

INVERSE KINEMATIC MULTIOBJECTIVE OPTIMIZATION FOR A VEHICLE-ARM ROBOT SYSTEM USING EVOLUTIONARY ALGORITHMS

HUMBERTO RODRÍGUEZ,

ILKA BANFIELD

*Mechanical Engineering Department, Universidad Tecnológica de Panama,
Panama City, Panama*

This work is aimed at discussing the solution of the inverse kinematic problem using Multi-Objective Evolutionary Algorithms (MOEA) for a vehicle-arm redundant robot. A simplified 5 DoF model was used to simulate the problem and the objective functions were properly selected assuming underwater operation. In addition, we present a review of the most important techniques used for solving the inverse kinematic problem, focusing at the end on the application of a Non-Dominated, Sorting, Elitist MOEA with nonlinear constraints.

1. Introduction

Currently, robotics plays an important role in increasingly complex engineering applications, with high demands in terms of dexterity, such as the performance of underwater robots; which in most cases are redundant [1-4]. In this work, we are interested in the case of redundant systems composed of a mobile platform and a manipulator arm, that could be used for ship's hull cleaning and maintenance.

The buildup of organisms on the side of boat hulls, propellers, and other infrastructure in marine environments increases fuel consumption and ecosystem's problems of invasive species. Many ship owners periodically deploy divers to inspect ship hulls and remove the buildup of organism, known as Biofouling [5]. Hull cleaners are autonomous or semi-autonomous underwater robot used to scrub hull clean while still in the water. The use of Robot cleaners can result in fuel savings and reduce the risk of the task itself.

There are several functional prototypes and patents developed for the inspection and cleaning of underwater surfaces. We can classify the different cleaning systems as, fastening methods to the hull surface and maneuverability mechanisms [6-9]. The results of this work will be applied to development of a manipulator on a free-floating submarine vehicle for inspection and cleaning of the target. Cui and others [10] reviewed the different challenges in terms of navigation of a mobile base and manipulation in conjunction with a robotic arm from four important aspects: positioning of the mobile robot by means of (GNSS), navigation based on vision and visual servo, robotic manipulation, planning and control of trajectories.

The present development analyzes the implementation of a method for the control of a redundant robot performing multiple prioritized tasks in the presence of limits in the joint range, speed and acceleration / torque efficiently.

The redundancy problem is described such that, given a desired trajectory, $r_d(t) \in R^m$, defined in a coordinate system for the end effector of a manipulator, it requires the corresponding displacements in the joint space of the robot, the sets of vectors $\theta(t) \in R^n$, where $m < n$, using the definition: $r_d(t) = f(\theta(t))$, where $f(\cdot)$ is the direct kinematics, which is nonlinear and differentiable with a structure and known parameters, for a given manipulator robot.

Several methods have been proposed as resolutions for the redundancy problem, and to characterize multiple approaches, two general methods have been proposed [11], indirect methods and direct methods. Indirect methods, such as the Pseudoinverse method, the Extended Jacobian method, kinematic optimization methods and the Lyapunov method or gradient method, are differential methods that are computationally intensive, so they are developed offline under previously specified optimization criteria. The main disadvantages with this type of methods are that solution of the optimization problem only guarantees to be locally optimal and that the algorithms used exhibit numerical difficulties when the manipulator is close to kinematic singularities [12-14]. Differential control algorithms must be initialized with some joint configuration, θ , from the set of possible solution. Then the manipulator is kinematically controlled by the desired trajectory while remaining within the set of solutions [15].

On the other hand, there are direct methods based on the approach of solving the inverse function of $f(\cdot)$ in a closed form or approximate form. Direct methods seek the explicit resolution of redundant degrees of freedom. These tend to be fast, but generally require strong assumptions regarding the structure of the set of possible solutions [11]. Generally, task dependent constraints need to be specified to accommodate the redundancy.

The inverse functions are local and cyclical and because their evaluation is frequently rapid, when these can be determined, they are computationally suitable for real time control [16]. It must be considered that expressions in closed form for inverse kinematics are difficult if not impossible to determine, however it is possible to adapt approximate forms of these using neural networks or other approximation methods for nonlinear functions.

