Science Arts & Métiers (SAM)
is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/18196

To cite this version :

Mohammad SAFEEA, Pedro NETO, Richard BEAREE - Collision Avoidance of Redundant Robotic Manipulators Using Newton's Method - Journal of Intelligent and Robotic Systems p.1-9 - 2020

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Collision Avoidance of Redundant Robotic Manipulators Using Newton's Method

Mohammad Safeea[1,2] · **Richard Béarée**[2] · **Pedro Neto**[1]

Abstract

This study investigates the application of Newton method to the problems of collision avoidance and path planning for robotic manipulators, especially robots with high Degrees of Freedom (DOF). The proposed algorithm applies to the potential fields method, where the Newton technique is used for performing the optimization. As compared to classical gradient descent method this implementation is mathematically elegant, enhances the performance of motion generation, eliminates oscillations, does not require gains tuning, and gives a faster convergence to the solution. In addition, the paper presents a computationally efficient symbolic formula for calculating the Hessian with respect to joint angles, which is essential for achieving realtime performance of the algorithm in high DOF configuration spaces. The method is validated successfully in simulation environment. Results for different methods (Newton, gradient descent and gradient descent with momentum) are compared in terms of quality of the path generated, oscillations, minimum distance to obstacles and convergence rate.

Keywords Collision avoidance · Newton · Optimization · Hessian

1 Introduction

Redundancy is a term used for referring to robots that have more DOF than the necessary to complete a given task. Extra DOF offer better flexibility and can be used to achieve multiple objectives besides to the required robot task [1]. As such, several studies have been dedicated to the problem of controlling redundant robots [2, 3]. In cluttered environments, with obstacles, redundancy plays an important role, allowing the robot to reach the goal while avoiding such obstacles. Yet, for achieving real-time path planning capacity for high DOF manipulators in unstructured environments, it is required to merge the redundancy control frameworks with some notion of attraction/repulsion vector fields [4, 5]. In such a case, task-level priorities are achieved through projections on the null space of the Jacobian [6], or by using more sophisticated projection criteria [7]. This method for formulating collision avoidance is proved to be computationally efficient. Nevertheless, in many cases this implementation requires a predefined trajectory of the end-effector (EEF), where the repulsion/attraction vectors are custom engineered with many parameters to tune. Moreover, this method suffers from local minima and oscillation problems, which become more evident for increasing number of obstacles and for higher DOF robots. For achieving better immunity to local minima other methods have been proposed, including the circular fields [8] and the Probabilistic Roadmaps (PRM) [9]. The circular fields is inspired by electromagnetism phenomena and has been applied successfully to control 7 DOF redundant manipulators [10]. PRM is a stochastic method which is composed of two phases. The first phase is called the learning phase, it is performed for stochastic generation of a roadmap of the scene. The second phase is called the query phase. In this phase, paths are queried, and the roadmap is searched for a feasible path. While PRM is a powerful method, it suffers two major problems. The first of which is due to computational cost, where according to the times

reported in [9] it can be concluded that for high DOF robots PRM is unsuitable for implementation in control loops with fast update rates. The second drawback comes from the fact that the resulting trajectories generated by this planner are not smooth and require processing before implementation on the robot [11]. The Dynamic Roadmaps method (DRM) is introduced in [12]. It is closely related to PRM method (implementing two phases, one for roadmap construction and another for query). However, in DRM the roadmap is updated according to changes in the environment. This allows using the algorithm for avoiding collisions with non-static obstacles. This algorithm was implemented in [13] for performing collision avoidance in human-robot collaborative scenario. In [14], the authors implemented DRM on a 7 DOF robot, in which they identified the main bottlenecks in DRM as such they achieved a 100 milliseconds update rate. Based on the variational principles, the authors in [15] presented an algorithm that generates collision free paths for redundant manipulators in constrained workspaces. Fuzzy control [16, 17] is used for controlling non-linear dynamical systems, including robots. Based on fuzzy rules [18], it is proposed a method for resolving internal joint angles in redundant manipulators. The method allows the EEF to follow the desired path, while the internal motion manifold is used to perform other objectives including collision avoidance with surrounding obstacles.

In this study we investigate the application of Newton method to the problem of collision avoidance for hyper redundant robotic manipulators. In such a case, the problem is reformulated as a second order optimization problem and the Newton method is utilized for performing the optimization. This offers various advantages due to the main features of the Newton method [19], namely:

1. Better numerical stability;
2. Faster convergence rates;
3. Better immunity to numerical oscillations that commonly appear in the first order (gradient descent) methods.

