Iranian Journal of Fuzzy Systems

Volume 19, Number 4, (2022), pp. 125-136

Original paper

University of Sistan and Baluchestan

A new concept of a fuzzy ontology controller for a temperature regulation

J. Benić[1], M. Krznar[2], T. Stipančić[3] and Ž. Šitum[4]

[1,2,3,4] *Department of Robotics and Production System Automation, Faculty of Mechanical Engineering and Naval Architecture University of Zagreb, Zagreb HR-10000, Croatia*

jbenic@fsb.hr, mkrznar@fsb.hr, tstipancic@fsb.hr, zsitum@fsb.hr

Abstract

In this paper, a new concept of the fuzzy logic controller is presented. The proposed concept is called fuzzy ontology controller and it's based on an ontology description of the Fuzzy Logic Controller (FLC). In this concept, the SPARQL query sent to the ontology replaced the fuzzification process, fuzzy interface, and rules of the fuzzy controller while defuzzification is done on an obtained response. Fuzzy ontology is designed with Protégé and hosted in the Virtuoso database on a remote server. Simulation results obtained with the proposed fuzzy ontology controller are compared to ones obtained with the classical fuzzy logic controller. Both controllers obtained the same simulation results due to the same membership function and defuzzification process. The results showed that this concept can be used as a part of some IoT solution within the vision of Industry 4.0. Also, results confirmed that the ontology-based controller works properly and gives the same output as FLC. The experiment is carried out on a heat exchange process controlled with a PLC. Modbus TCP protocol is used for communication between the server PC and a PLC. Experimental results show that only slow processes are controllable for now due to slow communication between PLC and server and the time needed for execution of SPARQL queries.

Keywords: Ontology, linked open data, fuzzy ontology control, fuzzy logic control, SPARQL, control theory, temperature regulation.

1 Introduction

Nowadays, advances in technology are increasingly affecting people's daily lives. Progress is particularly significant in artificial intelligence leading to the creation of new scientific fields, methods, technologies, and their applications including applications in blockchain technology [9], renewable energy [23, 34], electronics [35], energy consumption strategies [20], and robotics [22].

Fuzzy logic is used to a great extent on web or within expert (knowledge) systems based on ontologies. An ontology written in Ontology Web Language (OWL) [19] follows the principles of Open World Assumptions (OWA) [16]. Such ontology can respond to a query by providing more than one equally right answer resulting with hesitations or system indecisions. It is widely pointed out that the classical ontology model is not sufficient to deal with imprecise and vague knowledge strongly characterizing some real-world applications [30, 36]. To resolve these ambiguities, some authors use methods based on probabilistic reasoning under uncertainties like Bayesian Networks [29] or Fuzzy Logic [3]. Some authors also show that it is possible to have a mixture that can account for reasoning based on uncertainties, but also for other epistemologically relevant concepts, such as privacy, trust or belief [10, 28, 33, 36]. In the last decade, ontologies have become a standard for knowledge representation. Ontologies are increasingly used in all kinds of applications due to their structured and machine-interpretable form. They can be defined as a hierarchical description of relationships and entity dependencies and have a powerful expressive ability on knowledge representation. They are commonly used

Corresponding Author: J. Benić
Received: August 2021; Revised: May 2022; Accepted: June 2022.
https://doi.org/10.22111/IJFS.2022.7091

in linguistics [12], data mining [18], domain-specific manufacturing [8], etc. In [2] they are used for scene creating in the development of automated vehicles.

Today, Fuzzy Logic Controllers (FLC) are widely used by researchers. In [14] type-2 fuzzy logic controller was implemented for the control of a mobile two-wheeled inverted pendulum in combination with linear matrix inequalities. In [24] for maintaining the aerodynamic power of the speed wind energy system type-2 fuzzy logic controller is proposed. The adaptive type-2 fuzzy controller has been applied for variable speed control of doubly-fed induction generator wind turbine in [13]. Fuzzy optimized data management for big data analysis is presented in [21] and tested in real-time on the weather forecast dataset. The maneuverability of tethered space net robot for debris capture and removal was investigated in [37] and improved with fuzzy-based adaptive super-twisting SMC. Motorcycle traction control system was investigated in [7] and controlled with FLC where parameters of FLC were tuned manually and with an evolutionary algorithm. The adaptive fuzzy observer is used in [17] for estimation of state variables of uncertain nonlinear systems based only on measurable system output.

