing and explaining the behaviour of a structural model. Based on a scenario generated from fuzzy reasoning’s fuzzification process, fuzzy qualitative reasoning may be able to build a behavioural model automatically, and use this model to generate a behaviour description, acceptable by symbolic
systems, either by abstraction and qualitative simulation or as a comprehensive representation of all possible behaviours utilising linguistic fuzzy values.

Fuzzy reasoning theory and its applications have been greatly developed since a famous controversy of fuzzy logic in 1993 [14]. Fuzzy reasoning methods are becoming more and more popular in intelligent systems [15], [16], especially in applications which integrate hybrid methods including evolutionary computing [17], [18], decision trees [19], neural networks [20], data mining [21], and others [22–24]. Qualitative reasoning was reviewed in [12], [13], [25], [26]. The integration of fuzzy reasoning and qualitative reasoning (i.e. fuzzy qualitative reasoning) provides an opportunity to explore research problems (e.g. spatial reasoning) using the advantages of both the fuzzy reasoning and qualitative reasoning. Some of fuzzy qualitative reasoning contributions can be found in [27–31]. Shen and Leitch [27] use a fuzzy quantity space which allows for a more detailed description of the values of the variables. Such an approach relies on the extension principle and approximation principle in order to express the results of calculations in terms of the fuzzy sets of the fuzzy quantity space. Model-based methods have been successfully applied to a number of tasks in the process domain. However, while some effort has been expended on developing qualitative kinematics models, the results have been limited [32–34]. The basic requirement for progressing in this domain is the development of a qualitative version of the trigonometric rules. Buckley and Eslami [35] proposed a definition of fuzzy trigonometry from the fuzzy perspective without consideration of the geometric meaning of trigonometry. Some progress has been made in this direction by Liu [36] but, as with other applications of qualitative reasoning, the flexibility gained in variable precision by integrating the fuzzy and qualitative approaches plays a crucial role in the kinematics domain.

This paper proposes a novel representation of robot kinematics with the goal of solving the intelligent connection problem (also known as symbol grounding problem) for physical systems, in particular robotic systems. This problem is one of the key issues in AI robotics [3], [37] and relates to a wide range of research areas such as computer vision [38]. Based on FQ trigonometry [39], we present a derivative extension to FQ trigonometry. The derivative extension enables us further to derive FQ robot kinematics. This paper is organised as follows. Section II revisits FQ trigonometry, Section III derives the derivative extension to FQ trigonometry, Section IV derives FQ robot kinematics and Section V proposes an aggregation operator for calculation of the presented fuzzy qualitative kinematics with adjustable computational cost. We present a case study of a Puma robot in Section VI before we conclude this paper at Section VII.

II. FUZZY QUALITATIVE TRIGONOMETRY REVISIT

Robot kinematics actually is the combination of trigonometry and geometric constraints of physical robots. That is to say, trigonometry is the theoretical foundation of robot kinematics, which is the starting idea for the study on FQ trigonometry. FQ trigonometry [39] proposes a promising representation and transformation interface for the analysis of general trigonometry-related physical systems from an artificial intelligence perspective. FQ trigonometry provides theoretical foundations for the representation of trigonometric properties of physical systems, e.g. robotic systems, and it is able to represent numerical, symbolic and hybrid data and to transfer between these different types of data using fuzzy qualitative techniques. The section leads us through FQ trigonometry.

A. Fuzzy Qualitative Trigonometry

For implementation we have chosen the representation of the four-tuple fuzzy numbers because of its simplicity, high resolution and good compositionality [27], [40]. A four-tuple fuzzy number representation, namely \([a, b, \tau, \beta]\) with the condition \(a < b\) and \(a \times b > 0\), can be used to approximate the membership distribution of a normal convex fuzzy number. The four tuple set, \([a, b, \tau, \beta]\), can be defined by the following membership function, \(\mu_A(x)\),

\[
\mu_A(x) = \begin{cases} 
0 & x < a - \tau \\
\tau^{-1}(x - a + \tau) & x \in [a - \tau, a] \\
1 & x \in [a, b] \\
\beta^{-1}(b + \beta - x) & x \in [b, b + \beta] \\
0 & x > b + \beta 
\end{cases}
\]  

(1)

Note that if \(a = b\), equation 1 is a triangular fuzzy number with its membership as \(\mu_A(x) = [\tau^{-1}(x - a + \tau), \beta^{-1}(a + \beta - x)]\), where \(x \in [a - \tau, a + \beta]\). Besides, if \(a = b\) and \(\tau = \beta = 0\), it turns the 4-tuple fuzzy number into a real number. For instance \([1, 1, 0, 0]\) stands for the real number 1. In addition, the arithmetic operations on the 4-tuple parametric representation of fuzzy numbers are given in Appendix A, where \(\prec\) is the partial order \(<_\alpha\) when \(\alpha = 0\), which was used in paper [27]. The partial order has been given in a general form, i.e., \(\{a, b, \tau, \beta\} \in \{\prec, =, \leq, \geq, \in, \subseteq, \supseteq, \approx\}\). For instance, partial order \(<_\alpha\) is defined such that for \(m, n (m \neq n)\), we say \(m\) is \(\alpha\)-less than \(n\), \(m <_\alpha n\), iff \(a < b\), \(a \in m_\alpha\), \(b \in n_\alpha\) with \(m_\alpha\) and \(n_\alpha\) being the \(\alpha\)-cuts of \(m\) and \(n\), respectively. It is believed that fuzzy arithmetic is still a developing research topic, usage of the abovementioned fuzzy operators is constrained by the fact that an assignment of \(\alpha\) value is required in a calculation cycle.

FQ trigonometry actually is the FQ description of conventional trigonometry. The unit circle of conventional trigonometry has been modified by the introduction of quantity spaces for orientation and translation. Conventional trigonometric functions (e.g. a sine function) have been abstracted to obtain FQ versions of these operations. For a good survey on FQ reasoning please see [39]. The quantity space for every variable in the system is a finite and convex discretisation of the real number line. FQ quantity space \(Q\) consists of an orientation component \(Q^o\) and a translation component \(Q^d\).
It can be described as,

\[
Q^a = [QS_a(\theta_1), \cdots, QS_a(\theta_i), \cdots QS_a(\theta_m)]
\]
\[
Q^d = [QS_d(l_1), \cdots, QS_d(l_j), \cdots QS_d(l_n)]
\]

