

Path optimization for redundant manipulator to Travelling Salesman Problem (TSP)

Shinde V.B., Pawar P.J.

Department of Production Engineering, K.K. Wagh Institute of Engineering Education & Research, Nashik,
Savitribai Phule Pune University, Pune, Maharashtra, India
vbshinde11@rediffmail.com, pjpawar1@rediffmail.com

Abstract:

In this paper, a minimum time path planning approach is proposed for point to point (PTP) motions manufacturing problems in drilling/spot welding tasks. By optimizing the travelling schedule of the set points and the detailed transfer path between points, the minimum time manufacturing task is realized. According to the start-stop movement in drilling/spot welding task, the path planning problem can be converted into a traveling salesman problem (TSP) and a series of point to point minimum time transfer path planning problems. fmincon constrained nonlinear minimization Matlab function is used to parameterize the transfer path and then the path parameters are optimized to obtain minimum point to point transfer time. A new TSP with minimum time index is constructed by using point-point transfer time as the TSP parameter. The real coded genetic algorithm (RCGA) is applied to obtain the optimal travelling schedule. Point to point task of a 4 - DOF robotic manipulator is used as examples to demonstrate the effectiveness of the proposed approach.

Keyword:, Real Coded Genetic Algorithm (RCGA), transfer time, path planning

1. Introduction

This work presents a method for combining simple optimal motions in an optimal manner that makes it possible to plan an important class of complex manipulator tasks so that they can be performed in minimum time. Those tasks are composed of moves between points, where the manipulator comes to a stop at each point. Drilling, mechanical assembly, insertion of electronic components, and spot welding operations are common examples. The order in which manipulators must go to a number of positions and return to a starting point is not critical in many industrial operations, such as in most spot-welding tasks. However, that order can significantly affect the total time required to perform the task. If there are n points in the operation then there will be $(n - 1)!$ possible combinations; finding the order, or tour, with the shortest possible time by trial and error is not practical. This problem is nearly identical to the well-known Traveling Salesman Problem (TSP) in mathematics. There the objective is to find the tour of all cities in a set with the shortest total distance. A great deal of study has been done to develop algorithms that avoid

evaluating all $(n-1)!$ tours in finding the shortest tour. Some algorithms give exact solutions and some give approximate solutions. Path planning and control of robot manipulators require mapping from end effector Cartesian space coordinates into corresponding joint positions. This mapping is referred to as the inverse kinematics (IK) of the robot. Finding the position and orientation of the end-effector from the joint angles is called the forward-kinematics (FK) problem. Forward kinematics of a robot manipulator can easily be solved by knowing the link parameters and joint variables of a robot, while the inverse kinematics is a nonlinear and configuration dependent problem without unique solution. To achieve desired position and orientation of end effector or tool, so as to complete the pre-specified task is an important part of industrial robot arm. Hence, industrial manipulator can achieve a desired task or end effector position in more than one configuration. To achieve the above stated goal one should have the sound knowledge of inverse kinematic problem. The problem of getting inverse kinematic solution has been on the outline of various researchers

and is deliberated as thorough researched and mature problem.

To overcome difficulty of ineffective non redundant industrial robot manipulators, robots should possess at least one degree-of-freedom more than the number required for the general free positioning, i.e. they should be redundant. Redundant manipulators can be also most effective to resolve the task of the automatic assembly and demounting of the complex equipment, repair, and works which cannot effectively perform manually. As from the literature review, it is fair that, for Inverse kinematics problem of redundant robot manipulator under the conditions of restrictions, there is the strong need to develop methodology and optimization program, which will be able to consider any general configuration i.e. any number of links and any number of obstacles as input.

A problem of 4 Degree of freedom redundant robotic manipulator is considered. To find the tour for visiting the entire task points (once each one) so that minimum travel time is determined, taking into account the multiplicity of the robot configurations corresponding to every task point. The number of possible solutions depends at a great extent on the number of possible configuration.

