# Dynamic Fuzzy Force Field Based Force-Feedback for Collision Avoidance in Robot Manipulators

D. Wijayasekara, M. Manic
Department of Computer Science
University of Idaho
Idaho Falls, USA
dumidu.wijayasekara@gmail.com, misko@ieee.org

***Abstract*—Advanced remote teleoperation of robot manipulators enable complex tasks to be performed in hostile or inaccessible environments, without the physical presence of a human. For increased effectiveness of teleoperation, maintaining accuracy and speed of task while minimizing collisions is important. Visual and auditory inputs to the user aid in accurate control. However, to further increase the speed and accuracy, tactile and kinesthetic force-feedback information can be used. One of the most common methods of force-feedback generation is the virtual force field based method. However, in complex environments where increased accuracy is required, static force field based methods are insufficient. This paper presents a dynamically varying, virtual force field based force-feedback generation method for obstacle avoidance in remotely operated robot manipulators. The presented method utilizes a fuzzy logic model to dynamically vary a virtual force field surrounding the manipulator in real-time. The fuzzy controller utilizes the distance vectors to obstacles and the velocity vectors of the manipulator components to generate the force field in each axis. The generated force field is then used to calculate the final force-feedback that is sent to the user. The presented method was implemented on a simple 3-DOF robot manipulator, and compared to a typical static force field based force-feedback generation method. Test results show that the task completion time is significantly improved without significant loss of accuracy in certain tasks when the presented force-feedback method is used.**

***Keywords—Force feedback; Remote teleoperation; Robotics; Fuzzy control, Haptics***

## I. Introduction

Robot teleoperation entails controlling robots and interacting with the environment from a remote site, without direct physical contact. The primary advantage of teleoperation is the ability of controlling a robot in a situation where it is unsafe, difficult or inconvenient for a human to be physically present at the location [1], [2]. Furthermore, by using teleoperation extremely precise and accurate control can be achieved when the task of the robot is dynamic and complex such that it is difficult to complete autonomously [1], [3].

Minimizing collisions that may lead to adverse situations as well as equipment damage in these high complexity tasks is difficult. Thus, these tasks require precise movements and have low threshold for deviation. Therefore, minimizing collisions while maintaining high levels of accuracy and task completion times speed is essential [4].

In order to achieve these goals, providing accurate and useable information to the user about the robot position and orientation as well as the working environment is critical for successful and effective teleoperation [4]. The most widely used methods of information presentation to the user are visual and auditory [5], [6]. However, visual and auditory information might not be sufficient in many cases [7], [8].

Difficulty in modeling accurate information and the need for specialized devices makes it difficult to provide information in addition to audio visual information [5]. However, providing tertiary information via the sense of touch, known as haptics, has gained much interest in recent years [4], [6], [7], [9], [10]. Typically, in haptic applications, a haptic device which is a bi-directional human interface that provides sensory input to the user via the sense of touch while providing control inputs to the machine is used [5], [10]-[12]. Two types of haptic devices exist: tactile and kinesthetic [9], [13]. Tactile devices are based on sense of touch and enable the operator to feel textures, friction and rubbing forces, and consistency of objects [7], [9]. Kinesthetic devices reflect forces and increase the state awareness of operators about the work space [9].

Thus haptics has been used in a wide area of robot remote teleoperation tasks [14] such as teleoperation of mobile robots [15] operating industrial robotic manipulators [16], remote non-invasive surgery [17], [18], path planning [19] and virtual sculpting [10], [20]. Furthermore, improvement in time and accuracy of task completion has been shown by utilizing haptics as an additional sensory input to the operator [4], [6], [13], [17], [21].

Virtual force field based force-feedback generation where a force field is modeled surrounding the objects and the manipulator is used most commonly used for haptic applications [9], [11], [19]. Accurate physics based models [22] and mass-spring models where virtual springs surrounding obstacles are modeled [23] have also been used for collision avoidance while providing kinesthetic feedback. PD and PID control methods with varying gains have also been explored to provide accurate force-feedback to the user for collision avoidance [23]-[26].

