Precision Grasp Planning with Gifu Hand III based on Fast Marching Square

David Álvarez, Alejandro Lumbier, Javier V. Gómez, Santiago Garrido and Luis Moreno.
Robotics Lab, Department of Systems and Automation.
University Carlos III of Madrid, Leganés, 28911, Spain.
E-mail: {dasanche, alumbier, jvgomez, sgarrido, moreno}@ing.uc3m.es

Abstract— **This paper presents a novel methodology for planning the movements of a robotic hand when a precision grasp is going to be performed. The approach used is based on the standard Fast Marching Square (FM2) path planning method and its application to robot formations motion planning. In this case, the hand is considered to be a kinematic chain in which a mobile robot is located at every joint position. The robot formation is therefore deformable among the positions allowed by the mechanical limits of the joints. To perform a given precision grasp, the task is divided into two phases. First the hand approaches to the object. FM2 is used to calculate a fast and smooth path towards the object to be grasped. While the hand is covering it, the formation updates its shape according to the map of velocities calculated in FM2. The second phase consists on performing the precision grasp. Every finger is modelled as a robot formation and a path is calculated for each fingertip so that they reach the grasping points on the object. The position of the joints of the fingers is computed using an inverse kinematics algorithm. Simulations show the usefulness of this approach thanks to a good performance of the approaching and the grasping tasks.**

I. INTRODUCTION

Mobile manipulators deal with a wide variety of problems including: autonomous localization and mapping of the environment, path planning and trajectory following or object recognition and manipulation. When robots need to interact with the environmet, among a wide variety of actions they might need to be able to push or pull an object or even grasp and hold it to perform more complex tasks. When object manipulation is necessary, the difficulty of the planning process is usually very high. Although humans can grasp and manipulate objects very easily with their hands, robotic hands developed for research, such as Gifu Hand III [1], Shadow Hand [2] or HIT/DLR [3] often have a complex mechanical structure and a large number of degrees of freedom (DOF). Thus, achieving higher autonomy and dexterity for robotic hands in the manipulation of objects and tools are among the major challenges to be solved [4].

Among the problems found in object manipulation, grasp planning is an essential but very complex one since it involves the combination of open and close kinematic chains and redundant DOF. In order to find a solution, the problem can be divided into smaller tasks such us: given an object in an environment, determine how to approach to it, find a configuration of the hand that leads to a stable grasp and execute the chosen grasping strategy to hold the object.

According to Cutkosky [5], we can make a simple classification of grasping techniques: power grasps that ensure security and stability or precision grasps that emphasise on dexterity and sensitivity. Besides, moving along his taxonomy, one can choose among different kind of grasps depending on the purpose of the manipulation and the size of the objects. Furthermore, in general terms it can be said that while increasing the power of the grasping, there is a loss in the manipulability and dexterity, and the same applies if the relationship is inverted.

In the case of precision grasps, the problem is mainly defined as where and how to place the fingers so that the object is firmly held [6]. This type of grasps are essentially characteraized because they are not meant for manipulating heavy objects since only the fingertips of the hands would be in contact with them. All kind of grasps need to satisfy both dynamic and kinematic constraints. Dynamics because forces created between the hand and the object during the grasping process must be controlled in order to have a stable grasp and do not damage neither the object nor the robot. Besides, trajectories need to meet kinematic constraints since the contact points to be reached must be in the workspace of the fingers of the hand and, at the same time, collisions between the different parts of the hand must be avoided [7].

In this paper a new approach to solve the path planning problem in the case of a robotic hand performing precision grasps is presented. In order to achieve the goal, the problem has been divided into two phases in which two different path planning situations arise: approaching to the object to be grasped and achieving the predefined grasping points by the fingertips of the hand. In both situations, the Fast Marching Square (FM2) path planning method is used and the concept of control of robot formations based on FM2 is also applied. For this reason, the kinematic chain that defines a robotic hand has been considered as a robot formation in which mobile robots are located at the position of its joints and its fingertips and the geometry of the formations is updated based on an artificial potential field.

The next sections of the paper are organized as follows. Section II is a formal description of the Fast Marching Square technique and its application to path planning, in section III the control of robot formation and its application to the grasping problem is presented. In section IV conclusions and future work are addressed.