Traditionally, three models have been used to solve the inverse kinematic problem: geometric models, algebraic models and iterative models [17]. The use of geometric models is limited by the complexity of the structure, while algebraic models do not guarantee a closed form solution [18]. In iterative methods, convergence depends on the initial point of evaluation and are computationally prohibitive, which is why many researchers have focused their work on solving inverse kinematics using Artificial Neural Networks (ANN).

The application of Neural Networks to construct self-organized maps of the inverse kinematic problem, are well referenced [18-20]. In those works, the kinematics of non-redundant and redundant robots is solved after training the ANN for a period, yielding a unique solution corresponding to the desired position in the workspace. Recently, the work of hybrid approaches based on Neural

Networks has allowed the training of ANN with variable optimization criteria over time, which was not possible on earlier works [10]. As example, Jin, L. and Li, S [21], use a Dynamic Neural Network to solve a problem of optimization of the manipulability of a redundant robot, transforming it into the resolution of a set time variants non-linear equations. However, the major drawback of using ANN are still the difficulty of how to collect training sets, and that the training process when gradient-based learning algorithms is very slow, especially for a complex configuration, or a large set of training data.

2. Genetic Algorithms and Multi-Objective Evolutionary Algorithms

When formulating the redundancy problem, as a case of optimization, the techniques based on Genetic Algorithms (GAs), take relevance. The GAs are methods that solve constrained and unconstrained optimization problems and the popularity of their use is because they are naturally appropriate, with some modifications, for cases of multi-objective optimization. GA allows a population composed of many individuals to evolve under specific rules of selection towards a state that maximizes fitness (function aptitude) or cost function. The GA subsequently modifies the population of individuals, which represent the possible solutions. On successive generations, the population evolves towards an optimal solution. This method solves a variety of optimization problems such as discontinuous, non-differentiable, stochastic or highly non-linear functions, which are intractable under other optimization techniques [22].

Non-dominated Sorting Genetic Algorithm (NSGA) [22], was one of the first MOEAs, and from its use the following disadvantages have been highlighted [23]: a high computational cost, lacks elitism, and finally must specify the exchange parameter, which is a parameter that ensures diversity and equivalence in the solutions. From here on, various modifications and new MOEAs algorithms have been developed [23, 24].

2.1. MOEAs Constraints Management

This work involved the optimization of multiple cost functions to solve the kinematic redundancy of a robot manipulator, subject to several constrains equations, which are related to the task and the robot configuration itself. In general, an optimization problem can be defined as,

$$\begin{aligned}
 \min \text{ or } \max \quad & f_m(x), \quad m = 1, 2, \dots, M; \\
 \text{subject to} \quad & g_i(x) \geq 0, \quad i = 1, 2, \dots, I; \\
 & h_j(x) = 0, \quad j = 1, 2, \dots, J; \\
 & x_k^{inf} \leq x_k \leq x_k^{sup}, \quad k = 1, 2, \dots, K;
 \end{aligned} \tag{1}$$

where $f_m(x)$ are the fitness or cost functions and $g_i(x) \geq 0$ and $h_j(x) = 0$ are inequalities and equalities constraints, respectively. The constrained optimization divides the search space into two regions; feasible (those solutions that comply with the constraints) and not feasible. It's clear that the set of optimal solution, Pareto-optimal, must belong to the feasible region. A possible solution would be for the MOEA to assign more pressure to the group that violates the constraints to a lesser degree, so in this way the algorithm is provided with a direction in the search for a feasible region [22].

Another popular technique for handling constraints is the penalty function approach, in which, the objective and the normalized constraints are added and multiplied by a factor that penalizes them based on a minimization process. A drawback is that this factor needs to be updated according to some strategy. In the literature we can find several works [17-27], which present static and dynamic strategies for updating the penalty factor according to the objectives. The quality of the solution obtained depends on factor's quality. Recently, in the works of Fan, Z. *et al.* and Chehouri, A. *et al.* [27, 28], the authors claim that the penalization approach deviates from the philosophy of evolution of the algorithms and propose a technique that preserves the main concept of the GAs, developing rules that include the value variations of the constraints for the selection and generation of new individuals. In this work, the last approach was used for handling the nonlinear equality constraints.

