Journal of System Simulation

Volume 27 | Issue 8 Article 28

8-3-2020

Least-energy Maneuver of Five-link Manipulator Constrained within Tunnel Space Using Direct Collocation

Xiuqiang Pan
College of Information and Communications, Zhejiang Industry and Trade Vocational College, Wenzhou 325003, China;

Chengcai Mei
College of Information and Communications, Zhejiang Industry and Trade Vocational College, Wenzhou 325003, China;

Junjie Chen
College of Information and Communications, Zhejiang Industry and Trade Vocational College, Wenzhou 325003, China;

Least-energy Maneuver of Five-link Manipulator Constrained within Tunnel Space Using Direct Collocation

Abstract

Abstract: *Optimal control and designs least-energy maneuver control laws for a five-linked manipulator were applied in order to carry out designated tasks in a confined space*. Lagrange-Euler equation described the relationships between the actuators and system dynamics. Euler-Lagrange formulation indicates how optimization can be achieved when optimum occurs. *Direct collocation method was introduced in order to solve this highly nonlinear dynamic optimal control problem.* Simulations were done to exploit how the manipulator reacted to the constraint. In this study, the diameter of the cylindrical space was shrunken each time by 0.1 meters. The value of the cost function and final time observed.

Keywords

robotic control, optimization, direct collocation, confined workspace

Recommended Citation

Pan Xiuqiang, Mei Chengcai, Chen Junjie. Least-energy Maneuver of Five-link Manipulator Constrained within Tunnel Space Using Direct Collocation[J]. Journal of System Simulation, 2015, 27(8): 1844-1852.

第 27 卷第 8 期 系统仿真学报© Vol. 27 No. 8
2015 年 8 月 Journal of System Simulation Aug., 2015

Least-energy Maneuver of Five–link Manipulator Constrained within Tunnel Space Using Direct Collocation

Pan Xiuqiang, Mei Chengcai, Chen Junjie

（College of Information and Communications, Zhejiang Industry and Trade Vocational College, Wenzhou 325003, China）

Abstract: *Optimal control and designs least–energy maneuver control laws for a five–linked manipulator were applied in order to carry out designated tasks in a confined space*. Lagrange–Euler equation described the relationships between the actuators and system dynamics. Euler–Lagrange formulation indicates how optimization can be achieved when optimum occurs. *Direct collocation method was introduced in order to solve this highly nonlinear dynamic optimal control problem.* Simulations were done to exploit how the manipulator reacted to the constraint. In this study, the diameter of the cylindrical space was shrunken each time by 0.1 meters. The value of the cost function and final time observed.

Keywords: robotic control; optimization; direct collocation; confined workspace

基于直接配置的隧道五连杆机器手臂能耗优化控制

潘修强，梅成才，陈军杰

（浙江工贸职业技术学院信息传媒学院，温州 325003）

摘要：*使用最优控制法则，进行五连杆手臂在受限制空间中的最低能耗控制研究*。拉格朗日-欧拉方程描述了控制器与动态力学的关系，欧拉-拉格朗日公式规范了最优状况下的必要条件，针对此必要条件经常带出“两端点边界值问题”，*使用“直接定位法”用来寻找高度非线性最优控制问题的数值解*。针对诸多不同的限制场景，依序持续缩小隧道半径，每次缩小半径 0.1 m，观察记录值函数及完成任务所需时间，以此进行针对性仿真研究，并给出研究结果。

关键词：机器人控制；最优化；直接配置；受限空间

中图分类号：TP391.9 文献标识码：A 文章编号：1004-731X (2015) 08-1844-09

Introduction

Robotic manipulators have been applied to automobile industry production lines, hazardous object handling sites, payload deploying on space shuttle and so on. However, these manipulators do not necessarily inhabit any work–space which is irked as is in an undersea tunnel environment, and the cylindrical compartment environment in space station is another example. A manipulator inside of the space station has to complete its work without poking or scratching the inner surface of the compartment. This study would like to exploit how the performance of a manipulator arm may vary with and without work-space constraints.