II. FAST MARCHING AND PATH PLANNING

In this paper, the Fast Marching Method (FMM) has been chosen as path planner. It is a computational algorithm to solve the arrival time of expanding waves in every point of the space. Conceptually, it can be considered as a continuous version of the Dijkstra's algorithm [8]. It is based on the assumption that the information only flows outwards from the wave source. The FM2 method arises from the application of the FMM twice over the same map. The first time is used to create a map of velocities for the environment and the second time computes the time of arrival of the wave for every point when the wave is moving at the velocity computed in the previous step. The FM2 method is very versatile when is applied to motion planning problems.

A. *Fast Marching Method*

The FMM was proposed by J. A. Sethian in 1996 to approximate the solution of the Eikonal equation [9]. Let us assume a 2D map, where $\mathbf{x} = (x, y)$ is a point on the map with the coordinates in relation to a Cartesian referential, $T(\mathbf{x})$ is the frontwave arrival time function and $F(\mathbf{x})$ is the velocity of the wave propagation.

Let us assume that a wave starts propagating at time $T = 0$ with velocity F always non-negative. The Eikonal equation (1) defines the time of arrival of the propagating frontwave, $T(\mathbf{x})$, at each point $\mathbf{x}$, in which the propagation speed depends on the point, $F(\mathbf{x})$, according to:

$$|\nabla T(\mathbf{x})| F(\mathbf{x}) = 1 \tag{1}$$

Discretizing the gradient ∇T according to [10] it is possible to solve the Eikonal equation at each point $p(x_i, y_j)$, where i and j are the row and column of a grid map, as follows:

$$\begin{aligned} T_1 &= min(T_{i-1,j}, T_{i+1,j}) \\ T_2 &= min(T_{i,j-1}, T_{i,j+1}) \end{aligned} \tag{2}$$

$$\left(\frac{T_{i,j} - T_1}{\triangle x}\right)^2 + \left(\frac{T_{i,j} - T_2}{\triangle y}\right)^2 = \frac{1}{F_{i,j}^2} \tag{3}$$

The Fast Marching method consists on solving $T_{i,j}$ for every point of the map starting at the source point of the wave (or waves) where $T_{i_0,j_0} = 0$. The following iterations solve the value $T(i,j)$ for the neighbours of the points solved in the previous one. Using as an input a binary grid map, the output of the algorithm is a map of distances to obstacles as shown in figure 1. These distances are concretely the time or arrival of the expanding wave at every point of the map. FMM can be directly used as a path planner algorithm. By applying gradient descent from any point of the map of distances, a path will be obtained with the source of the wave as a goal point. This is valid only if one wave has been employed to generate the map of distances. Otherwise, local minima will appear. The main advantage of this method is that the path obtained is optimal in distance, like the example in figure 1.

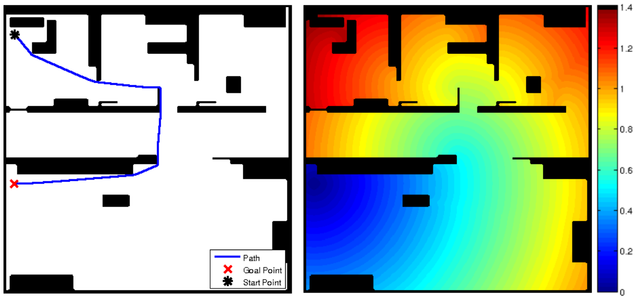


Fig. 1: Example of a path obtained with the FMM. The left side shows the original map and the path calculated. In the right there is the map of distances computed with FMM.

