

Precision Grasp Planning 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 wants to be performed. This approach is based on the standard Fast Marching Square (FM2) path planning method recently introduced for robot formations. A three-finger kinematic chain is considered as a robot formation to perform simulations. In order to achieve a precision grasp, the task is divided into two phases. In the first one, the hand has to move towards the object to be grasped and stops at a position from which the grasping points can be reached by the fingers of the hand. In the second one, given the contact points for a precision grasp, the movements of the fingers must be planned so that those points are reached by the corresponding fingertips. In both cases, the path planning method used is FM2, so smooth and fast paths are ensured due to the characteristics of FM2. In each phase, different control strategies for robot formations are used. The changes in the geometry of the formations are based on the velocities map calculated in FM2, ensuring collision avoidance and speeding up the grasping phase. Simulation results show the usefulness of this novel application of the method thanks to a good performance of the chosen planning strategy.

I. INTRODUCTION

Grasp planning is an essential problem in robot manipulation and at the same time is a very complex one, since it involves the combination of open and close kinematic chains, nonholonomic constraints and redundant degrees of freedom. While humans can grasp and manipulate objects very easily, robotic hands developed for research, such as Gifu Hand III [1], Shadow Hand [2] or HIT/DLR [3] often have a large number of degrees of freedom (DOF), which increments the planning and control complexity because, for example, a wide variety of joint configurations are possible to be reached.

The grasping problem can be defined as, given an object in an environment determine how to approach to this object, find a stable way to grasp it and perform the chosen grasping strategy to hold or manipulate it. According to Cutkosky [4], 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 [5]. The main characteristic of this kind of grasp is that only the fingertips are in contact with the object. When performing a precision grasp, we have to take into account basically two types of constraints. First, dynamic ones, since forces created between the hand and the object to be grasped must be compensated. This can be satisfied by force- or form closures, depending on whether the object is fully constrained by the fingers forces or by the finger positions, respectively [6]. Second, the plan needs to meet kinematic constraints since, obviously, 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 methodology for planning the necessary movements to perform a precision grasp of a three-finger robotic hand is presented. For achieving this goal, the grasping task has been divided into two phases: approaching to the object and reaching the given contact points. A three-finger hand is treated as a mobile robot formation with the leader-followers architecture, in which the geometry of the formation is updated by changing the position of the followers based on an artificial potential field. Two different situations are considered during the grasping process, in all cases the planning strategy is based on Fast Marching Square (FM2) and its application to control of robot formations.

The rest of the paper is 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 (FM) method has been chosen as path planner. At the beginning, this method suggested by J. Sethian in 1996 was proposed to approximate the solution of the Eikonal equation [8]. Let us assume that a wave starts propagating at time $T = 0$ with velocity F always non-negative. The Eikonal equation allows us to update the time of arrival of the wave T for each position x according to:

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

Discretizing the gradient ∇T according to [9] it is possible to solve the Eikonal equation at each point $p(x_i, y_j)$, which

corresponds to the row i and column j of a grid map as follows:

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

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

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 $T_{i_0,j_0} = 0$. Once it has finished, the algorithm outputs a distances map as shown in figure 1.

FM can be directly used as a path planner algorithm. By applying gradient descent from any point of the distance map, 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 distances map. Otherwise, local minima will appear. The main advantage of this method is that the path obtained is optimal in distance. An example of a path provided by Fast Marching is shown in figure 1.

A. Fast Marching Square Method

Although the paths provided by the FM method are optimal in distance terms, they do not accomplish the smoothness and safety constraints that most robotic applications require. These paths run too close to obstacles and walls and they have sharp curves. In view of these drawbacks, it turns out that the FM algorithm is not a good solution in most cases. However, the Fast Marching Square algorithm [10] solves these two main disadvantages.

The FM2 method is based on creating velocities (or slowness) maps depending on the environment map in which the velocity of the expanding wave varies depending on the distance to the closest obstacle. The FM method is applied in order to obtain these velocities maps. In this case, all the obstacles and walls are labeled as wave sources. The result is a distances map in which those cells in the grid that are farther from the obstacles have a higher value (figure 2 a)).