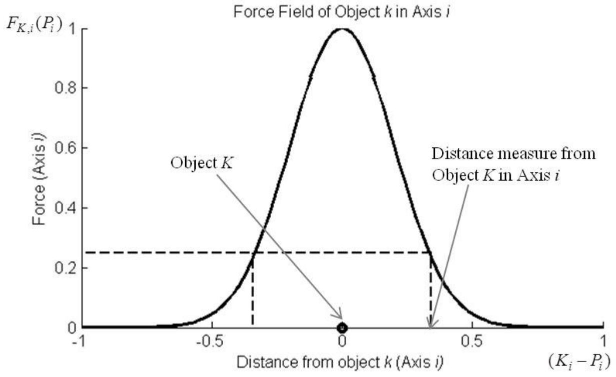


Fig. 1 Virtual force field representation for object *K*

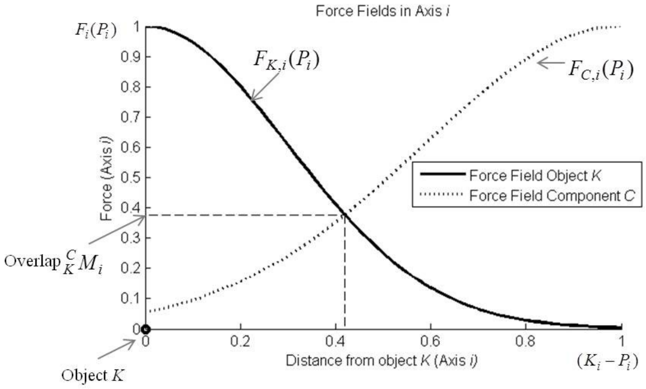


Fig. 2 Virtual force field overlap calculation (object *K* and component *C*)

When small precise movements are required from the robot manipulator, the force-feedback should be adjusted for easier maneuverability while maintaining state awareness. Static force field based methods, however, lack the flexibility required when operating in such conditions. Therefore, this paper presents a dynamic fuzzy logic based force field generation method for providing kinesthetic feedback to operator, thereby maintaining the state awareness of the operator while improving the accuracy of operation. The presented method utilizes the distance to obstacles and the speed of the robotic manipulator to dynamically vary the force field of the manipulator components to suit the environment and type of movement required for the situation. The presented method was compared to a static force field based method using a simple 3-Degree of Freedom (DOF) robotic manipulator. The experimental results show a significant improvement task completion time while maintaining the consistency of the task using the presented method.

This paper is organized as follows; Section II describes virtual force field based force-feedback generation. Section III details the presented dynamic force field based feedback generation method. Section IV presents the hardware and software implementation. Section V presents the experimental results while section VI concludes the paper.

## II. Virtual Force Field based Force Field for Force-Feedback Calculation

This section first details the virtual force field representation and then discusses the force field based force-feedback calculation.

### *A. Virtual Force Field Representation*

Virtual force field has been widely used as a method for generating force-feedback for obstacle avoidance in robot teleoperation applications [9], [11], [19]. In this paper the virtual force field of a given object *K*, is represented by a Gaussian function for each axis, centered at *K*. Thus the force field of object *K* in axis *i* ( $F_{K,i}$ ) can be expressed as:

$$F_{K,i}(P_i) = \exp\left(-\frac{(K_i - P_i)^2}{2\sigma^2}\right) \tag{1}$$

where, $K_i$ is the position of object *K* in axis *i* and $P_i$ is a point in axis *i*. The value $\sigma$ expresses the spread of the force field and can be assigned according to prior knowledge about the importance and sensitivity of the object. A force field where $\sigma = 0.2$ is shown in Fig. 1.

Since the Gaussian function is asymptotically approaching 0, in order to calculate the distance from an object *J* to the force field of object *K* ( $J \neq K$ ), a force threshold, $\delta_F$ is used:

$$\begin{aligned} T_{K,i} = \{&\min(P_i) \mid F_{K,i}(P_i) \geq \delta_F, \\ &\max(Q_i) \mid F_{K,i}(Q_i) \geq \delta_F\} \end{aligned} \tag{2}$$

where, $P_i$ and $Q_i$ are points in axis *i*, and $T_{K,i}$ is the distance threshold for object *K* in axis *i* (See Fig. 1).

### *B. Force-Feedback Calculation*

The force-feedback for each time step *t* is generated using the force fields of the obstacles and the robot manipulator at time step $t+1$. The force fields of the obstacles are assumed to be static (constant $\sigma$ ) and represented using the method described above. The force field of each component of the manipulator is dynamically generated at each time step using the method described in Section III.