The growing importance of vague information on ontologies demanded a standard way to represent such pieces of information which led to a fuzzy ontology. Integration of fuzzy logic in ontologies was studied in [1] where the new formalism for designing a fuzzy ontology was proposed. In [5, 31] a methodology for a fuzzy ontology was proposed and a prototypical fuzzy plug-in for Protégé was developed. The plug-in is only compatible with Protégé 4.1 and uses Gurobi Optimizer for calculations which makes it hard to implement. Next, in [4] fuzzy logic was used over WordNet ontology to represent semantic similarity and a hyponymic degree in a synonym set. A fuzzy interface for medical guidelines was developed in [32] where a domain ontology was refactored with additional fuzzy relations and properties. A fuzzy ontology for human activity representation was proposed in [26] and modeled using fuzzy Protégé plug-in from [5]. The proposed ontology showed better results during the recognition process than standard ontology. In [6] a new fuzzyDL reasoner was proposed for fuzzy ontologies. The fuzzyDL reasoner in combination with Protégé fuzzy plug-in gives new unique features to a fuzzy ontology. For the diabetes mellitus domain, the fuzzy ontology was proposed in [11] and it is considered as the first fuzzy case-based ontology in the medical domain. The proposed ontology covers the terminologies and logic of diabetes mellitus diagnosis.

In this paper, a Python-based Fuzzy Ontology Controller (FOC) is proposed. In the proposed concept, fuzzification, fuzzy interface, and rules are replaced with ontology while the Python Numpy module is used for membership interpretation and defuzzification based on given ontology output. The experimental setup is described in Section 2 while a dynamical model of the heat exchange process is given in Section 3. In Section 4 FOC is described and detailed steps for creating FOC from FLC are given. Simulation and experimental results are shown in Section 5 while conclusion and further work is given in Section 6.

2 Experimental setup

The experimental setup used in this research is shown on Figure 1. The setup consists of the Mitsubishi FX5U-32MT/ESS PLC (1) used for temperature measuring and generating of Pulse-Width Modulation (PWM) signal in the range of 0 - 24 VDC. Mitsubishi GS2107-WTBD HMI (2) is used for data representation. PWM signal from Programmable Logic Controller (PLC) is used for temperature regulation inside the heating chamber (3). Two 12 VDC, 18 W light bulbs are used for heating, while a 10k NTC thermistor is used for temperature measurement. The thermistor and light bulbs are placed inside a transparent plastic container. PLC communicates with the router (4) over the local Ethernet.

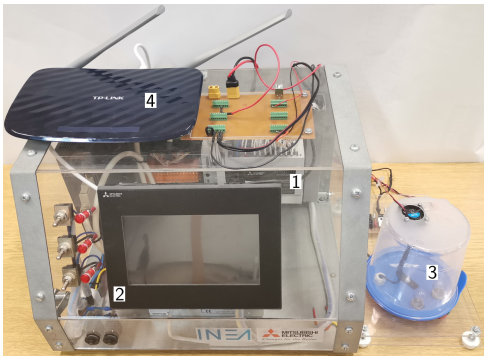


Figure 1: Experimental setup: 1 - PLC, 2 - HMI, 3 - heating chamber, 4 - router

Schematics representation of the communication process is shown on Figure 2. Internet access is provided to the PLC via a router. An external port is defined in a router that allows communication between the WEB server and PLC. The Modbus TCP is used as a communication protocol between PLC and the WEB server. The virtuoso database is hosted on the WEB server.

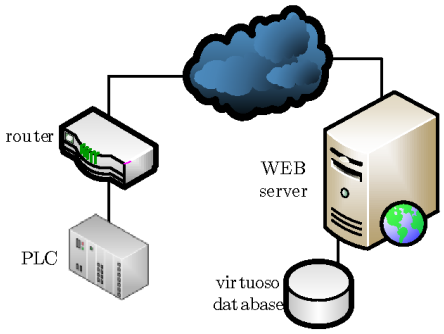


Figure 2: Schematics representation of the communication process

3 Dynamical model of a temperature control

The dynamical model of the proposed heat exchange process can be represented as a First-Order Process with Dead-Time (FOPDT). From [25] FOPDT model is given as:

$$G_p = \frac{K_p \mathrm{e}^{-T_d s}}{T_a s + 1}, \tag{1}$$

where K_p is process gain, T_a is time constant of the first-order lag term and T_d is equivalent dead-time. Furthermore, the heat exchange process is characterized by relatively slow aperiodic dynamics which can be represented by second-order aperiodic process where the dead time is approximated as:

$$\mathrm{e}^{-T_d s} \approx \frac{1}{T_d s + 1}. \tag{2}$$

Second order lag-term model of the heat exchange process is obtained from (1) and (2) as:

$$G_p = \frac{K_p}{(T_a s + 1)(T_d s + 1)} = \frac{K_p}{T_a T_d s^2 + (T_a + T_d)s + 1}. \tag{3}$$

A detailed description of the identification procedure for the proposed heat exchange process is given in [27].

4 Fuzzy controllers

The main components of an FLC are shown on Figure 3. Fuzzification is the first block inside an FLC which converts given inputs to fuzzy variables. The next block is a fuzzy interface in combination with rules on which fuzzy logic is done and the fuzzy output is obtained by the rules defined with linguistic variables. The last block is defuzzification where fuzzy outputs are converted to the output variable.