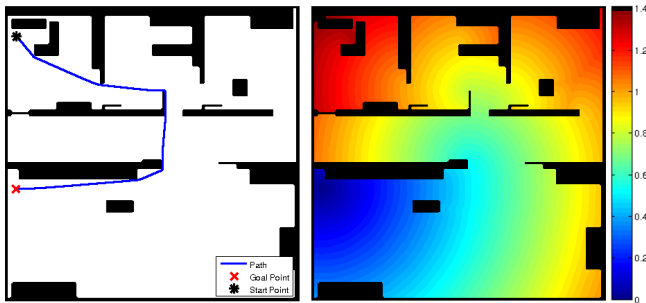


Fig. 1: Example of a path obtained with Fast Marching. The left side shows the original map and the path calculated by FM. In the right there is the distances map calculated in the first step of FM.

Once this distance map is computed, it is normalized in order to have values between 0 and 1 (which mean stopped or full propagation speed correspondingly). The FM method is then applied with the goal point as wave source. During the expansion, the wave will propagate with the velocities indicated in the map generated previously. The resulting distances map will be similar to the one obtained with the standard FM method, but with slight differences which make the paths very smooth when gradient descent is applied (figure 2 b)).

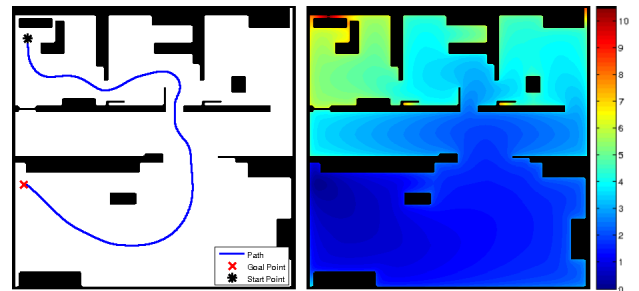
The effect the velocities map has over the expanding wave is exactly the same than the one the refraction index of a medium has over a light wave propagation. According to the Fermat's principle (least time principle): *the path taken between two points by a ray of light is the path that can be traversed in the least time.*

Apart from the smoothness and safety, FM2 has other properties worth to mention:

- *Absence of local minima.* As long as only one wave is employed to generate the distances map, FM ensures that there will only be one global minimum at the source point of the wave (goal of the path).
- *Completeness.* The method will find a path if it exists. It will point out also if there is not a possible path.
- *Fast response.* The velocities map has to be calculated only once and the FM method has complexity order of $O(n)$ [13]. Besides, the simplicity with which the environment is modelled avoids complex calculations or costly sensory treatment.



(a)



(b)

Fig. 2: a) Velocities map obtained from the map on the left side of figure 2 b). b) Left: original map and path calculated with FM2. Right: distances map after applying Fast Marching over the velocities map.

B. 3-Dimensional Fast Marching Square

Since the FM2 algorithm is based on the standard FM method, 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. 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 3 dimensions for a given environment using FM2.

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

Grasping an object can be considered as a two-phases action. In the first one, the goal is to approach to the object using a smooth path. This step finishes when the robotic hand is situated at a distance from which the grasping can be executed. Thus, at this point the hand configuration has to allow the necessary movements to be able to perform the selected grasping method. In the second phase the hand configuration evolves until the contact points between the robotic hand and the object are reached. In this paper an algorithm for each of these phases is presented, both of them based on FM2 and its application to the control of a robot formation in a 3D environment.

A. Geometry of the hand as a robot formation

A three-finger kinematic chain is considered as a robot formation in order to perform simulations of this methodology. The geometry proposed for the formation is based on the dimensions of the fingers of the Gifu Hand III [1] and is shown in figure 4. The leader of the robot formation is located at the centre of the palm of the hand, while the followers are situated at each joint of the fingers, and both the leader and the followers are considered as mobile robots. In this 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

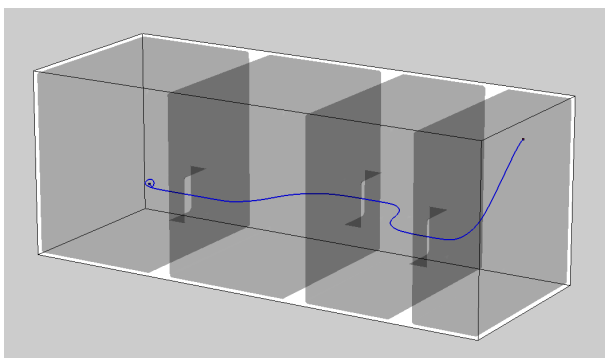


Fig. 3: Result of the FM2 algorithm applied in 3 dimensions.