Once the force fields are dynamically generated for all components of the manipulator, the forces acting upon each component of the manipulator for each axis are generated. This is done using the maximum overlap of component and obstacle force fields (See Fig. 2). The overlap between object *K* and component *C* in axis *i*, ${}^{C}_{K}M_i$ can be defined as:

$${}^{C}_{K}M_i = \max(F_{K,i}) \mid F_{K,i}(P_i) = F_{C,i}(P_i) \tag{3}$$

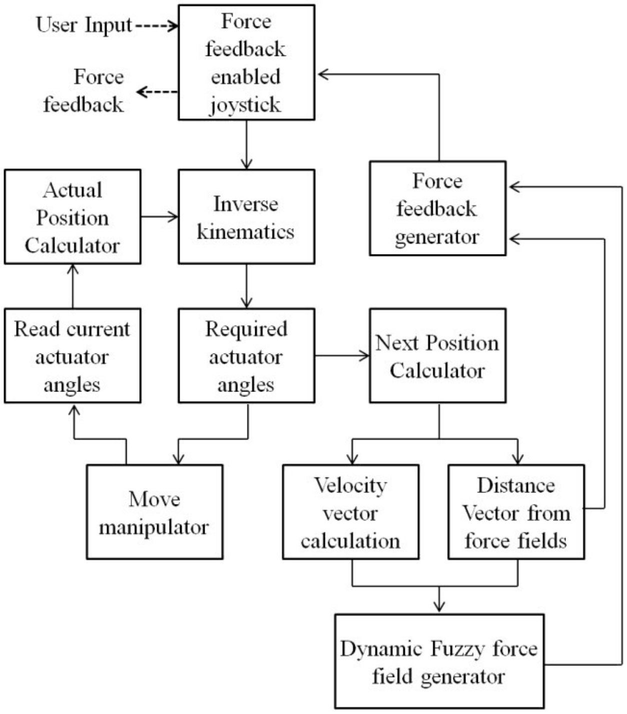

User Input
Force feedback
Force feedback enabled joystick
Actual Position Calculator
Inverse kinematics
Force feedback generator
Read current actuator angles
Required actuator angles
Next Position Calculator
Move manipulator
Velocity vector calculation
Distance Vector from force fields
Dynamic Fuzzy force field generator

Fig. 3 Overall framework of the presented dynamic fuzzy force field generator for obstacle collision avoidance

Thus, for each component in the manipulator, the overlap of force fields for each obstacle is calculated. For a given axis *i*, the force exerted by an obstacle *K* on the manipulator component *C* is thus proportional to the obstacle-component force field overlap in axis *i*.

$$ {}^{C}_{K}\hat{F}_i \propto {}^{C}_{K}M_i \tag{4}$$

where, ${}^{C}_{K}\hat{F}_i$ is the force exerted by obstacle *K* on component *C* in the axis *i*.

The forces acting on a given component for all obstacles are calculated and the final force for a component for axis *i* is the sum of all forces acting upon that component for axis *i*.

$$ {}^{C}\tilde{F}_i = \sum_{k=1}^{N} {}^{C}_{k}\hat{F}_i \tag{5}$$

where, ${}^{C}\tilde{F}_i$ is the force acting upon component *C* and *N* is the number of obstacles in the work space.

Finally the force acting upon each component for a given axis *i*, is aggregated and sent to the force-feedback input device. The forces are aggregated using a weight that is proportional to the velocity of that component, thus exerting more force to the user for components that are moving faster.

$$ \breve{F}_i = \sum_{c=1}^{M} {}^{c}\hat{F}_i \cdot w_{c,i} \tag{6}$$

where, $\breve{F}_i$ is the force sent to the user via the force-feedback device in axis *i*. *M* is the number of components in the manipulator, and $w_{c,i} \propto V_{c,i}$ where $V_{c,i}$ is the velocity of component *C* in axis *i*.

## III. Dynamic Fuzzy Force Field for Force-Feedback Generation

This section first details the overall architecture of the presented dynamic force field based force-feedback generation method and then discusses the fuzzy logic based system in detail.

### *A. Presented system*

The overall framework of the presented method is detailed in Fig. 3. A force-feedback enabled joystick device is used as the input device as well as providing kinesthetic feedback to the user. The position and size of the obstacles in the operating area is assumed to be known. As mentioned in Section II, each obstacle is surrounded by a static virtual force field.

The user inputs the desired location of the manipulator using the force-feedback enabled joystick device. Once a movement of the joystick is made at time *t*, the required actuator angles for the desired movement of the manipulator at time $t+1$, are calculated. In this paper, the actuator angles are calculated using inverse kinematics. Using the generated angles, the position of each component of the manipulator at time $t+1$ is calculated. This is performed before physically moving the manipulator.

Using the calculated position of the manipulator at time $t+1$, and the force fields of obstacles, the closest distance from the manipulator components to the obstacle force fields is calculated. This calculation is done for each axis separately. Thus for each axis *x*, *y* and *z*, the minimum distance from each of the manipulator components to an obstacle force field is calculated.