In robotics this translates into generating smoother paths and faster execution, as reported in [20, 21], where the modified Newton method is applied for the problem of robot navigation, where the mobile robot is approximated by a point in a plane and the potential field is an explicit function of the configuration variables, in such a case calculating the Hessian analytically is possible as noted in [22]. While the Newton method has been described in robotics literature for the problem of mobile robot navigation, this method has not been investigated yet for the problem of collision avoidance in robotic manipulators. This comes from the fact that the resulting potential fields, which are functions of Cartesian space variables, are implicit functions of configuration space variables, with high non-linearity due to the nature of the complex transform function between the spaces [23], this makes finding a closed-form solution for the Hessian in joint space a challenging task. In this study a systematic method for applying Newton technique for robotic manipulators, at the kinematic level, is addressed. We present a formula for efficient calculation of the Hessian matrix, by deducing a relationship between the Hessian in Cartesian space and the resulting Hessian in joint space. This allows to speed up the algorithm and makes it applicable for real-time implementation. In addition to the aforementioned advantages, the proposed solution contributes in the following:

1. Bigger optimization steps: In Newton method bigger optimization steps can be used, resulting in faster execution rates, as compared with the gradient method;
2. Faster convergence: Newton method offers better performance in terms of convergence rate than the gradient descent [24];
3. Elegant formulation: The proposed method is mathematically elegant, eliminates the need for tedious tuning of control parameters, and can be applied on hyper redundant manipulators. The method does not require pre-planned trajectory and keeps the number of control parameters to a minimum.

The proposed method can be used either for real-time control of the manipulator to avoid collisions with obstacles, or it can be used for off-line path planning. The application of the presented method extends easily to computer animations and to motion generation for kinematic chains in virtual environments with obstacles.

2 Mathematical Formulation

For achieving a robot motion that drives the EEF towards the goal, while avoiding collision with obstacles, the potential fields method by Khatib [25] is considered. In this method an attraction potential field u^{att} attracts the EEF towards the goal. This potential field is expressed in Cartesian space as a function of EEF position. In addition, repulsion potential fields repel the robot away from obstacles. As a result the robot is subjected to a total potential field u^{total}, given by:

$$u^{total} = u^{att} + \sum u^{rep}_{(i,j)} \tag{1}$$

Where $u^{rep}_{(i,j)}$ is a repulsion potential field that repels link i from colliding with obstacle j. The potential $u^{rep}_{(i,j)}$ is expressed in Cartesian space as a function of the minimum distance between link i of the robot and obstacle j. Given that the control algorithm of the robot is running in a loop, in each iteration the robot changes its configuration

slightly to minimize the total potential function, as a result the robot moves towards the goal while avoiding collisions with obstacles. Traditionally, the notion of attraction and repulsion vectors is used for performing the collision avoidance. This traditional method can be interpreted as minimizing the potential function using the gradient descent technique, where the gradients represent the repulsion and the attraction vectors. In contrast, we propose to use the Newton method for minimizing the potential function. Taking the second order approximation of the potential function yields the following update equation for the joint angles:

$$q^{m+1} = q^m - \alpha(\mathrm{H} + \lambda \mathrm{I})^{-1} g \tag{2}$$

Where q^k is the resulting vector of the angular positions calculated at time-step k, in real world scenario q^m represents the measurements of joints positions as acquired from the encoders, α is a scalar representing the minimization step, H is the Hessian of the potential field with respect to joint angles, g is the gradient of the potential field with respect to joint angles, I is the identity matrix, and λ is a damping factor. By adopting a similar approach to the already developed techniques in differential inverse kinematics, the damping factor can be calculated as a function of the minimum singular value [26], but unlike inverse kinematics methods, in our method the minimum singular value pertains to the singular value decomposition of the Hessian matrix H. For fast execution of the proposed algorithm, the Hessian shall be calculated analytically, by taking a second order approximation of the potential function (1), where we proved in Appendix that for robotic manipulators the Hessian is given by:

$$\mathrm{H} = \sum \mathrm{J}_k^{\mathrm{T}} \nabla^2 u_k \mathrm{J}_k \tag{3}$$

Where $\nabla^2 u_k$ is the Hessian matrix of the potential function in Cartesian space. Given that u_k is expressed in Cartesian space for attraction and repulsion fields, a closed-form solution of this matrix can be deduced easily by hand or by using symbolic math software. J_k is the partial Jacobian associated with the potential field u_k. In case of the attraction potential u^{att}, then J_{att} is the Jacobian associated with the EEF. In case of a repulsion potential $u^{rep}_{(i,j)}$ due to the effect of obstacle j on link i, then $\mathrm{J}_{(i,j)}$ is the partial Jacobian associated with control point $cp_{(i,j)}$, where $cp_{(i,j)}$ is the point of link i closest to obstacle j. The gradient (as shown in Appendix) is given by:

$$g = \sum \mathrm{J}_k^{\mathrm{T}} \nabla u_k \tag{4}$$

Where ∇u_k is the gradient of potential function u_k in Cartesian space. For achieving better numerical robustness and by analogy to differential kinematics approaches, the alternate Jacobian described in [27] can be utilized for calculating the partial Jacobians. As a result, it can be concluded that the presented mathematical formulation in this study forms a systematic way to integrate several potential fields:

- Attraction potential that attracts the EEF to the goal position;
- Repulsion potential field that repels the robot away from obstacles;
- Repulsion potential field that repels the robot away from self-collision and joint limits.

