

Continuous Path Planning of Kinematically Redundant Manipulator using Particle Swarm Optimization

Affiani Machmudah, Setyamartana Parman, M.B. Baharom
Mechanical Engineering
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, Tronoh, Perak, Malaysia

Abstract—This paper addresses a problem of a continuous path planning of a redundant manipulator where an end-effector needs to follow a desired path. Based on a geometrical analysis, feasible postures of a self-motion are mapped into an interval so that there will be an angle domain boundary and a redundancy resolution to track the desired path lies within this boundary. To choose a best solution among many possible solutions, meta-heuristic optimizations, namely, a Genetic Algorithm (GA), a Particle Swarm Optimization (PSO), and a Grey Wolf Optimizer (GWO) will be employed with an optimization objective to minimize a joint angle travelling distance. To achieve n -connectivity of sampling points, the angle domain trajectories are modelled using a sinusoidal function generated inside the angle domain boundary. A complex geometrical path obtained from Bezier and algebraic curves are used as the traced path that should be followed by a 3-Degree of Freedom (DOF) arm robot manipulator and a hyper-redundant manipulator. The path from the PSO yields better results than that of the GA and GWO.

Keywords—Path planning; redundant manipulator; genetic algorithm; particle swarm optimization; grey wolf optimizer insert

I. INTRODUCTION

Nowadays, researches in the robotic field are focusing on the use of robot to substitute human operator jobs in dangerous environment because it involves the high risk for human safety. Human operator is usually still used due to the task is conducted in a difficult environment involving a very complex geometrical path. The time to finish the job by manually programming, for example in the arc welding robotic system for manufacturing the large vehicle hull, is very long time [1] so that in this application, using the robotic system is still very challenging and high cost. Thus, the path planning to track the prescribed path is very important research to be conducted toward the automation in the manufacturing industry involving complex geometrical path.

Using the manipulator to track a complex geometry in the manufacturing industry, the goal is to achieve the high precision as well as satisfy the optimality criteria so that the production efficiency can be improved. For very challenging environmental conditions, an off-line tracking algorithm is considered as an efficient approach because an online path planning is only available for very simple tasks such as a pick and place [1]. Different with a point-to-point path planning where the end-effector path is free to be chosen, in the

continuous path planning, the arm robot manipulator needs to move along the prescribed end-effector path. Thus, the continuous path planning needs to solve an Inverse Kinematic (IK) problem for the entire traced path.

Solving the IK of the arm robot manipulator is a long-term research where it has been conducted since the past five decades. The Jacobian pseudoinverse technique was firstly employed to solve the IK problem of the robotic arm manipulator by Whitney in 1969 [2]. Baillieul [3] developed an extended Jacobian to achieve the cyclic properties. This approach imposed some additional constraints to be achieved along with the end-effector task to identify an appropriate solution. IK function approach by constructing a mathematical function to model the joint angle trajectories was presented by Wampler [4].

Burdick [5] presented a concept of self-motion manifold which considered global solution of the IK rather than instantaneous solution. Khatib [6] presented a generalized inverse which was consistent with the system dynamics namely, a dynamically consistent Jacobian. Tcho'n et al. [7] studied the design of the extended Jacobian algorithm which approximate the Jacobian pseudoinverse. Variational calculus and differential geometric were employed. 3-DOF manipulator and mobile robot were used in numerical examples.

In the case of the hyper-redundant robot, the computation of the Jacobian based approach becomes computationally expensive because of increasing the degrees of freedom. Furthermore, the Jacobian based methods are only suitable for serial link morphologies, and impractical for applications of locomotion and tentacle-like grasping [8]. Because of this drawback, the geometrical approach which does not require the computation of the Jacobian inverse becomes an alternative solution to the hyper-redundant manipulator IK [9].

Studies of the path planning to track the prescribed path have been conducted using both serial and parallel manipulators. Ebrahimi et al. [10] applied heuristic optimizations, such as genetic algorithm and particle swarm optimization to achieve an optimal path using the four-bar mechanism. Yue et al. [11] presented a computer-aided linkage design for tracking open and closed planar curves. Merlet [12] studied trajectory verification for Gough-Stewart parallel manipulator. The algorithm was proposed considering a real-

time method so that it may deal with any path trajectory and validity criterion. Yao and Gupta [13] studied the collision-free path planning of the arm robot manipulator whose end-effector traveled along a prescribed path. Yahya et al. [9] proposed the geometrical approach to the hyper-redundant robot to track the prescribed path. The angles between the neighboring links were arranged to be the same to avoid the singularities. Ananthanarayanan and Ordonez [14] proposed a novel approach to solve an IK problem for $2n+1$ degree of freedom manipulator. The IK was modeled as a constrained optimization considering collision-free and joint limits.

The complexity of the arm robot motion presents because the task is naturally delivered in the Cartesian coordinate while the motion is conducted in the joint space. This paper uses an interval analysis of the self-motion in generating joint space trajectories and applying the meta-heuristic optimization technique to choose the optimal solution among infinite possible solutions. The self-motion will be mapped into the interval of the angle domain variable, θ_g . The redundancy resolution exists within the angle domain boundary. To choose the best solution among infinite possible solutions, the meta-heuristic optimizations, which are the GA, the PSO and the GWO will be employed with the optimization objective is to minimize the total joint angle travelling distance. The 3-DOF planar series redundant manipulator and 6-DOF planar series hyper-redundant manipulator will be used to track the complex geometrical curve.

The present paper is organized as follows: Section 2 presents the self-motion of the arm robot manipulator. Section 3 describes the path planning optimization problem. Section 4 presents the methodology to solve the path planning optimization of the planar redundant and hyper-redundant manipulators. Section 5 presents the GA, PSO, and GWO algorithms to solve the path planning. The proposed method is applied to the planar redundant and hyper-redundant manipulators in the Section 6. The conclusion is presented in Section 7.

II. SELF-MOTION OF KINEMATICALLY REDUNDANT MANIPULATOR

The self-motion can be used to repair infeasible trajectories due to collision, singularity, and connectivity issues. The self-motion is the case where the end-effector does not move while the positions of the joints are moved. To generate the smooth trajectories, there is the requirement of the connectivity among sampling points of the generated trajectories from the initial configuration to the final configuration. The concept of connectivity is one of the important performances of the manipulator motion [15], [16].

Fig. 1(a) illustrates an example of inappropriate configurations because of disconnection of postures. Kinematically redundant manipulator has the self-motion capability which has an advantage in finding the proper posture. However, because there are many possible configurations to achieve single end-effector position for the redundant manipulator, finding the feasible trajectories is a challenging computational problem. The posture in Fig. 1(a) is possible to be repaired using the self-motion. Fig. 2 is the example of the feasible motion where the joint angle

trajectories, as shown in Fig. 2(b), are smooth and the postures of the robot, as shown in Fig. 2(a), have satisfied n -connectivity.