where \(QS_a(\theta_i)\) denotes the state of an angle \(\theta_i\), \(QS_d(l_j)\) denotes the state of a distance \(l_j\), and \(m\) and \(n\) are the number of the elements of the two components. The position measurement of \(P(QS_a(\theta_i), QS_d(l_j))\) determined by both the characteristics of the fuzzy membership functions of \(QS_a(\theta_i)\) and \(QS_d(l_j)\). The geometric meaning of FQ trigonometry is demonstrated in a proposed FQ unit circle, in which the motion is described by an orientation component and a translation component. A FQ unit circle is nothing but a conventional trigonometric circle whose axes are replaced by unit quantity spaces. The position state of a FQ point is defined by the projections of the point into FQ axes. For instance, let the \(X\) axis projection of a FQ point \(P\) be \(X_p\), and the \(Y\) axis projection as \(Y_p\), its FQ position can be described as \(P = (X_p, Y_p)\). It means that a point position in FQ coordinates is described by fuzzy sets, hence, fuzzy arithmetic and its characteristics can be applied to calculation of FQ states in FQ coordinates. In accordance with Cartesian coordinates convention, the position, \(P_o(X_o, Y_o) = ([0, 0, 0, 0], [0, 0, 0, 0])\), is the FQ origin, where the axes intersect. Counterclockwise is the positive orientation, the number of qualitative orientation states in a full circle starting from the \(X\) axis is denoted by \(p\), the number of qualitative translation states is denoted by \(q_{axis}\), (i.e., \(q_x, q_y\)). Though different functions can be employed to generate fuzzy numbers of a quantity space, two conditions must hold. First, fuzzy numbers in each component must be origin symmetric. Second, special real numbers (e.g., \(0, \pi/2, \pi, 3\pi/2\) in an orientation, and \(-1, 0, \pi\) in a translation) are the centres of some certain fuzzy numbers in their corresponding quantity space. For instance, a right angle in conventional trigonometry is corresponded to fuzzy number \(QS_a(p/4 + 1)\), whose centre is equal to \(\pi/2\). Please note that, for a 4-tuple fuzzy number (e.g., \([a, b, \tau, \beta]\)), its centre is defined by \(\left((a + b)/2\right) \times 2\pi\). Note that the fuzzy numbers in the orientation quantity space need to be multiplied by \(2\pi\) before they are applied to the arithmetic in Appendix A.

A MATLAB toolbox named XTRIG has been developed by [39] to implement the proposed FQ trigonometry. XTRIG has a function for the generation of 4-tuple fuzzy numbers, say, a fuzzy number \([a, b, \tau, \beta]\), the function is \(b - a = \kappa_0 \tau\) with the condition \(\tau = \beta\). \(\kappa_0\) is a threshold parameter to define the shape of fuzzy numbers. For instance, let the number of the quantity space of an orientation \(p\) be 17 for a FQ trigonometric circle, those of a translation, \(q_x\) and \(q_y\), be 23 each, and set \(\kappa_0\) as 5. XTRIG generates the corresponding quantity spaces of the orientation \(Q^a\) and translation \(Q^d\), which are shown in Fig. 1. It shows that the fuzzy-number generation function distributes the 16 fuzzy numbers in the translation component in the range \([0, 1]\), and two sets of the 21 fuzzy numbers in the range \([-1, 1]\). Fig. 2 shows the conversion from a fuzzy qualitative angle to its FQ position, where the qualitative angle is the third orientation state in the orientation quantity space \(Q^a\) (i.e., \([0.099, 0.151, 0.0104, 0.0104]\)). The conversion is carried out as follows. Firstly, we calculate the positions of crossing points \(A, B\) between the fuzzy number (i.e., the third orientation angle) and its adjacent fuzzy numbers; Secondly, we calculate crossing point positions \(A', B'\) between the unit circle curve and the two line segments \(OA\) and \(OB\); Thirdly, we obtain two line segments \(A_x'B_x'\) and \(A_y'B_y'\) by projecting points \(A', B'\) into \(X\) and \(Y\) coordinates, respectively. Finally we generate two sets of fuzzy numbers which totally or partially are in the range \(A_x'B_x'\) and \(A_y'B_y'\). The set of fuzzy numbers are within \(A_x'B_x'\) is the FQ position on \(X\) axis of the orientation angle (i.e., \(QS_a(3)_x\)); the other set within \(A_y'B_y'\) is for \(Y\) axis projection (i.e., \(QS_a(3)_y\)).

The FQ position is, 

\[
P_{QS_a(3)} = \left(\begin{array}{c}
P_{QS_a(3)_X} \\
P_{QS_a(3)_Y}
\end{array}\right)
\]

where 

\[
P_{QS_a(3)} = 
\begin{bmatrix}
0.5583 & 0.6417 & 0.0167 & 0.0167 \\
0.6583 & 0.7417 & 0.0167 & 0.0167 \\
0.7583 & 0.8417 & 0.0167 & 0.0167
\end{bmatrix}
\]

(Fig. 1. Quantity spaces described by fuzzy membership functions (\(p=17, q_x=23\) and \(q_y=23\))

Fig. 2. The relation between qualitative translation and orientation)
The FQ values of the third orientation angle in the X and Y axes are denoted by a set of three 4-tuple fuzzy numbers and a set of five fuzzy numbers generated from its translation quantity space when its \( \alpha \) cut is defaulted as zero. The conversion methodology can also be applied to the conversion from fuzzy numbers in its translation quantity space to those in its orientation quantity space.

### B. Fuzzy Qualitative Trigonometric Functions

FQ trigonometric functions are derived from conventional trigonometric functions based on the extension principle, which allows the extension of classical mathematical operators to the fuzzy domain. It means that an arithmetic operation performed between \( n \) fuzzy sets will yield a fuzzy set of the same form. Each trigonometric function is derived and illustrated using FQ trigonometry and the quantity spaces of the FQ coordinates. For simplicity, the symbols of quantitative trigonometric functions are used to describe their counterparts in FQ trigonometry but with qualitative variables instead (e.g., \( QS(S(j)) \), where \( i \in \{a, d\} \) and \( j \in \{p, q\} \)). In order to give a clear explanation of FQ trigonometric functions, let us consider the example for the third FQ angle in Fig. 2 as a general case of the \( i \)th orientation angle (i.e., \( QS(S(i)) \)). \( P_{QS(S(i))} \) stands for the FQ position of the intersection of \( QS(S(i)) \) to a unit circle curve, and its projected FQ positions on the axes of X and Y are \( QS_d(|A_x'B_x'|) \) and \( QS_d(|A_y'B_y'|) \). For now we are ready to define FQ trigonometric functions,

\[
\begin{align*}
\sin(QS_a(i)) &= \alpha QS_d\left(\left|\begin{array}{c}
A_y'B_y' \\
1 & 1 & 0 & 0
\end{array}\right|\right) = \alpha QS_d\left(\left|\begin{array}{c}
1 & 0 & 0 & 0
\end{array}\right|\cos(QS_a(i))\right) \\
\cos(QS_a(i)) &= \alpha QS_d\left(\left|\begin{array}{c}
A_x'B_x' \\
1 & 1 & 0 & 0
\end{array}\right|\right) = \alpha QS_d\left(\left|\begin{array}{c}
1 & 1 & 0 & 0
\end{array}\right|\right) \\
\sec(QS_a(i)) &= \alpha QS_d\left(\left|\begin{array}{c}
1 & 1 & 0 & 0
\end{array}\right|\right) = \alpha QS_d\left(\left|\begin{array}{c}
1 & 1 & 0 & 0
\end{array}\right|\right) \\
\csc(QS_a(i)) &= \alpha QS_d\left(\left|\begin{array}{c}
1 & 1 & 0 & 0
\end{array}\right|\right) = \alpha QS_d\left(\left|\begin{array}{c}
1 & 1 & 0 & 0
\end{array}\right|\right) \\
\arcsin(QS_d(|A_y'B_y'|)) &= \alpha QS_a(i) \\
\arccos(QS_d(|A_x'B_x'|)) &= \alpha QS_a(i)
\end{align*}
\]