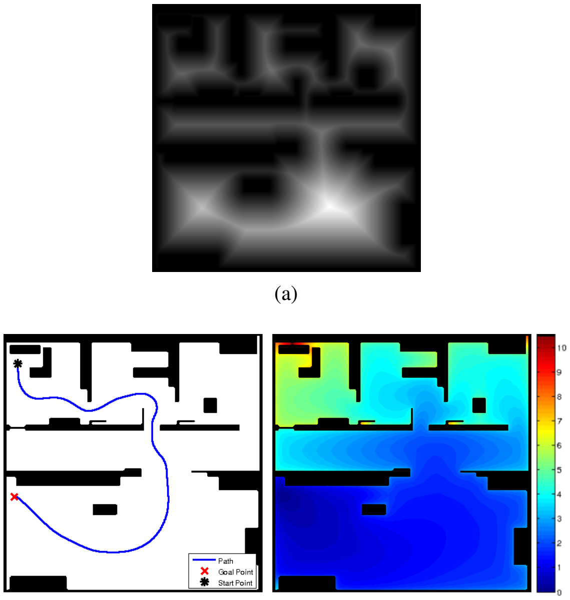
B. *Fast Marching Square Method*

Although the paths provided by the FMM are optimal in distance terms, they do not accomplish the smoothness and safety constraints that most of robotic applications require, since they run too close to obstacles and have sharp curves. In view of these drawbacks, the FMM algorithm is not a good solution in most cases. However, the FM2 algorithm [11] solves these two main disadvantages. It is based on creating a map of velocities in which the velocity of the expanding wave varies depending on the distance to the closest obstacle. The FMM is applied in order to obtain these maps of velocities. In this case, all the occupied positions in the grid are labeled as wave sources. The result is a map of distances in which those cells in the grid that are farther from the obstacles have a higher value (figure 2 a)).

Once the map of distances is computed, it is normalized in order to have values between 0 and 1 and interpreted as relative wave expansion speeds (meaning full stopped or full speed of the wave expansion, respectively). The FMM is applied once again with the goal point as wave source. During the expansion, the wave will propagate with the velocities indicated in the map generated previously. The propagation ends once the initial point of the path is reached. The resulting map of distances will be similar to the one obtained with the standard FMM, but with slight differences that make the paths very smooth when gradient descent is applied, as shown in figure 2. In summary, FM2 applies the FMM twice without any mathematical modification: the first step creates a map of velocities, $F(\mathbf{x})$, and the second step computes the time of arrival function, $T(\mathbf{x})$, in which gradient descent is applied to find the path.

In addition to the smoothness and safety, FM2 has other properties worth to mention:

- *No local minima*: as long as only one wave is employed to generate the map of distances, FMM ensures that there is a single global minimum at the source point of the wave (goal of the path).
- *Completeness*: the method finds a path if it exists and notifies in case of no feasible path.
- *Fast response*: If the environment is static, the map of velocities is calculated only once. Since the FMM can be implemented with a complexity order of $O(n)$ [12], building the map of velocities is a fast process.



(a)

(b)

Fig. 2: a) Map of velocities obtained from the initial map. b) Left: initial map and path calculated with FM2. Right: map of distances after applying FMM over the map of velocities.

C. 3-Dimensional Fast Marching Square

Since the FM2 algorithm is based on the standard FMM, it is extensible to more than 2 dimensions. Due to the fact that a grasping task is carried out in a 3 dimensional space, 3D FM2 algorithm is going to be applied. The algorithm is exactly the same as explained before but, in this case, the frontwave becomes a spatial curve.

All the properties of the FM2, such as smoothness or safeness, remain in a n-dimensional environment. This is the main fact that lead us to use this algorithm as path planner. In figure 3 it is shown an example of a path obtained in 3D for a given environment using FM2.

III. PATH PLANNING FOR THE HAND AS A ROBOT FORMATION TOWARDS PRECISION GRASP

In order to perform a precision grasp we divide the problem into two phases. Initially, we want the hand to get close to the object and stop at a point from which the grasp can be done, therefore, a path from the start position towards the object is calculated. While covering the path the configuration of the hand changes to make the next phase easier. For this purpose the hand is first opened to ensure that the grasping can be done and later the configuration closes slowly so that the grasping process is shorter. In the second phase, the given precision grasp points must be reached by the fingertips of the hand. In this case a path for every fingertip must be calculated. These paths are covered just moving the fingers of the hand and the palm remains in the same pose. In this paper an algorithm for each of these phases is presented, both of them based on FM2 path planning method and its application to the control of a robot formation in a 3D environment.

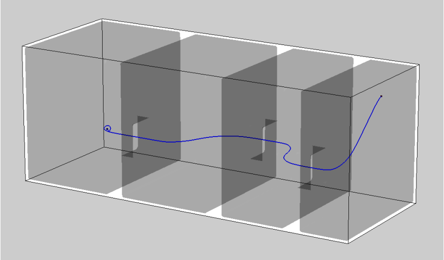


Fig. 3: Example of a 3D path result of the 3D FM2 algorithm.