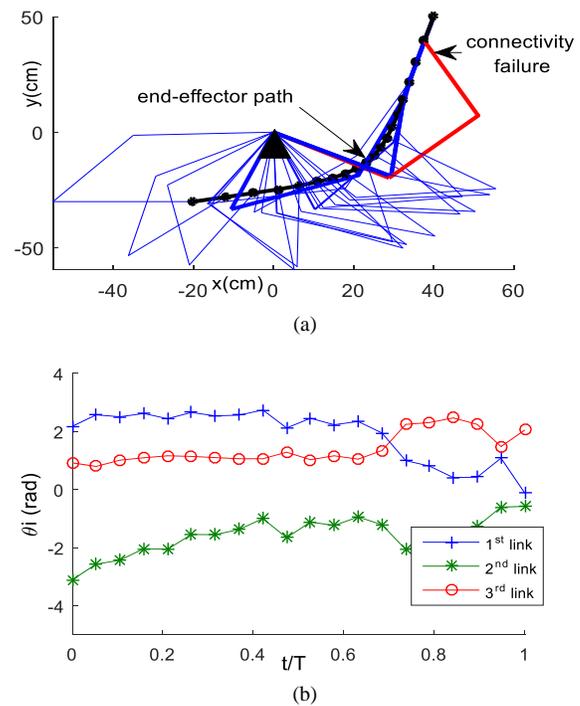


Fig. 1. Connectivity failure (a) improper posture (b) Joint angle trajectories of (a).

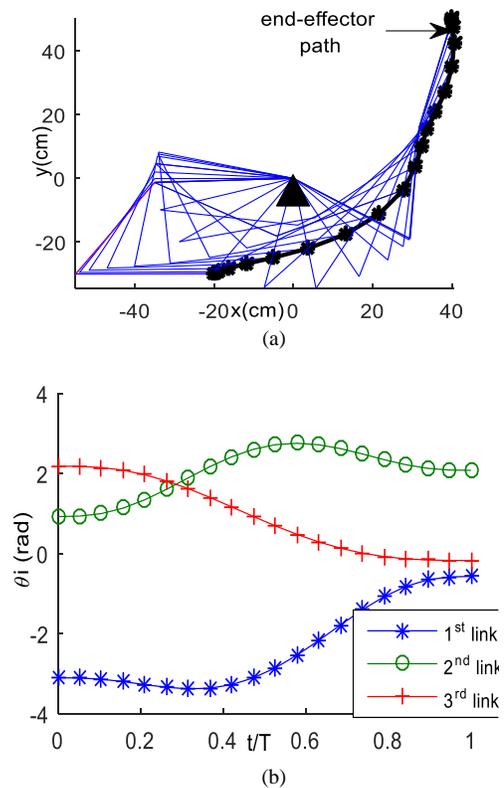


Fig. 2. Repair trajectories by self-motion (a) proper posture (b). Joint angle trajectories of (a).

III. PROBLEM FORMULATIONS

The continuous path planning is the problem to find the joint angle θ_i when the end-effector moves along the specified path. Since for kinematically redundant manipulator, there are infinite possible solutions to achieve the desired end-effector movement, the path planning can be modelled as the optimization problem to find the optimal solution based on the optimization objective.

This paper considers planar redundant and planar hyper-redundant manipulators where the path planning can be formulated as the optimization problem as follows:

Objective:

$$\text{Min } F_{\text{path}} = \sum_{i=1}^n \int_0^1 \sqrt{1 + \left(\frac{d\theta_i(r)}{dr}\right)^2} dr \quad (1a)$$

Constraints:

$$\theta_{i\min} \leq \theta_i \leq \theta_{i\max} \quad (1b)$$

$$(x_e, y_e) = (x, y) \quad (1c)$$

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + \dots + l_n \cos(\theta_1 + \theta_2 + \dots + \theta_n) \quad (1d)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + \dots + l_n \sin(\theta_1 + \theta_2 + \dots + \theta_n)$$

$$x_e = f(t) ; y_e = f(t) \quad (1e)$$

where F_{path} , r , n , l_i , θ_i , $\theta_{i\min}$, $\theta_{i\max}$, (x, y) , and (x_e, y_e) are the objective function, a linear time-scale, number of links, i th manipulator length, i th joint angle, minimum of θ_i , maximum of θ_i , actual end-effector path, desired end-effector path, respectively.

Equation (1a) is the joint angle traveling distance which is the objective of optimization. Equations (1b) and (1e) are the joint angle constraint and the forward kinematics of the manipulator, respectively.

IV. METHODS

Since the end-effector path is constrained, all possible trajectories are due to the contribution of the self-motion. This section will investigate the available solution of the IK for tracking the path in the area of nearest maximum reachable workspace by mapping the angle domain interval.

For the IK problem of 3-DOF planar robot, it has the analytic solution using the geometrical method in the following [17]:

$$w_x = x_p - l_1 \cos(\theta_g) ; w_y = y_p - l_1 \sin(\theta_g) \quad (2)$$

$$c_2 = \frac{(w_x^2 + w_y^2 - l_1^2 - l_2^2)}{2l_1 l_2} ; s_2 = \pm \sqrt{1 - c_2^2} \quad (3)$$

$$\theta_2 = a \tan 2(s_2, c_2) \quad (4)$$

where (x_p, y_p) , l_i , θ_g , θ_2 , c_2 , s_2 , are the position of end-effector in Cartesian coordinate, the length of i th link, the angle domain, the second joint angle, the cosines of θ_2 , and the sine of θ_2 , respectively.

First and third joint angles can be obtained by following equations:

$$\Delta = w_x^2 + w_y^2 ; s_1 = \frac{(l_1 + l_2 c_2)w_y - (l_2 s_2 w_x)}{\Delta} \quad (5)$$

$$c_1 = \frac{(l_1 + l_2 c_{21})w_x + (l_2 s_{21} w_y)}{\Delta} ; \theta_1 = a \tan 2(s_1, c_1) \quad (6)$$

$$\theta_3 = \theta_g - \theta_2 - \theta_1 \quad (7)$$

where θ_i , c_i , s_i and θ_3 are the first joint angles, the cosine of θ_i , the sine of θ_i , and the third joint angle, respectively.

Using (7), the self-motion of 3-DOF planar robot can be modelled as interval-valued function in the following:

For $P(x_p, y_p)$:

$$\theta_g = \theta_1 - \theta_2 - \theta_3 ; \theta_1 = (\theta_{1\min}, \theta_{1\max}) \quad (8)$$

$$\theta_2 = (\theta_{2\min}, \theta_{2\max}) ; \theta_3 = (\theta_{3\min}, \theta_{3\max})$$

By this approach, θ_g is a function of variables, θ_1 , θ_2 , and θ_3 where they are real numbers in radian with specific intervals. The problem becomes how to find these intervals for specific value of $P(x_p, y_p)$. Posture analysis will be employed to investigate these intervals in this section.