Furthermore, using the position of the manipulator components at time *t* and time $t+1$, the velocity vector for each component is also calculated.

Thus, for each of the components in the manipulator, for each axis, the minimum distance to an obstacle force field, and the velocity is known. This distance and velocity pair is then passed on to the appropriate fuzzy force field generation system. The fuzzy system then generates the appropriate force field of each component for each axis for time $t+1$. The fuzzy system and inference process is detailed in sub-section III.B.

The generated force fields of components for time $t+1$, and the force fields of obstacles is used to generate the force-feedback for time $t+1$, for each axis using the method described in Section II.

Once the force-feedback for each axis is generated it is sent to the user via the force-feedback enabled joystick device as a kinesthetic input. Simultaneously the manipulator is moves to the desired position for time $t+1$. Once the manipulator is moved, for increased accuracy, the actual angles of the

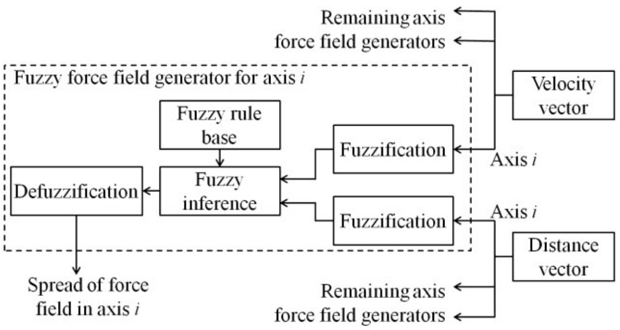

Remaining axis force field generators
Fuzzy force field generator for axis *i*
Fuzzy rule base
Fuzzification
Velocity vector
Axis *i*
Defuzzification
Fuzzy inference
Fuzzification
Axis *i*
Distance vector
Spread of force field in axis *i*
Remaining axis force field generators

Fig. 4 Fuzzy inference system for dynamic force field generation

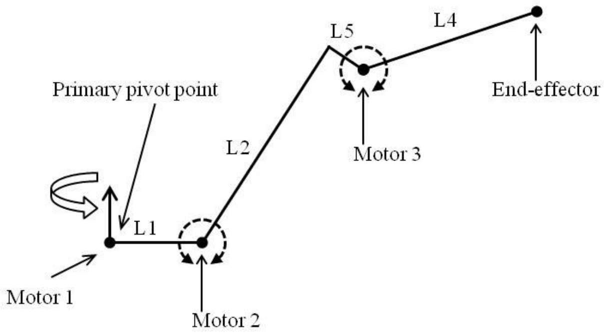


Fig. 5 Schematic of the implemented 3-DOF robotic manipulator

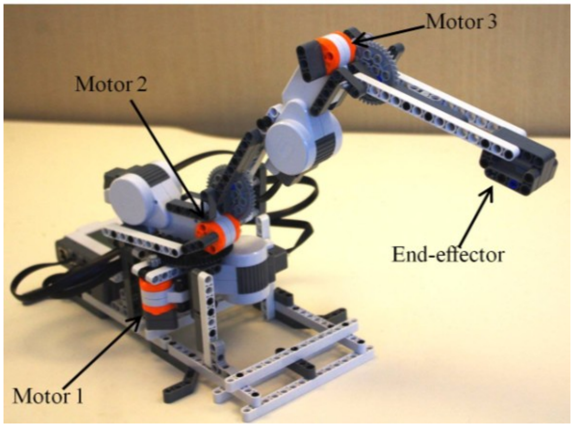


Fig. 6 The implemented 3-DOF robotic manipulator using Lego NXT

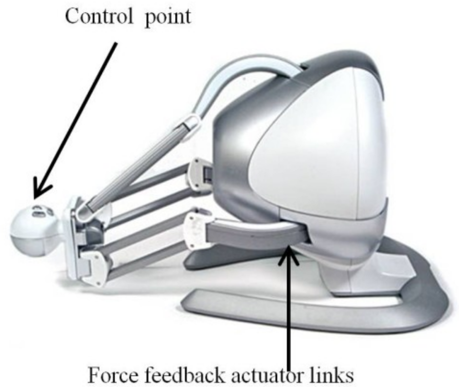


Fig. 7 The Novint Falcon 3-DOF force feedback joystick device

actuators are measured and the position in recalculated and these values will be used for inverse kinematics in time step $t+2$ .

### B. *Fuzzy Logic Based Force Field Generation*

The fuzzy logic based force field generator utilizes the generated distance vector at time $t+1$ and the velocity vector between time $t$ and $t+1$ to dynamically update the force fields of each component.