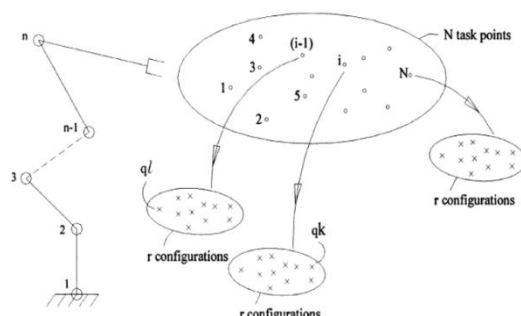


Fig.1 A schematic of a n-DOF Manipulator, the n-task points and the r configuration corresponding to every task point

The determination of fitness function for the inverse kinematic solution is based on the end effectors initial and desired position error which is also known as Euclidean distance norm. By solving Inverse Kinematics for each point optimized values

of joint angle of all joint coordinates are determined using fmincon constrained nonlinear minimization function in Matlab. Optimization variables of this function are the robot configuration corresponding to each task point. Real coded genetic algorithm (RCGA) is used for the determination of optimal sequence of a redundant manipulator's end effector, i.e. the sequence that guarantees the minimum cycle time, taking into consideration its multiple configurations.

In the next section, the optimization method Real Coded Genetic Algorithm (RCGA) is presented.

Real Coded Genetic Algorithm (RCGA)

Genetic algorithm (GA) is based on the concept of nature's law of survival of fittest, natural selection procedure and a population based stochastic search algorithm and. As per the principle of survival of fittest stronger ones live for many years, they reproduce many off-springs by passing on many attributes to their children, while weaker ones die early without reproduction (Goldberg, 1989). Binary code, Real Code and Integer code are types of the representation of Genetic Algorithm. Out of that, Real Coded Genetic Algorithm (RCGA) is used in this paper. In RCGA real numbers are used for the representation of the solution which avoids the discretization error and Hamming Cliff problem which is observed in binary coding.

The steps of RCGA are given below:

Step 1: The parameters like probability of Crossover (p), Probability of Mutation (q), and population size and termination criteria are defined.

Step 2: A set of possible solutions called the "Initial Population" is initialized such that the values range uniformly throughout the search space.

Step 3: The Fitness Values are calculated for each of the individuals and the population is sorted in the decreasing order of the fitness.

Step 4: Using the "Roulette Wheel" method of 'Selection', few individuals are selected to undergo crossover.

Step 5: according to Crossover Probability the 'Crossover' between two individuals produces two 'Child Chromosomes'.

Step 6: then using the 'Probability of Mutation', a few individuals are mutated.

Step 7: At the end of performing crossover and mutation, we obtain the crossover and the mutated child individuals. These along with the parent population form super set for the next generation population.

Step 8: Steps 3–7 are repeated until the 'Termination' condition is reached.

Step 9: The population obtained at the end of the specified number of iterations is the outcome of this process.

In the next section a case study of 4 degree of freedom redundant manipulator visiting 10 task points is considered.

Case study

In this paper, an optimization of 4 DOF redundant manipulator visiting 10 task points is considered. Inverse Kinematics solution for each point optimized values of joint angle of all joint coordinates are determined using fmincon constrained nonlinear minimization function in Matlab. The guaranteed minimum travel time sequence taking into consideration its multiple configurations is determined.

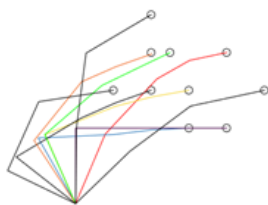


Fig.2. A schematic of 4- DOF Manipulator, 10 task points

The inverse Kinematics problem is converted to a minimization problem and then utilize a RCGA to find all the global minimums of the problem. In RCGA, a measure of the fitness of each individual is required to select the most potent individuals for crossover operation. This measure can

be defined as the difference between the end-effector position and orientation of the individual and that of the desired location. To measure the position error we use the Euclidean norm of the difference between the end-effector position of each individual and that of the desired point in the Cartesian space.

