A Quantum Fuzzy Inference Engine for Particle Accelerators Control

Roberto Schiattarella
Dept. of Physics "Ettore Pancini"
University of Naples Federico II
Naples, Italy
roberto.schiattarella@unina.it

Giovanni Acampora
Dept. of Physics "Ettore Pancini"
University of Naples Federico II
Naples, Italy
giovanni.acampora@unina.it

Michele Grossi
European Organisation for Nuclear Research
Espl. des Particules 1, 1211 Meyrin, Switzerland
michele.grossi@cern.ch

Micheal Schenk
European Organisation for Nuclear Research
Espl. des Particules 1, 1211 Meyrin, Switzerland
micheal.schenk@cern.ch

***Abstract*—Recently a Quantum Fuzzy Inference Engine (QFIE) that achieves an exponential advantage in computing fuzzy rules with respect to the classical counterpart has been proposed. The main goal of such a quantum engine is to pave the way for the implementation of fuzzy rule-based systems in complex environments where the number of fuzzy rules to compute is impractical for classical fuzzy systems. However, also due to the technological limitation of current Noisy Intermediate Scale Quantum (NISQ) devices, until now the quantum engine has been tested only in noiseless simulations and simplified environments. This work aims to overcome both of these limitations by using QFIE for the very first time to control real systems such as those related to particle accelerator facilities at the European Organization for Nuclear Research (CERN), as well as executing QFIE on real quantum hardware. As shown by a set of experiments carried out on the T4 target station at the CERN SPS fixed target physics beam line and on the Advanced Proton Driven Plasma Wakefield Acceleration Experiment (AWAKE), QFIE is able to control this kind of environment efficiently.**

***Index Terms*—Quantum computing, fuzzy control systems, Particle Accelerators.**

I. Introduction

Fuzzy sets and logic theory introduced by Lofti Zadeh [1], [2] has the capability of model classes of objects that do not have precisely defined criteria of membership, in a way to mimic human thinking on computers. Starting from Zadeh's theory Fuzzy Rule-Based Systems (FRBSs) have been developed and they have found a widespread set of applications in the field of automatic control and decision making [3], [4]. The reason of this success can be explained by the fact that expert knowledge is easily introduced in these systems by means of *fuzzy rules*. Despite their success, FRBSs suffer from the fuzzy rule explosion problem: the number of rules in a FRBS grows exponentially with the number of variables that makes up the system. This issue strongly limits the possibility of controlling environments characterized by an high number of variables by means of such systems. Very recently, to potentially overcome this important limitation of FRBSs, the new quantum computing paradigm has been exploited to implement a new generation of fuzzy inference engines. The first quantum fuzzy inference engine (QFIE) has been developed in [5]: thanks to the massive parallelism provided by quantum phenomena such as superposition and entanglement, QFIE is able to reach an exponential advantage in computing fuzzy rules with respect to the classical counterpart. However, in the aforementioned research, the authors focused on the theoretical aspects of this promising algorithm. On the other hand, also due to the current era of quantum computers characterized by a high level of noise and a small number of qubits available for the computation [6], until now QFIE has been tested by means of simulations just for the control of very easy environments such as the one related to an inverse pendulum system.

This work aims to validate experimentally the potentiality of QFIE, by using it for the control of real facilities such as those related to the particle accelerators at the European Organization for Nuclear Research (CERN). Moreover, some of the experimental tests carried out in this research aim to use QFIE for the very first time by means of a real quantum device from the IBM Q Family.

While the results achieved by this research experimentally prove the capabilities of this new quantum fuzzy inference engine, they also demonstrate how FRBSs can help particle physics experiments carried out at CERN. Indeed, CERN maintains a dense and diverse physics program with numerous experiments requiring more and more challenging control capability of particle accelerators. Whereas for many beam control problems, physics models are used at the CERN accelerators, various systems are still tuned manually due to the lack of models or beam instrumentation. Recently, sample-efficient control algorithms, such as reinforcement learning (RL), have been introduced for some of these cases [7], [8]. Sample efficiency is essential for any optimization algorithm in the context of accelerator operation to minimize the impact on beam time available for the physics experiments. This kind of sample efficiency is shown experimentally in this research also for QFIE. Therefore, this work represents outstanding proof

of the FRBSs approach as an alternative to the current RL scenario.

In detail, the experimentation carried out in this work focuses on two control problems related to two CERN facilities: the T4 target station at the CERN Super Proton Synchrotron (SPS) fixed target physics beam line and the much more complex Advanced Proton Driven Plasma Wakefield Acceleration Experiment (AWAKE).

In the former case study, the control problem was 1-dimensional, and the size of the quantum circuits implementing QFIE has enabled its execution on the *IBMQ Montreal* hardware. On the other hand, the AWAKE facility consists of a complex 10-dimensional control problem, and current NISQ devices are not ready to handle the resulting QFIE circuits. However, in this case, the simulated quantum circuits were tested on real data as an on-line controller of the beam line.

The remaining of the paper is structured as follows: in section II an analysis of the literature about the integration of fuzzy logic and quantum computing is carried out; section III summarizes the basic concepts related to the quantum computing and the main aspects of QFIE; section IV and section IV-B report the experimentation performed for the simulated T4 target station and AWAKE environments respectively; section IV-C shows the experimental test carried out on the real AWAKE facility. Finally, section V concludes the paper.

II. Related Works

Quantum Computing is becoming more and more a computational paradigm useful to develop a new generation of powerful and efficient computational intelligence algorithms: for instance, huge efforts have been done in the context of quantum neural networks [9], [10], as well as in quantum evolutionary computing [11], [12]. In both areas, these works show a potential advantage in using this new quantum computational intelligence framework over the classical counterpart. On the other hand, quantum fuzzy logic is a very recent and unexplored topic. Recently, several approaches exploiting the affinity of quantum mechanics and fuzzy sets theory have been presented, trying to interact to each other's advantage [13]–[16]. More generally, two lines of research have emerged in recent years: on the one hand, fuzzy sets theory has been used for simulating and modeling physical quantum systems [17], [18]; on the other hand, quantum computing approaches have been investigated to improve the performance of conventional fuzzy systems. Our interests focus on this second area, which is hereafter deeply analyzed. One of the first application of quantum computing to improve the computational performance of fuzzy systems is presented by Rigatos et al. in [19]. In this work, some operations performed in specific fuzzy inference engines are replaced by quantum operations. Although the theoretical results presented in [19] were very promising, at the time when that research was carried out no suitable quantum devices or simulators were available to allow experimental validation on the field. Successively, different researches focused on the quantum implementation of fuzzy logic operators. Some of them focused on the usage of a quantum circuital paradigm for implementing these operators [20], [21], while others exploited different quantum computational paradigms such as quantum annealing [22], [23]. Although all of the above research efforts show how quantum operations can be useful for performing basic logic operations between fuzzy sets, none of them implements an efficient quantum algorithm for executing fuzzy inference engines on quantum computers. An embryonic work in this direction is presented in [24], where Grover's algorithm was used to implement a fuzzy system based on a lookup table. Although this paper represents the first attempt to use a well-known quantum algorithm to implement fuzzy systems, its use is very limited since the input-output relationships present in the lookup must be generated using classical computation. This limitation does not allow obtaining a computational advantage from the quantum implementation of FRBSs. Finally, such a computational advantage was proven very recently in [5]. A deep description of the algorithm proposed in this paper is carried out in section III-B. However, the main goal of the aforementioned work was the theoretical proposal of an innovative QFIE, which has been tested in very easy case studies and just by noiseless simulations of the quantum circuits implementing the algorithm. According to this analysis of the state of the art, this work represents the very first attempt to use the quantum fuzzy inference engine presented in [5] for the control of real and complex environments, such as those related to particle accelerators at the CERN facilities. Moreover, the proposed work shows also the first real execution of the quantum fuzzy inference engines on real quantum devices from the IBM Q family.