3. Inverse Kinematic Solution of a Vehicle-Arm Robot

In this section we present the solution of the inverse kinematic problem of a vehicle-arm robot model using a Non-Dominated, Sorting, Elitist Algorithm MOEA II, developed by Deb, K. [23]. The fitness functions were selected to maximize manipulability and minimize joint angle average displacement when moving to consecutive points of the trajectory.

For the application of the MOEA II, the individuals of a population are defined as configurations, q , of the vehicle-arm, which will be used to evaluate two fitness (objective) function. Figure 1 shows the 5 DoF planar model and the generalized coordinates used.

In the context of evolutionary multi-objective minimization problems, the term dominance refers to the case when the evaluation of a fitness functions, g_i , for an individual q_1 gives smaller values than for other individual q_2 :

$$\begin{aligned} g_i(q_1) &\leq g_i(q_2) \quad \forall i \\ g_j(q_1) &< g_j(q_2) \quad \text{for some } j \end{aligned} \quad (2)$$

Thus, q_1 dominates q_2 . The group of individuals that have noninferior fitness function values are non-dominated by any other individual and its set is called a *pareto front*. Therefore, for each individual on the Pareto front, one fitness function can only be improved by degrading another.

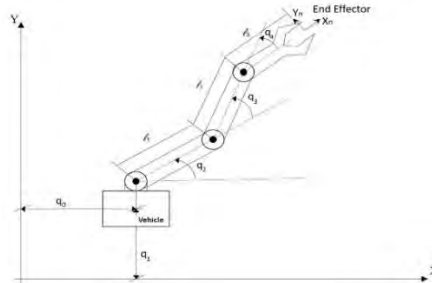


Figure 1. Vehicle – Manipulator Schematic: The generalized coordinates of the robot, q , are shown.

3.1. The MOEA II algorithm

The MOEA II was implemented in MATLAB. The first step in the algorithm is creating an initial population (200 in our case) which is feasible with respect to the nonlinear constraints and the joint angles bounds. In this algorithm both, the objective function and constraints values are used to obtain scores for the population.

After each next generation is created from selected parents, the children are obtained by mutation and crossover. Afterward, all infeasible individuals are assigned a lower rank than any feasible individual. Within the infeasible population, the individuals are sorted by an infeasibility measure, which we take as the value of the constraints function. For the new generation, the current population is combined with the children.

For a controlled elitist GA, it is important to maintain the diversity of population for convergence to an optimal Pareto front (23). The diversity is increase by keeping some of those individuals of the current population that are relatively far away of the pareto front. In addition, the number of individuals on the Pareto front (elite members) is limited using a fix fraction.

3.2. The Model and Simulation Results of the Vehicle-Arm system

The direct kinematic function $f(q)$, in equation 3, defines the position and orientation of the end effector (x, y, θ) in terms of the generalized coordinates q_0, q_1, q_2, q_3, q_4 , as shown in figure 1.

$$f(q) = \begin{bmatrix} q_0 + l_1 * \cos(q_2) + l_2 * \cos(q_2 + q_3) + l_3 * \cos(q_2 + q_3 + q_4) \\ q_1 + l_1 * \sin(q_2) + l_2 * \sin(q_2 + q_3) + l_3 * \sin(q_2 + q_3 + q_4) \\ q_2 + q_3 + q_4 \end{bmatrix} \quad (3)$$

For the intended application, it is important to assure a manipulability as high as possible to exert the require forces while following the desired trajectories on the vessel surface. Thus, Yoshikawa's manipulability index was used as the first