where \( \sin(QS_a(i)), \cos(QS_a(i)), \sec(QS_a(i)) \) and \( \csc(QS_a(i)) \) FQ functions are straightforward as shown in Fig. 2. The \( \text{i}th \) fuzzy qualitative position’s arcsine and arccosine functions shown in equations 4e and 4f, i.e., \( \arcsin(QS_d(i)) \) and \( \arccos(QS_d(i)) \), are the inverse of its sine and cosine functions given in equations 4a and 4b. Refer to Fig. 2, if points \( B_y' \) and \( Y \) are the crossing points of the \( \text{i}th \) fuzzy number and its adjacent fuzzy numbers in Y coordinate, the fuzzy numbers in the orientation quantity space within arc \( A_y'B_y' \) is its corresponding qualitative angle. FQ arcsine and arccosine functions, as their quantitative counterparts have, are constrained that \( \arcsin(QS_d(i)) \subseteq [-\frac{\pi}{2}, \frac{\pi}{2}] \), and \( \arccos(QS_d(i)) \subseteq [0, \pi] \). Further, the tangent of \( QS_a(i) \), written as \( tan(QS_a(i)) \), is defined as the ratio of the opposite to the adjacent sides, \( QS_d(|A_y'B_y'|) / QS_d(|A_x'B_x'|) \). It can clearly see that the tangent of \( QS_a(i) \) is equal to the sine of \( QS_a(i) \) divided by the cosine of \( QS_a(i) \). The same can apply to cotangent function, then we obtain

\[
\begin{align*}
\tan(QS_a(i)) &= \frac{QS_d(|A_y'B_y'|)}{QS_d(|A_x'B_x'|)} = \frac{\sin(QS_a(i))}{\cos(QS_a(i))} \\
\cot(QS_a(i)) &= \frac{QS_d(|A_x'B_x'|)}{QS_d(|A_y'B_y'|)} = \frac{\cos(QS_a(i))}{\sin(QS_a(i))}
\end{align*}
\]

where

\[
\cos^2(QS_a(i)) + \sin^2(QS_a(i)) = \alpha QS_d(p)
\]

With the setting of the example in Fig. 2, we obtain the following results,

\[
\begin{align*}
\cos(QS_a(114)) &= \left[0.7583 0.8417 0.0167 0.0167\right] \\
\sec(QS_a(14)) &= \left[4.1379 6.3158 0.2670 0.7430\right] \\
\tan(QS_a(3)) &= \left[0.6634 0.8462 0.0323 0.0415\right] \\
\cot(QS_a(3)) &= \left[1.1818 1.5075 0.0552 0.0772\right] \\
\arcsin(QS_d(19)) &= \left[0.0990 0.1510 0.0104 0.0104\right] \\
\arccos(QS_d(20)) &= \left[0.0365 0.0885 0.0104 0.0104\right]
\end{align*}
\]

The example above indicates two characteristics of FQ trigonometry. First, FQ trigonometric functions have period characteristic, e.g., \( \cos(QS_a(114)) = \cos(QS_a(2 + 16 \times 7)) \). Second, FQ tangent and cotangent functions are calculated based on FQ sine, cosine functions and the fundamental trigonometric identity. The use of the latter ensures the
elimination of those fuzzy numbers that are the results generated by FQ sine and cosine functions but they lose their geometric meaning.

III. DERIVATIVES OF FUZZY QUALITATIVE TRIGONOMETRY

Combining FQ trigonometry with the extension principle [41] and the principle of convex and normal fuzzy sets’ mapping [42], we have the lemmas below for the derivatives of FQ trigonometric functions.

**Lemma I:** If \( y = d^n(x) \) is a \( n \) order derivative function, then with \( A \) being a fuzzy set in quantity space \( X \) \((x \in \bigcup FQT_\alpha(Q_X))\), \( d^n() \) maps from \( A \) to a fuzzy set \( B \) in quantity space \( Y \) \((y \in \bigcup FQT_\alpha(Q_Y))\) such that

\[
\mu_B(y) = \sup_{x \in d^{-n}(y)} \mu_A(x) \tag{6}
\]

where \( FQT_\alpha() \) is one of FQT functions, \( \alpha \) stands for \( \alpha \)-cut and \( Q_x, Q_y \subset Q \).

Fuzzy numbers herein are fuzzy sets of the real line \( \mathbb{R} \) with a normal, fuzzy convex and continuous membership function of bounded support. In order to demonstrate the proposed extension, we have implemented it into XTRIG MATLAB toolbox. Recall the example used in Fig. 1. For first-order derivative, Lemma I can be rewritten as,

\[
\mu_B(y) = \sup_{x \in d^{-1}(y)} \mu_A(x) \tag{7}
\]

For instance, first order derivative of \( \cos(QS_\alpha(3)) \) can be calculated as,

\[
d^1(\cos(QS_\alpha(3))) = \alpha - \sin(QS_\alpha(3)) = 0
\]

\[
= \begin{bmatrix}
-0.7119 & -0.7966 & 0.0169 & 0.0169 \\
-0.8136 & -0.8983 & 0.0169 & 0.0169 \\
-0.9153 & -1.000 & 0.0169 & 0
\end{bmatrix}
\]

(8)

With the definition of the support of a fuzzy set, \( \text{supp}(A) \), Lemma I can be further developed as a lemma below for the mapping of a fuzzy set’s support.

**Lemma II:** With the convention in Lemma I, the following is obtained.

\[
d^n(\text{supp}(A)) = \text{supp}(B). \tag{9}
\]

Furthermore, applying Lemma II to the example, we have the fuzzy sets’ support version of FQ trigonometric first-order derivative as below,

\[
d^1(\sup (\cos(QS_\alpha(3)))) = 0
\]

\[
= \begin{bmatrix}
-0.7288 & -0.8135 \\
-0.8305 & -0.9152 \\
-0.9322 & -1.000
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-0.7288 & -1.000
\end{bmatrix}
\]

(10)

It shows that the support of the derivative output has two representations: local support and global support. The local support consists of the support values of the fuzzy numbers of the output; the global support is the support of a qualitative state. In the XTRIG toolbox, the column dimension is used to tell the representation of a fuzzy number from that of its support.

**Lemma III:** Quantity spaces in a unit circle must be \( x \)-axis or \( y \)-axis symmetric, or origin symmetric.

Lemma above is the condition that Lemma I and II hold; actually it is also the condition for FQ trigonometry. The characteristic of \( x \)-axis or \( y \)-axis symmetry is required for a translation quantity space, and that of origin symmetry is for an orientation quantity space.