Optimum path sequence is 5-1-2-6-4-3-10-9-7-8 and travel time is 9.0322 seconds

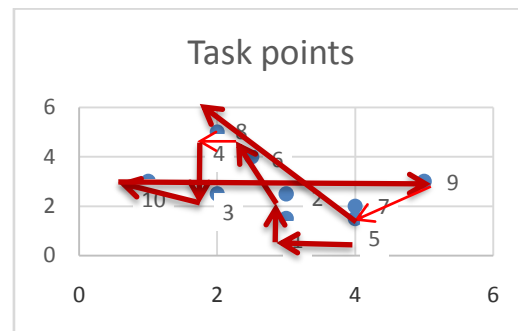


Fig.3.Optimum route for 10 points on the x–y plane for a 4 DOF manipulator.

Conclusion:

Proposed approach is found to be an effective and efficient method for determining the near-optimum sequence of a manipulator's end effector route visiting a number of task points taken into account the multiple solutions of the inverse kinematics problem. The proposed method is to be rather fast in finding an optimum or a near-optimum solution within an minimum time. Another advantage is that the multiple configurations of any non-redundant manipulator are easily included in the encoding of the RCGA.

At this point, it should be stressed that the problem is much more complex than the classical TSP problem, because the configurations of the robot are also taken into consideration. Considering the future research work, the proposed algorithm can be extended so that it can take into account the obstacle avoidance.

Table 1. Average time and joint angles difference (fmincon nonlinear minimization function)

Points	x	y	Initial angles				respective difference between angles				Avg Time Ta
			θ1	θ2	θ3	θ4	θ1	θ2	θ3	θ4	
1	3	2	2.0883	0.0459	0.1197	0.051	0.1013414	0.3363999	0.1327634	0.1249743	0.6644703
2	3	3	1.987	0.3823	0.2524	0.176	0.4890216	0.1581287	0.162926	0.1631128	0.9297985
3	2	3	2.476	0.5404	0.4153	0.339	0.3228892	0.3496109	0.0308515	0.0121475	0.6835979
4	2	4	2.1531	0.89	0.4462	0.351	0.5823103	0.8900029	0.4461979	0.3510924	2.168411
5	4	2	1.5708	0	0	0	0.4207901	0.7104088	0.4501378	0.4501379	1.9408994
6	2.5	4	1.9916	0.7104	0.4501	0.45	0.8235401	0.1073658	0.0647408	0.2402562	1.1807989
7	4	4	1.168	0.8178	0.5149	0.21	0.3918889	0.6247237	0.0183051	0.3233023	1.2976625
8	2	5	1.5599	1.4425	0.5332	0.533	0.793127	0.8021532	0.3229516	0.3229518	2.1412583
9	5	3	0.7668	0.6403	0.2102	0.21	1.9221747	0.5071523	0.0587057	0.0587057	2.4331896
10	1	3	2.689	1.1475	0.1515	0.152	0.6006674	1.1016341	0.0318694	0.1006687	1.7530314
1	3	2	2.0883	0.0459	0.1197	0.051	0.3876802	0.4945286	0.2956893	0.2880871	1.4006228
3	2	3	2.476	0.5404	0.4153	0.339					
1	3	2	2.0883	0.0459	0.1197	0.051	0.0647909	0.8441396	0.3265408	0.3002346	1.4672349
4	2	4	2.1531	0.89	0.4462	0.351					
1	3	2	2.0883	0.0459	0.1197	0.051	0.5175194	0.0458634	0.1196571	0.0508578	0.7011761
5	4	2	1.5708	0	0	0					
1	3	2	2.0883	0.0459	0.1197	0.051	0.0967292	0.6645454	0.3304808	0.3992801	1.4245562
6	2.5	4	1.9916	0.7104	0.4501	0.45					
1	3	2	2.0883	0.0459	0.1197	0.051	0.9202693	0.7719113	0.3952215	0.1590239	2.1462669
7	4	4	1.168	0.8178	0.5149	0.21					
1	3	2	2.0883	0.0459	0.1197	0.051	0.5283803	1.396635	0.4135266	0.4823262	2.695097
8	2	5	1.5599	1.4425	0.5332	0.533					
1	3	2	2.0883	0.0459	0.1197	0.051	1.3215074	0.5944818	0.0905751	0.1593744	2.0693681
9	5	3	0.7668	0.6403	0.2102	0.21					
1	3	2	2.0883	0.0459	0.1197	0.051	0.6006674	1.1016341	0.0318694	0.1006687	1.7530314
10	1	3	2.689	1.1475	0.1515	0.152	0.7020088	0.7652342	0.100894	0.0243056	1.5214419
2	3	3	1.987	0.3823	0.2524	0.176	0.1013414	0.3363999	0.1327634	0.1249743	0.6644703
1	3	2	2.0883	0.0459	0.1197	0.051					
2	3	3	1.987	0.3823	0.2524	0.176	0.1661323	0.5077397	0.1937775	0.1752603	0.9964106
4	2	4	2.1531	0.89	0.4462	0.351					
2	3	3	1.987	0.3823	0.2524	0.176	0.4161779	0.3822633	0.2524205	0.1758321	1.1720004
5	4	2	1.5708	0	0	0					
2	3	3	1.987	0.3823	0.2524	0.176	0.0046122	0.3281455	0.1977174	0.2743058	0.768899
6	2.5	4	1.9916	0.7104	0.4501	0.45					
2	3	3	1.987	0.3823	0.2524	0.176	0.8189278	0.4355114	0.2624582	0.0340496	1.4817965
7	4	4	1.168	0.8178	0.5149	0.21					
2	3	3	1.987	0.3823	0.2524	0.176	0.4270389	1.0602351	0.2807633	0.3573519	2.0306266
8	2	5	1.5599	1.4425	0.5332	0.533					
2	3	3	1.987	0.3823	0.2524	0.176	1.2201659	0.2580819	0.0421883	0.0344002	1.4855124
9	5	3	0.7668	0.6403	0.2102	0.21					
2	3	3	1.987	0.3823	0.2524	0.176	0.7020088	0.7652342	0.100894	0.0243056	1.5214419