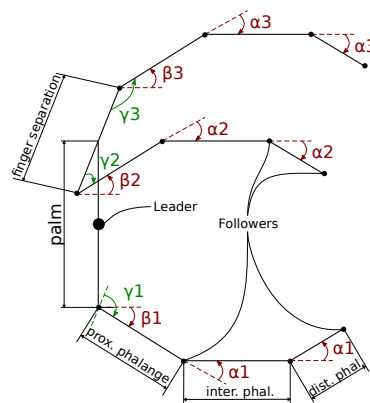


Fig. 4: Description of the geometry of the robot formation that simulates the hand.

following the specifications of the Gifu Hand III [1]. It is important to notice that, as in human hand, the fourth joint angle of each finger engages with the third joint angle linearly [14].

B. Approaching to the object to be grasped

As stated before, the first phase is the approaching one. In order to perform this approximation movement, the concept of robot formation planning based on FM2 [11], [12] is used to model the structure of the robotic hand. Thus, the three-finger hand is created using the leader-followers architecture, where the position that the followers adopt in the formation is deformable. The evolution of these positions depends on the location of the leader in the environment and the value of the velocities map calculated in FM2. In our case, the geometry of the formation will be modified by introducing changes in the angles described in figure 4, so that we control when the hand is opening or closing. These changes are generated in a very similar way to the algorithm proposed in [11].

As an initialization step of the algorithm, 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 to be grasped have some random terms, so that every time the path towards this object is different. The start position is defined by the centre of the palm of the hand, where the leader of the formation is placed. The angles defined in figure 4 are also initialised randomly, but always within certain limits that result in a configuration of the hand where it is partially closed. The goal position for the centre of the palm in the approaching phase is set in front of the object to be grasped in a way that ensures that the orientation of the palm of the robotic hand makes the grasping task possible. The selection of this points is out of the scope of this paper.

Once the environment is initialised FM2 is applied to obtain the path for the leader of the formation. Thanks to the characteristics of FM2 path planning algorithm we can assure that the resulting path is very safe and smooth. This

path is covered in an iterative loop in which, for each step, the angles describing the formation are updated and therefore the configuration of the hand changes. The final goal of this phase is to locate the hand in a position from which the precision grasp can be done, meaning that the grasping points are located in the working space of each finger. While the hand is covering the path, the algorithm is divided into two different phases.

- The first phase of the path occurs when the gradient of the grey level in the velocities map is positive or null. This means the leader is moving towards the object but it is still far from it. Figure 5 shows a two-dimensional projection of the path on the centre slice of the velocities map parallel to the YZ plane. The left part of the path surrounded by the blue ellipse of this figure corresponds with this phase. In this state the objective of the hand is to reach an open configuration since this ensures that afterwards the grasping can be performed. The limits of the open configuration are imposed by software ($\alpha_{j,max} = 0$, $\beta_{j,max} = 90$ and $\gamma_{j,max} = \pm 10$, where j indicates the finger and all of them are expressed in degrees) which imitate the mechanical limits of the system. Figure 6 shows the evolution of these angles when the hand is opening while approaching the object, which occurs following equations (4) and (5).

$$\alpha_{i,j} = \min(\alpha_{i-1,j}, (1 - grey_leader)\alpha_{j,max}) \quad (4)$$

$$\beta_{i,j} = \max(\beta_{i-1,j}, grey_leader\beta_{j,max}) \quad (5)$$

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

- The second phase starts when the hand is getting close to the object and, therefore, the sign of the gradient of the velocities map becomes negative. This happens

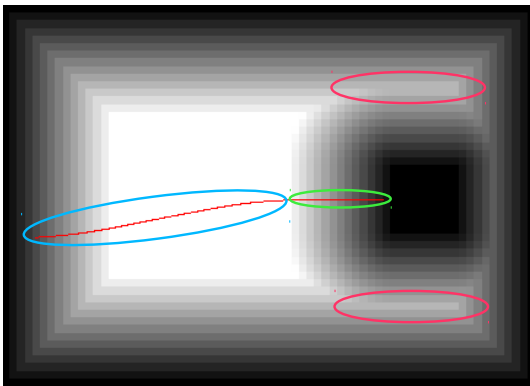


Fig. 5: Slice of the three-dimensional greymap located in the centre of the width of the environment.