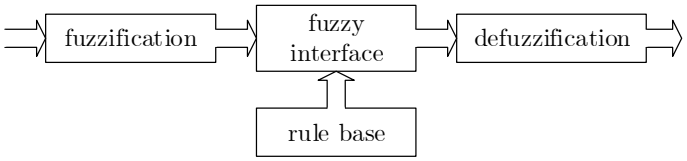


Figure 3: Structure of a fuzzy logic controller

4.1 Classical fuzzy controller

The classical FLC is shown in a Figure 4 where the input and output variables of the process are desired θ_d and measured temperature θ_m, respectively. Inputs to the fuzzy controller are temperature error e and error derivation de/dt while voltage u is the output variable of the FLC.

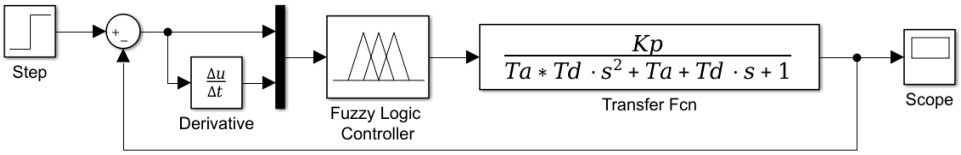


Figure 4: Classical FLC of the heat exchange process

The values of the linguistic variables of the FLC are defined as $[ln,\ sn,\ z,\ sp,\ lp]$ where ln means large negative, sn small negative, z zero, sp small positive and lp large positive error. Membership function for an input variables e and de/dt and output variable u are shown in Figure 5. The fuzzy rules used for decision-making are summarized in Table 1.

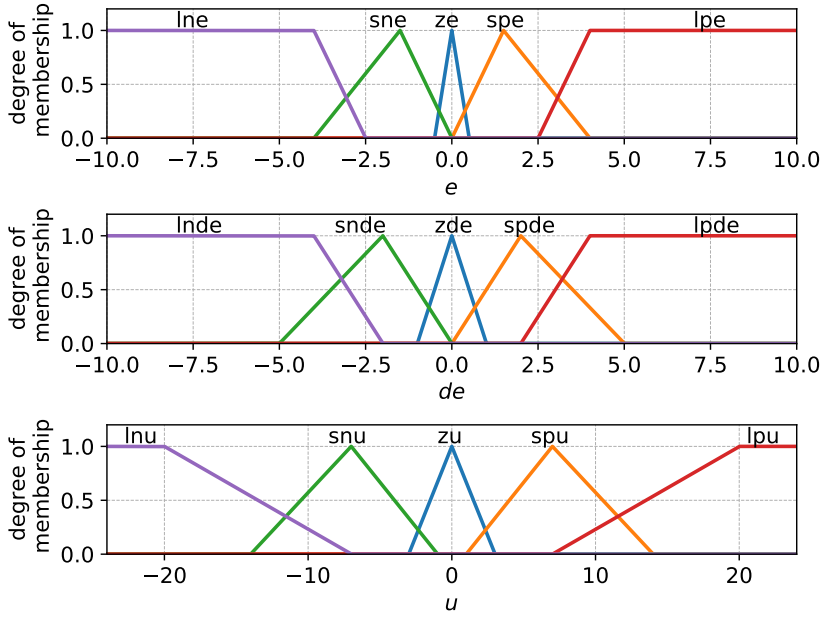


Figure 5: Membership functions

Table 1: FLC rules

de \ e	ln	sm	z	sp	lp
ln	ln	ln	ln	sn	z
sn	ln	ln	sn	z	sp
z	ln	sn	z	sp	lp
sp	sn	z	sp	lp	lp
lp	z	sp	lp	lp	lp

4.2 Fuzzy ontology controller

In this section, detailed steps for designing FOC are given and described. From an FLC three main classes can be distinguished and used for designing a FOC:

1. *Inputs* - inputs of FOC
2. *Outputs* - outputs of FOC
3. *Rules* - fuzzy rules used in FOC

These three classes are subclasses of the main class called *Fuzzy*. In the ontology representation, the FOC classes *Inputs*, *Outputs* and *Rules* need to be further divided. The class *Inputs* is divided on two subclasses, *E* and *De* which represents temperature error and error derivation. Class *Output* is further divided to subclass *U* which represents a voltage output of FOC while the class *Rules* is divided into five subclasses which represent fuzzy logic rules where:

- *R_lnu* implies on large negative output
- *R_snu* implies on small negative output
- *R_zu* implies on zero output
- *R_spu* implies on small positive output
- *R_lpu* implies on large positive output

Complete class hierarchy is shown on Fig 6.