III. Basic Concepts

This section aims to introduce the basic concepts of quantum computing, highlighting also the hardware limitations of the current quantum devices. Successively, a brief description of the QFIE proposed in [5] is carried out. Readers familiar with the basic concepts of quantum computing can therefore skip to section III-B.

A. Quantum Computing

Quantum computing manipulates and stores information by using qubits, the basic information unit of this innovative computational paradigm. The main peculiarity of qubits is that, unlike the classical bits, they can live in a superposition of states. Formally, a qubit is described by a vector of an Hilbert space $\mathcal{H}$: according to the bra-ket notation, such a vector is expressed as follows:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

where the set $\{|0\rangle, |1\rangle\}$ forms the basis state of $\mathcal{H}$, while α and β are complex coefficients. When a quantum state is measured it collapses in one of its basis states with a probability equal to the squared modulus of the related coefficient. The result of the measurement operation is therefore a classical bit 0 or 1 obtained with a probability $|\alpha|^2$ or $|\beta|^2$ respectively. One of the most powerful peculiarities of quantum computers

is the fact that when more qubits are used simultaneously, the size of the Hilbert space in which the quantum state lives increases exponentially with the number of qubits. This means that by considering n qubits, the quantum state describing the system is expressed as follows:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \tag{2}$$

where i denotes the integer representation of the n-dimensional bit string, and the coefficients α_i have the same meaning as the one-dimensional case described above.

Quantum states can be manipulated by quantum gates as well as classical bits can be manipulated by means of logical gates [25]. By definition quantum gates are represented by unitary matrices that act on a state as follows :

$$U|\psi\rangle = U \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \tag{3}$$

A more in deep description of these gates is out of the scope of this paper but can be found in [25]. Overall, a quantum algorithm is a collection of quantum gates acting on several qubits. In the end, the resulting quantum circuit is measured and usually, this procedure is repeated several times to esteem the distribution of probability encoded in the output quantum state.

For the past few years, various quantum hardware has been made available via the cloud by several major industries. However, these primordial quantum devices are denoted as Noisy Intermediate Scale Quantum (NISQ) computers [6]. The label refers to the limited amount of qubits of these devices and also to the high level of noise that occurs during the computation of quantum circuits. Recently, a lot of improvements have been done in the development of quantum hardware: for instance, IBM has just presented the first superconductive chip composed of more than 400 qubits[1]. However, this number of qubits is still too low for implementing directly on hardware error correction codes. On the other hand, attempts to simplify quantum circuits to be executed on actual NISQ devices have been presented. Crucial for this work is the D-NISQ reference model introduced in [26]. This architecture offers a reference model for distributing quantum computation in smaller quantum circuits that can be then executed on different quantum processors. The integration of the different outputs is in the end classically performed by means of the *information fusion* layer, which returns the output of the original problem to solve. As shown in section IV-B such a model is used in this work for enabling the development of a quantum fuzzy control system able to control a 10-dimensional environment such as the one of the AWAKE experiment at the CERN.

B. A Quantum Fuzzy Inference Engine

A fuzzy rule-based system (FRBS) is a rule-based system where fuzzy sets and fuzzy logic are used for modeling the relationships existing between its variables [27]. The general workflow of a FRBS is shown in Fig. 1. The knowledge base of the system is composed of a database and a rule-base. The former stores knowledge about the problem at hand in terms of fuzzy variables: a fuzzy variable X is composed of a set of m linguistic terms $X = \{T_1, T_2, \ldots, T_m\}$, where each linguistic term T_i (with $i = 1, 2, \ldots, m$) is described by a fuzzy set. On the other hand, a fuzzy rule-base is composed of a set of fuzzy if-then rules of the following form: IF $(X_1$ is $T_{s_1}) \bullet (X_2$ is $T_{s_2}) \bullet \ldots \bullet (X_n$ is $T_{s_n})$ THEN Y is T_{s_Y}, where T_{s_i} is the fuzzy set related to the s-th linguistic term defined in the universe of discourse of X_i and $\bullet$ is a general logic connector, such as and and or. These are implemented by means of *t-norm* and *s-norm* operators, respectively [2]. In general, the fuzzy proposition before the THEN keyword is denoted as *antecedent* preposition, while the following part is denoted as *consequent* preposition of the rule. Considering this, the action of a MISO (Multiple-Input Single-Output) FRBS composed of m input variables, can be summarized in the following steps:

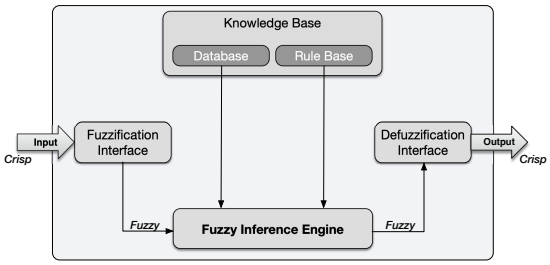


Fig. 1. Architecture of a fuzzy rule-based system

1) The m input crisp data are fuzzified by means of the fuzzification interface: for each input X_i the fuzzification interface computes the membership degrees $\alpha_{X_i}^{T_j}$ for all the T_j fuzzy sets defined in X_i.
2) The fuzzy inference engine (FIE) uses database and rule base to perform a nonlinear mapping from input and output fuzzy variables, through four sequential steps [28]:
    1) the evaluation of each fuzzy proposition belonging to the antecedent part of each rule by considering the system inputs;
    2) the computation of the degree of fulfillment of each rule obtained by aggregating the fuzzy propositions of the antecedent part;
    3) the computation of the fuzzy output of each rule obtained by applying the implication operator;
    4) the computation of the overall fuzzy output obtained by accumulating the fuzzy outputs of individual fuzzy rules.
3) The defuzzification interface converts the output fuzzy variable to a crisp value.