Many control schemes have shown their fortes in various applications. For example PID control

Received：2015-05-15 Revised: 2015-07-09;
Foundation： Expert Pioneer Project of Zhejiang (LJ2013146);
Biography: Pan Xiuqiang (1978-), Male, Yongjia, Zhejiang, China, Associate professor, Master, Optimal Control, Image processing; Mei Chengcai (1980-), Male, Pingyang, China Master; Chen Junjie (1960-), Male, Ph.D.

provides straightforward design for simple systems. Stochastic control allows the design to go on with statistic uncertain elements existing in the system. This study has its own concern–energy efficiency. Any robot deployed into fields on the Earth or orbits in the space is likely to operate on its battery power. The capacity of the battery array determines the length of operation time and this is a good reason why least energy maneuvering should be incorporated when laying out the control laws. Optimal control is the main color in this study and Euler–Lagrange formulation indicates how the control laws should look like when the optimum is reached. Optimal control problems are likely plagued with two-point boundary-values problem (T.P.B.V.P.) in many cases. The second goal of this study is to show how TPBVP can be done with direct collocation method in a breezy way.

1 Modelling the System

A heuristic five–linked manipulator arm is adopted as model. The manipulator links are assumed to be both inflexible and slender rods. This excludes the phenomenon of structure vibration. The Denavitt–Hartenburg(D–H) convention describes the geometry of the manipulator[1].The Lagrange–Euler equation formulates the relationships between control inputs and system dynamics. A dilemma looms. As the study goes on, the engineer has to make a decision on how the equations of motion should be derived. (1) is a dynamic equation involving total and partial derivatives. (2) is the Lagrangian of (1) which contains a three–fold cascade of summation with inertial matrix sitting in the middle of matrix multiplication. Other than the assumption that all links are rigid, this study makes no further simplification in deriving (1) and (2) so that the dynamics is best conserved and the manipulator system remains authentic.

Lagrange-Euler equation, which is also the equation of motion (E.O.M.), is a set of highly nonlinear differential equations, and the D–H convention furthermore complicates the situation. Yet it is still not the worst nightmare. Once the E.O.M. is plugged into the Euler-Lagrange formulation for optimal solution, the system is plagued with T.P.B.V.P. as most optimal control problems would encounter[2].

Direct collocation with nonlinear programming method (D.C.N.L.P.) is employed in this study as the major tool to crack the nutshell of the looming T.P.B.V.P. [3]This method converts E.O.M., which isa set of 10differential equations, into a set of $10\times N$ difference equations at N sub−intervals of time [4]. Links 1, 3 and 5 swivel side to side in their local horizontal planes. Links 2 and 4 pitch up and down in their local vertical planes. Fig. 1 shows the configuration of the manipulator and five homogeneous transformation matrices describe the geometry of the manipulator. The following homogeneous transformation matrices are used for describing the orientation and position of a link coordination frame. This transformation simplifies many operations when handling the mathematics of the manipulator arm. The D–H transformation matrix follows the following rules:

- Rotate about the $\vec{z}_{i-1}$ with an angle θ_i. Keep the $\vec{z}_{i-1}$ aligned with the $\vec{x}_i$ axis.
- Translate along the $\vec{z}_{i-1}$ with a distance of d_i in order to bring $\vec{x}_{i-1}$ and $\vec{x}_i$ into coincidence.
- Translate along the $\vec{x}_i$ a distance of α_i to bring two origins into coincidence.
- Rotate about the $\vec{x}_i$ an angle of α_i to bring the two frame systems into coincidence.