Fitness Function. A second function, the square root of the difference between consecutive configurations, $\hat{q}_i^t - \hat{q}_i^{t-1}$, is used to assure that the optimal configuration of two consecutive points in the trajectory are close. In addition, since it is easier to control the joint angles than the vehicle position in the water, a reducing gain, K_i , is used for the first two coordinates in the second fitness function to reduce as much as possible the motion of the vehicle or platform. Thus, the optimization problem is stated as:

$$\min \quad g = \frac{1}{Y(\mathbf{q}) + 1}; \text{ where } Y(\mathbf{q}) = \sqrt{J(\mathbf{q})J^T(\mathbf{q})}$$

$$S(\hat{\mathbf{q}}) = \sqrt{\sum_i K_i (\hat{q}_i^t - \hat{q}_i^{t-1})^2} \quad (5)$$

Subject to: $f(\mathbf{q}) = \hat{\mathbf{x}}_d$; $\mathbf{q}^{inf} \leq \mathbf{q} \leq \mathbf{q}^{sup}$.

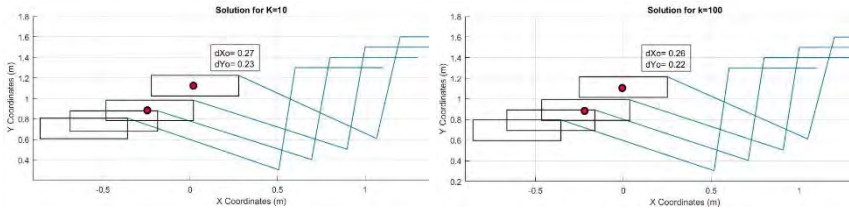


Figure 2. Effect of applying different gains to q_0 and q_1 , with $K=10$ (left) and $K=100$ (right),

Where $\hat{\mathbf{x}}_d$, is the desired position and orientation of the end effector, \mathbf{q}^{inf} and \mathbf{q}^{sup} are the lower and upper bound of the joint angles, and $J(\mathbf{q})$ is the Jacobian matrix, which obtained by direct differentiation of equation. Figure 2, shows some of the typical results obtained when a series of positions of the end effector are given and a fixed orientation pointing in the positive x direction is asked. The effect of the gain K can be appreciated, since the consecutive vehicle/platform positions stay closer when we increase its value. When only K_1 and K_2 are increased, the relative displacement of the vehicle between successive configurations is reduced (shown in the text box for last two), as expected.

With these results we have shown that a non-dominated elitist sorting evolutionary algorithm can be used to solved effectively the inverse kinematic problem of a redundant manipulator using multi-objective optimization. The fitness/objective functions have been selected bearing in mind the characteristics of the task and still need to be validated in actual operational conditions.

4. Conclusions and Future Work

The final individuals or joint coordinates were effectively selected out of the pareto front by choosing the best solution for the smoothness cost function, since the values of the manipulability does not vary much for the different solutions of

the front. It is worth to mention that, to generate populations that expand into feasible and not feasible regions the number of initial individuals was increased up to 200. The next step in this research is to simulate the complete 12 DoF system (6 DoF for the vehicle and 6 DoF on the arm) and implement an Impedance Control System on a ground vehicle + arm manipulator, having the kinematic optimization technique presented in this work as an offline process.