A. Interval Analysis

This paper considers tracking the end-effector path in the case of nearest maximum reachable workspace, which is defined as the end-effector trajectories in the range of radius in the following:

$$R^*_{\min} \leq R < R_{\max} = (l_1 + l_2 + l_3) \quad (9)$$

$$R^*_{\min} = \sqrt{(l_1 + l_2)^2 - l_3^2} \quad (10)$$

where R and R_{\max} are radius or distance of the end-effector trajectories from the base and the maximum reachable radius, respectively.

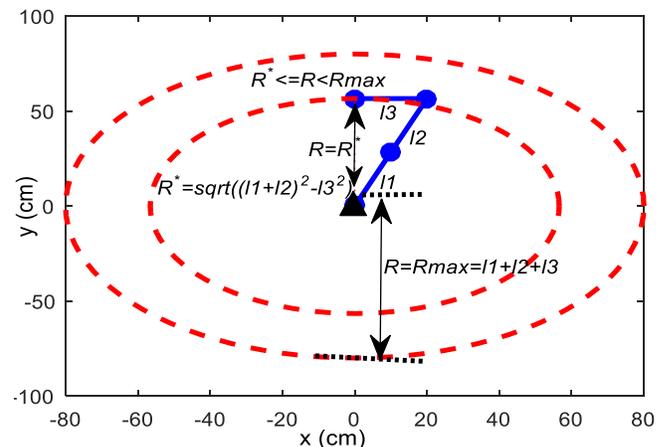


Fig. 3. Nearest maximum reachable workspace area.

Fig. 3 illustrates the nearest maximum reachable workspace defined in this paper. In this nearest maximum reachable workspace, the self-motion capability is reduced due to the limitation of the arm robot geometry. Analysis the feasible range of the trajectories is very important step in the path planning, since to track the entire path, the joint space trajectories should have connectivity without any error in position. Any joint space trajectories that are outside this feasible interval will contribute to the tracking position errors.

For the nearest maximum reachable workspace, regarding the range of reachable point by the arm robot manipulator, there are the maximum and minimum configurations contributing the maximum and minimum angle domain as illustrated in Fig. 4. Posture A is the maximum configuration while posture B is the minimum configuration. These are the maximum and minimum configurations that can be reached by the arm robot at the curve point (x_p, y_p) . These maximum/minimum postures can be mapped into interval of the angle domain. For the near maximum reachable workspace at curve point (x_p, y_p) , the angle domain solution lies within this interval. Outside this interval, the arm robot cannot reach due to geometrical limitation so that it will contribute to the position error.

Since the value of joint angle is periodic with period 2π , the minimum value should be chosen in constructing the proper interval of the angle domain. The angle domain for maximum and minimum configurations can be expressed in the following:

$$\theta_{g \min/\max} = \theta_{1\max/\min} + \theta_{3\max/\min} \quad (11)$$

where $\theta_{g\min/\max}$, $\theta_{1\max/\min}$, and $\theta_{3\max/\min}$ are the maximum or minimum of theta global, the first joint angle at maximum or minimum posture, and the third joint angle of maximum/minimum posture, respectively.

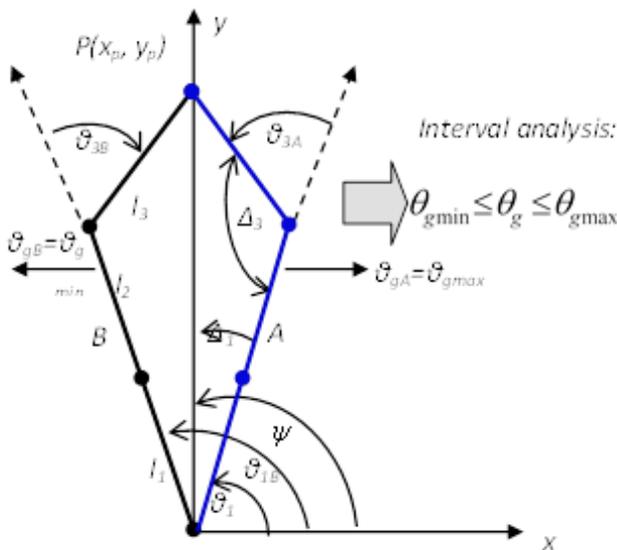


Fig. 4. Mapping the feasible postures into interval analysis.

Refer to Fig. 4, the value of $\theta_{1\max/\min}$ and $\theta_{3\max/\min}$ can be obtained using trigonometric rule as follows:

$$\theta_{1\max} = \psi - \Delta_1; \theta_{1\min} = \psi + \Delta_1 \quad (12)$$

$$\theta_{3\max} = 2\pi - \Delta_3; \theta_{3\min} = -(2\pi - \Delta_3) \quad (13)$$

$$\theta_{g\min} = \theta_{1\min} + \theta_{3\min} \quad (14)$$

$$\theta_{g\max} = \theta_{1\max} + \theta_{3\max} \quad (15)$$

where ψ , $\theta_{1\max}$, $\theta_{1\min}$, $\theta_{3\max}$, and $\theta_{3\min}$ are the angle of radius of curve point (x_p, y_p) from x-axis, the first joint angle of maximum configuration, the first joint angle of minimum configuration, the third joint angle of maximum configuration, and the third joint angle of minimum configuration, respectively.

The value of Δ_1 and Δ_3 can be obtained from geometrical analysis of triangular of maximum-minimum configurations using cosine rule.

The posture of the point in the near maximum reachable workspace area then can be mapped into interval of the angle domain in the following:

$$(\theta_{g \min} \pm p.2\pi) \leq \theta_g \leq (\theta_{g \max} \pm p.2\pi) \quad (16)$$

where p is an integer value, respectively.

The inverse trigonometric function is not injective function. The value is periodic with period 2π so that the angle domain interval is expressed in (16). Making constraint in interval of first and second joint angles range within $[-\pi, \pi]$ is necessary to avoid the disconnection problem of the sampling points. By making this interval constraint, the value of p in (16) will be zero, and (16) can be expressed in the following

$$\theta_{g \min} \leq \theta_g \leq \theta_{g \max} \quad (17)$$

$$-\pi \leq \theta_1, \theta_2, \theta_3 \leq \pi \quad (18)$$

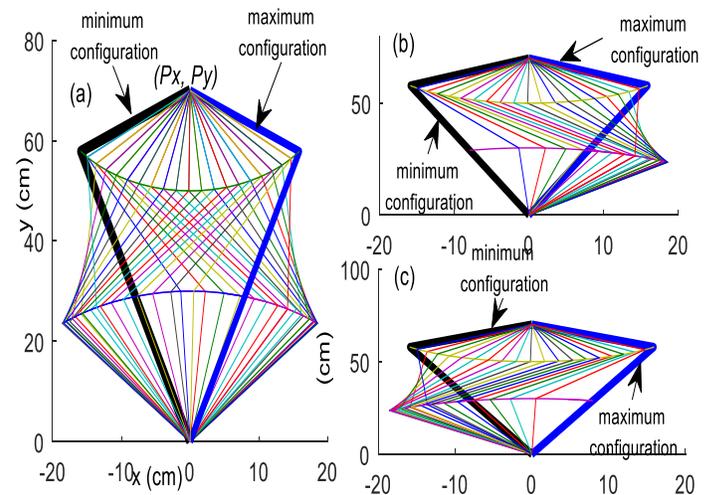


Fig. 5. (a) Posture interval for curve point (P_x, P_y) for both positive and negative sign of (3) (b) Posture plots for positive value of (3) (c) posture plots for negative value of (3).