	10	1	3	2.689	1.1475	0.1515	0.152	0.2129872	0.6071054	0.26382	0.1874184	1.2146474
3	2	3	2.476	0.5404	0.4153	0.339	0.9051995	0.540392	0.4153464	0.3389449	2.1017989	
5	4	2	1.5708	0	0	0						
3	2	3	2.476	0.5404	0.4153	0.339	0.4844094	0.1700168	0.0347914	0.1111193	0.7647235	
6	2.5	4	1.9916	0.7104	0.4501	0.45						
3	2	3	2.476	0.5404	0.4153	0.339	1.3079494	0.2773826	0.0995322	0.1290632	1.7330517	
7	4	4	1.168	0.8178	0.5149	0.21						
3	2	3	2.476	0.5404	0.4153	0.339	0.9160605	0.9021064	0.1178373	0.1942391	2.0352643	
8	2	5	1.5599	1.4425	0.5332	0.533						
3	2	3	2.476	0.5404	0.4153	0.339	1.7091875	0.0999531	0.2051143	0.1287127	2.0474213	
9	5	3	0.7668	0.6403	0.2102	0.21						
3	2	3	2.476	0.5404	0.4153	0.339	0.2129872	0.6071054	0.26382	0.1874184	1.2146474	
10	1	3	2.689	1.1475	0.1515	0.152						
4	2	4	2.1531	0.89	0.4462	0.351	0.1615201	0.1795941	0.0039399	0.0990455	0.4242991	
6	2.5	4	1.9916	0.7104	0.4501	0.45						
4	2	4	2.1531	0.89	0.4462	0.351	0.9850602	0.0722283	0.0686807	0.1412107	1.2106814	
7	4	4	1.168	0.8178	0.5149	0.21						
4	2	4	2.1531	0.89	0.4462	0.351	0.5931712	0.5524954	0.0869858	0.1820917	1.3516664	
8	2	5	1.5599	1.4425	0.5332	0.533						
4	2	4	2.1531	0.89	0.4462	0.351	1.3862983	0.2496578	0.2359658	0.1408601	1.9230402	
9	5	3	0.7668	0.6403	0.2102	0.21						
4	2	4	2.1531	0.89	0.4462	0.351	0.5358764	0.2574945	0.2946715	0.1995659	1.230199	
10	1	3	2.689	1.1475	0.1515	0.152						
5	4	2	1.5708	0	0	0	0.4027499	0.8177746	0.5148786	0.2098817	1.8585525	
7	4	4	1.168	0.8178	0.5149	0.21						
5	4	2	1.5708	0	0	0	0.010861	1.4424984	0.5331837	0.533184	2.4073826	
8	2	5	1.5599	1.4425	0.5332	0.533						
5	4	2	1.5708	0	0	0	0.803988	0.6403451	0.2102321	0.2102322	1.7816537	
9	5	3	0.7668	0.6403	0.2102	0.21						
5	4	2	1.5708	0	0	0	1.1181867	1.1474974	0.1515265	0.1515265	2.4542075	
10	1	3	2.689	1.1475	0.1515	0.152						
6	2.5	4	1.9916	0.7104	0.4501	0.45	0.4316511	0.7320896	0.0830459	0.0830461	1.2705408	
8	2	5	1.5599	1.4425	0.5332	0.533						
6	2.5	4	1.9916	0.7104	0.4501	0.45	1.2247782	0.0700637	0.2399057	0.2399057	1.6955285	
9	5	3	0.7668	0.6403	0.2102	0.21						
6	2.5	4	1.9916	0.7104	0.4501	0.45	0.6973966	0.4370886	0.2986114	0.2986114	1.6544981	
10	1	3	2.689	1.1475	0.1515	0.152						
7	4	4	1.168	0.8178	0.5149	0.21	0.4012381	0.1774295	0.3046465	0.0003505	0.8442655	
9	5	3	0.7668	0.6403	0.2102	0.21						
7	4	4	1.168	0.8178	0.5149	0.21	1.5209366	0.3297228	0.3633521	0.0583552	2.1710511	
10	1	3	2.689	1.1475	0.1515	0.152						
8	2	5	1.5599	1.4425	0.5332	0.533	1.1290477	0.295001	0.3816572	0.3816575	2.0898377	
10	1	3	2.689	1.1475	0.1515	0.152						