References

1. Rui Y., Benoit C., Ali M., Ming L., Nailong W., Modeling of a complex-shaped underwater vehicle for robust control scheme. *J Intell Robot Syst.* 2015; 80:491-506.
2. Azorin-Lopez J., Saval-Calvo M., Fuster-Guillo A., Oliver A., A predictive model for recognizing human behavior based on trajectory representation. In 2014 International Joint Conference on Neural Networks (IJCNN), pages 1494-1501. IEEE, July 2014.
3. Calinon S., Sardellitti I., Caldwell D. G. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *Proceedings of IROS* pages 249-254. IEEE, October 2010.
4. Cambon S., Alami R., Gravot F., A Hybrid Approach to Intricate Motion Manipulation and Task Planning. *The International Journal of Robotics Research*, 28(1):104-126, January 2009.
5. Coutts AD, Valentine JP, Edgar GJ, Davey A, Burgess-Wilson B., Removing vessels from the water for biofouling treatment has the potential to introduce mobile nonindigenous marine species. *Marine pollution bulletin*, 60(9):1533–1540, (2010).
6. Michael J Maloney & Ronald C Pennock. Boat hull cleaning apparatus. US Patent 5,222,452, June 29 1993.
7. William George Ecklund. “Ship hull cleaning apparatus and method of use”, March 10 2004. US Patent App. 10/798,528.
8. Goldberg, D. “Huxley: a flexible robot control architecture for autonomous underwater vehicles”. *OCEANS*, 2011 IEEE-Spain, pages 1–10. IEEE, 2011.
9. Muelling, K., Venkatraman, A., Valois, J-S., Downey, J. y Weiss, J., *Autonomy Infused Teleoperation with Application to BCI Manipulation*. *Auton Robot*, 41: 1401-1422, 2017.
10. Cui L, Tan T, Duc Do K y Teunissen P. Challenges and Solutions for Autonomous Robotic Mobile Manipulation for Outdoor Sample Collection. *Journal of Electrical and Electronic Engineering*. 3(5): 156-164 (2015).
11. DeMers D. Kreutz-Delgado K. *Inverse Kinematics of Dextrous Manipulator*, Neural Systems for Robotics Academic Press, pp. 75-116 (1997).
12. P. Kinematic singularities of Robot Manipulators. 5772/9548. (2010).

13. Lara-Molina, F.A., Rosario, J.M., Dummur, D., Multi-Objective Design of Parallel Manipulator Using Global Indices, *The Open Mechanical Engineering Journal*, 4, 37-47 (2010).
14. Taghirad H., Nahon M., Kinematic Analysis of a Macro-Micro Redundantly Actuated Parallel Manipulator, *Advanced Robotics* 22, 657-687 (2008).
15. Zhang, T., An, H., Ma, H., Joint torque and velocity optimization for a redundant leg of quadruped robot, *International Journal of Advanced Robotic Systems*, pp 1-12, (2017).
16. DeMers D. Kreutz-Delgado K, Canonically Parameterized families of Inverse Kinematic Functions for Redundant Manipulators, *Robotics and Automation*, pp. 1881-1886, (1994).
17. Korein, J., Balder N., Techniques for generating the goal-directed motion of articulated structures, *IEEE Computer Graphics and Applications*, 2(9), 71-81, (1982).
18. Alavandar, S., Nigam, M., Inverse Kinematics Solutions of 3DOF Planar Robot using ANFIS, *Int. Journal of Computers, Communications and Control*, (3)150-155, (2008).
19. Ding, H., Wang, J., Recurrent Neural Network for Minimum Infinity-Norm Kinematic Control of Redundant Manipulators, *IEEE Trans Syst Man Cybern* 29(3)269-276, (1999).
20. Mao, Z., Hsia, T., Obstacle Avoidance Inverse Kinematics Solution of Redundant Robots by Neural Networks. *Robotica* 15(1)3-10, (1997).
21. Jin, L., Li, S., Manipulability Optimization of Redundant Manipulators Using Dynamics Neural Networks, *IEEE Transactions of Industrial Electronics*, (2016).
22. Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley and Sons, Inc. New York, NY, USA, (2001).
23. Deb, Kalyanmoy, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA II. *IEEE Transactions on Evolutionary Computation* 6(2) 182-197, (2002).
24. Deb, Kalyanmoy, An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation* 18(4) 577-601, (2014).
25. Coello, C., Use of a Self-adaptive penalty approach for engineering Optimization Problems. *Computation Industrial* 41 133-127.
26. Yeniyay, O., Penalty Function Methods for Constrained Optimization with Genetic Algorithms. *Mathematics Computing Applic.* 10 45-56 (2005).
27. Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Push and Pull Search for Solving Constrained Multi-objective Optimization Problems. *Neural and Evolutionary Computing*, (2017).
28. Chehouri, A., Younes, R., Perron, J., Ilinca, A., A Constraint-Handling Technique for Genetic Algorithms using a Violation Factor. *Journal of Artificial Intelligence of Computer Sciences*. 12(7) 350-362. (2016).