3 Experimental Tests and Results

To evaluate the performance of the proposed method, real-time simulations using MATLAB are implemented, where the attraction potential field u^{att} is chosen to be a quadratic function with a minima at the goal position. Consequently, u^{att} causes the EEF to move towards the goal:

$$u^{att} = \frac{1}{2} k_{att} (x_{ef} - x_g)^{\mathrm{T}} (x_{ef} - x_g) \tag{5}$$

Where k_{att} is the attraction constant, x_{ef} is the position of the EEF and x_g is the position of the goal point. On the other hand, the repulsion potential field chosen for the simulations is identical to the one in [25], where according to the proximity between obstacle j and link i the repulsion potential $u^{rep}_{(i,j)}$ is given by:

$$u^{rep}_{(i,j)} = \begin{cases} 0 & \rho_{(i,j)} > \rho_1 \\ \frac{k_{rep}}{\rho_{(i,j)} - \rho_0} - \frac{k_{rep}}{\rho_1 - \rho_0} & \rho_{(i,j)} \leq \rho_1 \end{cases} \tag{6}$$

Where k_{rep} is a repulsion constant, $\rho_{(i,j)}$ is the minimum distance between obstacle j and link i of the robot, ρ_1 is the distance at which the repulsion potential starts acting on the robot, ρ_0 defines the forbidden area around the obstacle where the robot cannot penetrate. Similar repulsion fields could be added for joint limits avoidance and for self-collision avoidance. Under the previous potential functions a planar hyper redundant manipulator, with 15 DOF, is navigating through 10 obstacles towards the goal. The goal is marked by a green circle and obstacles are marked by red circles, as shown in Fig. 1, where the Newton method is applied to perform the collision avoidance and the motion generation. The figure shows a sequence of configurations taken by the robot during its motion towards the goal. In the beginning of the simulation, the robot is in the vertical upright configuration, links are fully stretched. Afterwards the robot starts bending and reaching towards the goal while navigating through the obstacles. In the simulation, the Hessian matrix was calculated efficiently using (3), as such

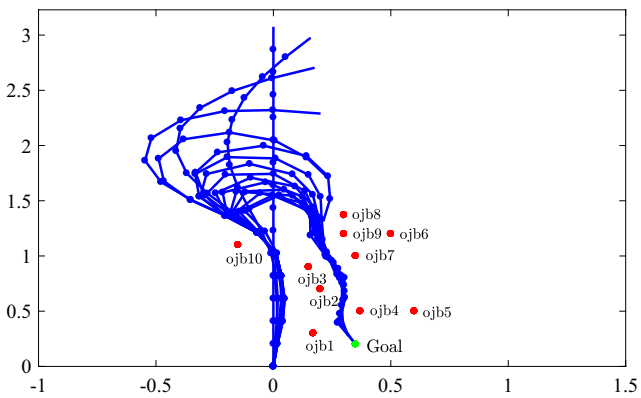

Fig. 1 Sequence in time of collision avoidance simulation for hyper redundant planar manipulator using Newton method

an update time lower than 5 milliseconds was achieved on dual core laptop without code optimization.

We want to note here that our choice of using a hyper redundant manipulator for the tests is motivated by the following:

1. Calculating the Hessian of a hyper redundant manipulator in the conventional way is very difficult. Therefore, we demonstrated the ability of the proposed algorithm to calculate the Hessian efficiently and in real-time, even in high dimensional spaces.
2. The oscillation problem is more evident in high dimensional spaces. Therefore, we demonstrate the effectiveness of the proposed algorithm over the conventional gradient descent to reduce oscillations.
3. The difference in configurations taken by the robot for the various methods is more visible for hyper redundant manipulators.

For comparing the proposed implementation of the Newton method as opposed to the traditional gradient descent based method, two different tests are performed:

- Test 1: In this test a comparison between Newton method and gradient descent is carried out. The comparison criteria is in terms of the quality of the generated path, and the minimum reach distance between the last link of the manipulator and the obstacles;
- Test 2: In this test a comparison between three different methods for performing the optimization is carried out. Those methods are (1) Newton method, (2) gradient descent and (3) gradient descent with momentum. The comparison criteria is in terms of convergence rate.

In both tests the manipulators used are the same, planner, 15 DOF robots, with revolute joints and 0.2 meter link-length. However, the positions of the goal point and the obstacles are not the same.