A. Geometry of the hand as a robot formation

For the purpose of this paper, a hand is modelled as a kinematic chain so that simulations can be performed. The model used is based on the kinematic characteristics of Gifu Hand III which is shown in figure 4. We have chosen this model because is a five-finger hand and is very similar to a human one. In the structure, the lengths of the links between different joints (palm, finger separation and proximal/inter/distal phalanges) are constant, while the angles between these links (α, β and γ) are variable. These angles are limited by software following the specifications of the Gifu Hand III [1]. It is important to notice that, as in a human hand, the fourth joint of each finger engages with the third one linearly [13]. The same idea is applied for the thumb (although in the Gifu Hand III this joints in the thumb are independent). In order to consider the hand as a robot formation, mobile robots are located at the position of the joints and the fingertips of the hand. The hand is considered as a leader-followers formation in which the followers will change their location while performing the grasping process depending on their position in the environment. For each phase of the algorithm, the configuration of the formation is changed so that the different objectives can be achieved.

B. Approaching to the object to be grasped

As stated before, the first phase of the algorithm consists of an approaching movement towards the object to be grasped. In order to perform this approximation, the concept of robot formation planning based on FM2 [14], [15] is used to select the different configurations adopted by the structure of the robotic hand. Thus, the five-finger hand model is created using the leader-followers architecture. The leader is a mobile robot located at the center of the palm. The followers are located at each joint of the hand creating a robot formation that is deformable in the range of the mechanical limits of the real robotic hand. When the values of the joints change, the position of the followers evolve creating a different configuration of the hand. In the algorithm presented, this changes are generated in a very similar way to the one proposed in [15] but with a slight difference: instead of depending only on the position of the leader, the shape of the formation depends also on the location of the fingertips in the environmet and their value in the map of velocities calculated in FM2.

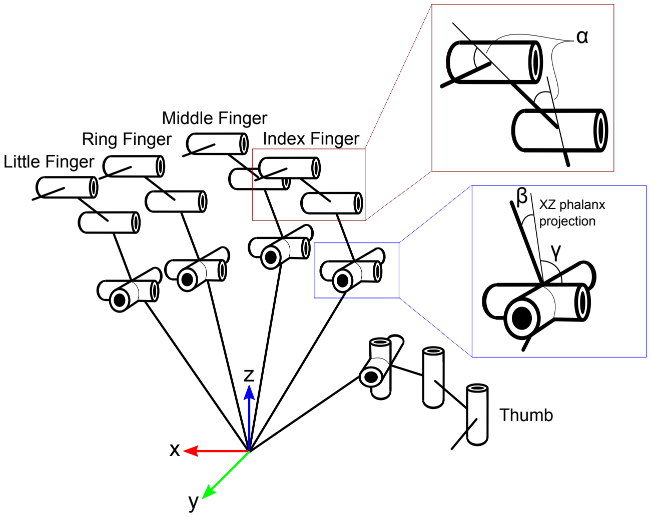


Fig. 4: Kinematic model of the Gifu Hand III.

The algorithm starts with an initialization step. Both the hand and the object are located in a binary three-dimensional environment in which free space is represented by zeros and objects or obstacles are represented by ones. In order to introduce some stochasticity, the start position for the hand and the location of the object are calculated introducing some random terms, so that every time the path towards this object is different. The leader of the formation (centre of the palm) is located at the start position and the position of the followers is calculated using the forward kinematics of the hand as a function of α, β and γ. These angles are initialized randomly to obtain a different evolution of the configuration of the hand every time. The goal position of the centre of the palm is located in front of the object to be grasped in a place from which the grasping phase can be performed. The computation of this position is out of the scope of the proposed algorithm, so it has been selected manually.

From this point we can start the first phase of the algorithm. A path is calculated for the leader of the formation using FM2. Thanks to its characteristics we can assure that the resulting path is very safe and smooth. This path is covered in an iterative loop. For each iteration, the value of the joints of the kinematic model of the hand are updated so the position of the followers in the formation changes. This approach allows us to control the 3D position of the 15 DOF of the robotic hand calculating only one three-dimensional path. In order to have a good configuration when the hand is close to the object, the evolution of the hand geometry is divided into two different phases:

- One of the stages corresponds to the part of the path where the leader is moving towards free space, which usually corresponds to the first part of the movement. This phase is detected analyzing the gradient of the grey level in the map of velocities for the position of the leader of the formation, which will be positive or null for this stage. Figure 5 shows a two-dimensional projection of the path on the centre slice of the map of velocities parallel to the YZ plane. The left part of the path surrounded by the blue ellipse corresponds with this phase. Since we know the hand is still far from the object and we are getting far from obstacles, the purpose in this phase is to evolve to an open configuration of the hand which ensures that the grasping points can be reached afterwards. Maximum opening of the hand is obtained for values: $\alpha_{j,max} = 0°$, $\beta_{j,max} = 90°$ and $\gamma_{j,max} = \pm 10°$, where j indicates the finger and all of them are expressed in degrees. Figure 6 shows the evolution of these angles which occurs following equations (4) and (5).