The QFIE proposed in [5] replaces the classical FIE of the above architecture. It is based on a formulation of the fuzzy rule set as a Boolean oracle and on an encoding procedure of

[1]https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two

fuzzified values in quantum states. These features are hereafter described.

1) Oracle-based FRBS: For sake of simplicity, let us consider the design of the oracle-based FRBS of a MISO fuzzy system $\mathcal{S}$ with n input fuzzy variables $X_0, X_1, \ldots, X_{n-1}$ and one output fuzzy variable Y. Each input variable X_j is defined by using m_j linguistic terms: $X_j = \{T_0^j, T_1^j, \ldots, T_{m_j-1}^j\}$, with $j = 0, 1, \ldots, n-1$. The output variable Y is defined by using m_Y linguistic terms: $Y = \{T_1^Y, T_2^Y, \ldots, T_{m_Y}^Y\}$. The oracle-based design of an inference fuzzy system requires that both input and output linguistic terms are encoded by binary strings. Then, for input variables, let $\{B_I^j\}_{j=0}^{n-1}$ be a family of sets, whose j-th component is defined as follows: $B_I^j = \{b_i^j | i = 0, 1, \ldots, m_j - 1\}$, where b_i^j is the binary encoding of the number i, with $i = 0, 1, \ldots, m_j - 1$, computed by using the conventional decimal to binary conversion, and such that $\bar{\eta}_j = |b_i^j| = \lceil \log_2(m_j) \rceil$ is the number of bits used to encode above strings. In this way, the j-th element of $\{B_I^j\}_{j=0}^{n-1}$ contains the binary label b_i^j of the linguistic terms T_i^j related to the i-th input variable, where $i = 0, 1, \ldots, n-1$. For the output variable, let B_O be a set defined as follows: $B_O = \{c_i | i = 1, 2, \ldots, m_Y\}$, where c_i is the binary encoding of the number i, with $i = 1, 2, \ldots, m_Y$, computed by using a one-hot encoding: c_i is a m_Y-bit binary string consisting of all 0's except for a 1 in the i-th position from left, and such that $|c_i| = m_Y$ is the number of bits used to encode above strings. Thus, B_O contains the binary representation c_i of the linguistic term T_i^Y related the output variable, where $i = 1, 2, \ldots, m_Y$. Now, let $\mathcal{A}_\mathcal{S}$ and $\mathcal{C}_\mathcal{S}$ be two new sets defined as follows:

$$\mathcal{A}_\mathcal{S} = \prod_{j=0}^{n} B_I^j; \quad \mathcal{C}_\mathcal{S} = B_O \cup \{c_0\} \tag{4}$$

where $\mathcal{A}_\mathcal{S}$ is a Cartesian product that contains the binary encoding of all possible antecedents that can be defined with the input variables $\{X_j\}_{j=0}^{n-1}$; $\mathcal{C}_\mathcal{S}$ contains the binary encoding of all possible consequent parts that can be defined with the output variable Y; and $c_0 = \{0\}^{m_Y}$ is a m_Y-bit string containing all 0's. Thus, the function $f : \mathcal{A}_\mathcal{S} \to \mathcal{C}_\mathcal{S}$ is a Boolean oracle that maps the antecedent parts to the consequent parts to compose fuzzy rules. For a given $a \in \mathcal{A}_\mathcal{S}$, f returns a binary string other than $\{0\}^{m_Y}$ if a is a suitable antecedent part present for the system $\mathcal{S}$, $\{0\}^{m_Y}$ otherwise.

2) Encoding Procedure:: QFIE requires that the data on which a FIE operates, i.e., fuzzified values, be modeled through quantum states. This goal is achieved by using an amplitude encoding procedure.

For sake of simplicity, let us suppose that for each input variable the terms $\{T_0^j, T_1^j, \ldots, T_{m_j-1}^j\}$ are a fuzzy partition, i.e. given a crisp value x_j of X_j then $\sum_{i=0}^{m_j-1} \alpha_i^j = 1 \quad \forall j \in [0, n-1]$. Under this condition, a quantum register composed of $\eta_j = \lceil \log_2(|X_j|) \rceil$ qubits for each input variable can be initialized as:

$$|\psi_j\rangle = \sum_{i=0}^{m_j-1} \sqrt{\alpha_i^j} \left| b_i^j \right\rangle + \sum_{i=m_j}^{2^{\eta_j}-1} 0 \cdot \left| b_i^j \right\rangle \tag{5}$$

where the set $\{b_i^j\}_{j=m_j}^{2^{\eta_j}-1}$ is composed of the Boolean strings in excess of B_I^j. If X_j is not a fuzzy partition, a dummy fuzzy set can be used for ensuring the normalization in Eq. (5). In order to define a quantum state that takes into account all n fuzzy variables involved, the tensor product of the individual states $\{|\psi_j\rangle\}_{j=0}^{n-1}$ is computed:

$$\begin{aligned} &|\psi_0, \psi_1, \ldots \psi_{n-1}\rangle = \\ &= \sum_{j=0}^{m_0-1} \cdots \sum_{l=0}^{m_{n-1}-1} \sqrt{\alpha_j^0 \cdot \ldots \cdot \alpha_l^{n-1}} \left| b_j^0, \ldots, b_l^{n-1} \right\rangle = \\ &= \sum_{a \in \mathcal{A}_s} \sqrt{F_a} \left| a \right\rangle \end{aligned} \tag{6}$$

where F_a is the fire strength of the antecedent a computed with the product T-norm [29]. This quantum state has an important property: its basis states correspond to all possible antecedent parts of fuzzy rules that can be generated with the variables $X_0, X_1, \ldots, X_{n-1}$. Consequently, the squared amplitude of each basis state represents the degree of fulfillment of the related antecedent part. As for the output fuzzy variable Y, it is embodied in a quantum state $|\psi_Y\rangle$ composed of m_Y qubits, each of which is initialized to $|0\rangle$: $|\psi_Y\rangle = |0_0, 0_1, \ldots, 0_{m_Y-1}\rangle = |\bar{0}\rangle$. Let $\mathcal{O}_f$ be the unitary gate implementing the oracle $f : \mathcal{A}_\mathcal{S} \to \mathcal{C}_\mathcal{S}$. Then, $\mathcal{O}_f$ acts as follows:

$$\begin{aligned} &|\psi'_Y\rangle = \mathcal{O}_f \left| \psi_0, \psi_1, \ldots, \psi_{n-1} \right\rangle |\bar{0}\rangle = \\ &= \sum_{i=1}^{m_Y} \sum_{a \in \mathcal{A}_\mathcal{S}^i} \left(\sqrt{F_a} \left| a \right\rangle \left| c_i \right\rangle \right) + \sum_{a \in \mathcal{A}_\mathcal{S}^0} \left(\sqrt{F_a} \left| a \right\rangle |\bar{0}\rangle \right) \end{aligned} \tag{7}$$