The maximum and minimum of the angle domain represent the maximum and minimum configurations that can be reached by arm robot manipulator at such corresponding point. Illustration of all postures that can be reached by the arm robot from minimum configuration to maximum configuration is shown in Fig. 5.

Using this posture analysis, interval-valued function in (8) can be expressed in the following:

$$\theta_g = \theta_1 - \theta_2 - \theta_3 ; (\theta_{g \min}, \theta_{g \max}) \quad (19)$$

In the case there is joint angle constraint, it needs to check the feasible interval according to the corresponding joint angle constraint. The angle domain interval may decrease as compared to the case when the manipulator does not have the joint angle limit.

For the hyper-redundant manipulator, using the concept of a moving base of the previous 3-link component, the geometrical approach (2-7) can be employed. The local angle domain boundary is computed with respect to the moving base by employing the interval analysis of the self-motion of the tracked path and the trajectories of the local angle domain are generated inside the boundary.

B. Sinusoidal Function as the Joint Angle Trajectories

This paper uses the continuous function of the angle domain to achieve the n -connectivity among the sampling points of the generated trajectories.

The angle domain as function of time can be expressed as composition function of the angle domain profile and linear time-scale as follows:

$$\theta_g(t) = \theta_g(r) \circ r(t) \quad (20)$$

$$r(t) = \frac{t}{T} \quad (21)$$

where $\theta_g(t)$, $\theta_g(r)$, $r(t)$, t , and T are the theta global, the joint angle profile, linear time-scale, the time, and the total travelling time, respectively.

The angle domain trajectories will be generated in the form of the sinusoidal function as follows:

$$\theta_g(r) = \theta_{gi} + A \sin 2\pi fr \quad (22)$$

where $\theta_g(t)$, $\theta_g(r)$, $r(t)$, t , and T are the angle domain, the angle domain profile, linear time-scale, the time, and the total travelling time, respectively.

The path planning problem then can be reduced as the problem to find the feasible angle domain from parameter 0 to 1. Using the sinusoidal function, f is the parameter to be searched in the optimization step.

C. Boundary of the Angle Domain Representing the Feasible Zone Trajectories

Computing the angle domain for the entire traced path will construct the boundary as illustrated in Fig. 6. The angle domain trajectories during the motion should be maintained inside this boundary to avoid the position error. Using interval analysis of the self-motion, the angle domain boundary for the

path in Fig. 6(a) can be illustrated as Fig. 6(b). The boundary lines are composed from the minimum angle domain trajectories and the maximum angle domain trajectories.

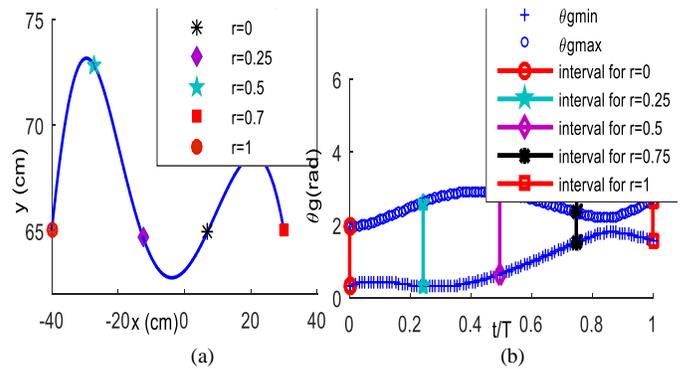


Fig. 6. (a) Traced path (b) the angle domain boundary of traced path in (a).

D. Algorithm

Based on previously presented analyses, the IK algorithm for manipulator continuous path planning can be computed using the following procedure:

- 1) Compute the angle domain boundary using the interval analysis to obtain minimum and maximum angle domains trajectories.
- 2) In the case of hyper-redundant manipulator, defined the additional path in such a way so that such path is used as the moving base of 3-link component and compute the angle domain boundary using interval analysis with respect to the fixed base and the moving base.
- 3) Consider the interval-limited as the interval of interest so that the analysis can be focused on this interval to avoid an ambiguity of values of the angle domain trajectories.
- 4) Completely map the trajectories of minimum/maximum angle domains from the initial configuration to the final configuration to construct the angle domain boundary.
- 5) Choose the initial configuration inside the angle domain boundary.
- 6) Optimize the joint angle path by generating the angle domain trajectories, (22), inside the angle domain boundary using the meta-heuristic optimization with the optimization objective is to minimize the joint angle traveling distance.
- 7) The joint angle trajectories can be obtained using (2-7). For the hyper-redundant manipulator, compute the joint angle trajectories with respect to fixed base and the moving.

V. PATH PLANNING OPTIMIZATION

For kinematically redundant manipulator, there will be many possible solutions to achieve the desired motion. Using the proposed approach, the redundancy resolution of the continuous path planning lies within the angle domain boundary. There will be one searching parameter, f , as (22). To choose the best solution, the meta-heuristic optimizations, which are the GA, the PSO and the GWO will be employed in this paper to search the optimal solution with the optimization objective is to minimize joint angle traveling distance as (1a).

A. Genetic Algorithm

There are three main operators in the GA: reproduction, crossover, and mutation. The searching parameter is represented into the chromosome to be coded to find the best individual with the best fitness value. Fig. 7 gives the illustration of the GA procedure to solve the continuous path planning of the arm robot manipulator.

B. Particle Swarm Optimization

The PSO is firstly proposed by Kennedy and Eberhart [18]. The searching parameters are denoted as particles in the PSO. The particle moves with a certain velocity value. Fig. 8 illustrates the PSO procedure to solve the continuous path planning of the arm robot manipulator.

Velocities and positions are evaluated according to the local and global best solutions. The velocity for each particle is updated and added to the particle position. If the best local solution has better fitness than that of the current global solution, then the best global solution is replaced by the best local solution.

Eberhart and Shi [19] proposed the velocity by utilizing constriction factor, χ , in the following:

$$v_{t+1} = \chi \{ \omega v_t + \varphi_1 \beta_1 (p_i - x_i) + \varphi_2 \beta_2 (p_g - x_i) \} \quad (23)$$

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}; \varphi = \varphi_1 + \varphi_2 \quad ; \quad \varphi > 4$$

where v_t , v_{t+1} , φ_1 , and φ_2 are the velocity, the update velocity, cognitive parameter, social parameter, respectively. β_1 and β_2 are independent uniform random number, p_i and p_g are best local solution, and best global solution, while x_i is the current position in the dimension considered.