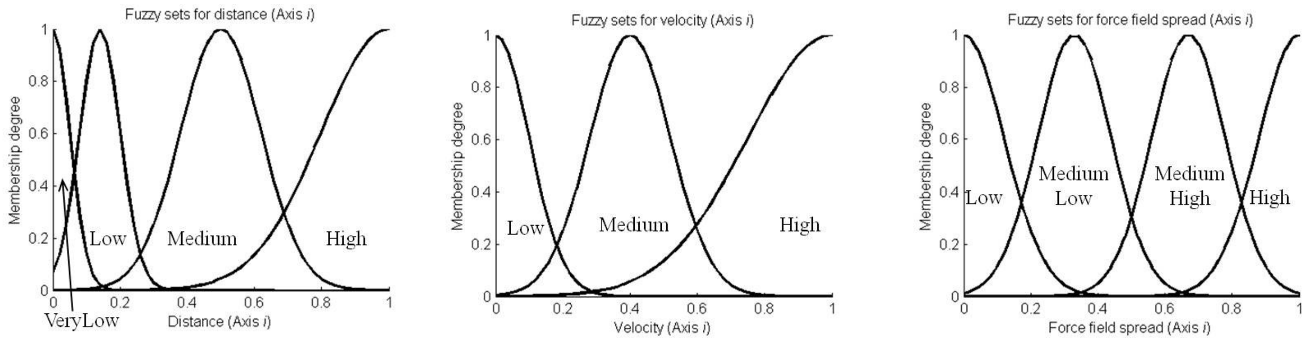
The force field is dynamically updated for components to enable dynamic force-feedback according to the environment and movement speed. This is achieved by weighing the spread of the Gaussian function ( $\sigma$ in (1)) of manipulator component force fields.

At higher speeds of movement, the likelihood of collisions increase. Furthermore, earlier warning is required for the user to react when the actuator is moving faster. Thus the fuzzy system increases the spread of the force field at higher manipulator speeds. When obstacles are further away from the manipulator the spread of the force field can be increased for increased awareness of the environment, without affecting the accuracy.

However, for precise and accurate movements in constricted environments, the larger force field is unsuitable as it yields higher force-feedback that increases the difficulty of precise movements. Thus at lower speeds and with closer obstacles, the spread of the component force fields are reduced, thus enabling movements with higher accuracy, with minimal loss of state awareness.

For axis *x, y* and *z*, a separate force field generator is used, which generates the force field on that axis for the specific component. Each force-feedback generator takes the axis component of the velocity vector and the axis component of the distance vector. The axis component of the distance vector is the distance from the manipulator to the closest obstacle in that axis. The basic block diagram of the fuzzy inference process is depicted in Fig. 4.

By utilizing separate controller for each axis, the dynamic behavior of the force field in that axis can be controlled differently. Differences in fuzzy controllers entail different rule bases as well as different fuzzy sets that granulizes the input and output spaces differently. For example, a manipulator with limited movement in one axis can have a different force field generator for that axis. Similarly, for manipulator components that behave differently, different force field generators can be used.



(a) (b) (c)

Fig. 8 Input and output fuzzy sets used: (a) input distance, (b) input velocity, (c) output (force field spread)

TABLE I
FUZZY RULE BASE FOR AXIS *I*

| Velocity axis *i* / Distance axis *i* | **Low** | **Medium** | **High** |
|---|---|---|---|
| **Very Low** | Medium Low | Medium Low | Medium Low |
| **Low** | Low | Medium Low | Medium Low |
| **Medium** | Medium Low | Medium High | Medium High |
| **High** | Medium High | Medium High | High |

For space considerations the steps of the fuzzy inference process will not be detailed in this paper. Specific implementation used for this paper is detailed in Section IV.

## IV. IMPLEMENTATION

This section details the specific hardware and software implementation of the presented method.

### A. *Hardware Implementation*

A simple 3-DOF robot manipulator with 3 actuators that was implemented using Lego NXT [27] and utilized in [4] was used in this paper for testing. The schematic of the implemented manipulator is depicted in Fig. 5. Fig. 6 shows the actual implemented robot. Inverse kinematics was used to derive the angles of the actuators for a given end-effector position. The actuator angles can be read via the NXT interface for calculating the actual position of the robot after a movement has been made.

As the force-feedback enabled joystick device, the Novint Falcon device (Fig. 7) was selected [28], [29]. The Novint Falcon device has 3 degrees of freedom and 3 actuators work in conjunction to provide powerful and accurate kinesthetic or tactile feedback to the user, while maintaining highly accurate control.