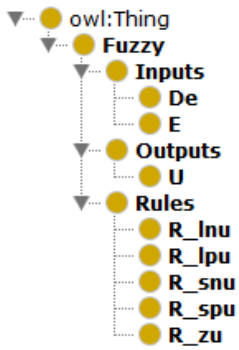


Figure 6: Classes hierarchy of a FOC

Object properties used for designing FOC are shown in Figure 7. Only three of them were used and they are defined as:

1. *hasInput* - used in rule instances for determining which instance is active
2. *hasOutput* - gives the output of a selected rule
3. *hasError* - used for determining which input variable of a FOC is given

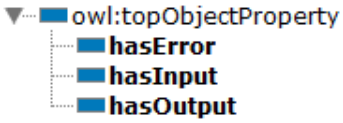


Figure 7: Properties hierarchy of a FOC

Data properties used in FOC are shown in Figure 8. Three data properties were used and assigned to an instance. Domain of a given membership function is defined with *min* and *max* data properties while *universe* property defines domain of a given input and output variables.



Figure 8: Data property hierarchy of a FOC

Individuals defined in classes *E*, *De* and *U* are linguistics variables with prefix [*ln*, *sn*, *z*, *sp*, *lp*] and extension of lower-case classes name (e.g. *lnde* means large negative derivation error). Individuals in subclass of a class *Rules* have lover-case subclass name with extension *_x* where *x* is a number of input-input combination for that rule from Table 1 (e.g. *r_lnu_1* is first combination of inputs which gives output *lnu*). Individuals defined in classes *Inputs* and *Outputs* are inputs (*e* and *de*) and output (*u*) of the FOC, respectively. Each input and output linguistic variable is annotated as fuzzy OWL 2 datatype with its corresponding membership function. An example of fuzzy OWL 2 annotation for individual *lne* is given on Figure 9.

```
<fuzzyOwl2 fuzzyType="datatype">
        <Datatype type="leftshoulder" a="−4" b="−2.5" />
</fuzzyOwl2>
```

Figure 9: Fuzzy OWL 2 datatype for individual *lne*

Relations between individuals and their data properties are shown on Figure 10, 11 and 12. Corresponding input linguistic variable of FOC is linked with *hasError* data property to an input type variable. An example of an input variable of FOC is shown in Figure 10.

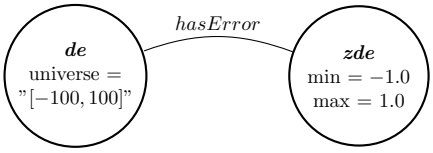


Figure 10: Object and data properties of the *zde* individual

An example of the output linguistic variable is shown in Figure 11. They are linked to an output type of the FOC with *hasOutput* data property.

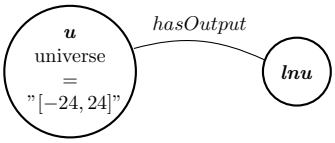


Figure 11: Object and data properties of the *lnu* individual

A rule is defined for each combination of inputs. Each rule indicate to an input linguistic variable with *hasInput* data property while given rule is linked to corresponding output with *hasOutput* data property as shown on Figure 12.

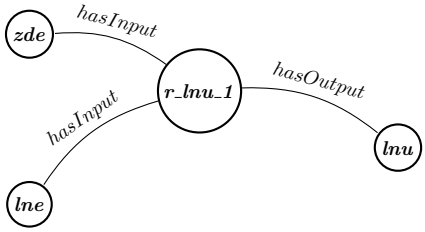


Figure 12: Object properties of the *r_lnu_1* individual

4.3 Fuzzy ontology controller algorithm

Python is used for obtaining inputs, sending SPARQL queries to a virtuoso database, and for results processing. The algorithm used in FOC is summarized in Algorithm 1. Firstly, inputs and outputs need to be initialized using *getParameters* function by using the query shown in Figure 13. Afterward, the FOC instance is created using obtained input and output parameters, database location, and search query, then a *foc.run* method is called with temperature error and error derivation. Method *foc.run* searches the database with given inputs and determines which inputs are active and their implication to outputs. From obtained outputs, by using Mamdani min-max definition of fuzzy logic implication and summation [15], aggregated output μ_{agg} membership function is obtained as:

$$\mu_{agg}(e, de, u) = \max\left\{\min\left\{\mu_{e,1}(e), \mu_{de,1}(de), \mu_{u,1}(u)\right\}, \ldots, \min\left\{\mu_{e,N}(e), \mu_{de,N}(de), \mu_{u,N}(u)\right\}\right\}, \tag{4}$$

where μ_e, μ_{de}, and μ_u are error, error derivation, and output membership functions, respectively, and index N is a number of active rules for given inputs. The centroid defuzzification of the aggregated output membership function is used for the defuzzification process in Algorithm 1 given as:

$$u_0 = \frac{\int u \cdot \mu_{agg}(e, de, u) du}{\int \mu_{agg}(e, de, u) du}. \tag{5}$$

Reinitialization of the inputs and outputs is possible inside the proposed algorithm if new membership functions are added to the Virtuoso database.