$$\alpha_{i,j} = max(\alpha_{i-1,j}, grey\_leader \times \alpha_{j,max}) \tag{4}$$

$$\beta_{i,j} = max(\beta_{i-1,j}, grey\_leader \times \beta_{j,max}) \tag{5}$$

where j indicates the finger, i corresponds to the iteration and the grey level of the leader is the value of the map of velocities in the position where the leader is located in the environment.

- The second stage starts when the hand is getting close to the object. Since the value of the map of velocities becomes smaller when the position is closer to an obstacle, the sign of its gradient for consecutive positions of the leader becomes negative when we are close to it. This part of the path is marked in figure 5 surrounded by a green ellipse. In this phase we try to close the fingers so that the position of each fingertip gets as close as possible to its local maximum in the map of velocities, which are indicated by the top and bottom red ellipses in figure 5. This way, we ensure that the position of every fingertips is safe and at the same time make the grasping phase shorter since part of the movement is already done. To detect when the local maximum has been reached, the gradient of value of the map of velocities for the fingertips is checked. Figure 7 shows the evoluion of the geometry in this stage. The update of the angles of the joints follows equations (6) and (7).

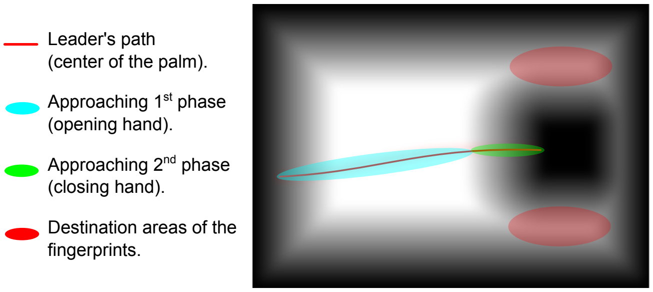


Fig. 5: Slice of the 3-dimensional map of velocities located in the centre of the width of the environment.

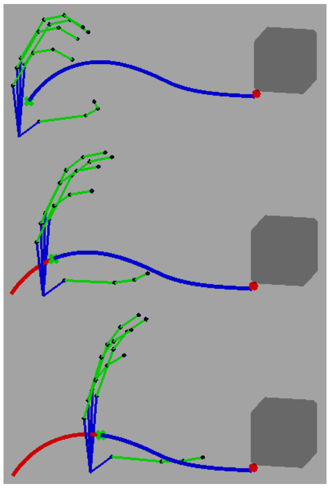


Fig. 6: Evolution of the hand towards an open configuration. Blue and red lines correspond to remaining and covered path respectively. The small black dots are followers in the formation and the green one in front of the palm is the leader. The dark grey cube is the object to be grasped.

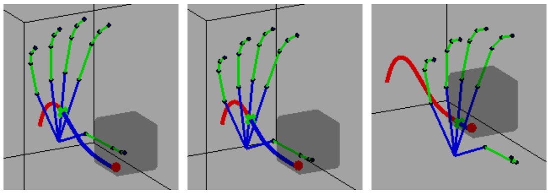


Fig. 7: Second phase of the approaching step.

$$\alpha_{i,j} = \alpha_{i-1,j} + K_1 grad\_grey\_level_{i,j} \tag{6}$$

$$\beta_{i,j} = \beta_{i-1,j} + K_2 grad\_grey\_level_{i,j} \tag{7}$$

where K_1 and K_2 are constants that definde the speed of the closing movement.

The first phase of the algorithm ends when the leader of the formation reaches a position from which the grasping can be done, which means that the grasping points on the object form part of the workspace of each finger. For being able to compute this point, an off-line analysis of the workspaces of the fingers and the thumb has been carried out to calculate an aproximate distance at which the hand has to stop. Again the grey level of the map of velocities is analyzed for the position of the leader to stop at the right distance to the object.