Table 1 Coordinates of obstacles/goal for Test 1

Object	(x, y) coordinates in meters
Goal	(0.35,0.2)
Obstacle 1	(0.17,03)
Obstacle 2	(0.2,0.7)
Obstacle 3	(0.15,0.9)
Obstacle 4	(0.37,0.5)
Obstacle 5	(0.6,0.5)
Obstacle 6	(0.5,1.2)
Obstacle 7	(0.35,1)
Obstacle 8	(0.3,1.37)
Obstacle 9	(0.3,1.2)
Obstacle 10	(-0.15,1.1)

3.1 Test 1

In this test the Newton method applied to the problem of collision avoidance and path planning for high DOF manipulators is compared with the gradient descent method. For this purpose the following MATLAB simulation is proposed. Two identical planar robots with 15 DOF are navigating towards the same goal point, while avoiding collision with 10 circle-shaped obstacles. The (x,y) coordinates of the obstacles and the goal point in Test 1 are as shown in Table 1. In the beginning of the simulation, the two manipulators are coincident with each other, dashed line in Fig. 2. The blue manipulator is controlled using the Newton method and the red manipulator is controlled using the gradient descent method. For performing the comparison the same goal, obstacles, potential fields are used for both manipulators. The only difference is in the optimization direction used, where for the red robot the gradient method is used to calculate the optimization direction, while for the blue manipulator the Newton method is used to calculate the

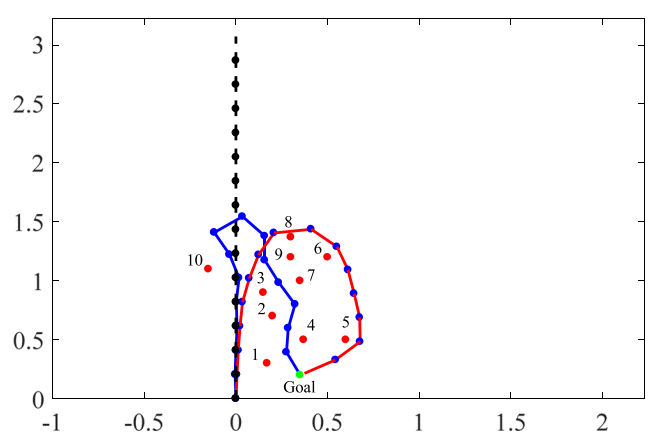

Fig. 2 Test 1: comparison between configurations of the two manipulators after a period of time from the beginning of the simulation

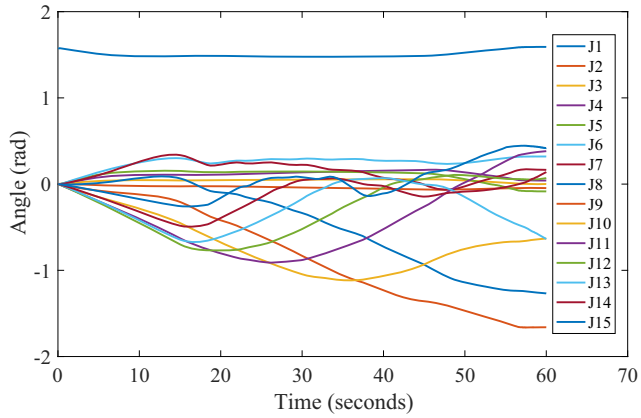

Fig. 3 Test 1: joint angles (Newton method)

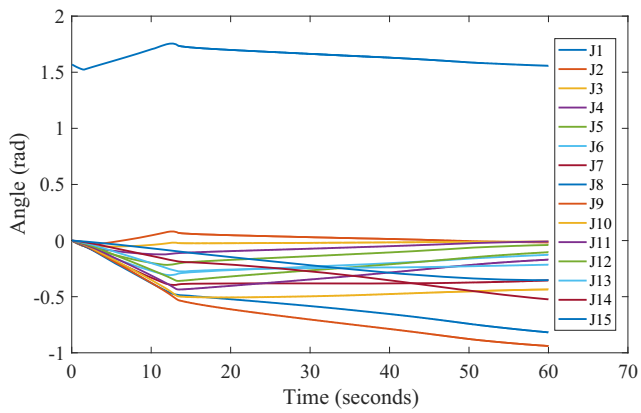

Fig. 4 Test 1: joint angles (gradient method)

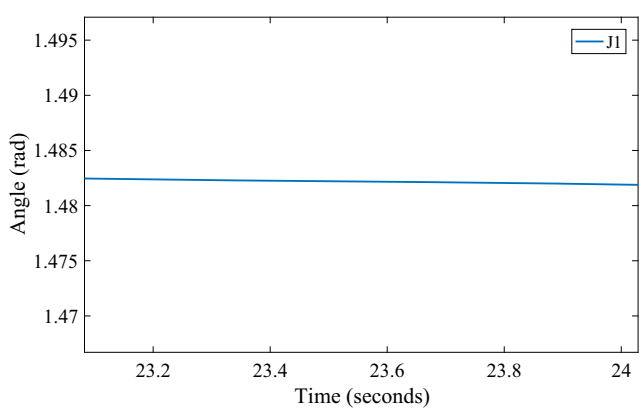

Fig. 5 Test 1: first joint angle close up (Newton method)