Say \( A \) is a fuzzy set. \([A]_\alpha \) is a compact subset of \( \mathbb{R} \) and \( M(A) \) is a closed interval bounded by the lower and upper possibilistic mean values \([M_\alpha(A), M^*(A)]\) of \( A \) [43], then it leads to the following definition,

**Definition I:** \( \alpha \)-centre of a fuzzy set is its possibilistic mean value.

\[
C(A_\alpha) = \frac{M^*(A_\alpha) + M_\alpha(A_\alpha)}{2} \tag{11}
\]

where

\[
M^*(A_\alpha) = \int_0^1 \text{Pos}[A > b(\alpha)] \times \max[A]^* \, d\alpha
\]

\[
M_\alpha(A_\alpha) = \int_0^1 \text{Pos}[A \leq b(\alpha)] \times \min[A]^* \, d\alpha
\]

and \( \text{Pos} \) denotes possibility.

**Definition II:** \( \alpha \) power support of a fuzzy set is the proportion of its \( \alpha \)-power to its \( \alpha \)-height.

\[
PS(A_\alpha) = \left\{ \begin{array}{l} 
\int_R \mu^\alpha_A(x) \, dx \quad \alpha = 1 \\
\frac{\int_R \mu^\alpha_A(x) \, dx}{\int_0^1 \mu^\alpha_A(x) \, dx} \quad \alpha \neq 1
\end{array} \right. \tag{12}
\]

**Definition III:** A fuzzy qualitative state \( QS(A) \) in a \( \alpha \)-level of a fuzzy set \( \mu_A(X) \) can be described by its \( \alpha \)-centre and scaled \( \alpha \) power support.

\[
QS(A) = \alpha \left[ C(A), \ \gamma PS(A) \right] \tag{13}
\]

where \( \gamma \in [0 \ 1] \) is a scaling factor. Hence, we have two representations for a qualitative state, four-tuple fuzzy numbers and centre-power-support fuzzy intervals. For instance, given a four tuple fuzzy number \( A = [a \ b \ τ \ β] \), its centre-power-support representation can be obtained below with default value \( \alpha = 0 \),

\[
QS(A) = \alpha \left[ C(A), \ \gamma PS(A) \right] = \alpha \left[ \frac{a + b}{2}, \ \ b - a + \frac{τ + β}{2} \right]
\]

Let us consider \( QS_\alpha(3) = [0.2034 \ 0.2881 \ 0.01069 \ 0.01069] \), its fuzzy qualitative state at \( \alpha_0 \) level is [0.2458, 0.1016] as its centre, 0.1016 as its scalable radius. A fuzzy qualitative state’s linguistic meaning or precision can be controlled by the combination of the adjustment of its scaling factor and fuzzy quantity spaces. Further, arithmetic for FQ states at a \( \alpha \) level can be defined in terms of midpoint-radius (MR) representation and Definition III, for literature on interval MR representation, please see [44], [45].
Definition IV: For $QS(A) =_{\alpha} [C(A) \gamma_A PS(A)]$ ($QS(A) \in Q$), $QS(B) =_{\alpha} [C(B) \gamma_B PS(B)]$, ($QS(B) \in Q$), $\gamma_A$ and $\gamma_B$ are scaling factors, respectively, we define

$$QS (A) + QS (B) =_{\alpha} [ C (A) + C (B) , \ \gamma_A PS (A) + \gamma_B PS (B) ]$$ (14a)

$$QS (A) - QS (B) =_{\alpha} [ C (A) - C (B) , \ \gamma_A PS (A) + \gamma_B PS (B) ]$$ (14b)

$$QS (A) \times QS (B) =_{\alpha} [ C (A) \times C (B) , |\rho(C (A))| \times \gamma_B PS (B) + |C (B)| \gamma_A PS (A) + \gamma_A \gamma_B PS (A) PS (B) ]$$ (14c)

where

$$\{ \begin{array}{ll} \rho = 1 \ & \text{strong MR multiplication} \\
\rho = \frac{2}{3} \ & \text{weak MR multiplication} \end{array}$$

and

$$I_{QS(A)} =_{\alpha} \left[ \frac{C(B)}{D}, \frac{\gamma_B PS(B)}{D} \right]$$ (14d)

where

$$D =_{\alpha} C^2 (B) - \gamma_B^2 PS^2 (B), \ \text{and} \ QS (B) \neq 0$$

and

$$QS (B) =_{\alpha} QS (A) \times \frac{I}{QS (B)} \ \text{for} \ QS (B) \neq 0$$ (14e)

Due to Rump’s overestimation theory in [46] that the overestimation of MR interval arithmetic compared to interval set operations is uniformly bounded by a factor 1.5 in radius, we introduce an index to distinguish strong and weak multiplication in terms of interval MR representation, see equation 14c. Strong multiplication with the index as 1 includes the overestimation caused by interval MR multiplication, on the other hand, weak multiplication with the index as 2/3 does not. Generally speaking, interval computation in MR representation is faster, especially matrix operations, than infimum-supremum representation. This overestimation characteristic provides a method of controlling tradeoffs for the selection of interval representations for interval-related applications, e.g., real-time humanoid systems. The algorithms in MR representation and its conversion on four-tuple-fuzzy-number-based computation have been implemented into the XTRIG toolbox. We use MR-based computation in the rest of this paper, not only because its faster computation nature is suitable for robotics application, but also because MR-based representation is more flexible to control precision in data conversion among numerical, symbolic and hybrid data, thanks to the fact that fuzzy quantity space with domain background allows convert parameters among different types of data.

IV. FUZZY QUALITATIVE ROBOT KINEMATICS

Based on FQ trigonometry, we first establish definitions for FQ vectors, and then introduce FQ transformations, finally derive FQ robot kinematics. The notation of conventional robotics in Craig’s book [47] has been used in this paper.

A. Fuzzy Qualitative Transformations

Necessary definitions of FQ transformation are provided in this section.

Definition V: A FQ point vector is

$$QS_d (p) = QS_d (x) \ i + QS_d (y) \ j + QS_d (z) \ k,$$

where $i, j, k$ are unit vectors along $x, y, z$ FQ coordinate axes, respectively. It can be represented in homogeneous FQ coordinates as,

$$QS_d (p) = \begin{bmatrix} QS_d (x) & QS_d (y) & QS_d (z) & 1 \end{bmatrix}^T,$$

where the superscript $T$ indicates the transpose of the row vector into a column vector and $I$ is a unit vector.

Definition VI: Given a fuzzy qualitative point $p_0$, its rigid transformation $p$ by a transformation matrix $H$ is represented by the matrix product,

$$QS_d (p) = H \cdot QS_d (p_0).$$ (15)