C. *Planning the movements of a precision grasp*

Once the hand is close enough to the object, the precision grasp can be performed. The grasping points are selected imitating a grasping configuration of a real hand on a cube. Although there could be several different grasping positions including a different number of fingers, we have selected the one showed in the right picture of figure 8 since we want to use all the fingers for this process. Besides, we have assumed that if this grasp is stable for a human hand, it should also be valid for a robotic hand. In order to perform the grasping movements, the setting of the formation is changed. We switch from one to five formations, one for each finger, in which the leader will be its respective fingertip. Besides, each new formation has three followers, mobile robots, located at the joints of the fingers that are connected by the proximal, intermediate and distal phalanges. In the same way as stated before, changes in the angles of the joints will lead to new configurations of the fingers. For this set of leaders, we perform the following grasping algorithm in which the pose of the palm is never changed:

1) A new path for every new leader, the fingertips, is calculated. These paths start at the stop position of the previous phase and end at the corresponding grasping point on the cube. In order to ensure that the paths are in the workspace of each finger, it is necessary to add some virtually occupied positions at the external edge of the workspace of the index and the little fingers. For these two fingers new maps of velocities need to be calculated because their environment has changed, but the others the same map of velocities that was calculated for the first phase of the algorithm is used. Once we have set these virtual obstacles, FM2 is used to calculate the five paths. Figure 8 shows the environment around the object to be grasped with the hand and the calculated paths.
2) After the paths have been obtained, the leaders cover them in an iterative loop applying the idea of robot formation control in a similar way as we did before. In this case, for each iteration, the joint angles are updated using the inverse kinematics of each finger. With these new values the followers of the formation change to their new position. The loop continues untill the grasping points have been reached. There are several approaches for the computation of the inverse kinematics, such us analytical [16], numerical [4] or iterative [17] algorithms. Although any method would be valid for this application, we use an analytical solution. It is important to point out that the movements of the fingers are made so that all the fingers arrive at their contact point at the same time.

Figure 9 shows several iterations of the algorithm while the fingertips are covering the paths towards the precision grasping points. Finally, figure 10 shows the final position

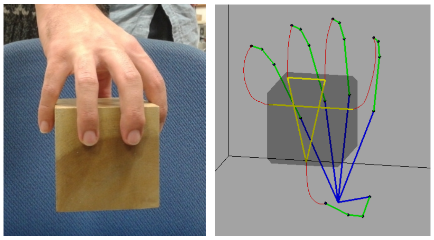


Fig. 8: Left: a human hand holding a cube. Right: computed paths towards the grasping points.

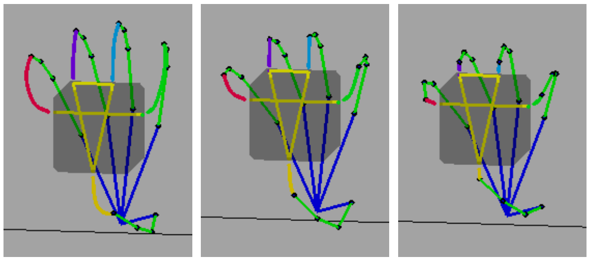


Fig. 9: Precision grasp phase. The fingertips cover the paths towards the contacts points with the object.

of the grasping algorithm from a different point of view.

IV. CONCLUSIONS AND FUTURE WORK

This paper presents a novel application of the standard FM2 path planning method recently introduced for the control of robot formations in a three dimensional space. In this application, FM2 method has been applied for obtaining both the path to get close to the object to be grasped and the necessary paths to reach precision grasp points on it. While covering these paths, the kinematic chain that models the Gifu Hand III is considered to be a leader-followers robot formation which shapes evolves with respect to the map of velocities computed by FM2.

The simulations, carried out with Matlab, show that it is possible to consider a robotic hand as a robot formation in order to decrease the complexity of the planning process. Using this approach, we do not need to compute a path in the 3D space for each DOF (15 in our case). Instead, only one path is calculated and the positions of the other DOFs are updated using the map of velocities. In the grasping phase, only the paths for the fingertips are calculated.

Since FM2 is used for the computation of the different paths, we can assure that they are smooth, safe and fast thanks to the characteristics of the path planning method. Besides, not only the positions in this path are obtained, but also the desired velocity each robot should reach at these positions, which makes it easier for the movement controller.

The future work focuses on more complex simulation including dynamics, extending the experimentation to more complex environments and evaluating the algorithm with a real robotic hand-arm system.