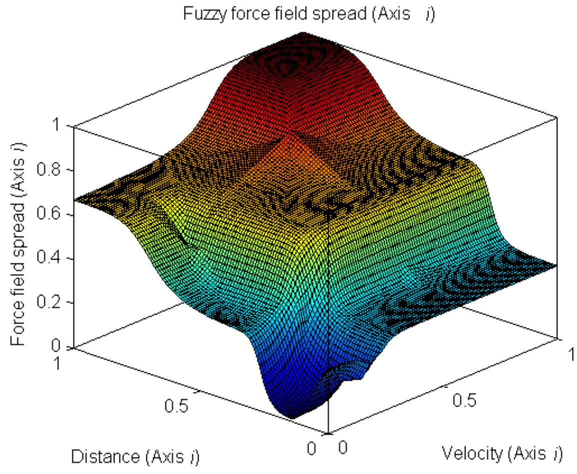


Fig. 9 Output force field spread surface

### B. *Software Implementation*

As mentioned in Section III, the force field of each axis in the control space is controlled by a separate fuzzy controller. Each controller can be customized according to the freedom of movement in that axis. For the sake of simplicity and space, in this paper, the same fuzzy system (fuzzy rules and fuzzy sets) was used for each axis.

Gaussian membership functions were used for representing each of the input and the output dimensions. Fig. 8 shows the input fuzzy sets for the distance (Fig. 8(a)), velocity (Fig. 8(b)), and the output force field spread (Fig. 8(c)). The fuzzy rule base utilized is shown in Table I. The force field spread surface generated by the using the fuzzy sets and the rule base is depicted in Fig. 9.

As shown by Fig. 9, at low distances and speeds the force field spread is reduced to enable more accurate movements. As the velocity increases the spread is also increased. Similarly, with increasing distances to obstacles, the spread of the force field is increased so that the state awareness of the user is increased. At very low distances the spread of the force field is increased to avoid collisions.

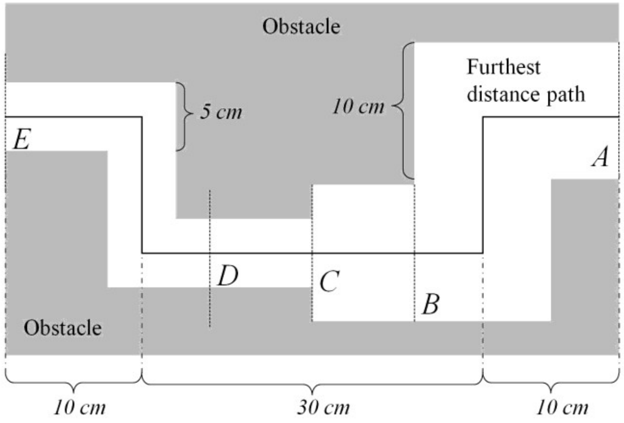


Fig. 10 Experimental task for comparison

The fuzzy sets and rules were selected by testing the effectiveness of the fuzzy system on the implemented hardware setup several times. As mentioned, the fuzzy system for dynamically changing the force field spread can be implemented according to the specific application.

## V. Experimental Results

A task using the implemented system detailed in section IV was setup for evaluating the presented dynamic force field based feedback method. The presented method was compared to a typical system where the force fields are static. The task was completed 10 times using each method and the results were averaged.

The selected task comprised of moving the end effector of the manipulator through an area surrounded by an obstacle and is shown in Fig. 10. The end effector is entered into the obstacle area starting at point *A* (Fig. 10). At this point the width of the opening is 10 cm. At point *C*, the width of the opening is reduced to 5 cm simulating a constricted work area. The work area was divided in to three areas: 1) *A-B*: lower restriction allowing more freedom of movement, 2) *B-D*: transitioning from a lower restriction area to a higher restriction area, and 3) *D-E*: high restriction area where more precision of movement is required.

The time to complete each area as well as the complete task was recorded. Furthermore, the accuracy of the movement of the end effector was calculated as a measure of distance from the preset furthest distance path shown in Fig. 10. This was calculated by dividing the accumulated closest distance to the furthest distance path at a given time by the number of time intervals.

Tables II and III present the averaged time taken to complete the task and the averaged accuracy for the two methods tested, respectively. The accuracy and time taken to complete the less restricted area *A-B* is comparable in both methods. However, when using a static force field, the time taken for moving within the more restricted areas *B-D* and *D-E* is significantly increased. In the static case, the larger force field leads to increased forces acting upon the user. Thus movement in the more restricted area is difficult which leads to increased completion times.