The state returned by the oracle application is then used by the inference engine to compute the values P_{c_i}, corresponding to the probability of measuring the output quantum state as $|c_i\rangle$, to cut the membership function related to the output linguistic term T_i^Y, with $i = 1, 2, ..., m_Y$:

$$\begin{aligned} &P_{c_i} = \langle \psi'_Y | \, \mathcal{M}_{c_i}^\dagger \mathcal{M}_{c_i} \left| \psi'_Y \right\rangle = \\ &= \sum_{a \in \mathcal{A}_\mathcal{S}^i} \sum_{b \in \mathcal{A}_\mathcal{S}^i} \left(\sqrt{F_a} \left\langle a \right| \left\langle c_i \right| \right) \cdot \left(\sqrt{F_b} \left| b \right\rangle \left| c_i \right\rangle \right) = \\ &= \sum_{a \in \mathcal{A}_\mathcal{S}^i} \sum_{b \in \mathcal{A}_\mathcal{S}^i} \delta_{a,b} \sqrt{F_a} \sqrt{F_b} \left\langle a | b \right\rangle = \sum_{a \in \mathcal{A}_\mathcal{S}^i} F_a \end{aligned} \tag{8}$$

Where $\mathcal{M}_{c_i}$ represents the projector operator on the basis state $|c_i\rangle$. At this point, the final fuzzy set is obtained by the inference operator which applies two sequential operations, namely implication and aggregation: starting from the values $\{P_{c_i}\}_{i=1}^{m_Y}$, the implication operator computes the output fuzzy set $\mu_i^Y(x)$ related to the i-th linguistic term of the output variable Y: $\mu_i^Y(x) = \min(P_{c_i}, \mu_{T_i^Y}(x)) \; \forall x \in U_i$ where U_i is the universe of the discourse of the fuzzy set $\{(x, \mu_{T_i^Y}(x)) | x \in U_i\}$ that describes the linguistic term T_i^Y of the output variable Y, with $i = 1, 2, \ldots, m_Y$. The aggregation operation

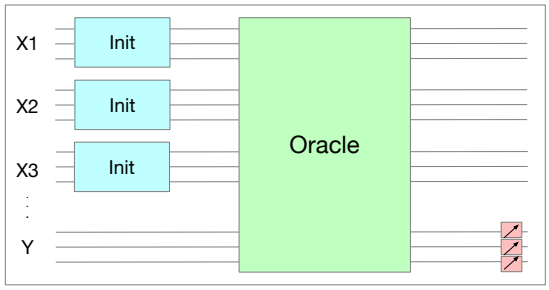


Fig. 2. Quantum circuit implementing QFIE.

combines, the fuzzy membership functions $\{\mu_i^Y(x)\}_{i=1}^{m_Y}$ to compute the output fuzzy set $S^Y = \{(x, \mu_Y(x))|x \in \cup_{i=1}^{m_Y} U_i\}$ as follows: $\mu_Y(x) = \max_{i=1,2,\ldots,m_Y}(\mu_i^Y(x))\ \forall x \in \bigcup_{i=1}^{m_Y} U_i$. Then, the crisp output value can be obtained by applying the preferred defuzzification method on the fuzzy set S^Y, such as the CoG approach reported in [30]. Figure 2 represents a schematic view of the quantum circuit implementing QFIE.

IV. Experimental Results

This work aims to solve some control problems at the CERN accelerator complex. The environments studied are two existing CERN beam lines representing control problems of different degrees of complexity. The former beam line describes a target steering task for a proton beam with a single control variable. In contrast, the second one represents the AWAKE electron beam line featuring ten control parameters. For both tasks, accurate simulations exist, which will be used mostly to develop and test the implementation of QFIE for their control. The environments are built on top of the OpenAI GYM template [31], while QFIE is implemented by using the Python Package available on GitHub[2]. Finally, section IV-C shows the performance of the QFIE based FRBS on the real AWAKE accelerator.

A. QFIE for the Target Steering Environment Control

This section aims at showing the application of QFIE in controlling the one-dimensional beam target steering environment based on the beam optics of the TT24-T4 transfer line at CERN [32]. This line is about 170 m long and transports protons with a momentum of 400GeV/c from the Super Proton Synchrotron (SPS) to some of the fixed-target physics experiments installed in the CERN North Area. TT24 is equipped with several dipole and quadrupole magnets to steer and focus the beam, various beam position monitors (BPM), and the actual target, which is placed at the end of the line. The objective of the task is to optimize the number of particles hitting the target by tuning the first dipole magnet in the line to maximize the event rates in the particle detectors. The left-hand side of Fig. 3 shows the relevant elements of TT24 together with horizontal beam trajectories obtained from tracking simulations for three different settings of the main bending dipole (orange). Depending on the dipole deflection angle, the particles hit the target (grey, hatched) at different horizontal positions, as illustrated by the zoomed view on the right-hand side of the figure. There are focusing (purple) and defocusing (olive) quadrupoles along the beam line to keep the beam particles confined.

Overall, the QFIE-based controller implemented aims to deflect the beam via the magnetic dipole according to two input variables such as the position reading of one of the BPMs installed in the beam line (cyan) X_{bmp} and the desired target position X_T. The allowed range of deflection angles Y is [-140, 140] μrad.

Two scenarios have been considered: in the first configuration (C1) X_T was any value in the range $[-1.5, 1.5]$ mm, while in the second configuration (C2), X_T was set to zero, and therefore X_{bpm} was the only input variable of the system. This simplification of the environment enables the construction of smaller quantum circuits for QFIE that can be reliably executed on real quantum hardware. Finally, to assess how well the beam is hitting the target, the following reward function is considered:

$$\mathcal{R} = -(1 - I) \tag{9}$$

where I represents the intensity of the Gaussian beam in the range $X_T \pm 3\sigma$, where σ is the beam size following from a fixed emittance of 11.8 nm. The beam is hitting the perfect target position when $\mathcal{R} = 0$.

The fuzzy partitions and the rule base used by QFIE in controlling C1 and C2 are reported in Fig. 4 and Fig. 9 respectively. For C1, five linguistic terms were considered for each variable: Negative (N), Medium Negative (MN), Zero (Z), Medium Positive (MP), and Positive (P). As a consequence, the number of fuzzy rules having as antecedent an *and* connection of two linguistic terms related to the different input variables is 25. For C2, three linguistic terms were considered for the input and the output variable: Negative (N), Zero (Z), and Positive (P). This leads to a set of 3 fuzzy rules. The limited number of fuzzy rules in C2 enables the execution of QFIE also on real NISQ devices, while their high level of noise forces the control of C1 via noiseless simulations of QFIE. In detail, the transpiled[3] quantum circuit implementing QFIE for C1 control has a depth of 3182 with 1554 CNOT gates, while the transpiled quantum circuit of QFIE for C2 control has an overall depth of 42 and a number of CNOTs equal to 20.