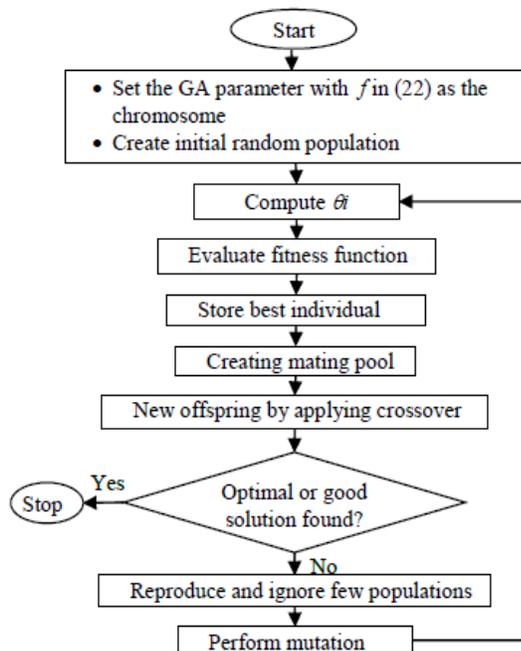


Fig. 7. GA algorithm to solve the continuous path planning.

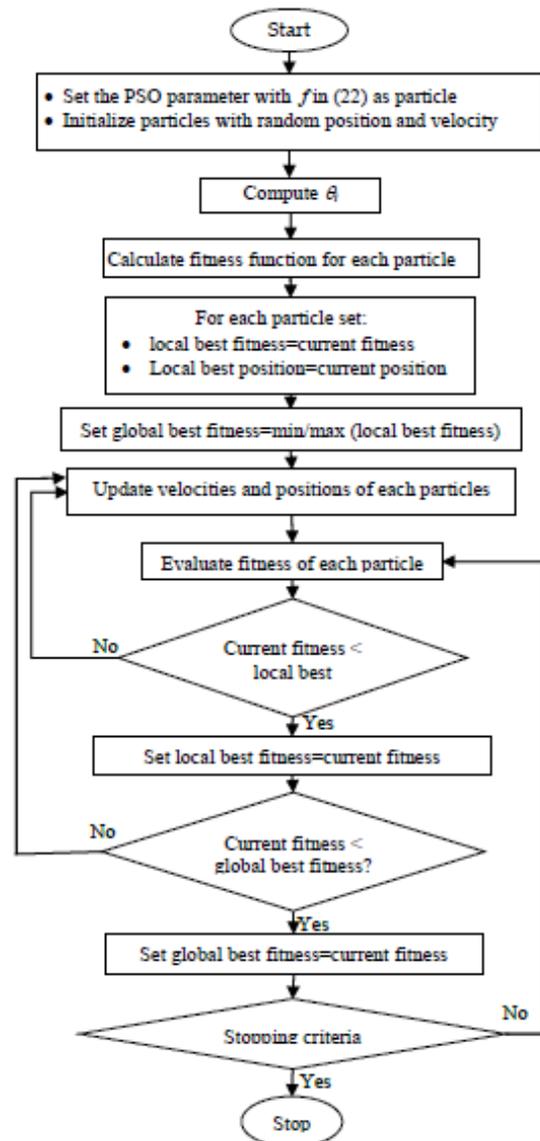


Fig. 8. PSO algorithm to solve the continuous path planning.

C. Grey Wolf Optimizer

The GWO is the meta-heuristic technique developed by Mirjalili et al in 2014 [20]. It is inspired from the leadership hierarchy and hunting mechanism of grey wolves in nature. There are four types of grey wolves which are alpha, beta, delta, and omega. It involves three steps, namely hunting, searching for prey, encircling prey, and attacking prey. Fig. 9 illustrates the path planning algorithm using the GWO.

The encircling prey by the grey wolves is modelled as follows:

$$D = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (24)$$

$$\vec{X}_p(t+1) = |\vec{X}_p(t) - \vec{A} \cdot \vec{D}|$$

where t , \vec{A} , \vec{C} , \vec{X}_p , \vec{X} are the current iteration, coefficient vector, coefficient vectors the position vector of the prey, and the position vector of a grey wolf, respectively.

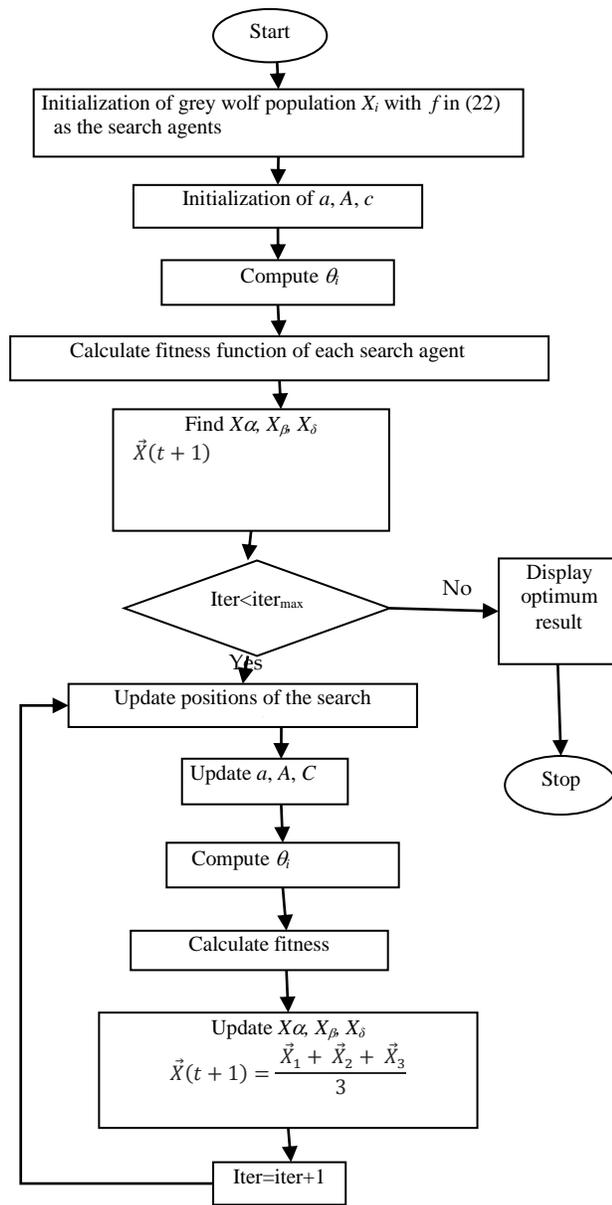


Fig. 9. GWO algorithm to solve the continuous path planning.

The vectors \vec{A} , and \vec{C} are calculated as follows:

$$\vec{A} = 2\vec{A} \cdot \vec{r}_1 - \vec{a} \quad (25)$$

$$\vec{C} = 2\vec{r}_2$$

\vec{a} are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in [0, 1].