TABLE II
Average task completion time for the two methods tested (s)

| **Method tested** | **Measure** | **A-B** | **B-D** | **D-E** | **Total** |
|---|---|---|---|---|---|
| Presented (dynamic force field based) | Mean | 7.26 | 2.24 | 7.58 | 17.08 |
| | SD | 0.24 | 0.16 | 0.23 | 0.31 |
| Typical (Static force field based) | Mean | 7.28 | 3.76 | 9.6 | 20.64 |
| | SD | 0.21 | 0.77 | 0.4 | 1.14 |

TABLE III
Average accuracy compared to furthest path for the two methods tested (cm)

| **Method tested** | **Measure** | **A-B** | **B-D** | **D-E** | **Total** |
|---|---|---|---|---|---|
| Presented (dynamic force field based) | Mean | 2.08 | 0.84 | 0.96 | 3.88 |
| | SD | 0.08 | 0.17 | 0.11 | 0.19 |
| Typical (Static force field based) | Mean | 2.06 | 0.82 | 0.88 | 3.76 |
| | SD | 0.16 | 0.22 | 0.13 | 0.21 |

The accuracy of the presented method is slightly reduced in the areas *B-D* and *D-E*. This is because in the presented method the force field spread is reduced in this area allowing more free movement.

## VI. Conclusion

This paper presented a dynamically varying, virtual force field based force-feedback generation method control for obstacle collision avoidance in remotely operated robot manipulators. The presented method utilizes a fuzzy logic model to dynamically vary a virtual force field surrounding the manipulator in real-time. Distance vectors to obstacles and velocity vectors of the manipulator components are used by the fuzzy controller to dynamically vary the force field of components in each axis.

The presented method was implemented and tested using a simple 3-DOF robot manipulator coupled to a commercially available 3-DOF force-feedback enabled joystick. The presented method was compared to a typical non-dynamic force field based force-feedback generation method. Test results show that the task completion time is significantly decreased without significant loss of accuracy using the presented method when the obstacle distances are changed.

Future work entails identifying the added computational complexity of the presented method compared to typical non-dynamic force field based methods. Furthermore, the presented method will be tested in more complex environments performing real-world tasks. Finally, the presented method can be further extended to include other factors such as friction forces and the manipulator position error forces for more accurate control.

# REFERENCES

[1] H. Hu, J. Li, Z. Xie, B. Wang, H. Liu, G. Hirzinger, “A robot arm/hand teleoperation system with telepresence and shared control,” in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, pp. 1312-1317, 2005.

[2] A. Bartl, M. Diaz-Cacho, A. Barreiro, E. Delgado, “Passivity framework and traffic reduction for the teleoperation of a gantry crane,” in *Proc. of Annual Conf. of the IEEE Ind. Eleclectronics Society, (IECON)*, pp. 3675-3680, Nov. 2013.

[3] H. Saafi, M. A. Laribi, S. Zeghloul, M. Y. Ibrahim, “Development of a spherical parallel manipulator as a haptic device for a tele-operation system: Application to robotic surgery,” in *Proc. of Annual Conf. of the IEEE Ind. Eleclectronics Society, (IECON)*, pp. 4097-4102, Nov. 2013.

[4] D. Wijayasekara, M. Manic, “Fuzzy Logic Based Force-Feedback for Obstacle Collision Avoidance of Robot Manipulators,” Accepted for publication in *IEEE Int. Conf. on Human System Interaction (HSI)*, Jun. 2014.

[5] T. Kwon, J. Song, “Force display using a hybrid haptic device composed of motors and brakes,” in *Mechatronics*, vol. 16, no. 5, pp. 249-257, June 2006.

[6] B. Osafo-Yeboah, S. Jiang, R. Delpish, Z. Jiang, C. Ntuen, “Empirical study to investigate the range of force feedback necessary for best operator performance in a haptic controlled excavator interface,” in *International Journal of Industrial Ergonomics*, vol. 43, no. 3, pp. 197-202, May 2013.

[7] T. Miyagi, S. Katsura, “Frequency model of haptic information in rubbing motion,” in *Proc. of Annual Conf. of the IEEE Ind. Eleclectronics Society, (IECON)*, pp. 4132-4137, Nov. 2013.

[8] C. Antonya, “Force Feedback in String based Haptic Systems,” Procedia Computer Science, vol. 25, pp. 90-97, 2013.