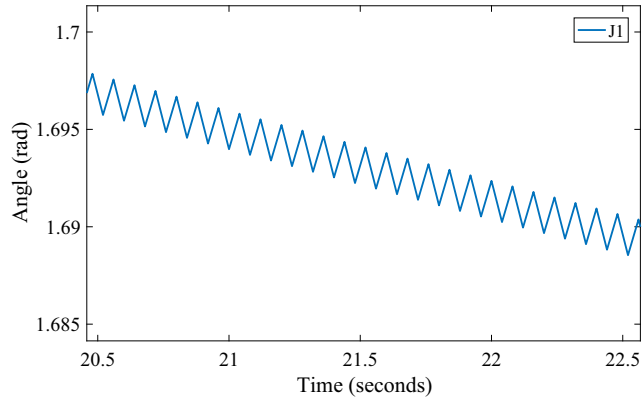

Fig. 6 Test 1: first joint angle close up (gradient method)

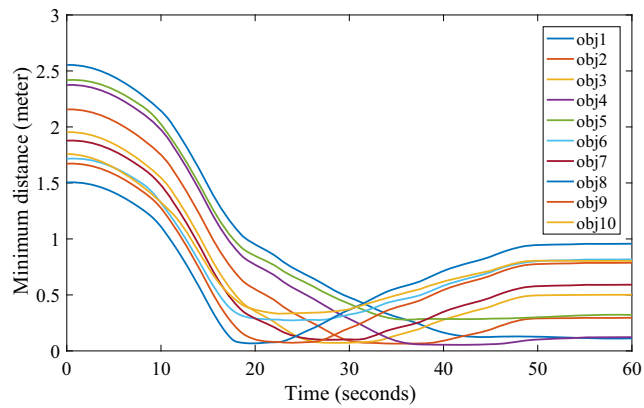

Fig. 7 Test 1: minimum distance between last link and obstacles (Newton method)

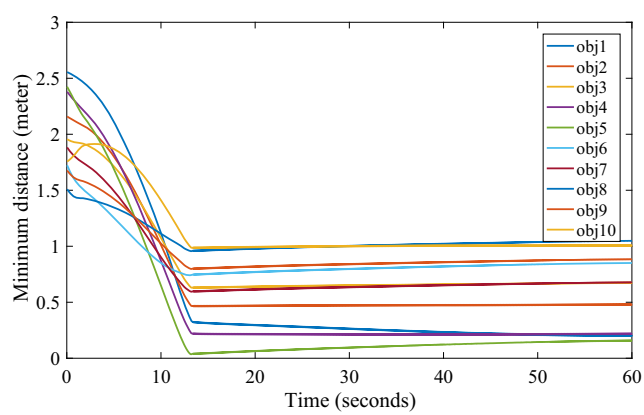

Fig. 8 Test 1: minimum distance between last link and obstacles (gradient method)

Table 2 Test 1. Minimum reach-distance $\rho_{(15,j)} - \rho_0$ between last link of the manipulator and obstacles

Method	Minimum distance (m)
Gradient descent	0.0356
Newton	0.0540

optimization direction. For both robots the magnitude of the optimization step α is the same, in other words the update equation for both robots is:

$$q^{m+1} = q^m - \alpha s \tag{7}$$

Where s is the optimization direction. In the case of the Newton method s is given by:

$$s = \frac{(\mathrm{H} + \lambda \mathrm{I})^{-1} g}{\left\| (\mathrm{H} + \lambda \mathrm{I})^{-1} g \right\|} \tag{8}$$

In case of the gradient method:

$$s = \frac{g}{\|g\|} \tag{9}$$

Figure 2 shows the final configurations taken by the two manipulators. Figures 3 and 4 show joint angles for this simulation as recorded in the Newton and the gradient descent methods. After performing the simulation the two methods are compared in terms of:

1. Quality of the path: The Newton method generated an oscillation free path. On the other hand, the gradient method suffered intermittent oscillations. This is evident by comparing Figs. 5 and 6, where the angular position of the first joint is plotted during the Newton and during the gradient tests. It is worth mentioning that the oscillations, or the zig-zag phenomena as known