Time Matrix

	1	2	3	4	5	6	7	8	9	10
1		0.6645	1.401	1.467	0.701	1.425	2.146	2.695	2.069	1.753
2	0.664		0.93	0.996	1.172	0.769	1.482	2.031	1.486	1.521
3	1.401	0.9298		0.684	2.102	0.765	1.733	2.035	2.047	1.215
4	1.467	0.9964	0.684		2.168	0.424	1.211	1.352	1.923	1.23
5	0.701	1.172	2.102	2.168		1.941	1.859	2.407	1.782	2.454
6	1.425	0.7689	0.765	0.424	1.941		1.181	1.271	1.696	1.654
7	2.146	1.4818	1.733	1.211	1.859	1.181		1.298	0.844	2.171
8	2.695	2.0306	2.035	1.352	2.407	1.271	1.298		2.141	2.09
9	2.069	1.4855	2.047	1.923	1.782	1.696	0.844	2.141		2.433
10	1.753	1.5214	1.215	1.23	2.454	1.654	2.171	2.09	2.433	

Solution of RCGA

1) Initial random solutions

20	1	2	3	4	5	6	7	8	9	10	Obj
1	1	2	3	5	9	4	6	7	8	10	12.39335978
2	1	9	5	2	7	3	6	4	8	10	12.86839695
3	5	6	2	3	8	4	7	9	1	10	12.90387392
4	1	9	5	2	4	3	7	6	8	10	12.97725971
5	1	6	3	5	8	4	7	9	2	10	13.11202866
6	1	2	4	5	7	6	9	10	8	3	15.12246338
7	8	3	6	5	7	1	4	2	9	10	15.12805394
8	1	2	4	5	8	3	7	6	9	10	15.31450753
9	1	8	6	5	2	4	7	3	9	10	15.49929206
10	6	9	4	5	2	3	7	1	8	10	16.55303178