For C1 evaluation 20 different initial beam positions were considered, while the target X_T was randomly selected in the range $[-1.5, 1.5]$mm. Both X_{bpm} and Y have been normalized in the simulations in the range [-1, 1]. For sake of space, the reported results refer just to a particular target position, but the performances are very similar for all the different target positions tested. For C2 evaluation, due to the limitation in time for the availability of IBM quantum devices, just 10

[2]https://github.com/Quasar-UniNA/QFIE

[3]Transpilation is the process by which a quantum circuit is decomposed into basic gates for be executed on a specific quantum hardware. More details can be found in [33].

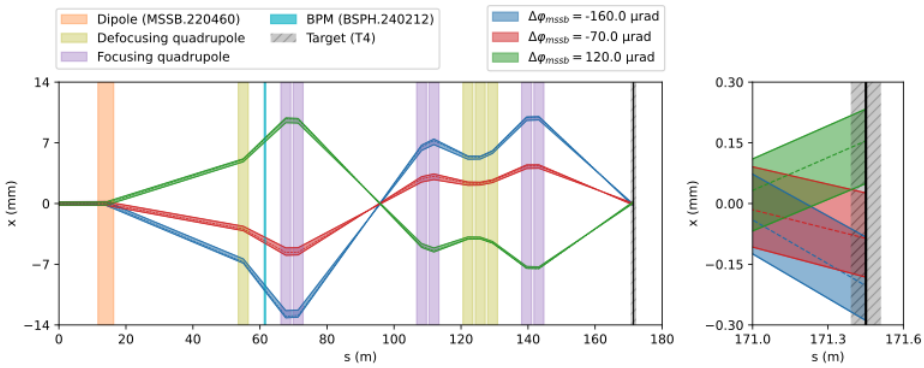
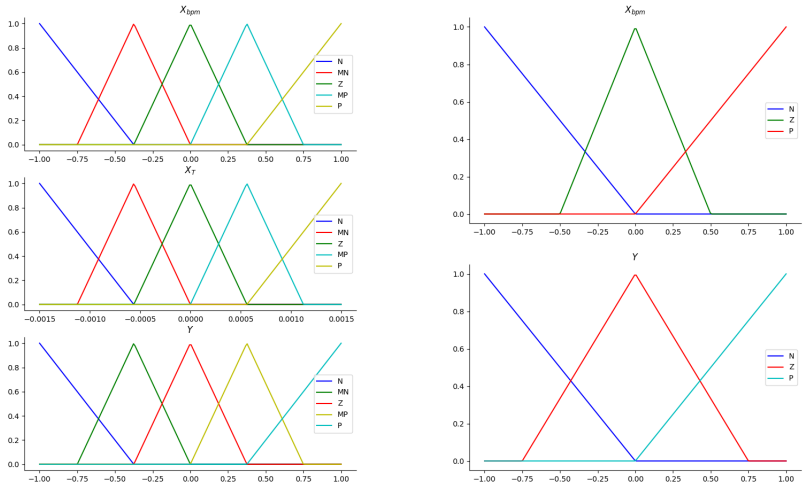


Fig. 3. One-dimensional beam target steering task at the CERN TT24-T4 beam line. Left: Horizontal beam trajectories obtained from tracking simulations are shown for three different settings of the main deflecting dipole (orange). Right: Zoomed view on the target (grey, hatched) region showing the horizontal position of impact of the beam for the three settings of the main dipole [8].



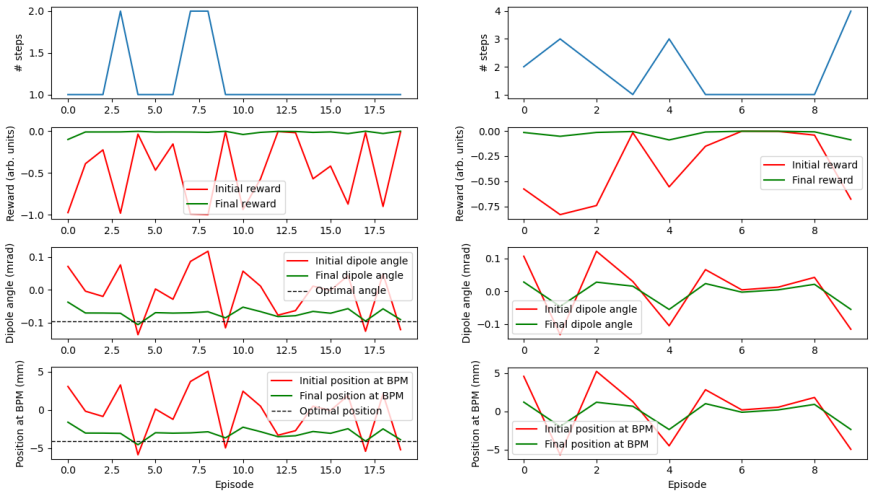
(a) (b)

Fig. 4. Membership functions for C1 (a) and C2 (b)

different initial beam positions were considered. In particular, the experiments have been carried out on the IBMQ Montreal device. The results obtained for both C1 and C2 are reported in Fig. 5(a) and 5(b), respectively. In detail, the upper plots represent the number of controller actions (steps) required to achieve the target position. The second plot reports the value of the reward function $\mathcal{R}$ at the beginning of the episode (red line) and at the end of it (green line). The two boxes on the bottom of the figure, represent the dipole angle in mrad (Y) and the beam position at the BPM in mm respectively. The dashed lines represent the ideal output of the system, while the red and green lines show the initial and final values. For configuration C2, the ideal output for both these plots is 0, where the target was set. It can be seen that QFIE is able to control the environment in all the episodes and for both configurations. The absence of noise during the simulation for C1 enables the control of the beam with two steps at worst. On the other hand, the noise in computing the QFIE action that occurs in C2 and the lower granularity of the input fuzzy partition, make more steps required for achieving the optimal solution. Overall, these results prove the suitability of QFIE in controlling this kind of environment, both by means of simulations of the quantum circuits and by means of execution on real quantum hardware.

B. QFIE for the Electron Beam Line Control

The Advanced Wakefield Experiment (AWAKE) at CERN uses high intensity 400 GeV proton bunches from the Super Proton Synchrotron (SPS) as a plasma wakefield driver. Electron bunches are simultaneously steered into the plasma cell to be accelerated by the proton induced wakefields. Electron energies up to 2GeV have been demonstrated over a plasma



(a) (b)

Fig. 5. Experimental results for the target steering environment control in C1 (a) and C2 (b)

X_{bpm}

X_T	N	MN	Z	MP	P
N	Z	MN	N	N	N
MN	MP	Z	MN	N	N
Z	MP	MP	Z	MN	MN
MP	P	P	MP	Z	MN
P	P	P	P	MP	Z

(a)

X_{bpm}

N	Z	P
P	Z	N

(b)

Fig. 6. Rule set for C1 (a) and C2 (b). The conjunction of the elements of the first row and column represents the antecedent part of a fuzzy rule having as consequent the corresponding matrix element. For instance, the first rule in C1 corresponds to the sentence *If X_{bpm} is Negative and X_T is Negative then the correction angle is Zero.*

cell of 10m length corresponding to an electric field gradient of 200MV/m [34]. The ultimate goal for AWAKE is to reach a field gradient of 1GV/m. These numbers are to be compared to conventional accelerating structures using radio-frequency (rf) cavities in the X-band regime, which are currently limited to accelerating field gradients of about 150 MV/m [35].