FQ transformations are used to describe the fuzzy qualitative position and orientation of objects. For a rigid body, its transformations includes translation transformations $T$ and orientation transformations $R$, whose descriptions are given as,

$$T = \text{Trans}(QS_d (p_x), QS_d (p_y), QS_d (p_z))$$

$$= \begin{bmatrix} I & O & O & QS_d (p_x) \\
O & I & O & QS_d (p_y) \\
O & O & I & QS_d (p_z) \\
O & O & O & I \end{bmatrix}$$ (16)

and

$$R = \text{Rot}(axis, QS_d (\theta_{axis})) \ \text{for} \ axis \in \{x, y, z\}. \ (17)$$

where

$$\text{Rot}(x, QS_d (\theta_x)) = \begin{bmatrix} I & O & O & O \\
O & \cos (QS_d (\theta_x)) & -\sin (QS_d (\theta_x)) & O \\
O & \sin (QS_d (\theta_x)) & \cos (QS_d (\theta_x)) & O \\
O & O & O & I \end{bmatrix} \ (18a)$$

$$\text{Rot}(y, QS_d (\theta_y)) = \begin{bmatrix} \cos (QS_d (\theta_y)) & O & \sin (QS_d (\theta_y)) & O \\
O & I & O & O \\
-\sin (QS_d (\theta_y)) & O & \cos (QS_d (\theta_y)) & O \\
O & O & O & I \end{bmatrix} \ (18b)$$

$$\text{Rot}(z, QS_d (\theta_z)) = \begin{bmatrix} \cos (QS_d (\theta_z)) & -\sin (QS_d (\theta_z)) & O & O \\
\sin (QS_d (\theta_z)) & \cos (QS_d (\theta_z)) & O & O \\
O & O & I & O \\
O & O & O & I \end{bmatrix} \ (18c)$$

where the elements of FQ transformations are 4-tuple fuzzy numbers in this paper, for simplicity, though those can be replaced by any types of fuzzy numbers. Note that the elements $O$ and $I$ in the MR representation are

$$O = \begin{bmatrix} 0, & 0 \end{bmatrix} \ \text{and} \ I = \begin{bmatrix} 1, & 0 \end{bmatrix}. $$
where the length of vectors $0$ and $1$ is determined by the maximum length of the output of FQ trigonometric functions at the same row as they are in a transformation matrix in equations 16 and 17. Due to the fact, unlike conventional trigonometry, the output of a FQ trigonometric function is a subset of a normalized quantity space instead of a unique real number; the propagation of trigonometric functions in FQ terms allows FQ transformations to transform among real numbers, fuzzy intervals and linguistic variables.

Let us consider a transformation of a FQ point vector from $p_0$ to $p_e$ by first rotating by the $y$ axis with an angle $QS_{\theta}(\theta)$, then translating by a position vector $QS_{d}(t)$. For demonstration purpose, parameters of its quantity space $p$ and $q$ are set as 16 and 4, that is to say, 16 qualitative states are used to describe a full orientation and 4 qualitative states to a unit length. The transformation is described in equation 19a, it generates all possible FQ states for the given motion. In order to compare the proposed method with conventional space transformation, equation 19a has been rewritten as equation 19b in terms of Euclidean geometry. The simulation shows that the initial position $p_0$ is labelled as ‘△’, its Euclidean transformation is labelled as ‘□’ in Fig. 3: the result of the fuzzy qualitative transformation are given in two representations, one is shown by a set of centres labelled as ‘□’ in Fig. 3 and the other is represented by a set of overlapped interval patches in 4. Fig. 3 shows that the Euclidean position ‘□’ is not in the normally distributed centre of the fuzzy qualitative positions ‘○’, this caused by fuzzy qualitative arithmetic [39]. This demonstrates that fuzzy qualitative transformation has extended Euclidean transformation in terms of qualitative data propagation instead of a simple replacement of Euclidean transformation. We expect that this characteristic to play a key role in the problem of robot intelligent connection.

$$p_e = \text{Rot}(y, QS_{\theta}(\theta)) \cdot \text{Trans}(QS_{d}(t)) \cdot p_0$$
$$= \text{Rot}(y, QS_{\theta}(0.79, 0.39)) \times \text{Trans}((4.25, 0.25], [11.25, 0.25], [4.25, 0.25]) \times \begin{bmatrix} 4, 0 \quad 3, 0 \quad 5, 0 \end{bmatrix}^T$$

$$(19a)$$

$$p_e = \text{Rot}(y, \theta') \cdot \text{Trans}(t') \cdot p_0$$
$$= \text{Rot}(y, 0.79) \times \text{Trans}(4.25, 11.25, 4.25) \begin{bmatrix} 4, 3, 5 \end{bmatrix}^T$$

$$(19b)$$

B. Fuzzy Qualitative Robot Kinematics

We first define a general robot kinematics in terms of FQ trigonometry, then consider its relation and advantages over conventional robotic kinematics. Denote a $n$-link spatial robot with its FQ home position as $QS_{d}(p_0)$, its end-effector position $QS_{d}(p)$ can be obtained by the transformations of its link components $H_i$ or the transformations $i^{-1}H_{DH}$ in

$$QS_{d}(p) = \prod_{i=1}^{n} H_i \cdot QS_{d}(p_0) = \prod_{i=1}^{n} i^{-1}H_{DH} \cdot QS_{d}(p_0)$$

$$(20)$$

where $i^{-1}H_{DH}$ is FQ Denavit-Hartenberg kinematics structure. By replacing conventional trigonometric function with FQ trigonometry, we have the following,

$$i^{-1}H_{DH} = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & -d_{\alpha_i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & C\theta_i S\alpha_{i-1} & \alpha_{i-1} \\ 0 & O & O & 1 \\ O & O & O & 1 \end{bmatrix}$$

$$(21)$$

where $C$ stands for cos; $S$ stands for sin. $\theta$ stands for $QS_{\theta}(\theta)$, $\alpha$ stands for $QS_{\alpha}(\alpha)$, $d$ stands for $QS_{d}(d)$, $\alpha$ stands for $QS_{\alpha}(\alpha)$, $d$ stands for $QS_{d}(d)$, $\theta_i$ is the rotation angle from $X_{i-1}$ to $X_i$ measured about $Z_i$, joint variable from revolute joints, $QS_{\alpha}(\alpha_{i-1})$ is the twist angle from $Z_{i-1}$ to $Z_i$ measured about $X_{i-1}$, $QS_{d}(d_i)$ is the distance from $X_{i-1}$ to $X_i$ measured along $Z_i$, joint variable for prismatic joints, $QS_{d}(a_{i-1})$ is the link offset, distance from $Z_{i-1}$ to $Z_i$ measured along $X_{i-1}$. $X_i, Y_i, Z_i$ is the coordinate axes of $i$th coordinates system mounted on $i$th link segment. For a graphical representation please see Craig’s book [47]. Further, robot kinematics in equation

Fig. 3. Comparison of a numeric position’s Euclidean transformation and the centres of interval patches of its fuzzy qualitative transformation

Fig. 4. Interval patches of the fuzzy qualitative transformation in Fig. 3

D-H parameter terms, respectively,
20 can be rewritten for the end-effector of a robot as follows,
\[
QS_d(p_\epsilon) = \prod_{i=1}^{n} H_{D,H} \cdot QS_d(p_0) \\
= \left[ R_{x3}(QS(\theta, \alpha_{i-1})) \cdot P_{x3}(QS(\theta, \alpha_i, d_i)) \right] \cdot QS_d(p_0)
\]

(22)

Furthermore, the FQ velocity of the end-effector of a serial robot can be derived as,
\[
QS^1_d(p_\epsilon) = QS_d(p_\epsilon) \\
= \frac{\partial \left( \prod_{i=1}^{n} H_{D,H} \right)}{\partial QS_d(p_\epsilon)} \cdot QS_d(p_0) \\
= \frac{\partial \left( \prod_{i=1}^{n} H_{D,H} \right)}{\partial QS_d(p_\epsilon) \cdot \partial QS_d(p_\epsilon) \cdot \partial QS_d(p_\epsilon) \cdot \partial QS_d(p_\epsilon)} \cdot QS_d(p_0) \\
= QS(J) \cdot QS_d(p_0)
\]