To simulate the hunting behavior of grey wolves, the first three best solutions obtained are saved and the other search

agents (including the omegas) need to update their positions according to the position of the best search agents. The following formulas are proposed:

$$\vec{D}_\alpha = \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}; \vec{D}_\beta = \vec{C}_2 \cdot \vec{X}_\beta - \vec{X}; \vec{D}_\delta = \vec{C}_3 \cdot \vec{X}_\delta - \vec{X};$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha); \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta); \quad (26)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta);$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$

The grey wolves finish the hunting by attacking the prey when it stops moving. Approaching the prey is modeled by decreasing the value of \vec{a} . \vec{A} is a random value in $[-2a, 2a]$ where a is decreased from 2 to 0. When random values of \vec{A} are in $[-1, 1]$, the next position of the search agent lays between its current position and the position of the prey. Grey wolves search for prey based on the position of the alpha, beta, and delta to model divergence. \vec{A} values are chosen as random values greater than 1 or less than -1. \vec{C} vector contains random values in interval [0, 2].

VI. RESULTS AND DISCUSSIONS

A numerical experiment has been conducted in MATLAB environment by writing a computer program. The PSO used cognitive and social parameters 1.5 and constriction factor 1. For the GA, the real value coded is used with the selection and mutation rates are 0.5 and 0.1, respectively. The GA, PSO, and GWO are evaluated using 100 iterations and 20 individuals in the population. The computation of the path planning algorithm used 1000 sampling points to conduct the motion from the initial point to the final point.

A. 3-DOF Manipulator

Bezier curve, which is frequently used in the manufacturing, will be employed as the end-effector path to be tracked by the 3-DOF planar series manipulator which has the lengths $l=[30 \ 30 \ 20]$. A fifth-degree Bezier curve is utilized as the tracked curve as illustrated in Fig. 10(a). Detail of these tracked curves can be seen in Table I. Using the interval analysis of the self-motion, Fig. 10(b) shows the result of the angle domain boundary of the Bezier curve.

This paper considers the first and third joint angle does not have joint limits and only the second joint angle, θ_2 , has constraint as follows:

$$0 \leq \theta_2 \leq \pi \quad (27)$$

Considering the above joint angle constraint, the second joint angle trajectories from the negative root of (3) is not feasible, so that only the positive root of (3) is considered in the optimization.

TABLE I. BEZIER CURVE TRACKED PATH CONTROL POINTS

B_0	B_1	B_2	B_3	B_4	B_5
(50,-10)	(85,30)	(50,50)	(60,-10)	(58,30)	(70,-10)

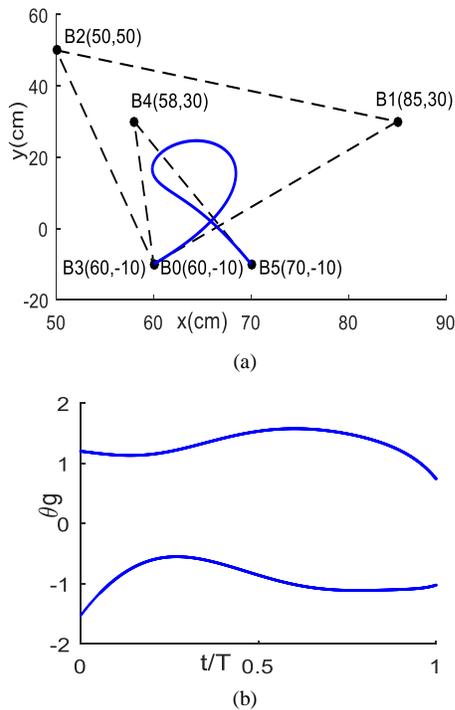


Fig. 10. (a) Fifth Bezier tracked path (b) Boundary of angle domain of path (a).

The angle domain trajectories should be kept inside the angle domain boundary. This requirement can be achieved by choosing the proper value of amplitude A and initial angle domain, θ_{gi} , in (22) in such way so that the generated angle domain trajectories lie within such boundary. This paper uses the value of $A=0.4$ and $\theta_{gi}=0.3$. Fig. 11 shows the result of the fitness value evolution during 100 iterations for the GA, the PSO, and the GWO. The searching area for f is $[0, 15]$. Detail of the path planning results is tabulated in Table II. According to the fitness value, the PSO yields better result than that of the GA and the GWO. The GWO result is near to the value of the PSO result while the result from the GA is quite far from the PSO value. Fig. 12(a) illustrates the joint angle results of the optimal solution obtained by the optimal result obtained from PSO. The posture during the motion to track the Bezier curve using this optimal value is shown in Fig. 12(b).

TABLE II. PATH PLANNING RESULTS

	Fitness	f
GA	4.8486	0.5562
PSO	4.84829399	0.5329
GWO	4.84829414	0.5330

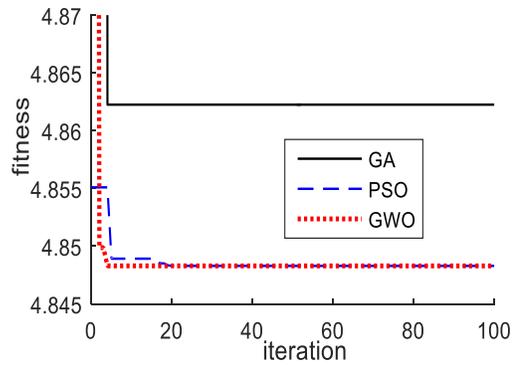


Fig. 11. Fitness value evolution (a) GA (b) PSO (c) GWO.

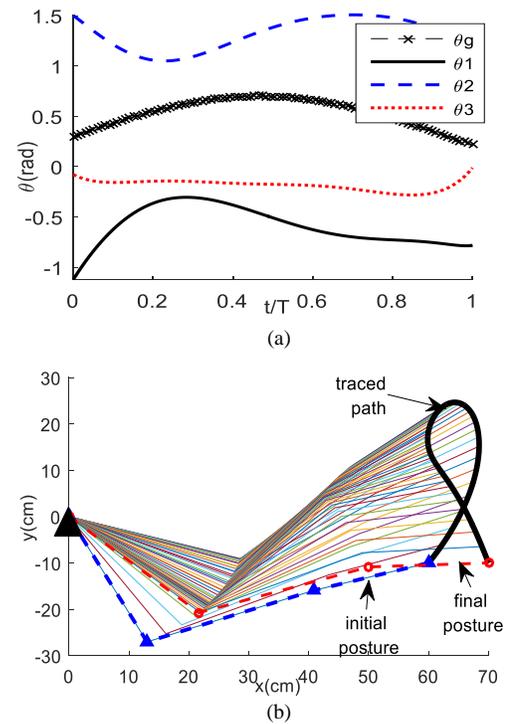


Fig. 12. Optimal result by PSO (a) joint angle, $f=0.5329$ (b) configuration.

B. Hyper-Redundant Manipulator

This section applies the proposed approach to the 6-link planar series manipulator. The manipulator has the same length for each link, $l=[30\ 30\ 30\ 30\ 30\ 30]$ cm.