The AWAKE electron source and beam line are particularly interesting for algorithm preparation and testing due to the high repetition rate and insignificant damage potential in case of losing the beam at accelerator components. The AWAKE electrons are generated in a 5 MV photocathode rf gun, accelerated to 18 MeV and then transported through a beam line of 12m to the AWAKE plasma cell. The trajectory is controlled with 10 horizontal and 10 vertical steering dipoles according to the measurements of 10 BPMs, see Fig. 8. The BPM electronic read-out is at 10 Hz and acquisition through the CERN middleware at 1 Hz.

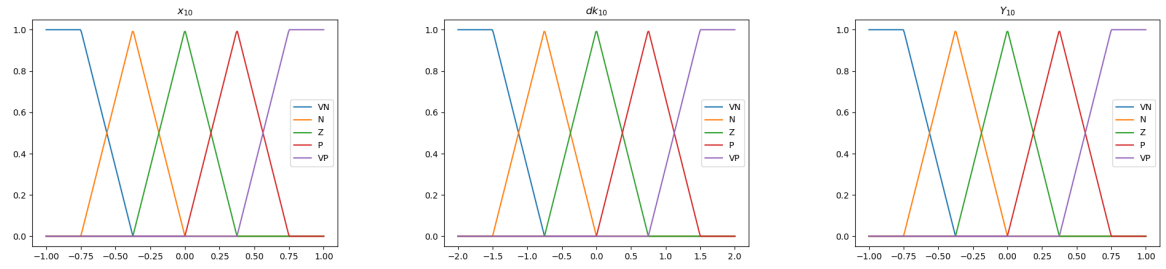
The goal was to develop a QFIE based controller able to correct the horizontal trajectory with similar accuracy as the response matrix-based singular value decomposition (SVD) algorithm that has been traditionally used [36].

The input state of the controller is formalized as a ten-dimensional vector of horizontal beam position measured with respect to the reference trajectory. Accordingly, the controller action is a ten-dimensional vector of corrector dipole magnet kick angles within a range of $\pm 300 \mu$rad. To evaluate the performance of the controller, it is used a reward function consisting of the negative root-mean-squared (rms) of the measured beam trajectory with respect to the reference at all the BPMs.

Developing a single QFIE controlling simultaneously all the corrector dipole magnets along the AWAKE trajectory would reflect in a quantum circuit too big for being classically simulated or executed on a current NISQ device. Therefore to solve the control problem an approach based on the D-NISQ reference model proposed in [26] has been exploited: the original 10-dimensional problem was divided into ten 1-dimensional control problems, where each corrector dipole magnet K_i with $i \in [1, 10]$ is controlled by a QFIE, $QFIE_i$ with $i \in [1, 10]$. Each $QFIE_i \quad \forall i \in [2, 10]$ acts considering two input variables x_i and dk_i, where the former refers to the distance from the ideal position of the beam registered by the corresponding BPM_i, while the latter refers to the sum of the deviation carried out by the magnets that are placed previously to the i-th magnet on the AWAKE beam line. Formally, denoting with $\hat{y}_i$ the corrector dipole magnet kick angles computed by $QFIE_i$, the dk_m input variable for $QFIE_m$ is defined as follows:

$$dk_m = \sum_{i=1}^{m-1} \hat{y}_i. \tag{10}$$

The action of $QFIE_1$ depends just from the position of the particle beam at the first beam position monitor along the trajectory. In detail, dk_i is defined in an interval [-2,2] $\forall i \in [2, 10]$; x_i is defined in an interval [-1,1] $\forall i \in [1, 10]$; the



(a) (b) (c)

Fig. 7. Fuzzy Partitions for $QFIE_{10}$ Input 7(a), 7(b) and Output 7(c) variables.

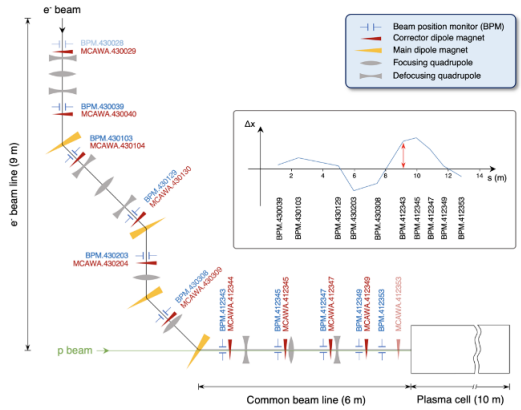


Fig. 8. Representation of the CERN AWAKE beam line with 10 trajectory correctors and 10 beam position readings along the line.

output corrector dipole magnet kick angles y_i are defined in the normalized interval [-1,1] $\forall i \in [1, 10]$.

The fuzzy partitions used for the variables of each $QFIE_i$ are the same. In particular, Fig. 7 shows them for $QFIE_{10}$. Moreover, Fig. 9(a) shows the fuzzy rule base for $QFIE_i \quad \forall i \in [2, 10]$, while Fig. 9(b) shows the fuzzy rule base for $QFIE_1$.

x_i \ dk_i	VN	N	Z	P	VP
VN	VP	VP	VP	P	Z
N	VP	VP	P	Z	N
Z	VP	P	Z	N	VN
P	P	Z	N	VN	VN
VP	Z	N	VN	VN	VN

(a)

x_1				
VN	N	Z	P	VP
VP	P	Z	N	VN

(b)

Fig. 9. Rule set for $QFIE_i \ \forall i \in [2, 10]$ (a) and $QFIE_1$ (b). The conjunction of the elements of the first row and column represents the antecedent part of a fuzzy rule having as consequent the corresponding matrix element. For instance, the first rule in $QFIE_i$ corresponds to the sentence *If dk_i is Very Negative and x_i is Very Negative then the correction angle is Very Positive*.

To minimize the number of interactions of the whole controller with the environment a bias factor b has been multiplied by the ten $QFIE$s output. Formally, denoting with $\hat{y}_i$ the output computed by $QFIE_i$, the final corrector dipole magnet kick angle y_i used to modify the environment state is obtained as follows:

$$y_i = \begin{cases} \hat{y}_i \cdot b & \text{if } \hat{y}_i \cdot b \in [-1, 1] \\ 1 & \text{if } \hat{y}_i \cdot b > 1 \\ -1 & \text{if } \hat{y}_i \cdot b < -1 \end{cases} \tag{11}$$

In our experimentation, b has been set to 10.