(23)

where \(QS(J)\) is FQ Jacobian matrix which is able to transform FQ velocity between the configuration space of the orientation angles and the Cartesian space of the end-effector. Note that variable transformed herein could be numeric, symbolic or hybrid and physical constant robotic parameters, e.g., \(d_i\) and \(d_i\), are not considered as FQ parameters. FQ Jacobian matrix \(QS(J)\) consists of FQ trigonometric functions of \((QS(\theta, \alpha_{i-1}))\) and their FQ derivatives. FQ robot kinematics can be mimicked as conventional robot kinematics, for instance, when the scaling factor \(\gamma\) in equation 13 is set as zero and its quantity space is sufficient big, e.g., 360 elements in the MR representation. Its advantage over conventional kinematics is that it is able to handle hybrid parameters in the context of robot kinematics. FQ robot kinematics could provide a base for solving the perception-action problem which is one of the bottleneck problems leading to intelligent robots.

V. QUANTITY VECTOR AGGREGATION

Quantity vector aggregation is proposed to extract robot behaviour description of FQ states generated by FQ trigonometry and its variants. The foundation of quantity vector aggregation is quantity point vector aggregation because quantity point vector aggregation is superset of their point vector aggregation. Both point and patch vectors can be aggregated by either point vectors or patch vectors. Understanding the geometrical motion of quantity vectors benefits the mathematical description of variable aggregation for handling and transferring different types of parameters in reasoning systems. Understanding the mathematical nature of reasoning methods such as qualitative reasoning, fuzzy reasoning and probabilistic reasoning will speed up the development of automatic specification and verification tools. It will also help to overcome limitations such as state explosion and undecidability.

For simplicity and without loss of generality, vector aggregation in a 3-dimensional quantity space is considered. Point vector aggregation can be used to describe rigid transformations of quantity point vectors. Craig's book [47] provides a thorough background in transformation terminology in Euclidean space. The general transformation of a point quantity vector \(P(P_1, P_2, P_3)\) consists of translation transformations \(P_T\) and orientation transformations \(P_R\). These transformations can be defined by either numerical vectors or quantity vectors. There is no propagation when transforming a quantity vector to another numerical location by applying real numbers to the transformation matrices because it is calculated in Euclidean space. A translation matrix of quantity vectors and an orientation matrix of either real numbers or quantity vectors are considered in FQ calculation in this section. Fuzzy quantity vector propagation in Section III generates complete possible states or trajectory states using the proposed quantity arithmetic with inputs described in terms of numerical, intervals or symbols. However, in order to describe the propagated states by a qualitative description, aggregation operators are introduced to select both suitable states and symbolic functions. Aggregation operators model operations such as conjunction, disjunction and averaging on intervals and fuzzy sets [48].

One of aggregation operator families is ordered weighted averaging operators \(\text{OWA}\) and their variants [49–52], originally studied by Yager [53]. Their general mathematical description is given below
\[
\text{OWA}(P_1, P_2, \cdots, P_n) = \sum_{i=1}^{n} w_i P_{\sigma(i)},
\]

(24)

where \(\sigma\) is an ordinal sequence, \(w_i \geq 0\) and \(\Sigma w_i = 1\). The \(\text{OWA}\) operators provide a parameterised family of aggregation operators which can be used for many of the well-known operators by choosing suitable weights. It includes the average \(\frac{1}{n} \sum_{i=1}^{n} P_i\), the quadratic mean \(\sqrt{\frac{1}{n} \sum_{i=1}^{n} P_i^2}\), the harmonic mean \(\frac{n}{\sum_{i=1}^{n} \frac{1}{P_i}}\), the geometric mean \(\left(\prod_{i=1}^{n} P_i\right)^{\frac{1}{n}}\), the maximum operator and minimum operator. The \(\text{OWA}\) weights can be selected based on domain knowledge or application contexts where the aim is to adjust the \(\text{OWA}\) operator to generate suitable symbolic or qualitative functions. In addition to the method proposed in this section, there are a wide variety of other information aggregation and fusion techniques. For example, [54] could be adapted for quantity spaces and could be used to generate symbolic functions as possible to describe requested qualitative as accurately as possible. Note that It has shown that normalized quantity spaces can be used to produce grounded symbols with the aid of domain knowledge.

Quantity vector aggregation can be manipulated by the \(\text{OWA}\) and their variants when the dimension of the quantity vector is low or its online computation has required low priority. The issue of computational cost has been raised for quantity vector aggregation in higher dimensions, especially for real-time robotic systems. A \(k – \text{AGOP}\) aggregation operator is proposed to estimate the aggregation of quantity vectors of a higher dimension using a smaller \(k\)-dimensional subset. The \(\text{OWA}\) aggregation operators manipulate a single dataset or a data vector whereas the \(k – \text{AGOP}\) aggrega-
tion operator has the ability of handling multiple datasets using the proposed quantity arithmetic. Note that apart from domains where quantity space is used, calculations between datasets can be easily replaced by standard arithmetic. A \( k - \text{AGOP} \) aggregation operator of dimension \( n \) is a two-step mapping as follows,

\[
\Gamma : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^k \rightarrow \mathbb{R}
\]

Function \( \Gamma \) has an associated weighting vector \( W \) of dimension \( n \) whose elements have the property:

\[
\sum_{i=1}^{k} \sum_{j=1}^{k_1} \frac{w_{kj}}{k} = 1, \text{ for } w_{kj} \in [0, 1]
\]

The mapping is given by

\[
\Gamma_k(P_m, P_n) = W_k^T (P_m^i OP_n^j), \text{ for } i, j \in \{k_s, k_l\}
\]

where \( P_m^i OP_n^j \) is a \( k \)-dimensional column vector in ascending order and \( O \) is the quantity arithmetic operator. If the dimensions of the two vectors \( P_m \) and \( P_n \) are \( m \) and \( n \) respectively then \( k_s = \min(m, n) \) and \( k_l = \max(m, n) \). \( k_s \) and \( k_l \) can be obtained from the equation 26, where \( \text{ceil}(X) \) rounds the element of \( X \) to the nearest integer greater than or equal to \( X \).

\[
k_s = \text{ceil} \left( \frac{\min(\text{length}(P_m), \text{length}(P_n))}{\max(\text{length}(P_m), \text{length}(P_n))} \right)
\]

\[
k_l = k - k_s
\]

The \( k - \text{AGOP} \) aggregation operator consists of the calculation of the quantity vector and the aggregation operation. First, \( \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^k \) selects two sets of intervals from respective quantity vectors and uses them to calculate a \( k \)-dimensional quantity vector using the proposed quantity vector arithmetic. Then \( \mathbb{R}^k \rightarrow \mathbb{R} \) aggregates the \( k \)-dimensional quantity vector into a unique numerical value using the \( \text{OWA} \) aggregation operators. In order to estimate the quantity vector boundary in the first-step mapping, the first and last elements of the \( k_s \) - or \( k_l \)-dimension quantity vectors are selected as the minimum and maximum elements of the \( m \)- or \( n \)-dimension quantity vectors. The remaining \( k_s - 2 \) and \( k_l - 2 \) elements can be chosen using a probability distribution or by other criteria.