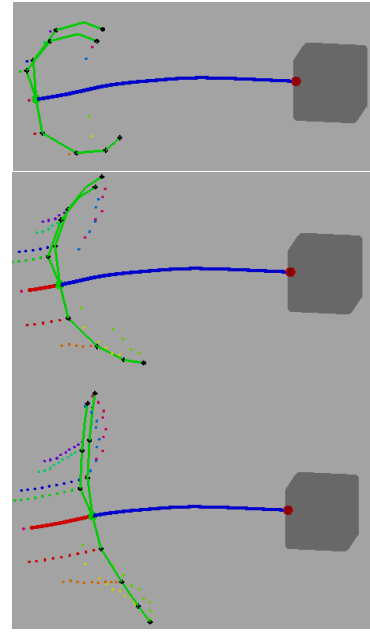


Fig. 6: The hand evolves towards an open configuration. The blue and red lines correspond to remaining and covered path respectively. The small black dots are follower robots in the formation located in the joints of the hand. Green lines simulate the links of the hand. The dark grey cube is the object to be grasped.

because the velocities in the map become smaller when we approach to any obstacle, in this case, the object to be grasped. This part of the path is shown in figure 5 surrounded by a green ellipse located in the centre of the figure. 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 grey level map, which are indicated by the top and bottom red ellipses in figure 5. This way, we ensure that the position of the fingertips is safe, and at the same time make the grasping phase shorter since a part of the movement is already done. To know when the local maximum has been reached, the gradient of grey level of the fingertips in the velocities map is checked. Figure 7 shows the evolution of the geometry in this phase. The change of the angles of the joints follows equations (6) and (7).

$$\alpha_{i,j} = \alpha_{i-1,j} - K_1 grad_gray_level \quad (6)$$

$$\beta_{i,j} = \beta_{i-1,j} + K_2 grad_gray_level \quad (7)$$

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

This first part of the algorithm ends when the leader of the formation reaches a position from which the grasping phase can be performed, in other words, every desired contact point

of the precision grasp forms part of the workspace of the corresponding finger. The stop distance is determined by an off-line analysis of the workspaces of the fingers and it is detected using the velocities map aforementioned, since a low grey level in this map means we are close to the object.

C. Planning the movements of a precision grasp

The last part of the algorithm consists on performing a given precision grasp. The grasp points are chosen so that they form a triangle whose centre of masses, also known as barycentre, is located at the same point of the centre of masses of the object to be grasped. Although the calculation of the contact points is not an objective of this paper, we expect the selected ones cause the forces applied by the robotic hand to compensate among them and we get a stable grasp. This triangle can be seen in figure 8 painted with green lines. From this moment the leader of the formation changes, instead of having one leader located at the palm, we have three leaders, one for each fingertip. The new formations are formed by the proximal, intermediate and distal phalanges of each finger, which have a fixed value. Again, the variables that change their value while performing the grasping are the joints of the fingers. For these set of leaders, we perform the following grasping algorithm in which the pose of the palm is never changed:

- 1) First, a path for each fingertip is calculated using FM2. These paths start in the position where the fingertips were located in the last iteration of the approaching phase and end in their corresponding contact point with the object. To ensure that the points in the paths calculated with FM2 are reachable by the fingers, three subspaces delimiting the workspace of each finger are placed in the environment so that these paths can never go out of their corresponding workspace. These matrixes act like walls, this way the FM algorithm never explores positions out of the workspace of the fingers. Since the environment has changed, the two phases of FM algorithm must be performed. This means, first we calculate the new velocities map for each leader, and then obtain the new paths, again they are smooth and safe paths thanks to the characteristics of FM2. Figure 8 shows the environment around the object to be grasped after these subspaces are added.
- 2) After the grasping paths have been calculated, the new leaders start covering them in an iterative loop. For

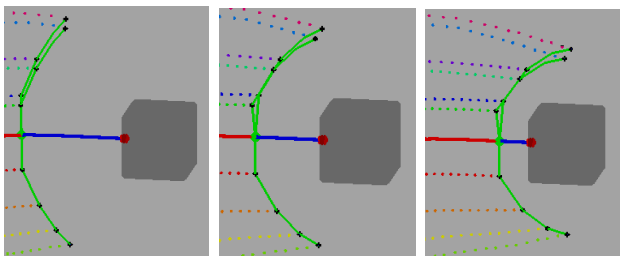


Fig. 7: Second phase of the approaching step.