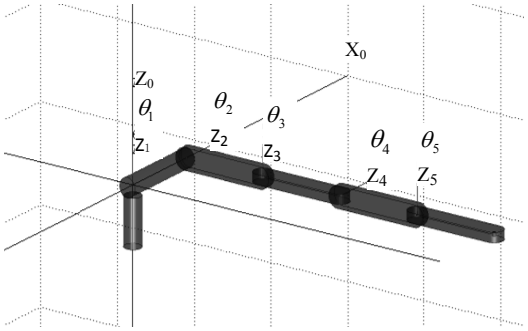

Fig. 1 Geometry of the five-link manipulator

$$ {}^{0}A_1 = \begin{bmatrix} \cos\theta_1(t) & -\sin\theta_1(t) & 0 & 0 \\ \sin\theta_1(t) & \cos\theta_1(t) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}A_2 = \begin{bmatrix} 0 & 0 & 1 & L_1 \\ -\cos\theta_2(t) & \sin\theta_2(t) & 0 & 0 \\ -\sin\theta_2(t) & -\cos\theta_2(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}A_3 = \begin{bmatrix} \cos\theta_3(t) & -\sin\theta_3(t) & 0 & L_2 \\ 0 & 0 & -1 & 0 \\ \sin\theta_3(t) & \cos\theta_3(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}A_4 = \begin{bmatrix} \cos\theta_4(t) & -\sin\theta_4(t) & 0 & L_3 \\ 0 & 0 & -1 & 0 \\ -\sin\theta_4(t) & -\cos\theta_4(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{4}A_5 = \begin{bmatrix} \cos\theta_5(t) & -\sin\theta_5(t) & 0 & L_4 \\ 0 & 0 & -1 & 0 \\ -\sin\theta_5(t) & -\cos\theta_5(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

where $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$, $\theta_4(t)$ and $\theta_5(t)$ denote the angular displacement at each joint. L_1 , L_2 , L_3 , L_4 and L_5 denote the length of each link.

2 Equations of Motion

Since the manipulator is a mechanism of five linkages, Lagrange–Euler formulation is introduced to describe the relationships between the system dynamics and control inputs. The formulation and its Lagrangian are listed as in (1) and (2).

$$ \vec{\tau}(t) = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\vec{\theta}}}\right) - \frac{\partial L}{\partial \vec{\theta}} \tag{1} $$

$$ L = \frac{1}{2}\sum_{i=1}^{5}\sum_{j=1}^{i}\sum_{k=1}^{i}[Tr(U_{ij}J_iU_{ik}^T)\cdot\dot{\theta}_j\dot{\theta}_k] + \sum_{i=1}^{5} m_i\vec{g}({}^{0}A_i\cdot{}^{i}\vec{r}_i) \tag{2} $$

where

$$ J_i = \begin{bmatrix} \frac{1}{3}m_iL_i^2 & 0 & 0 & \frac{1}{2}m_iL_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}m_iL_i & 0 & 0 & m_i \end{bmatrix} $$

$$ U_{ij} = \begin{cases} {}^{0}A_{j-1}Q_j\,{}^{j-1}A_i & 0 \leqslant j < i \\ 0 & j > i \end{cases}, 1 \leqslant i, j \leqslant 5 $$

$$ \frac{d^{j-1}A_j}{d\theta_j} = Q_j({}^{j-1}A_j) $$

$$ \vec{g} = [0 \quad 0 \quad -9.81 \quad 0] $$

$\vec{\tau}(t)$ denotes the control inputs generated by the five actuators at each joint and J_i denotes moment of inertia pertaining to each link. Rearrange (1) and it appears that(1) can be reduced to a symbolic form (3).

$$ \vec{\tau}(t) = D(\vec{\theta})\cdot\ddot{\vec{\theta}}(t) + \vec{h}(\vec{\theta},\dot{\vec{\theta}}) + \vec{c}(\vec{\theta}) \tag{3} $$

Further algebraic manipulations make the Lagrange–Euler formulation (3) more consenting for state space operations as is shown in (4).

$$ \ddot{\vec{\theta}}(t) = D^{-1}(\vec{\theta})\cdot(-\vec{h}(\vec{\theta},\dot{\vec{\theta}}) - \vec{c}(\vec{\theta}) + \vec{\tau}(t)) \tag{4} $$

The state variable vector is accordingly

$$ \vec{x}(t) = [\theta_1(t), \dot{\theta}_1(t), ..., \theta_5(t), \dot{\theta}_5(t)]^T $$