In this paper, it is assumed that the data in a dataset is uniformly distributed across its dimensions, which results in:

\[
P_m \rightarrow P_{km} \left( p_{1 \min}, \ldots, p_{1}, \ldots, p_{km \max} \right)
\]

\[
P_n \rightarrow P_{kn} \left( p_{1 \min}, \ldots, p_{s}, \ldots, p_{kn \max} \right)
\]

where \( i = j \times \text{round} \left( \frac{m}{k_m - 1} \right) \), \( s = t \times \text{round} \left( \frac{k_m}{k_m - 1} \right) \) for \( k_m \in \{k_s, k_l\} \) and \( j = [2, \ldots, k_m - 1] \), \( t = [2, \ldots, k_m - 1] \). \( P_{km}(p_{1 \min}) \) is the minimum of quantity vector \( P_{kn} \) and \( P_{kn}(p_{km \min}) \) is the maximum, it is similarly applied for \( P_{km} \). Hence the first-stage mapping \( \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^k \) can be decomposed into \( \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^{km} \times \mathbb{R}^{km} \rightarrow \mathbb{R}^k \). It is obvious that the proposed \( k - \text{AGOP} \) aggregation operator satisfies the general requirements of the \( \text{OWA} \) operator such as monotonicity, commutativity and bounded conditions which are described in [52].

In order to implement the proposed aggregation operator to robotic motion behaviour description for intelligent robotics, control parameters \( \lambda_i \) have been presented for \( P_i \in \mathbb{R}^k \), where \( i \in (x, y, z) \).

\[
\begin{align*}
\begin{bmatrix}
x \\ y \\ z
\end{bmatrix} &= \begin{bmatrix}
\lambda_2 \text{Max}(P_x) + (1 - \lambda_2) \text{Min}(P_x) \\
\lambda_2 \text{Max}(P_y) + (1 - \lambda_2) \text{Min}(P_y) \\
\lambda_2 \text{Max}(P_z) + (1 - \lambda_2) \text{Min}(P_z)
\end{bmatrix}
\end{align*}
\]

(27)

The control parameters represent the degree of the aggregated values related to the two boundary values, i.e., \( \text{Max}(P_i) \) and \( \text{Min}(P_i) \), \( i \in (x, y, z) \). Knowledge-based methods and learning algorithms can be employed here to produce the control parameters, e.g., generate motion behaviour subject to observed environmental information.

VI. CASE STUDY

In order to demonstrate the ability of FQ robotic kinematics handling the conversion among numeric, symbolic and hybrid data in the context of intelligent robotics, we have implemented the proposed theory into a PUMA 560 robot whose parameters shown in Table II. It is common in behaviour-related research to assume that robots have rigid bodies and to consider the joint angles as variables only. Hence we denote the joint angles as \( \theta \) variables and the other parameters for the PUMA 560 as numeric, i.e., \( a_2=432\text{mm}, a_3=20\text{mm}, d_3=124\text{mm} \) and \( d_4=432\text{mm} \). The orientation parameter \( p \) is set as 60, the translation parameter \( q \) as 40 and the initial configuration is given as,

\[
\begin{bmatrix}
QS_{a}(\theta_1) \\
QS_{a}(\theta_2) \\
QS_{a}(\theta_3)
\end{bmatrix} = \begin{bmatrix}
0.007, 0.015 \\
0.391, 0.017 \\
0.024, 0.016
\end{bmatrix}
\]

(28)

A joint trajectory of a PUMA 560 robot is provided in Table I, note that the joint trajectory is described in terms of qualititative states, which can be connected to symbols through domain knowledge.

After implementing equation 22 using the parameters in Table II, the centre of the PUMA tool frame is obtained as

\[
\begin{align*}
\mathbb{Q}_{S_d}(p_{tcp}) &= \begin{bmatrix}
C_{\theta_1} [a_2C_{\theta_2} + a_3C_{\theta_2+3} - d_3S_{\theta_2+3}] - d_3S_{\theta_1} \\
S_{\theta_1} [a_2C_{\theta_2} + a_3C_{\theta_2+3} - d_3S_{\theta_2+3}] + d_3C_{\theta_1} \\
- a_2S_{\theta_2+3} - a_3S_{\theta_2} - d_4C_{\theta_2+3}
\end{bmatrix},
\end{align*}
\]

(29)

where \( \theta_{2+3} \) stands for \( QS_{a}(\theta_2) + QS_{a}(\theta_3) \).

The simulation results of \( Q_{S_a} \) positions are given in Figs 5, 6 and 7. The positions of the initial robotic configuration in
equation 28 are demonstrated in Fig. 5 in interval terms. It shows that Joint position $J(QS(1), QS(24), QS(2))$ leads to a qualitative state for the end-effector’s position, which is then represented by a numeric value aggregated by the proposed aggregation operator in equation 27. Further, the corresponding end-effector’s positions of the trajectory of qualitative states in Table I are given in Fig. 6 using the MR representation. It demonstrates that each qualitative position is composed of the centres of overlapped FQ intervals which are labelled using circles. It also shows that the qualitative positions do not have the same interval regions, which indicates that each qualitative position has different position behaviours. In order to extract a suitable behaviour from the qualitative trajectory, we employ max, min and mean aggregation operators and equation 27. Corresponding to Fig. 6, there are six positions of the end-effector illustrated in Fig. 7, each of which is represented in both terms of conventional robot kinematics and the proposed method. For instance, qualitative state $QS_0$, its qualitative state’s profile is qualitatively described by symbols ‘$\times$’, ‘$\triangle$’ and ‘$\triangledown$’, which are positions based on combinatorial $x,y,z$ coordinates in equation 27 with $\lambda_i \in \{0, 1\}$. Positions labelled by ‘$\triangledown$’ (where $\lambda_i=1$) and ‘$\triangle$’ (where $\lambda_i=0$) are used to label the maximum and minimum positions of the qualitative state, and the qualitative state’s mean position labelled by ‘□’. Positions with symbols ‘$\circ$’ are those generated by conventional robot kinematics with defaulted $\alpha$-centre values of input joint trajectory in Table I. It shows that positions ‘$\circ$’ are within the profile of the qualitative states but not coincident with the mean values; a set of aggregated values labelled as ‘$\times$’ calculated by equation 27 is provided with $\lambda_x = 0.2$, $\lambda_y = 0.5$ and $\lambda_z = 0.8$. The control parameters can be either assigned or learned by contextual information, e.g., recognized obstacle objects. Aggregated values can also be used for representing their corresponding qualitative states. This demonstrates the connection of numeric values and qualitative states through contextual information, the former can be employed to a low-level control system; the latter can be used to construct a symbol-based reasoning system. Additionally, each group of positions are lined up to show the trajectory sequence, trajectory I stands for the aggregated trajectory produced by the proposed method, trajectory II denotes the trajectory by conventional robot kinematics and trajectory III stands for the trajectory by the mean values of the qualitative states. Suitable interpolation techniques can be applied to generate smooth trajectories for non-symbolic modules such as motion control modules.