[9] P. García-Robledo, J. Ortego, M. Ferre, J. Barrio, M. Sánchez-Urán, “Segmentation of Bimanual Virtual Object Manipulation Tasks Using Multifinger Haptic Interfaces,” in *IEEE Trans. on Instrumentation and Measurement*, vol. 60, no. 1, pp. 69-80, Jan. 2011.

[10] A. Peer, M. Buss, “A New Admittance-Type Haptic Interface for Bimanual Manipulations,” in *IEEE/ASME Trans. on Mechatronics*, vol. 13, no. 4, pp. 416-428, Aug. 2008.

[11] T. Endo, M. Yasue, H. Kawasaki, “Experimental investigation of a collision avoidance controller for a bimanual multi-fingered haptic interface,” in *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2664-2677, Nov. 2013.

[12] H. Park, J. Lee, “Adaptive impedance control of a haptic interface,” in *Mechatronics*, vol. 14, no. 3, pp. 237-253, Apr. 2004.

[13] J. Gwilliam, M. Mahvash, B. Vagvolgyi, A. Vacharat, D. Yuh, A. Okamura, “Effects of haptic and graphical force feedback on teleoperated palpation,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 677-682, May 2009.

[14] K. Okiyama, T. Murakami, “Integration of robot task and human skill with bilateral control,” in *Proc. of Annual Conf. of the IEEE Ind. Eleclectronics Society, (IECON)*, pp. 5882-5887, Nov. 2013.

[15] O. Linda, M. Manic, “Fuzzy force-feedback augmentation for manual control of multirobot system,” in *IEEE Trans. on Industrial Electronics*, vol. 58, no. 8, pp. 3213-3220, 2011.

[16] P. Chotiprayanakul, D. Wang, N. Kwok, D. Liu, “A haptic base human robot interaction approach for robotic grit blasting,” in *Proc. Int. Symp. on Automation and Robotics in Construction, (ISARC)*, pp. 148-154, June 2008.

[17] G. Tholey, J. P. Desai, A. E. Castellanos, “Force feedback plays a significant role in minimally invasive surgery,” in *Annals of Surgery*, vol. 241, no. 1, pp. 102-109, 2005.

[18] T. P. James, J. J. Pearlman, A. Saigal, “Predictive force model for haptic feedback in bone sawing,” in *Medical Engineering & Physics*, vol. 35, no. 11, , pp. 1638-1644, Nov. 2013.

[19] X. He, Y. Chen, “Haptic-aided robot path planning based on virtual tele-operation,” in *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 5, pp. 792-803, Oct. 2009.

[20] K. Laehyun, G. S. Sukhatme, M. Desbrun, “A haptic-rendering technique based on hybrid surface representation,” in *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 66-75, 2004.

[21] M. A. Srinivasan, C. Basdogan, “Haptics in virtual environments : taxonomy, research status, and challenges,” in *Comput . & Graphics*, vol. 21, no. 4, pp. 393-404, 1997.

[22] G Gonzalez-Badillo, I. H. Medellin-Castillo, T. Lim, “Development of a Haptic Virtual Reality System for Assembly Planning and Evaluation,” in *Procedia Technology*, vol. 7, pp. 265-272, 2013.

[23] W. Chen, Q. Zhu, Y. Chen, “Research on the fuzzy PID-based control of haptic system,” in *Proc. IEEE Int. Conf. on Computer Science and Network Technology*, vol. 2, pp. 1000-1003, 2011.

[24] T. Xinxing, Y. A. M. A. D. A. Hironao, A. A. Yusof, “Virtual reality-based master-slave control system for construction tele-operation robot,” in *Proc. IEEE Int. Conf. on Intelligent Computing and Intelligent Systems*, vol. 2, pp. 823-827, 2009.

[25] G. Wen, H. Zhang, “Research on improved force feedback control method for construction telerobot,” in *Proc. IEEE Int. Conf. on Intelligent Computing and Intelligent Systems*, vol. 2, pp. 57-61, 2010.

[26] V. Minh Hung; T. Quang Trung, “Modeling and impedance control of a 6-DOF haptic teleoperation system,” in *Proc. Int. Conf. on Control, Automation and Information Sciences*, pp. 134-139, Nov. 2013.

[27] LEGO MindStorms Webpage [URL], Available: http://mindstorms.lego.com/, from March 2014.

[28] S. Martin, N. Hillier, “Characterization of the Novint Falcon Haptic Device for Application as a Robot Manipulator,” in *Proc. Australasian Conference on Robotics and Automation (ACRA)*, Dec. 2009.

[29] Novint Falcon Webpage [URL], Available: http://home.novint.com/, from March 2014.