Motion planning optimization using hierarchical algorithms for multi-robot cells

Stefano Mutti[1], Giorgio Nicola[2], Nicola Pedrocchi[1]

***Abstract*— In manufacturing, motion planing of multi-robot cells is a demanding activity. The issue of work-piece positioning is still mostly solved using the human competence, by trial and error procedures, making feasible solutions hard to find, and making it a time consuming procedure. Surely, the optimization of a custom objective function along a path is complex, due to the dimension of the feasible-configuration space, its high non-linearity, and to the collision probability along complex trajectories. Furthermore, a human approach to the problem, makes the exploit of the task redundancy, often found in industry, a complex matter. This work proposes a two-layer iterative optimization method that integrates an external Whale Optimization and an internal Ant Colony Optimization algorithm, allowing the optimization of a user-defined objective function along a working redundant path, to produce a quasi-optimal, collision free solution within the feasible-configuration space. The trials result a good trade-off between computing time and solution optimality.**

I. Introduction

Multiple robots configurations are often used in industry for a huge variety of tasks (finishing, cutting, painting, *etc.*) [1], [2], thanks to the dexterity and re-configurability offered(see for example Figure 1). In such a setup, the position of the work-piece greatly affects the feasibility (*e.g.*, respect of the joints speed limits) and the execution accuracy in term of path tracking as consequence of the different robot kinematics parameters(*e.g.*, different joints speed, accelerations, friction, *etc.*). Furthermore, the trajectory has to be collision free, which is a hard requirement for complex paths, especially when the tool path involves re-orientations.

In case of a single robot configuration, a common approach is the optimization of the robotic dexterity on the trajectory [3]–[7], ignoring problems such as backlashes or transmissions elasticity. To cope with such complications, [8]–[12] proposed methods tailored on specific tasks and setup that may be extended to multi-arm work-cells. An interesting similar practice is to optimize the work-cell layout [13], [14], which is hard to carry out in industry, where cells are used for many different tasks, and are hardly modifiable. The main problematic to be tackled in the matter, is the domain search space hugeness, and its non-connected manifold since the constraints are always expressed in the Cartesian space but the inverse kinematics is not a bijective function.

[1] Institute of Intelligent Industrial Technologies and Systems, National Research Council of Italy {name.surname}@stiima.cnr.it

[2] Università di Padova, Department of Information Engineering (DEI), via Gradenigo 6/A, Padua, Italy {name.surname}@unipd.it

contact author: stefano.mutti@stiima.cnr.it

In such a field of research, the here presented work, discussed in the next Section, is an evolution of [15], where the methodology consists of splitting the problem in two sub-problems, and to run iteratively two nested optimizers: first the problem of the object positioning is solved (*e.g.*, definition of the Robot-2 holding position), then, the Robot-1 redundant path is optimized. In order to improve the work, the task redundancy and collision avoidance are now taken into account, furthermore, the inner algorithm is adapted to the graph complexity, trough the adoption of a better algorithm design, and to dynamic parameters tuning.

II. Method

The method is composed by an external **W**ale **O**ptimization **A**lgorithm, in charge of the workpiece positioning, while an internal **A**nt **C**olony **O**ptimization algorithm maximizes the objective function along the robotic path. Given a starting position outputted by each WOA particle at every iteration, a directed weighted graph is formed joining all the consecutive redundant solutions for every frame of the working trajectory, where the task redundancy is discretized in the Cartesian space using Halton points. The graphs nodes weights values are then assigned, based on a kinematic property of the nodes, and solved using a Min-Max Elitist version of the ACO algorithm, which is proven to carry out better results with huge and complex graphs. The solutions are then forwarded to the WOA for the computation of the next workpiece pose. The internal optimization loop ensures the lack of collisions, in order to optimize in proximity of feasible paths only. The parameters for the method are the iterations number of WOA, number of the whales particle, iterations of ACO, and number of ACO particles, while the setting of the heuristics internal parameters is dynamically done according to [16].

Fig. 1: Laser welding performed by three cooperative Robot.

III. Performance Analysis

The software architecture designed, takes advantage of the ROS-Industrial, and it also employs a modified version of the ikfast openrave [17] for parallel computation on GPU. Its architecture is shown in Fig 2. The setup is composed by two ABB IRB4600 configured for pipes laser cutting. The objective function implemented consist of the maximization of the inverse of the quadratic sum of the joints speed along the trajectory. Redundancies have been introduced for the rotation around the cutting head Z axis and, specifically, the head Z axis can freely move inside a given cone with a small aperture. The tested trajectory are an elliptic hole cutting on a pipe side, and a complex pipe tip trimming with a full re-orientation of the cutting head.

All the computations outputted a collision-free solution. At the begin, the algorithm explores the working robot joint space, the hold solutions are diverse for each WOA particle, and the value of the objective function increases considerably, changing at every iteration, due to the WAO algorithm nature and its parameters value. Then, the WOA algorithm particles move in proximity of their best solutions and the exploration is concluded, the search is now focusing on optimizing the best solutions found so-far, settling on the nearest optimum points. The solutions show a huge exploit of the available redundancy, changing the orientation during the movement in order to decrease joint speeds and accelerations during the work head re-orientations. Averagely, the solution frames drifts from the Frenet frame by 40°.

The algorithm running time depends linearly on N_{whales} and N_{WOA}, Figure 3, where the average time and its standard deviation are plotted for different parameters. The running time deviation value is due to the collision checking phase, which is the non-deterministic part of the algorithm in terms of execution time. The bottom part of figure 3 shows the trend of the objective value per whales and whales iteration numbers, showing that it has an asymptotic behaviour, meaning that adding resources to the problem does not increase linearly the solution goodness. This behaviour highlights the fact that the optimization problem is well solved with a limited amount of resources, that the solution joint space is explored adequately by the WOA and the redundancy trajectory graph is fully explored by the ACO.