ACKNOWLEDGMENT

This work is funded by the project number DPI2010-17772, by the Spanish Ministry of Science and Innovation.

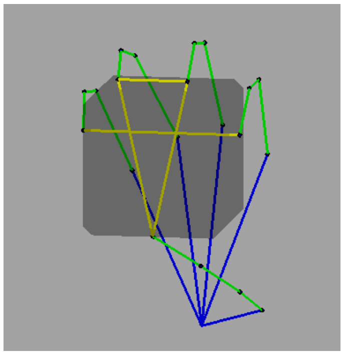


Fig. 10: End position of the fingertips in the grasping algorithm.

REFERENCES

[1] T. Mouri , H. Kawasaki, K. Yoshikawa, J. Takai and S. Ito, Anthropomorphic Robot Hand: Gifu Hand III, *International Conference on Control, Automation and Systems*, 2002.

[2] Shadow Robot Company, Shadow Dextrous Hand C5 Technical Specification [Online] Available: http://www.shadowrobot.com/hand/ (March 2013).

[3] H. Liu, P. Meusel, G. Hirzinger, M. Jin, Y. Liu and Z. Xie, The Modular Multisensory DLR-HIT-Hand: Hardware and Software Architecture *IEEE/ASME Transactions on Mechatronics*, pp. 461–469, volume 13, 2008.

[4] M. Abderrahim, C. Bensalah and J. G. Gómez, A New Finger Inverse Kinematics Method for an Anthropomorphic Hand, *Proceedings of the IEEE International Conference on Robotics and Biometrics*, 2011.

[5] M. Cutkosky, Human Grasp Choice and Robotic Grasp Analysis, *IEEE Transactions on Robotics and Automation*, pp. 269–279, chapter, volume 5, no. 3, 1989.

[6] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi and Y. Miura, Patterns of Static Prehension in Normal Hands, *Americal Journal of Occupational Therapy*, pp. 437–445, volume 34, 1980.

[7] C. Rosales, J. M. Porta, R. Suarez and L. Ros, Finding All Valid Hand Configurations for a Given Precision Grasp, *International Conference on Robotics and Automation Pasadena*, 2008.

[8] E. W. Dijkstra, A Note on Two Problems in Connexion With Graphs, *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.

[9] J. A. Sethian, A Fast Marching Level Set Method for Monotonically Advancing Fronts, *Proceedings of the National Academy of Science*, pp. 1591–1595, volume 93, no. 4, 1996.

[10] S. Osher and J. A. Sethian, Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*, no. 79, pp. 12–49, 1988.

[11] S. Garrido, L. Moreno, M. Abderrahim and D. Blanco, FM2: A Real-time Sensor-based Feedback Controller for Mobile Robots *International Journal of Robotics and Automation*, pp. 3169–3192, volume 24, no. 1, 2009.

[12] L. Yatziv, A. Bartesaghi and G. Sapiro, A Fast O(n) Implementation of the Fast Marching Algorithm, *Journal of Computational Physics*, pp. 393-399, volume 212, no. 1, 2005.

[13] H. Kawasaki, T. Komatsu, M. Suda, and K. Uchiyama, Development of an Anthropomorphic Robot Hand Driven by Built-in Servo-motors, *Proceedings of the 3rd Int. Conf. On ICAM, Vol.1, pp. 215-220*, 1998.

[14] S. Garrido, L. Moreno and P. U. Lima, Robot Formation Motion Planning Using Fast Marching, *Journal Robotics and Autonomous Systems*, pp. 675-683, volume 59, no. 9, 2011.

[15] J. V. Gómez, A. Lumbier, S. Garrido and L. Moreno, Planning Robot Formations with Fast Marching Square Including Uncertainty Conditions, *Journal Robotics and Autonomous Systems*, pp. 137-152, volume 61, no. 2, 2013.

[16] V. Bundhoo, Design and Evaluation of a Shape Memory Alloy-based Tendon-driven Actuation System for Biomimetic Artificial Fingers, *PhD Thesis, University of Mauritius*, 1999.

[17] J. Lasenby, A. Aristidou, FABRIK: A fast, Iterative Solver for the Inverse Kinematics Problem, *Journal Graphical Models*, pp. 243-260, volume 73, no. 5, 2011.

[18] Andrew T. Miller, GraspIt!: A Versatile Simulator for Robotic Grasping, *PhD Thesis, Department of Computer Science, Columbia University*, June 2001.