Fig. 13 shows the illustration of the developed approach applied to the 6-link serial manipulator. Tracking point A' can be carried out using the same procedure as in a 3-DOF planar series manipulator with respect to point A. There will be a moving coordinate system: (x_o', y_o') . In the case of the first 3-DOF planar series robot, the coordinate is fixed because the base does not move. The moving coordinate or moving frame should be kept inside the first three-link manipulator workspace. 6-DOF planar series robot will consist of 3-DOF planar series robot with fix base and virtual 3-DOF planar series robot with moving base.

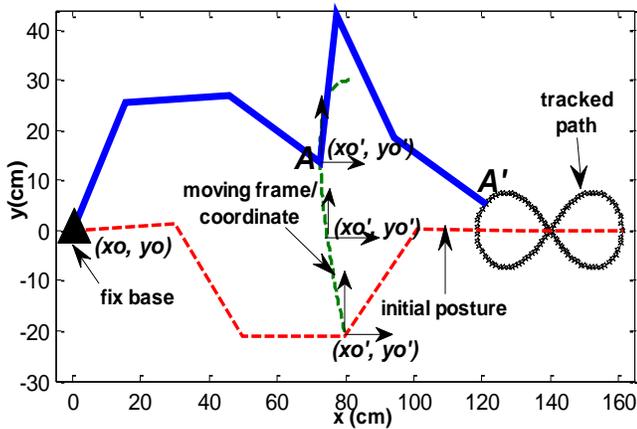


Fig. 13. Moving frame/coordinates of 6-DOF manipulator.

A complex geometrical curve, namely, generalized clothoid [12], is used as the traced curve. The clothoid is generalized using the polynomial function as follows:

$$x(t) = x_c + k \int_0^t \sin(p(u)) du \quad (28)$$

$$y(t) = y_c + k \int_0^t \cos(p(u)) du$$

where (x_c, y_c) , k , and $p(u)$ are the center of the curve, the scale factor, and the polynomial function, respectively.

This paper uses $(x_c, y_c) = (110, 20)$, $k = 25$, $t = [-4.712, 0]$ and the polynomial function in the following:

$$p(u) = (0.33u^3 - 4u) \quad (29)$$

The manipulator has constraints of the second and fifth joint angle as follows:

$$0 \leq \theta_2 \leq \pi ; 0 \leq \theta_5 \leq \pi \quad (30)$$

Considering the above constraints, only the positive root of (3) is feasible. The moving base for the virtual 3-DOF planar robot needs to be determined. This paper models the moving base using a cubic Bezier curve with the control points: $B_0(80, 30)$, $B_1(70, 30)$, $B_2(70, 10)$, and $B_3(80, -21)$. This moving base should be kept inside the workspace of the first 3-DOF planar series manipulator.

Computing the angle domain boundary for the first 3-link manipulator to track the cubic Bezier curve of the moving base, the angle domain boundary is illustrated in Fig. 14(a). Fig. 14(b) shows the angle domain boundary to track the clothoid curve with respect to the moving base. The value of amplitude used is 0.4 for both the fixed base and the moving base. For the value of θ_{gi} , this paper uses 0.35 and 0 for the fixed base and the moving base, respectively. The searching area for f is $[0, 15]$. Detail of the path planning results is presented in Table III. As in the 3-DOF planar series manipulator, for the fixed base, the PSO has outperformed the GA and the GWO where the PSO yields the lowest fitness

value. For the moving base, the PSO and the GWO converge to the same optimal value. They have lowest fitness value as compare with the GA result.

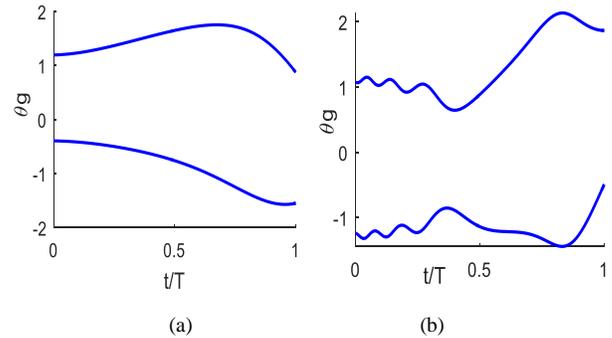


Fig. 14. Angle domain boundary (a) fixed base (b) moving base.

TABLE III. PATH PLANNING OF CLOTHOID PATH BY 6-LINK MANIPULATOR

	Fix base		Moving base	
	Fitness	f	Fitness	f
GA	5.1635	0.5304	9.6343	0.1748
PSO	5.147048672	0.5524	9.6279	0.1551
GWO	5.147048758	0.5523	9.6279	0.1551

Fig. 15 shows the fitness value evolution during 100 iterations to track the Bezier curve with respect to the fixed base for the GA, the PSO, and the GWO. Fig. 16 shows the fitness value evolution during 100 iterations to track the clothoid curve with respect to the moving base for the GA, the PSO, and the GWO.

Fig. 17(a) and 17(b) show the joint angle domain trajectories for the first 3-link and the second 3-link using the optimal value of f , respectively. Here, the fourth joint angle is in the form of the absolute angle where the positive direction is calculated counter clockwise from the x -axis. The configuration of the optimal path to track the clothoid using the 6-DOF hyper-redundant manipulator is illustrated in Fig. 17(c).

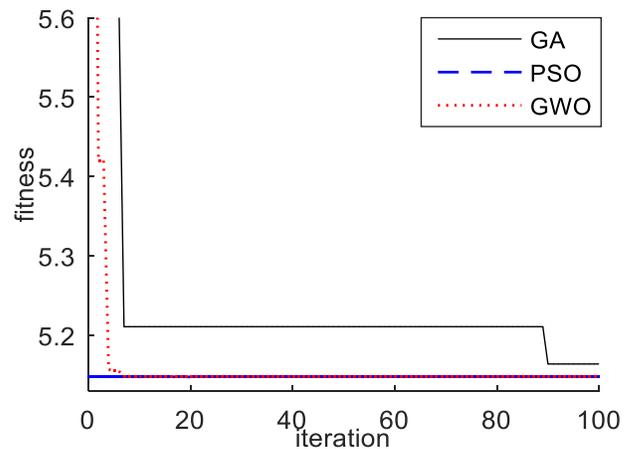


Fig. 15. GA, PSO, GWO fitness value evolution for fix based.

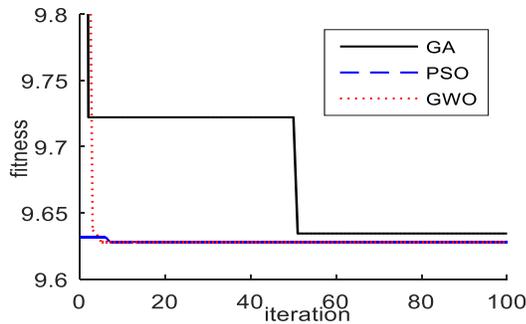


Fig. 16. GA, PSO, GWO fitness value evolution for moving base.