Fig. 8: Environment matrix with the addition of the workspaces of the fingers.

each position of the fingertips in the path, the inverse kinematics of each finger is calculated so that we obtain the new values for the angles in the geometry, defined in figure 4, and the position of the followers in the formation is updated. For the purpose of this algorithm, any method for computing the inverse kinematics (analytical, numerical, optimization algorithms, etc.) is valid.

- 3) Finally, for each leader, we check if the grasping point has been reached. This is done looking at the grey level of the velocities map since, as explained before, it can be used to determine distances. If the goal point has not been reached, the algorithm continues with the next point in the path.

Figure 9 shows several iterations of the algorithm while the fingertips are covering the paths towards the precision grasp points. The little dots painted in different colours correspond to previous positions of the members of the formation. Finally, figure 10 shows the final position of the grasping algorithm from a different point of view. The submatrixes added between the two phases of the algorithm are not shown for easier understanding of these figures.

IV. CONCLUSIONS AND FUTURE WORK

This paper presents a novel application of the standard FM2 path planning method recently introduced for robot formations. In this application a robotic hand is considered as a robot formation with a geometry described by the forward kinematics of a three-finger hand. Although the considered hand model is a simple one, the simulations, carried out with Matlab, show that it is possible to consider a robotic hand as a robot formation in order to plan the necessary path to get close to an object and, meanwhile, set the control points of the position of the fingers of the hand while the path is being covered. It is also shown that this method can be applied for planning the necessary movements to perform a precision grasp once the pre-grasp position has been reached. It is also important to point out that this methodology can be easily extended to a five-finger robotic hand.

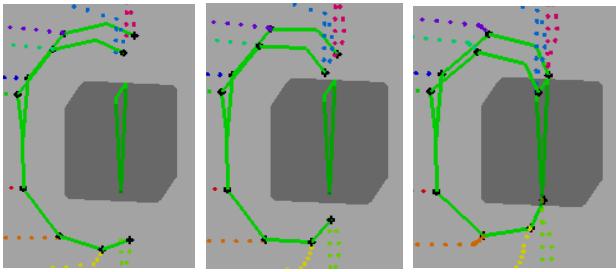


Fig. 9: Evolution of the precision grasp phase. The fingertips cover the calculated path towards the contacts points with the object.

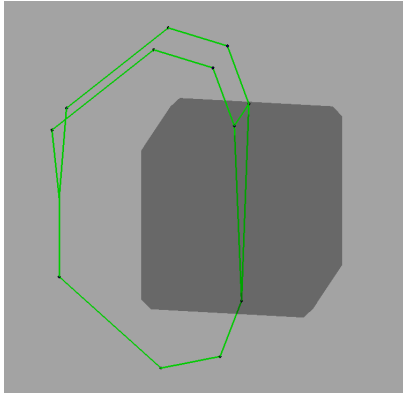


Fig. 10: End position of the fingertips in the grasping algorithm.

Due to the characteristics of FM2, we can assure that the paths obtained for the palm of the hand in the approaching step and for the fingertips in the grasping phase are smooth, safe and fast.

Also it is important to point out that, since we consider the hand as a robot formation, the complexity of the path planning phase decreases because we do not plan the path for all the degrees of freedom in the hand.

The future work related to this paper focuses on expanding its application to a five-finger robotic hand, using a more complex formation, and extending the experimentation to more complex environments.

ACKNOWLEDGMENT

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

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/> (May 2012).
- [3] H. Liu, P. Meusel, G. Hirzinger, M. Jin, Y. Liu and Z. Xie, The Modular Multisensory DLR-HIT-Hand: Hardware and Software Architecture *Mechatronics, IEEE/ASME Transactions on*, pp. 461–469, volume 13, 2008.
- [4] M. Cutkosky, Human Grasp Choice and Robotic Grasp Analysis, *IEEE Transactions on Robotics and Automation*, pp. 269–279, chapter, volume 5, number 3, 1989.

- [5] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi and Y. Miura, Patterns of Static Prehension in Normal Hands, *American Journal of Occupational Therapy*, pp. 437–445, volume 34, 1980.
- [6] A. Bicchi, On the Closure Properties of Robotic Grasping", *International Journal of Robotic Research*, pp. 319–334, volume 14, number 4, 1995.
- [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] 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, number 4, 1996.
- [9] S. Osher and J. A. Sethian, Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*, number 79, pp. 12–49, 1988.
- [10] 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, number 1, 2009.
- [11] 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, issue 9, 2011.
- [12] 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, issue 2, 2013.
- [13] 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, number 1, 2005.
- [14] 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.