According to Park, any direct derivation of (1) and (2) for manipulator with more than five links without any kind of simplification is unheard[5]. It has always been a challenge to derive the Lagrange–Euler formulation with pencil and paper when the number of link is larger than two. Not only it is very liable to introduce unanticipated mistakes during the derivation, but also the mistakes, once transplanted during the deriving, are very difficult to identify, locate and thus remove. Some approaches

assume the angular velocities are small and thus ignore them. It greatly simplifies the complexity of (1) but this assumption is not necessarily true. The Newton–Euler formulation might be efficient in computation but it is difficult to represent the model when the system gets advanced. This study derives Lagrange–Euler formulation without any form of simplification by using Mathematica. Symbolic programming excludes all the possible hand-made mistakes. Note that in (1) there is a total derivative term. Mathematica software does a good job in performing the total derivative faithfully while some other symbolic programing software can only do a partial differentiation. Without the function of total derivative, the symbolic derivation cannot be completed.

3 Optimal Control Theory

Once the E.O.M. (4) is laid out, this study applies optimal control theory to design the control laws which minimize the consumption of the energy consumed by the five actuators[6-8].(4) can be rewritten as (5).(5) is also known as state equations. According to Bryson and Ho, the necessary conditions for the optimum to occur are:

$$\dot{\vec{x}}(t) = \vec{f}_{10\times1}(\vec{x}(t), \vec{\tau}(t), t) \tag{5}$$

$$\dot{\vec{\lambda}}(t) = -\frac{\partial H}{\partial \vec{x}^*} \tag{6}$$

$$0 = \frac{\partial H}{\partial \vec{\tau}^*} \tag{7}$$

$$\vec{\lambda}(tf) = \left(\frac{\partial \phi}{\partial \vec{x}^*}\right)\Bigg|_{t_f}, \text{ and} \tag{8}$$

$$\text{some } x_i(t_0),\ x_j(t_f) \text{ and } \lambda_k(t_f) \text{ are given} \tag{9}$$

H is the Hamiltonian of the manipulator system. $\vec{\tau}^*$ is the optimal control. ϕ is the final conditions of the state variables, if any. $\vec{\lambda}$ is the Lagrange multiplier vector.

As soon as the simultaneous differential equation set of (5) and (6) is solved, the optimal control laws are obtained. Yet looking closely, one can tell that (5) and (6) together are by all means a two–point boundary–value problem (TPBVP) since the initial/final conditions of the state variables and Lagrange multipliers are prescribed incompletely at t_0 and t_f according to (8) and (9).

4 Direct Collocation with Nonlinear Programming

The abovementioned incompleteness of initial/final conditions complicates the situation for numerical iteration approach[9,10]. In order to activate the numerical iteration for the problem, indirect methods hazard guesses on the missing elements in the initial condition array and propagate the inexact initial conditions forward anyway along the time line. The guessed initial condition elements are amended according to the offsets between the propagated and prescribed final values. It goes on until the prescribed final values are satisfied. However, indirect methods do not guarantee convergence which turns out to be their most significant drawback. Instead of tweaking around the guesses on initial conditions, this study turns to DCNLP for robust approach.

4.1 State equation constraints

DCNLP discretizes the time history of a continuous TPBVP into *n* segments[11,12]. See Fig. 2. In the figure the new variables in each segment are denoted as $\vec{X}_i$ where $\vec{X}_i = [\vec{x}(t_i) \quad \vec{\tau}(t_i)]^T$. By Hermite interpolation, the cubic polynomials in each segment are determined not only by the values of $\vec{X}_i$ and $\vec{X}_{i+1}$ but also by the values of the time derivatives $\dot{\vec{X}}_i$ and $\dot{\vec{X}}_{i+1}$. Gauss–Leba to interpolation is also a reasonable approach because it provides better approximation than Hermit–Simpson interpolation does. However Gauss–Lebato requires more computations and thus is not a candidate in this study at this moment.