Algorithm 1 FOC

```
initialize inputs = getParameters()                    ▷ inputs: e, de
initialize outputs = getParameters()                   ▷ outputs: u
foc ← FOC(inputs, outputs, database, query)
u ← foc.run(e, de)
procedure FOC.RUN(e, de)
    initialize u = list()
    f ← (e, de)
    R ← sparql.query(database, query.format(f))
    for each r ∈ R do
        p ← parse r
        u.add(p.output)
    end for
    defuzzification of u by e, de
    if needed then
        reinitialize inputs
        reinitialize outputs
    end if
    return u
end procedure
```

Input and output parameters are obtained via function *getParameters* summarized in Algorithm. 2. The function accepts three parameters: database location, SPARQL query, and parameters used for parsing the obtained results.

Algorithm 2 Function for obtaining parameters from virtuoso database

```
function GETPARAMETERS(database, query, p)
    initialize results = dict()
    R ← sparql.query(database, query)
    for each r ∈ R do
        results ← parse r by p
    end for
    return results
end function
```

The SPARQL Query example for obtaining parameters is given on Figure 13. The query returns the list of three parameters where the first one is a linguistic variable, the second is a fuzzy OWL 2 fuzzy datatype membership function with parameters, and the last one is the universe of a variable.

```
PREFIX fuzzy: <http://elte.fsb.hr/fuzzy.owl#>
SELECT ?e ?e_f ?e_u WHERE{
        ?e fuzzy:hasError fuzzy:e ; fuzzy:fuzzyLabel ?e_f ; fuzzy:hasError ?e_g .
        ?e_g fuzzy:universe ?e_u .
}
```

Figure 13: SPARQL query example for obtaining error parameters

The SPARQL query which determines relations between activation functions and their output functions is given on Figure 14. It searches for active input functions based on given error and error derivation, which are represented in query with %(e)s and %(de)s. The output functions are determined based on active input functions.

```
PREFIX fuzzy: <http://elte.fsb.hr/fuzzy.owl#>
SELECT DISTINCT ?e ?de ?u WHERE{
        ?e fuzzy:hasError fuzzy:e ; fuzzy:min ?e_min ; fuzzy:max ?e_max .
        Filter( ?e_min <= %(e)s && ?e_max >= %(e)s )
        ?de fuzzy:hasError fuzzy:de ; fuzzy:min ?de_min ; fuzzy:max ?de_max .
        Filter( ?de_min <= %(de)s && ?de_max >= %(de)s )
        ?p fuzzy:hasOutput ?u ; fuzzy:hasInput ?e, ?de .
}
```

Figure 14: SPARQL fuzzy query

The fuzzyDL reasoner is written in JAVA and provides Java API for the users. Also, it has its native fuzzyDL syntax, but it supports Fuzzy OWL 2 syntax as well. In comparison to fuzzyDL reasoning, the proposed algorithm uses standard SPARQL syntax for communication with a fuzzy ontology and is written in Python. Also, the Gurobi optimization solver is removed together with complex tableau rules which decompose complex concept expressions into simpler ones. From the aspect of the computational complexity, the NP problem is still obtained, the same as for fuzzyDL.

5 Results

In this section, simulation results are presented for the FOC and compared with FLC. Experimental results are obtained for the FOC with the time needed for obtaining the result from the server which includes the time needed for communication, queries, membership interpretation, and the defuzzification process.

5.1 Simulation results

The comparative study is carried out between the FLC and FOC and it is based on a simulation. The study is necessary to certify that the proposed concept of the FOC is working the same as the FLC. To ensure the fairness of the test, both controllers have identical rules and membership functions, where the subject of interest is not the way of choosing the rules and membership functions but the functionality of the proposed concept.

Numerical simulations are carried out with Python because FOC is not supported in Matlab. Simulation results are shown on Figure 15 while the error between desired and measured trajectory is given on Figure 16. Both controllers obtained the same results with a small overshoot of approximately 1°C, and a settling time of 40 seconds. Results showed that both controllers have the same response due to identical membership functions and the defuzzification process. Additionally, results proved that the proposed concept of FOC is working properly and it obtained the same result as FLC but in a different way with the use of ontology. The benefit of such a concept is in using the same controller for multiplay similar processes without the need of compiling the algorithm on the PLC every time the parameters of the controller are changed. Also, a knowledge database can be shared with other autonomous processes and easily integrated into modern IoT concepts.