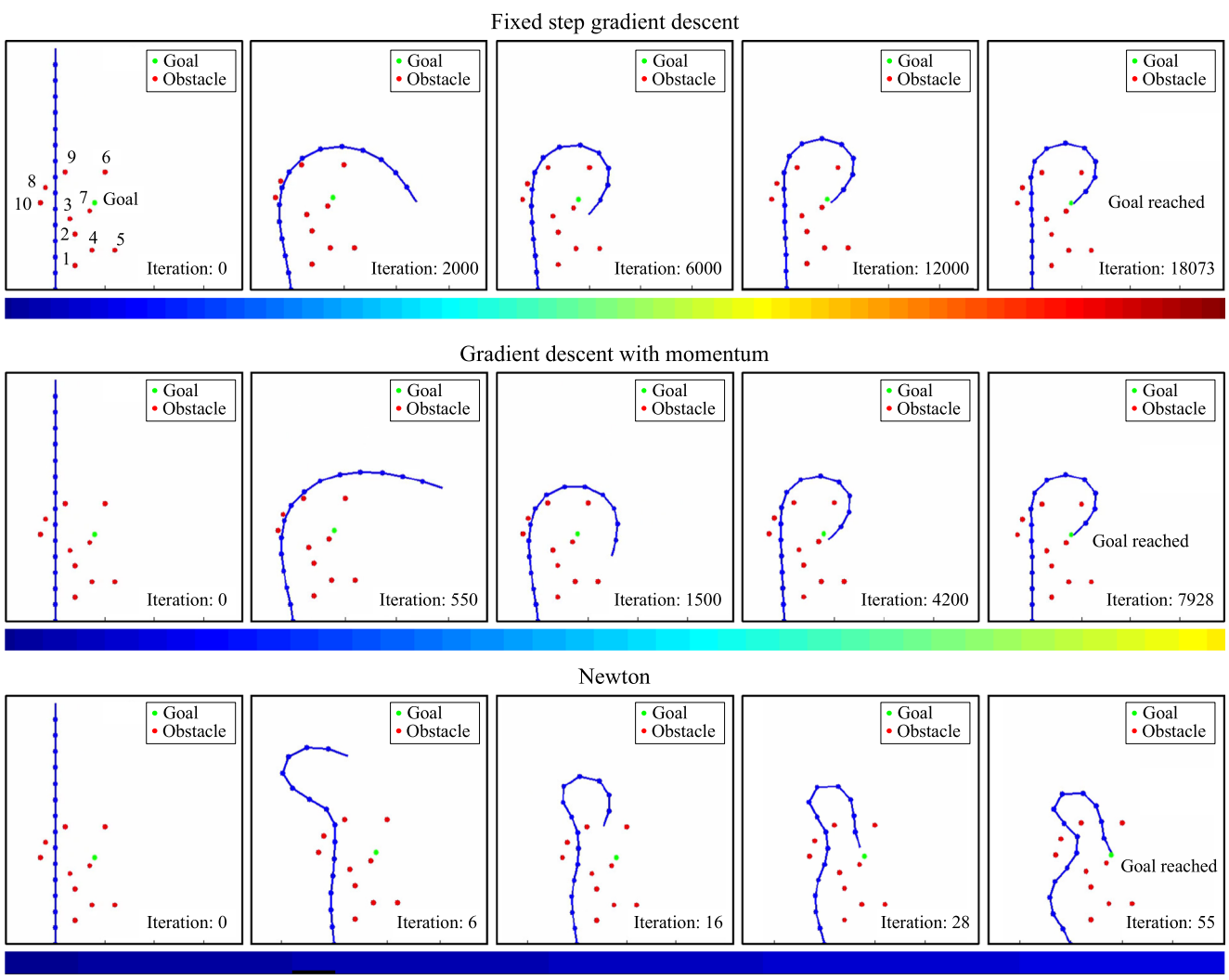

Fig. 9 Test 2: time line during simulations of 15 DOF planar manipulator for three different optimization methods

Table 3 Coordinates of obstacles/goal for Test 2

Object	(x, y) coordinates in meters
Goal	(0.4,1.1)
Obstacle 1	(0.2,0.3)
Obstacle 2	(0.2,0.7)
Obstacle 3	(0.15,0.9)
Obstacle 4	(0.37,0.5)
Obstacle 5	(0.6,0.5)
Obstacle 6	(0.5,1.5)
Obstacle 7	(0.35,1)
Obstacle 8	(-0.1,1.3)
Obstacle 9	(0.1,1.5)
Obstacle 10	(-0.15,1.1)

in the optimization community, happens in the gradient method when one component of the gradient keeps flipping direction during the minimization, as such the minimization point keeps overshooting in the vicinity of a cleavage on the optimization surface [19], this phenomena does not appear in the Newton method.

2. Distance between robot and obstacles: In Newton method the robot keeps longer distance away from obstacles than in the gradient method where the robot reaches closer to obstacles, despite the fact that identical potential fields, geometries, and optimization steps are used. This is demonstrated in Figs. 7 and 8, where the distances $\rho_{(15,j)} - \rho_0$ between the last link of the robot (link 15) and the obstacles, $j = 1, 2..10$, are plotted for the Newton method in Fig. 7, and for the gradient method in Fig. 8. Numerical values of the minimum distance reached during the simulations are listed in Table 2.