IV. Conclusion

The method provides a quasi-optimal solution, and it is efficient in resource usage and robustness. The running time is satisfactory but relevant, due to the high amount of offline data to compute, although, the method is highly parallelizable so the calculation time is expected to decrease with usage of high-end technologies. Results show that the exploitation of the task redundancy is a key element to the optimization.

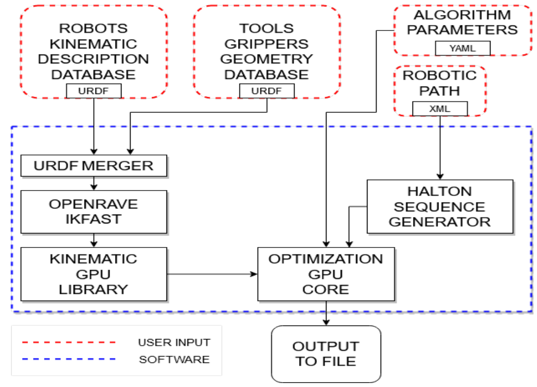

Fig. 2: Software architecture

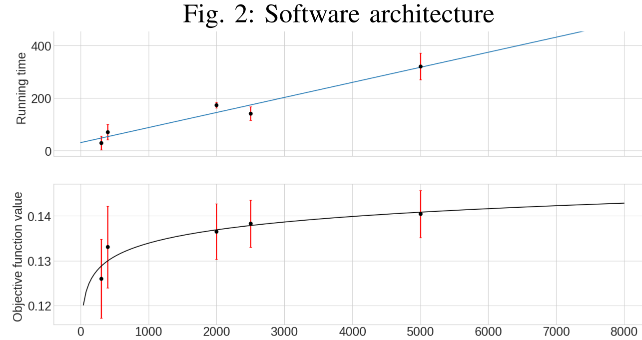

Fig. 3: Algorithm running time (top) and objective function value(bottom), iterations on the abscissa.

References

[1] Y. Chen and F. Dong, "Robot machining: recent development and future research issues," *Adv. Manu. Tech.*, vol. 66, no. 9-12, pp. 1489–1497, 2013.

[2] W. Ji and L. Wang, "Industrial robotic machining: a review," *Int. J. Adv. Manu. Tech.*, vol. 103, no. 1-4, pp. 1239–1255, 2019.

[3] T. Yoshikawa, "Manipulability and redundancy control of robotic mechanisms," *Proc. 1985 IEEE Int. Conf. on Rob. and Aut.*, vol. 2, pp. 1004–1009, 1985.

[4] J. Angeles and C. S. López-Cajún, "Kinematic Isotropy and the Conditioning Index of Serial Robotic Manipulators," *Int. J. Robotics Research*, vol. 11, no. 6, pp. 560–571, 1992.

[5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics, Modelling, Planning and Control*. Springer, 2009.

[6] G. Legnani *et al.*, "The "point of isotropy" and other properties of serial and parallel manipulators," *Mec. and Mach. Theory*, vol. 45, no. 10, pp. 1407 – 1423, 2010.

[7] J. Lachner *et al.*, "The influence of coordinates in robotic manipulability analysis," *Mec. and Mach. Theory*, vol. 146, 2020.

[8] J. T. Feddema, "Kinematically optimal robot placement for minimum time coordinated motion," *IEEE Int. Conf. on Rob. and Aut.*, 1996.

[9] G. Pamanes and S. Zeghloul, "Optimal placement of robotic manipulators using multiple kinematic criteria," *Proc. IEEE Int. Conf. on Rob. and Aut.*, vol. 1, no. April, pp. 933–938, 1991.

[10] B. Kamrani *et al.*, "Optimal robot placement using response surface method," *Int. J. Adv. Manuf. Tech.*, vol. 44, no. 1-2, pp. 201–210, 2008.

[11] R. R. dos Santos, V. Steffen, and S. d. F. Saramago, "Optimal Task Placement of a Serial Robot Manipulator for Manipulability and Mechanical Power Optimization," *Intelligent Information Management*, vol. 02, no. 09, pp. 512–525, 2010.

[12] G. C. Vosniakos and E. Matsas, "Improving feasibility of robotic milling through robot placement optimisation," *Rob. and Comp.-Int. Man.*, vol. 26, no. 5, pp. 517–525, 2010.

[13] M. Tay and B. Ngoi, "Optimising robot workcell layout," *Int. J. Adv. Manu. Tech.*, vol. 12, no. 5, pp. 377–385, 1996.

[14] S.-W. Son and D.-S. Kwon, "A convex programming approach to the base placement of a 6-dof articulated robot with a spherical wrist," *Int. J. Adv. Manu. Tech.*, vol. 102, no. 9-12, pp. 3135–3150, 2019.

[15] G. Nicola, N. Pedrocchi, S. Mutti, P. Magnoni, and M. Beschi, "Optimal task positioning in multi-robot cells, using nested metaheuristic swarm algorithms," *Proc. CIRP*, vol. 72, pp. 386–391, 2018.

[16] P. Pellegrini, D. Favaretto, and E. Moretti, "On MAX – MIN Ant System's Parameters," in *Int. Conf. on Swarm Intelligence*. Springer, Berlin, Heidelberg, 2006, pp. 203–214.

[17] OpenRAVE, http://openrave.org/docs/0.8.2/openravepy/ikfast/, 2020, accessed: 8.9.2020.