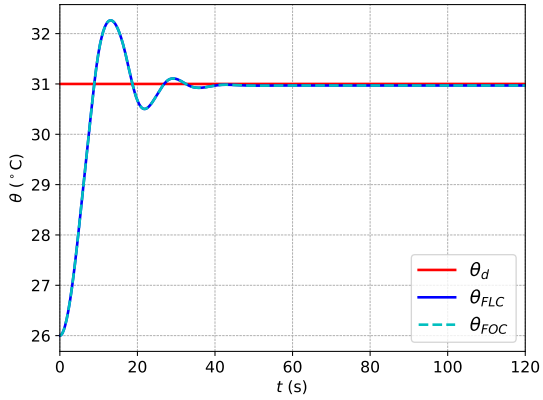


Figure 15: Experimental results - regulated temperature

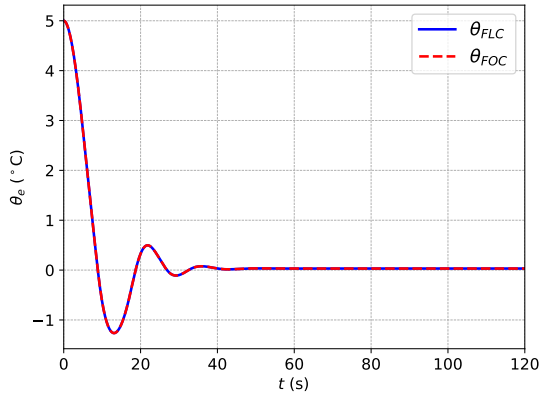


Figure 16: Error of the proposed experimental setup

5.2 Experimental results

Obtained experimental results of the proposed heat exchange process are shown in Figure 17 and Figure 18. Results are obtained with PD FOC. The response showed that there exists a small overshot while the absolute value of the steady-state error is smaller than 0.1 degrees Celsius.

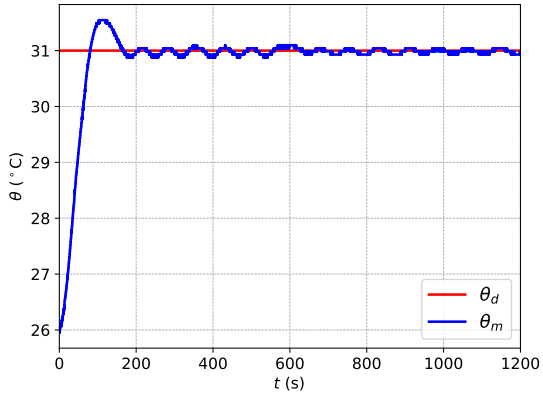


Figure 17: Experimental results - regulated temperature

Time taken for obtaining the control variable is shown in Figure 19. This time takes into account the time needed for reading PLC registries over Modbus TCP protocol, running SPARQL queries, and writing control variable back in the PLC. For the experiment duration of 1200 seconds, the server made 9974 queries. Each query includes reading and

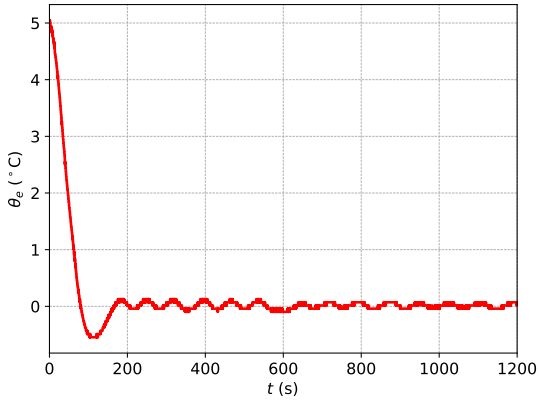


Figure 18: Error of the proposed experimental setup

writing data to PLC and running SPARQL queries. For the 9974 queries made by the server, the mean time was 0.12 seconds while the standard deviation is 0.018 seconds. Obtained results are acceptable for control of a given process but in comparison to the FLC 0.12 seconds is extremely slow for control of fast processes. The main reason for such slow response time is in communication between PLC and server and the time needed for execution of SPARQL queries to the database. All overall, the proposed concept works and it can easily control slow processes.

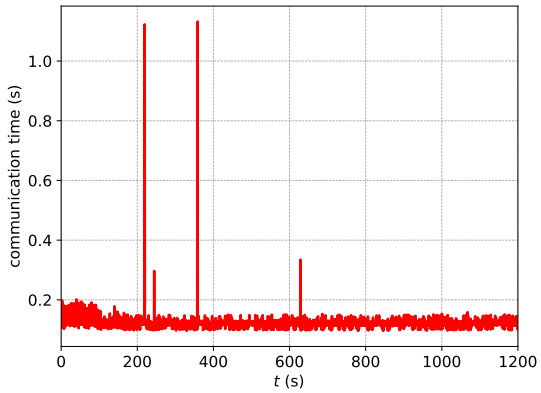


Figure 19: Time needed for PLC to obtain control variable

6 Conclusion

Simulation results show that a novel FOC based on linked open data and Industry 4.0 principles is theoretically possible. The proposed controller obtained the same simulation results as the FLC. Experimental results showed that implementation of the proposed concept is possible, but the time necessary for obtaining an output signal of a FOC is extremely long due to a slow internet connection. Because of that, only a slow process can be successfully controlled with FOC.