3.2 Test 2

Test 2 is performed to asses the convergence rate for the Newton method against the gradient descent and the gradient descent with momentum [28] as applied to the problem of collision avoidance and path planning. The three methods are applied in simulation environment, Fig. 9. In all cases the simulation scenes are identical, the coordinates of the obstacles and the goal points are as shown in Table 3. The distance between the EEF and the goal position is used as the convergence criteria, when the EEF is closer than 0.02 meter from the goal position, the algorithm is terminated and the iterations number is reported. In all three methods the step size α is taken as big as possible, given that the algorithm does not cause the robot to overshoot through any of the obstacles. From the tests the Newton method achieved the best convergence per iteration, followed by the momentum and lastly the gradient method. The convergence rates achieved are as shown in Table 4.

Table 4 Test 2. Convergence rate for three different optimization methods as applied to collision avoidance and path planning problems for robotic manipulators

Method	Iterations
Newton	55
Fixed step gradient descent	18073
Gradient descent with momentum	7928

4 Conclusion

This study addressed the application of Newton's method to the standard artificial potential fields for performing collision avoidance and path planning in high DOF manipulators. Using mathematical optimization, the collision avoidance problem is reformulated. The Newton method is utilized for performing the optimization. An efficient symbolic formula for calculating the Hessian of robotic manipulators is also presented. For evaluating the performance of the Newton method, a comparison with gradient based methods is performed in two different tests. In test 1 the Newton method and the gradient method are compared in terms of quality of the path generated, oscillations, and the minimum distance to obstacles. In test 2 three different methods, (1) Newton, (2) gradient descent, and (3) gradient descent with momentum are compared in terms of the convergence rate. According to the presented results the proposed method achieves better performance. Future work will be dedicated to applying the proposed technique for controlling industrial robotic manipulator in human centered environments to achieve real-time collision avoidance with dynamic obstacles.

Acknowledgments This research was partially supported by Portugal 2020 project DM4Manufacturing POCI-01-0145-FEDER-016418 by UE/FEDER through the program COMPETE 2020, and the Portuguese Foundation for Science and Technology (FCT) SFRH/BD/131091/2017 and COBOTIS (PTDC/EME-EME/32595/2017).

Appendix: Gradient and Hessian Formulas Expressed in Joint Space

In this section, the formulas for the first and the second order derivatives, gradient (4) and Hessian (3), of the potential field with respect to manipulator's joint angles are deduced. Let $u_k(x)$ be a potential field, this potential field is described in Cartesian space. It could be an attraction potential that attracts the EEF to the goal, or a repulsion potential that

repels the robot away from obstacles. Then the second order approximation of the potential field $u_k(x)$ near the current position x_0 is given by:

$$u_k(x) \approx u_k(x_0) + \triangle x^{\mathrm{T}} \nabla u_k + \triangle x^{\mathrm{T}} \nabla^2 u_k \triangle x \tag{10}$$

Where:

- $u_k(x)$ is the approximate value of the potential function at the Cartesian position x ;
- $u_k(x_0)$ is the value of the potential function at the Cartesian position x_0 ;
- ∇u_k is the gradient of the potential function u_k with respect to x taken at the point x_0 ;
- $\triangle x = x - x_0$;
- $\nabla^2 u_k$ is the Hessian of the potential function u_k with respect to x taken at the point x_0 .

In addition, differential kinematics gives the following relationship between elemental displacements in Cartesian space and elemental displacements in joint space:

$$\triangle x = \mathrm{J} \triangle q \tag{11}$$

Where J is the Jacobian associated with the point of the manipulator x_0, and $\triangle q = q - q_0$. By substituting (11) in (10) and fixing:

$$u_k(x) \approx u_k(x_0) + \triangle q^{\mathrm{T}} \mathrm{J}^{\mathrm{T}} \nabla u_k + \triangle q^{\mathrm{T}} \mathrm{J}^{\mathrm{T}} \nabla^2 u_k \mathrm{J} \triangle q \tag{12}$$

From the previous equation the gradient and the Hessian formulas are deduced:

- The gradient is associated with the first order term of the approximation:

$$g_k = \mathrm{J}^{\mathrm{T}} \nabla u_k \tag{13}$$

- The Hessian is associated with the second order term of the approximation:

$$\mathrm{H}_k = \mathrm{J}^{\mathrm{T}} \nabla^2 u_k \mathrm{J} \tag{14}$$

Finally, in case of a manipulator subjected to several potential fields simultaneously, the total potential field is the sum of all of the potentials, and the total gradient g is the sum of the individual gradients:

$$g = \sum g_k \tag{15}$$

The total Hessian is the sum of the individual Hessians:

$$\mathrm{H} = \sum \mathrm{H}_k \tag{16}$$

References

1. Nakamura, Y.: Advanced robotics: redundancy and optimization. Addison-Wesley (1990)
2. Sciavicco, L., Siciliano, B., Villani, L.: Lagrange and Newton-Euler dynamic modeling of a gear-driven robot manipulator with inclusion of motor inertia effects. Adv. Rob. **10**(3), 317–334 (1995)
3. Flacco, F., De Luca, A., Khatib, O.: Control of redundant robots under hard joint constraints: Saturation in the null space. IEEE Trans. Rob. **31**(3), 637–654 (2015)
4. Flacco, F., Kröger, T., De Luca, A., Khatib, O.: A depth space approach to human-robot collision avoidance. In: 2012 IEEE International Conference Onrobotics and Automation (ICRA), pp. 338–345 (2012). https://doi.org/10.1109/ICRA.2012.6225245
5. Polverini, M.P., Zanchettin, A.M., Rocco, P.: Real-time collision avoidance in human-robot interaction based on kinetostatic safety field. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4136–4141 (2014). https://doi.org/10.1109/IROS.2014.6943145
6. Žlajpah, L., Petrič, T.: Obstacle avoidance for redundant manipulators as control problem. In: Kucuk, S. (ed.) Serial and Parallel Robot Manipulators, pp. 10–13. IntechOpen (2012). https://doi.org/10.5772/32651
7. Flacco, F., De Luca, A.: A reverse priority approach to multi-task control of redundant robots. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2421–2427 (2014). https://doi.org/10.1109/IROS.2014.6942891
8. Singh, L., Stephanou, H., Wen, J.: Real-time robot motion control with circulatory fields. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2737–2742 (1996). https://doi.org/10.1109/ROBOT.1996.506576
9. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 566–580 (1996)
10. Haddadin, S., Belder, R., Albu-Schäffer, A.: Dynamic motion planning for robots in partially unknown environments. In: IFAC Proceedings, pp. 6842–6850 (2011). https://doi.org/10.3182/20110828-6-IT-1002.02500
11. Kim, J., Pearce, R.A., Amato, N.M.: Extracting optimal paths from roadmaps for motion planning. In: 2003 IEEE International Conference on Robotics and Automation (ICRA), pp. 2424–2429 (2003). https://doi.org/10.1109/ROBOT.2003.1241956
12. Leven, P., Hutchinson, S.: A framework for real-time path planning in changing environments. Int. J. Rob. Res. **21**, 999–1030 (2002)
13. Dittrich, F., Puls, S., Wörn, H.: A modular cognitive system for safe human robot collaboration: design, implementation and evaluation. In: Robotic Assistance Technologies in Industrial Settings (2013)
14. Kunz, T., Reiser, U., Stilman, M., Verl, A.: Real-time path planning for a robot arm in changing environments. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5906–5911 (2010). https://doi.org/10.1109/IROS.2010.5653275
15. Shukla, A., Singla, E., Wahi, P., Dasgupta, B.: A direct variational method for planning monotonically optimal paths for redundant manipulators in constrained workspaces. Rob. Auton. Syst. **62**, 209–220 (2013)
16. Qiu, J., Sun, K., Wang, T., Gao, H.: Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance. IEEE Trans. Fuzzy Syst. **27**(11), 2152–2162 (2019)
17. Sun, K., Mou, S., Qiu, J., Wang, T., Gao, H.: Adaptive fuzzy control for nontriangular structural stochastic switched nonlinear systems with full state constraints. IEEE Trans. Fuzzy Syst. **27**(8), 1587–1601 (2019)
18. Palm, R.: Control of a redundant manipulator using fuzzy rules. Fuzzy Set Syst. **45**, 279–298 (1992)
19. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
20. Ren, J., McIsaac, K.A., Patel, R.V.: Modified newton's method applied to potential field-based navigation for mobile robots. IEEE Trans. Rob. **22**(2), 384–391 (2006)

21. Ren, J., McIsaac, K.A., Patel, R.V.: Modified newton's method applied to potential field-based navigation for nonholonomic robots in dynamic environments. Robotica **26**(1), 117–127 (2008)
22. McIsaac, K.A., Ren, J., Huang, X.: Modified Newton's method applied to potential field navigation. In: 42nd IEEE International Conference on Decision and Control, pp. 5873–5878 (2003)
23. Denhavit, J., Hartenberg, R.S.: A kinematic notation for lower-pair mechanisms based on matrices. ASME J. Appl. Mech. (1955)
24. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. Wiley, New Jersey (2013)
25. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Rob. Res. **5**(1), 90–98 (1986)
26. Klein, C.A., Blaho, B.E.: Dexterity measures for the design and control of kinematically redundant manipulators. Int. J. Rob. Res. **6**(2), 72–83 (1987)
27. Buss, S.R.: Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. https://www.math.ucsd.edu/sbuss/ResearchWeb/ikmethods/iksurvey.pdf. Accessed 6 December 2009 (2009)
28. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Networks **12**(1), 145–151 (1999)