Fig. 10 reports the experimental results obtained by simulating the AWAKE environment controlled by QFIEs. The simulations stop when the reward objective reaches an rms value of 2mm. As shown by the plots, considering 50 different episodes, where the initial condition of the beam is far away from the threshold rms value, the quantum fuzzy control system is able to align the particle beam to the ideal trajectory in 100% of the episodes. Moreover, in all the episodes the desired trajectory is obtained at the worst by means of two interactions of the controllers with the environment.

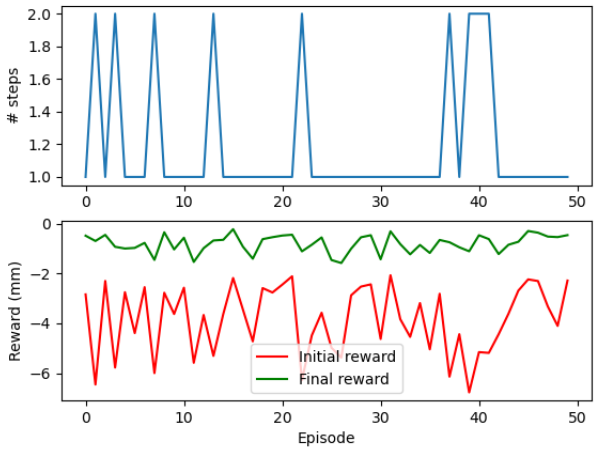


Fig. 10. Experimental results on the simulated AWAKE environment.

C. Online Tests on Real AWAKE Environment

The QFIE based FRBS was also evaluated in the real AWAKE environment to test sim-to-real transfer. The same configuration of QFIE considered for the simulated environment has been exploited. Fig. 11 shows the histograms reporting the obtained results: QFIE has been tested by considering four different levels of reward objective threshold value, -2mm (Fig. 11(a)), -1.6mm (Fig. 11(b)), -1.2mm (Fig. 11(c)) and -0.8mm (Fig. 11(d)). A lower value of rms reflects in a more precise control of the particle beam. For each threshold value, 20 independent episodes have been collected. Histograms in Fig. 11 report respectively the distribution of the number of interactions environment-control required to reach the desired rms value, the distribution of the 20 initial state reward values and the distribution of the 20 final state reward values. As highlighted by the plots, the QFIE based controller is able to solve both the tasks in the simulated and real AWAKE environment. Indeed the objective reward value is reached in the 100% of the episodes considered. The number of steps required for achieve such an impressive result is in line with the simulated environment, except for an outlier in the rms threshold equals to -0.8mm situation.

V. Conclusion

In this work, the innovative QFIE introduced in [5] has been experimentally tested for the very first time for the implementation of FRBSs useful to control real-world environments, such as those related to particle physics accelerators at CERN facilities. The main result obtained from this research is twofold: on the one hand, it has been shown that QFIE is able to control these complex environments both by means of simulations and executions on real hardware of the quantum circuits implementing the algorithm; on the other hand, as shown on the test carried out on the real AWAKE beam line, it has been proved that FRBSs could be a valid tool for the control of particle accelerators for the physics experiments at CERN.

In detail, the research was carried out on two successive scenarios, two existing CERN beam lines representing control problems of different degrees of complexity. In the first scenario, the QFIE controller implemented aims to deflect the beam via the magnetic dipole in an environment based on the TT24-T4 transfer line at CERN: in this simpler context, the whole algorithm has been executed on real IBM Quantum computers, proving the feasibility of QFIE in controlling this kind of environment on real quantum devices. The second scenario consists of the AWAKE use case, where the 10-dimensional environment is much more complex and current NISQ devices are not ready to handle the resulting QFIE circuits. However, in this case, the simulated quantum circuits were tested on real data as an online controller of the beam line. This result proves for the very first time the capability of a FRBS to control a real particle accelerator.

In the future QFIE based FRBS will be developed and tested for more complex experiment and environments, where no analytical solutions are available to control the systems. Moreover, further tests on real quantum hardware execution of quantum circuits implementing QFIE will be carried out.

VI. Acknowledgments

This research is supported by the IEEE Computational Intelligence Society Graduate Student Research Grants obtained by the author Roberto Schiattarella in 2022. Moreover, we thank the ATLAS collaboration at CERN.

References

[1] L. A. Zadeh, "Fuzzy sets,"information and control", 1965, vol. 8, 338–353," *Fuzzy Algorithms,"Information and Control*, vol. 12, pp. 94–102, 1968.

[2] ——, "Fuzzy sets," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 394–432.

[3] M. Sugeno, "An introductory survey of fuzzy control," *Information sciences*, vol. 36, no. 1-2, pp. 59–83, 1985.

[4] R.-E. Precup and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in industry*, vol. 62, no. 3, pp. 213–226, 2011.

[5] G. Acampora, R. Schiattarella, and A. Vitiello, "On the implementation of fuzzy inference engines on quantum computers," *IEEE Transactions on Fuzzy Systems*, pp. 1–15, 2022.

[6] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[7] V. Kain, S. Hirlander, B. Goddard, F. M. Velotti, G. Z. Della Porta, N. Bruchon, and G. Valentino, "Sample-efficient reinforcement learning for cern accelerator control," *Physical Review Accelerators and Beams*, vol. 23, no. 12, p. 124801, 2020.

[8] M. Schenk, E. F. Combarro, M. Grossi, V. Kain, K. S. B. Li, M.-M. Popa, and S. Vallecorsa, "Hybrid actor-critic algorithm for quantum reinforcement learning at cern beam lines," *arXiv preprint arXiv:2209.11044*, 2022.

[9] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.

[10] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, "An artificial neuron implemented on an actual quantum processor," *npj Quantum Information*, vol. 5, no. 1, pp. 1–8, 2019.

[11] G. Acampora, R. Schiattarella, and A. Vitiello, "Using quantum amplitude amplification in genetic algorithms," *Expert Systems with Applications*, vol. 209, p. 118203, 2022.

[12] G. Acampora and A. Vitiello, "Implementing evolutionary optimization on actual quantum processors," *Information Sciences*, vol. 575, pp. 542–562, 2021.

[13] Y.-P. Huang, P. Singh, W.-L. Kuo, and H.-C. Chu, "A type-2 fuzzy clustering and quantum optimization approach for crops image segmentation," *International Journal of Fuzzy Systems*, vol. 23, no. 3, pp. 615–629, 2021.

[14] J. Li and Z. Hao, "A quantum probabilistic linguistic term framework to multi-attribute decision-making for battlefield situation assessment," *International Journal of Fuzzy Systems*, vol. 24, no. 1, pp. 495–507, 2022.

[15] M. Hou, S. Zhang, and J. Xia, "Quantum fuzzy k-means algorithm based on fuzzy theory," in *International Conference on Adaptive and Intelligent Systems*. Springer, 2022, pp. 348–356.