The proposed concept uses a knowledge database for its configuration. That allows an easy reconfiguration of the FOC for use in the different control processes. There is no need for recompiling the algorithm when the new rules or membership functions are introduced to the FOC. That is done on the fly. An optional feature, of data logging on the remote server, can be easily implemented into the proposed algorithm. This feature has been used in this paper for obtaining experimental results.

In future work, additional investigation of the control process needs to be carried out to see how fast the process has to be so that it can't be controllable. The impact of the network infrastructure on the latency needs to be further investigated, specialty the use of 100 gigabit Ethernet or 5G networks. Also, steps such as database optimization and optimization of fuzzification and defuzzification process should be further considered.

Acknowledgment

This work has been supported in part by Croatian Science Foundation under the project Affective Multimodal Interaction based on Constructed Robot Cognition - AMICORC (UIP-2020-02-7184). We would also like to thank for the constructive advice to our dear professor M. Essert.

References

[1] H. Agrebi, A. Bahri, R. Bouaziz, *Fuzzy ontologies model for semantic web*, Proceeding of the Second International Conference on Information, Process, and Knowledge Management, St. Maarten, 2010.

[2] G. Bagschik, T. Menzel, M. Maurer, *Ontology based scene creation for the development of automated vehicles*, Proceeding of the IEEE Intelligent Vehicles Symposium (IV), Changshu, (2018), 1813-1820.

[3] A. Bahri, R. Bouaziz, F. Gargouri, *Fuzzy ontology implementation and query answering on databases*, Proceeding of the Annual Meeting of the North American Fuzzy Information Processing Society, Cincinnati, Ohio, Jun 2009.

[4] F. Bobillo, J. Gómez-Romero, P. L. Araúz, *Fuzzy ontologies for specialized knowledge representation in WordNet*, Communications in Computer and Information Science, Springer Berlin Heidelberg, (2012), 430-439.

[5] F. Bobillo, U. Straccia, *Fuzzy ontology representation using OWL 2*, International Journal of Approximate Reasoning, **52** (2011), 1073-1094.

[6] F. Bobillo, U. Straccia, *The fuzzy ontology reasoner fuzzy DL*, Knowledge-Based Systems, **95** (2016), 12-34.

[7] J. A. Cabrera, J. J. Castillo, E. Carabias, A. Ortiz, *Evolutionary optimization of a motorcycle traction control system based on fuzzy logic*, IEEE Transactions on Fuzzy Systems, **23** (2015), 1594-1607.

[8] H. Cheng, P. Zeng, L. Xue, Z. Shi, P. Wang, H. Yu, *Manufacturing ontology development based on industry 4.0 demonstration production line*, Proceeding of the Third International Conference on Trustworthy Systems and their Applications, Wuhan, China, Sept., (2016), 42-47.

[9] M. Dehghani, M. Ghiasi, T. Niknam, A. Kavousi-Fard, M. Shasadeghi, N. Ghadimi, F. Taghizadeh-Hesary, *Blockchain-based securing of data exchange in a power transmission system considering congestion management and social welfare*, Sustainability, **13** (2020), 90.

[10] D. Dubois, J. Mengin, H. Prade, *Chapter 6 possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion*, Fuzzy Logic and the Semantic Web, **1** (2006), 101-113.

[11] S. El-Sappagh, M. Elmogy, *A fuzzy ontology modeling for case base knowledge in diabetes mellitus domain*, Engineering Science and Technology, an International Journal, **20** (2017), 1025-1040.

[12] S. Farrar, W. Lewis, T. Langendoen, *A common ontology for linguistic concepts*, Proceeding of the Knowledge Technology Conference, Seattle, Washington, (2002), 1-9.

[13] S. M. Hosseini, M. Manthouri, *Type 2 adaptive fuzzy control approach applied to variable speed dfig based wind turbines with mppt algorithm*, Iranian Journal of Fuzzy Systems, **19**(1) (2022), 31-45.

[14] J. Huang, M. Ri, D. Wu, S. Ri, *Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum*, IEEE Transactions on Fuzzy Systems, **26** (2018), 2030-2038.

[15] S. S. Izquierdo, L. R. Izquierdo, *Mamdani fuzzy systems for modeling and simulation: A critical assessment*, Journal of Artificial Societies and Social Simulation, **21**(3) (2018), 1-18.

[16] B. Jerbic, T. Stipancic, T. Tomasic, *Robotic bodily aware interaction within human environments*, Proceeding of the Intelligent Systems Conference, London, UK, (2015), 305-314.