2) Roulette wheel selection

Obj	Reci	Fitness	Cum	Rand	paste	Sol
12.39335978	0.08069	0.113378	0.113378	0.655999	0.057422	1
12.86839695	0.07771	0.109193	0.222571	0.236604	0.763406	8
12.90387392	0.0775	0.108892	0.331463	0.427413	0.689549	7
12.97725971	0.07706	0.108277	0.43974	0.755234	0.901972	9
13.11202866	0.07627	0.107164	0.546904	0.085791	0.889325	9
15.12246338	0.06613	0.092917	0.639821	0.802671	0.835873	9
15.12805394	0.0661	0.092883	0.732703	0.608186	0.376062	4
15.31450753	0.0653	0.091752	0.824455	0.431396	0.178408	2
15.49929206	0.06452	0.090658	0.915113	0.717597	0.037874	1
16.55303178	0.06041	0.084887	1	0.897665	0.596963	6

3) Selection

20	1	2	3	4	5	6	7	8	9	10	20
1	1	2	3	5	9	4	6	7	8	10	1
2	1	2	4	5	8	3	7	6	9	10	2

3	8	3	6	5	7	1	4	2	9	10	3
4	1	8	6	5	2	4	7	3	9	10	4
5	1	8	6	5	2	4	7	3	9	10	5
6	1	8	6	5	2	4	7	3	9	10	6
7	1	9	5	2	4	3	7	6	8	10	7
8	1	9	5	2	7	3	6	4	8	10	8
9	1	2	3	5	9	4	6	7	8	10	9
10	1	2	4	5	7	6	9	10	8	3	10

4) Solutions after mutation

	1	2	3	4	5	6	7	8	9	10	Obj
1	1	2	3	5	9	4	6	7	8	10	12.39335978
2	1	9	5	2	7	3	6	4	8	10	12.87
3	5	6	2	3	8	4	7	9	1	10	12.90387392
4	1	9	5	2	4	3	7	6	8	10	12.97725971
5	1	6	3	5	8	4	7	9	2	10	13.11202866
6	1	2	4	5	7	6	9	10	8	3	15.12
7	8	3	6	5	7	1	4	2	9	10	15.13
8	1	2	4	5	8	3	7	6	9	10	15.31450753
9	1	8	6	5	2	4	7	3	9	10	15.50
10	6	9	4	5	2	3	7	1	8	10	16.55303178

5) Optimum solution

1	2	3	4	5	6	7	8	9	10	Objective
5	1	2	6	4	3	10	9	7	8	9.032207

References:

- Bhosale, K. C. and Pawar, P. J. (2018) 'Material flow optimisation of production planning and scheduling problem in flexible manufacturing system by real coded genetic algorithm (RCGA)', Flexible Services and Manufacturing Journal. doi: 10.1007/s10696-018-9310-5.
- Goldberg, D. E. (no date) 'Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking 2 Past Use of Real-Coded Genes', pp. 1–21.
- P.Th. Zacharia, N.A. Aspragathos 'Optimal robot task scheduling based on genetic algorithms' Robotics and Computer-Integrated Manufacturing journal 2005 doi:10.1016/j.rcim.2004.04.003
- Q. Zhang and M.-Y. Zhao, "Minimum time path planning of robotic manipulator in drilling/spot welding tasks," Journal of Computer Design Engineering. vol. 3, no. 2, pp. 132–139, 2016.
- S. Stevo. I. Sekaj and M. Dekan, "Optimization of Robotic Arm Trajectory Using Genetic Algorithm," *Int. Fed. Autom. Control*, vol. 8, pp. 24–29, 2014.
- M. Galicki, "Path-constrained control of a redundant manipulator in a task space," *Rob. Auton. Syst.*, vol. 54, no. 1, pp. 234–243, 2006.
- C. Nearchou and N. A. Aspragathos, "Point-To-Point motion of redundant

- manipulators,” vol. 31, no. 3, pp. 261–270, 1996
8. X. F. Zha, “Optimal pose trajectory planning for robot manipulators,” Mech. Mach. Theory, vol. 37, no. 1, pp. 1063–1086, 2002.
 9. J. R. and A. Rubiano, “ ’Optimization of Inverse Kinematics of a 3R Robotic Manipulator using Genetic Algorithms,” World Acad. Sci. Eng. Technol., pp. 1425–1430, 2011.