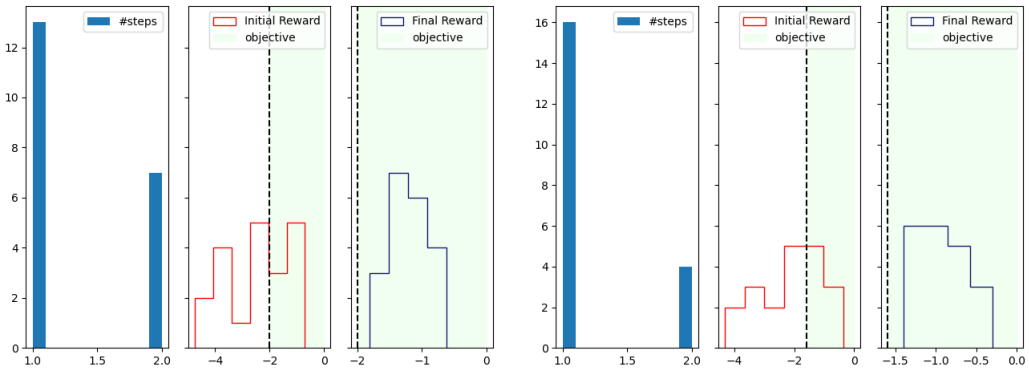
[16] S. Ishikawa and K. Kikuchi, "Quantum fuzzy logic and time," *Journal of Applied Mathematics and Physics*, vol. 9, no. 11, pp. 2609–2622, 2021.

[17] R. Leporini, C. Bertini, and F. C. Fabiani, "Fuzzy representation of finite-valued quantum gates," *Soft Computing*, vol. 24, no. 14, pp. 10 305–10 313, 2020.

[18] D. Aerts, T. Durt, and B. Van Bogaert, "A physical example of quantum fuzzy sets and the classical limit," *Tatra Mt. Math. Publ*, vol. 1, pp. 5–15, 1993.

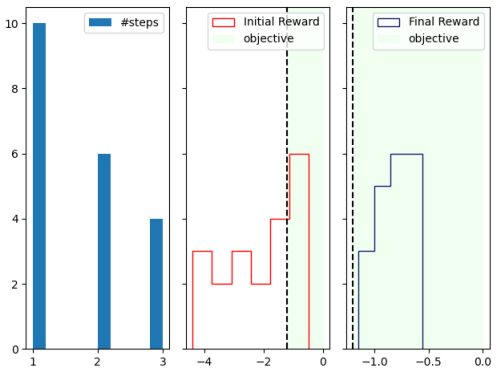
[19] G. G. Rigatos and S. G. Tzafestas, "Parallelization of a fuzzy control algorithm using quantum computation," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 451–460, 2002.

[20] L. Visintin, A. Maron, R. Reiser, and V. Kreinovich, "Aggregation operations from quantum computing," in *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2013, pp. 1–8.

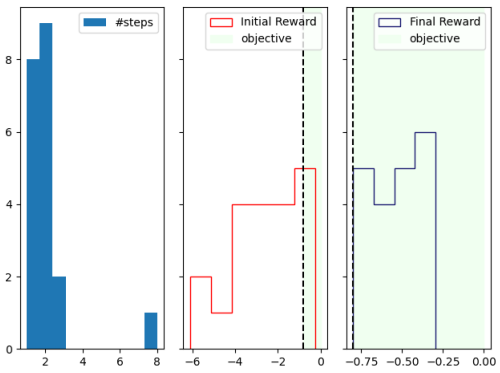


(a) rms = 2mm

(b) rms = 1.6mm



(c) rms = 1.2mm



(d) rms = 0.8mm

Fig. 11. Online experimentation on the real AWAKE accelerator. Each plot refers to a different value of rms threshold.

[21] A. Ávila, M. Schmalfuss, R. Reiser, and V. Kreinovich, "Fuzzy xor classes from quantum computing," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2015, pp. 305–317.

[22] A. Pourabdollah, G. Acampora, and R. Schiattarella, "Fuzzy logic on quantum annealers," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2021.

[23] ——, "Implementing defuzzification operators on quantum annealers," in *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2022, pp. 1–6.

[24] G. Acampora, F. Luongo, and A. Vitiello, "Quantum implementation of fuzzy systems through grover's algorithm," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

[25] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[26] G. Acampora, F. Di Martino, A. Massa, R. Schiattarella, and A. Vitiello, "D-nisq: a reference model for distributed noisy intermediate-scale quantum computers," *Information Fusion*, vol. 89, pp. 16–28, 2023.

[27] L. Magdalena, "Fuzzy rule-based systems," in *Springer Handbook of Computational Intelligence*, J. Kacprzyk and W. Pedrycz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 203–218.

[28] R. Jager, *Fuzzy logic in control*. Rene Jager, 1995.

[29] M. M. Gupta and J. Qi, "Theory of t-norms and fuzzy inference methods," *Fuzzy sets and systems*, vol. 40, no. 3, pp. 431–450, 1991.

[30] Z. Kovacic and S. Bogdan, *Fuzzy controller design: theory and applications*. CRC press, 2018.

[31] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, and J. Schulman, "Jie tang and wojciech zaremba. openai gym," *arXiv preprint arXiv*, vol. 1606, 2016.

[32] G. L. D'Alessandro, J. Bernhard, A. Gerbershagen, L. Gatignon, M. Brugger, S. M. Gibson, F. M. Velotti, N. Doble, D. Banerjee, and B. Rae, "Jacow: Target bypass beam optics for future high intensity fixed target experiments in the cern north area," *JACoW IPAC*, vol. 2021, pp. 3046–3048, 2021.

[33] G. Acampora and R. Schiattarella, "Deep neural networks for quantum circuit mapping," *Neural Computing and Applications*, vol. 33, no. 20, pp. 13 723–13 743, 2021.

[34] E. Adli, A. Ahuja, O. Apsimon, R. Apsimon, A.-M. Bachmann, D. Barrientos, F. Batsch, J. Bauche, V. Berglyd Olsen, M. Bernardini *et al.*, "Acceleration of electrons in the plasma wakefield of a proton bunch," *Nature*, vol. 561, no. 7723, pp. 363–367, 2018.

[35] R. Agustsson, P. Carriere, O. Chimalpopoca, V. Dolgashev, M. Gusarova, S. Kutsaev, and A. Y. Smirnov, "Experimental studies of a high-gradient x-band welded hard-copper split accelerating structure," *Journal of Physics D: Applied Physics*, vol. 55, no. 14, p. 145001, 2022.

[36] Y. Chung, G. Decker, and K. Evans, "Closed orbit correction using singular value decomposition of the response matrix," in *Proceedings of International Conference on Particle Accelerators*. IEEE, 1993, pp. 2263–2265.