[17] A. A. Kalat, V. Mokhtari, *Robust output feedback adaptive sliding mode control for a class of uncertain nonlinear systems using robust adaptive fuzzy observer*, Iranian Journal of Fuzzy Systems, **18**(1) (2021), 171-183.

[18] C. M. Keet, A. Ławrynowicz, C. d'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, M. Hilario, *The data mining OPtimization ontology*, Journal of Web Semantics, **32** (2015), 43-53.

[19] H. Knublauch, R. W. Fergerson, N. F. Noy, M. A. Musen, *The protégé OWL plugin: An open development environment for semantic web applications*, The Semantic Web – ISWC, Springer Berlin Heidelberg, (2006), 229-243.

[20] J. Liu, C. Chen, Z. Liu, K. Jermsittiparsert, N. Ghadimi, *An IGDT-based risk-involved optimal bidding strategy for hydrogen storage-based intelligent parking lot of electric vehicles*, Journal of Energy Storage, **27** (2020), 101057.

[21] G. Manogaran, P. M. Shakeel, S. Baskar, C. H. Hsu, S. N. Kadry, R. Sundarasekar, P. M. Kumar, B. A. Muthu, *FDM: Fuzzy-optimized data management technique for improving big data analytics*, IEEE Transactions on Fuzzy Systems, **29** (2021), 177-185.

[22] A. Meghdari, M. Alemi, *Recent advances in social and cognitive robotics and imminent ethical challenges*, Proceedings of the 10th International RAIS Conference on Social Sciences and Humanities, Princeton, USA, (2018), 75-82.

[23] M. Mehrpooya, N. Ghadimi, M. Marefati, S. A. Ghorbanian, *Numerical investigation of a new combined energy system includes parabolic dish solar collector, stirling engine and thermoelectric device*, International Journal of Energy Research, **45** (2021), 16436-16455.

[24] K. A. Naik, C. P. Gupta, *Type-2 fuzzy logic based pitch angle controller for fixed speed wind energy system*, Iranian Journal of Fuzzy Systems, **17**(1) (2020), 77-90.

[25] D. Pavković, S. Polak, D. Zorc, *PID controller auto-tuning based on process step response and damping optimum criterion*, ISA Transactions, **53** (2014), 85-96.

[26] N. D. Rodríguez, M. P. Cuéllar, J. Lilius, M. D. Calvo-Flores, *A fuzzy ontology for semantic modeling and recognition of human behavior*, Knowledge-Based Systems, **66** (2014), 46-60.

[27] P. Šarac, *Design and implementation digital pid-type controllers for thermal chamber temperature control*, Master's Thesis, University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, 2020.

[28] A. Selvaraj, S. Sundararajan, *Evidence-based trust evaluation system for cloud services using fuzzy logic*, International Journal of Fuzzy Systems, **19** (2016), 329-337.

[29] T. Stipancic, B. Jerbic, P. Curkovic, *A context-aware approach in realization of socially intelligent industrial robots*, Robotics and Computer-Integrated Manufacturing, **37** (2016), 79-89.

[30] U. Straccia, *Chapter 4 a fuzzy description logic for the semantic web*, Fuzzy Logic and the Semantic Web, Elsevier, (2006), 73-90.

[31] U. Straccia, *Foundations of fuzzy logic and semantic web languages*, Taylor and Francis Ltd., 2016.

[32] P. Szwed, *Application of fuzzy ontological reasoning in an implementation of medical guidelines*, Proceeding of the 6th International Conference on Human System Interactions (HSI), Sopot, (2013), 342-349.

[33] C. Thomas, A. Sheth, *Chapter 1 on the expressiveness of the languages for the semantic web — making a case for "a little more"*, Fuzzy Logic and the Semantic Web, Elsevier, (2006), 3-20.

[34] Z. Yang, M. Ghadamyari, H. Khorramdel, S. M. S. Alizadeh, S. Pirouzi, M. Milani, F. Banihashemi, N. Ghadimi, *Robust multi-objective optimal design of islanded hybrid system with renewable and diesel sources/stationary and mobile energy storage systems*, Renewable and Sustainable Energy Reviews, **148** (2021), 111295.

[35] H. Ye, G. Jin, W. Fei, N. Ghadimi, *High step-up interleaved dc/dc converter with high efficiency*, Energy Sources, Part A: Recovery Utilization, and Environmental Effects, (2020), 1-20.

[36] F. Zhang, J. Cheng, Z. Ma, *A survey on fuzzy ontologies for the semantic web*, The Knowledge Engineering Review, **31** (2016), 278-321.

[37] F. Zhang, P. Huang, *Fuzzy-based adaptive super-twisting sliding-mode control for a maneuverable tethered space net robot*, IEEE Transactions on Fuzzy Systems, **29**(7) (2020), 1739-1749.