The results in this section have shown that the developed approach have succeeded to solve the continuous path planning of the redundant manipulator. The approach has also succeeded to be applied to the hyper-redundant manipulator. The proposed approach is based on the interval analysis of the self-motion of the 3-link serial manipulator. The connectivity of the generated path is achieved by modeling the angle domain trajectories inside the angle domain boundary. The proposed approach is very promising for solving the redundant/hyper-redundant manipulator since it does not require the matrix inversion. To select the best solution among many possible solutions, the meta-heuristic optimization is employed with the objective is to minimize the joint angle traveling distance. The optimization can be focused in keeping the angle domain trajectories within the angle domain boundary of the traced path while satisfying the optimization criterion. In general, the PSO has better performance than that of the GA and the GWO, except for the case of hyper-redundant manipulator in tracking the clothoid curve where the PSO and the GWO converge to the same fitness value for the moving base.

For future works, developing the computational strategy to generate the angle domain trajectories inside the boundary which give better results than that of the sinusoidal function is an open research to be conducted.

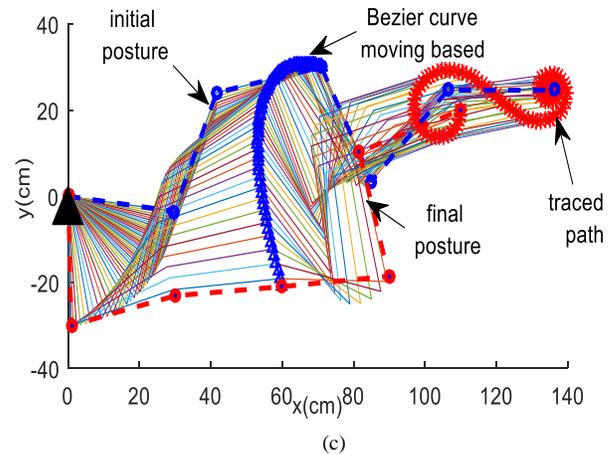
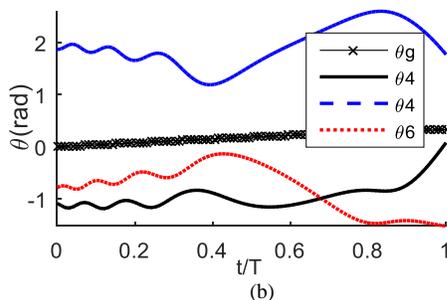
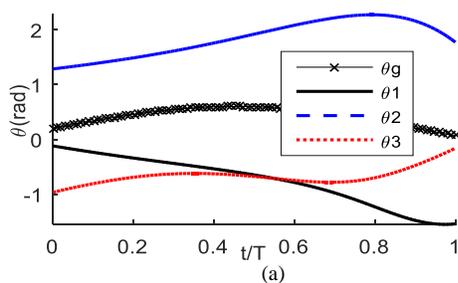


Fig. 17. Optimal result (a) joint angle of fix base, $f=0.5524$ (b) joint angle of moving base, $f=0.1551$ (c) configuration of 6-DOF manipulator to track the clothoid curve.

VII. CONCLUSIONS

The path planning algorithm of the redundant and hyper-redundant manipulators to track the complex geometrical path has been developed. The self-motion of the traced path was mapped into the interval of the angle domain and the redundancy resolution existed within the angle domain boundary. Generating the angle domain trajectories using the continuous function, namely the sinusoidal function, the connectivity among the sampling points can be achieved where the generated joint angle trajectories and posture were smooth. To solve kinematic redundancy problem, the meta-heuristic optimization, namely the GA, the PSO, and the GWO was employed to achieve the optimal solution with the objective optimization is to minimize the joint angle travelling distance. Results showed that the PSO had better performance than that of the GA and GWO where during 100 iterations the PSO yielded the lowest fitness value.

REFERENCES

- [1] X. Pan, J. Polden, N. Larkin, S. V. Duijn, J. Norrish, "Recent progress on programming methods for industrial robots," *Robot. Comput.-Integr. Manuf.*, Vol. 28, pp. 87-94, 2012.
- [2] D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, Vol. 10, pp.47-53, 1969
- [3] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 722-728, 25-28 March 1985.
- [4] C. W. Wampler, "Inverse kinematic function for redundant manipulators," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 610-617, March 1987
- [5] J. W. Burdick, "On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds," *proceedings IEEE International Conference on Robotics and Automation*, pp. 264-270, May 1989.
- [6] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *Int. J. Robot. Res.* Vol. 14, pp.19-36, 1995.
- [7] K. Tcho'n, J. Karp'nska, M. Janiak, "Approximation of Jacobian inverse kinematics algorithms", *Int. J. Appl. Math. Comput. Sci.* Vol. 19, pp. 519-531, 2009

- [8] G. S. Chirikjian, J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics Robot," IEEE Trans. Robot. Autom. Vol. 10, p. 343-354, 1994.
- [9] S. Yahya, M. Moghavvemi, H. A. F. Mohamed, " Geometrical approach of planar hyper-redundant manipulators: Inverse kinematics, path planning and workspace," Simul. Model. Pract. Theory, Vol. 19, pp. 406-422, 2011.
- [10] S. Ebrahimi, P. Payvandy, "Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms," Mech. Mach. Theory Vol. 85, pp.189-204, 2015
- [11] C. Yue, H-J. Su, Q. J. Ge, "A hybrid computer-aided linkage design system for tracing open and closed planar curves," Comput.-Aided Des. Vol. 44, pp. 1141-1150, 2012.
- [12] J. P. Merlet, "A generic trajectory verifier for the motion planning of parallel robot," J. Mech. Des., Vol. 123, pp. 510-515, 2001.
- [13] Z. Yao, K. Gupta, "Path planning with general end-effector constraints," Robot. Auton. Syst. Vol. 55, pp. 316-327, 2007.
- [14] H. Ananthanarayanan, R. Ordonez, "Real-time inverse kinematics of (2n+1) DOF hyper-redundant manipulator arm via a combined numerical and analytical approach," Mech. Mach. Theory, Vol. 91, pp. 209-226, 2015
- [15] P. Wenger, P. Chedmail, "Ability of a robot to travel through its free workspace," Int. J. Robot. Res. Vol. 10, pp. 214-227, 1991.
- [16] P. Wenger, "Performance Analysis of Robots." in Robot manipulators: modeling, performance, analysis, and control, E. Dombre, W. Khalil, Eds. ISTE Ltd, London, UK, 2007, pp. 141-183
- [17] J. Angeles, "Fundamental of Robotic : Mechanical Systems: Theory, Methods, and Algorithms," Springer, New York, NY, 2002
- [18] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization," Proceedings IEEE International Conference on Neural Networks, pp. 1942-1948, Nov 1995.
- [19] R. C. Eberhart, Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," Proceedings IEEE Congress Evolutionary Computation, pp. 84-88, July 2000.
- [20] S. Mirjalili, S. M Mirjalili, A. Lewis, "Grey wolf optimizer," Adv. Eng. Softw., Vol. 69, pp. 46-61, 2014.