Further we obtain the FQ velocity formula below for the centre of the PUMA tool frame using equation 23:

$$QS_d(\dot{tcp}) = J_{3 \times 3} \cdot QS_d(\dot{\theta})$$

(30)

where $QS_d(\dot{\theta})$ is FQ vector which describes the angular velocity of the PUMA and is given by

$$QS_d(\dot{\theta}) = [QS_a(\dot{\theta}_1) \; QS_a(\dot{\theta}_2) \; QS_a(\dot{\theta}_3)]^T$$

where $J_{3 \times 3}$ is its FQ Jacobian matrix which is defined as

$$J_{3 \times 3} = [J_1 \; J_2 \; J_3]$$

(31)
to extract the behaviours of serial robots, an example of a
Ordered weighted averaging operators have been adapted
developed a FQ version of robot kinematics in FQ terms.
XTRIG MATLAB toolbox. Based on the extension, we have
derivative to FQ trigonometry and implemented it into the
trajectory with
The simulation result of the FQ velocity of the PUMA robot’s
modules for a better solution to intelligent connection problem.
Trajectory II is based on the mean values of the qualitative states, which
provides a reference for the comparison. A trajectory point labelled by ‘○’ illustrates that the velocity calculated by
conventional robot kinematics is within its qualitative profile.

The proposed method can be easier adapted to robotic sys-
tems for a better solution to intelligent connection problem. For instance, Chan et al [55] has adapted this method to
model human motion in image sequences as stick model in order to analyse their motion behaviour. In addition,
the proposed method potentially could be used to construct behaviour vocabularies for the development of skill-level
interfaces for autonomous humanoid robots in [6].

VII. CONCLUDING REMARKS AND FUTURE DIRECTIONS

We have proposed an extension named FQ trigonometry
derivative to FQ trigonometry and implemented it into the
XTRIG MATLAB toolbox. Based on the extension, we have
developed a FQ version of robot kinematics in FQ terms.
Ordered weighted averaging operators have been adapted to extract the behaviours of serial robots, an example of a
PUMA robot has been presented to support the proposed
methods.

The advantage of the proposed FQ robot kinematics is
twofold. It can be used as a tool to measure robotic cal-
ibration issues, e.g., accuracy, repeatability and calibration
[56]. More importantly, it provides an opportunity to connect
numerical data and interval symbols together, these elements
can work as atomic behaviour functions that could be used
to build up behaviour vocabularies of a robot and its motion
control. It paves the way for providing a feasible approach
to intelligent connection problem. The connection problem
is crucial for research in the field of AI and robotics because
intelligent robots commonly comprises of non-symbolic sub-
systems and symbolic subsystems. The state of art in intel-
ligent robotics including technical needs by a wider range
of robots requires their symbolic or behaviour description.
In practice, almost all existing robotic systems consisting of
at least a motion control module and a reasoning/symbol-
based module, hardcode methods are the dominant methods
to connect these two types of modules; it is one of key
barriers to intelligent robotics because symbols could have
different meanings in various contexts. The proposed method
provides a fundamental way to replace the hardcode methods
because of its relation to conventional robot kinematics and
fuzzy qualitative reasoning. In addition, our future work
targets two research lines: one is the computational cost
issue, which deals with how efficiently the proposed interval
computation meets the requirements of robotic systems, or
systems modelled as robots (e.g., stick models for human
motion). The other is to explore new methods to improve how
effective aggregation operators or new learning operators
would be in their behaviour description. The priority of
the latter would be given to optimally tune the propagated
fuzzy qualitative parameters and its effect on robot behaviour
description.
### APPENDIX A: ARITHMETIC OPERATIONS WITH FOUR-TUPLE FUZZY NUMBERS

**References**


### Table: Arithmetic Operations with Four-Tuple Fuzzy Numbers

<table>
<thead>
<tr>
<th>Operation</th>
<th>Result Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$[\alpha_1, -\alpha_1, 0, 0]$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$[\beta_1, 0, -\beta_1, 0]$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>$[\gamma_1, 0, 0, -\gamma_1]$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>$[\delta_1, 0, 0, -\delta_1]$</td>
</tr>
</tbody>
</table>

**Conditions**

- $m < n$
- $n < m$
- $m = n$
- $m > n$
- $n > m$

**Arithmetic Operations**

- $\alpha + \beta = \alpha_1 - \alpha_1 + \beta_1 - \beta_1$
- $\alpha - \beta = \alpha_1 - \alpha_1 - \beta_1 + \beta_1$
- $\alpha / \beta = \alpha_1 / \beta_1$
- $\alpha \times \beta = \alpha_1 \times \beta_1$
- $\alpha^\beta = \alpha_1 \times \beta_1^\beta_1$
- $\alpha^{-\beta} = \alpha_1 \times \beta_1^{-\beta_1}$

**For Positive Real Numbers**

- $\alpha^\beta = \alpha_1 \times \beta_1^\beta_1$
- $\alpha^{-\beta} = \alpha_1 \times \beta_1^{-\beta_1}$

**For Negative Real Numbers**

- $\alpha^\beta = \alpha_1 \times \beta_1^\beta_1$
- $\alpha^{-\beta} = \alpha_1 \times \beta_1^{-\beta_1}$

**For Complex Numbers**

- $\alpha^\beta = \alpha_1 \times \beta_1^\beta_1$
- $\alpha^{-\beta} = \alpha_1 \times \beta_1^{-\beta_1}$

**For Fuzzy Numbers**

- $\alpha^\beta = \alpha_1 \times \beta_1^\beta_1$
- $\alpha^{-\beta} = \alpha_1 \times \beta_1^{-\beta_1}$

**Output Result Conditions**

- $m < n$
- $n < m$
- $m = n$
- $m > n$
- $n > m$
C.S. Chan, H. Liu, and D.J. Brown. Human motion classification using
qualitative normalised templates. *Journal of Intelligent and Robotic

T. Calvo, R. Mesiar, and R.R. Yager. Quantitative weights and

**Honghai Liu** (M’02-SM’06) received his PhD in Robotics from King’s College London, UK, in 2003. He joined the University of Portsmouth, UK in September 2005. He previously held research appointments at Universities of London and Ab-
erdeen, U.K.

He has published over 80 research papers in-
cluding two Best Paper Awards. His research
interests are computational intelligence methods
and applications with a focus on those approaches
which could make contributions to the intelligent
connection of perception to action.

**David J. Brown** received his PhD in motion con-
trol from Southampton University, Southampton,
UK, in 1985. He came back to academia after
spending about 20 years in industry.

He has published numerous refereed journal and
conference papers in artificial intelligence area.
He won engineering excellence prize from Royal
Academy of Engineering in 2004. His research
interests are in artificial intelligence and their ap-
plications to intelligent systems.

**George M. Coghill** received his Ph.D. in fuzzy
qualitative reasoning from Heriot-Watt University
UK, in 1996 and is currently a Senior Lecturer in
Computing Science at the University of Aberdeen.

He has published numerous journal and confer-
ence papers in fuzzy qualitative reasoning. His re-
search interests focus on fuzzy model-based tech-
nology and its application to simulation, diagnosis,
system identification and systems biology.