第 27 卷第 8 期 系统仿真学报 Vol. 27 No. 8
2015 年 8 月 Journal of System Simulation Aug., 2015

Once the cubic polynomials are determined, the slopes of the polynomials at the center of segment *i* can be easily acquired. They are denoted as $\vec{X}_c'$. See (10). In another hand, (5) produces another set of slope $\vec{f}_c$ at the center of the segment. See Fig. 3. $\vec{X}_c'$ is the interpolated slope vector at the center of the segment *i*, and $\vec{f}_c$ is the time derivative vector of the E.O.M. which represents the dynamics of the manipulator. DCNLP forces the interpolated derivatives $\vec{X}_c'$ to agree with the differential equations $\vec{f}_c$ by making $\vec{\Delta}_i$ a zero vector where

$$\vec{\Delta}_i = \vec{f}_c - \vec{X}_c' = \vec{f}_c + \frac{3}{2\Delta t}(\vec{X}_i - \vec{X}_{i+1}) + \frac{1}{4}(\vec{f}_i + \vec{f}_{i+1}) \quad (10)$$

Since there are 10 state equations in (5), it yields 10 defect functions for a single time segment. This study divides the time history into 100 segments, hence there are exactly 1000 defect functions originating from the state equations.

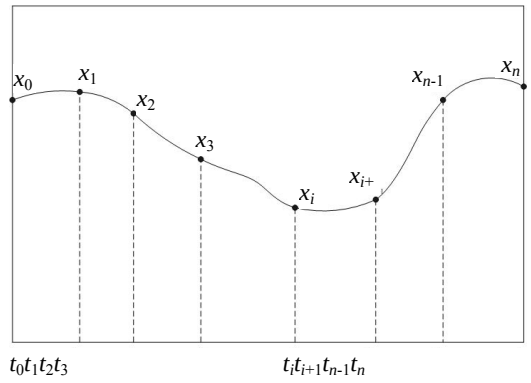

Fig. 2 Solution time history discretization.

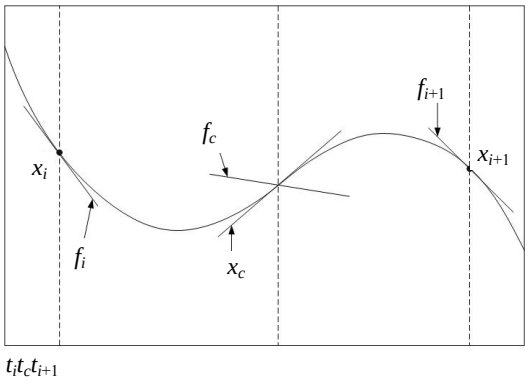

Fig. 3 Simpson's system constraint formulation.

4.2 End–effectors constraints

The state equations (5) may be treated as Newton's Laws constraints derived from Lagrange–Euler formulation. Yet there are still also other constraints. In this study, the manipulator is requested, say, to weld along a designated helix in the inner surface of acylindrical space. By common sense, other than the end-effectors, no part of the manipulator is allowed to scratch the inner surface of the tunnel. The trajectory of the helix is

$$\vec{R}(t)_{helix} = \begin{bmatrix} x(t)_{helix} \\ y(t)_{helix} \\ z(t)_{helix} \\ 1 \end{bmatrix} \quad (11)$$

where $t \in [0.0, 1.0]$ second.

The coordinate of the end–effectors of the robot manipulator at any time is

$$\vec{R}(t)_{\text{end}} = \begin{bmatrix} x(t)_{\text{end}} & y(t)_{\text{end}} & z(t)_{\text{end}} & 1 \end{bmatrix}^T = {}^0A_1 \bullet {}^1A_2 \bullet {}^2A_3 \bullet {}^3A_4 \bullet {}^4A_5 \bullet {}^5\vec{r}_{end} \quad (12)$$

where ${}^5\vec{r}_{\text{end}} = [L_5 \quad 0.0 \quad 0.0 \quad 1.0]^T$. In each time segment, $\vec{R}(t)_{\text{end}}$ must coincide with $\vec{R}(t)_{helix}$ to make certain that the welding task is correctly done. Thus three defect functions are created in (13) as

$$\vec{\Delta}(t)_{weld} = \vec{R}(t)_{end} - \vec{R}(t)_{helix} \quad (13)$$

(13) repeats through the 100 segments and thus produces 300 additional defect functions. DCNLP must do its best to push (13) to be zero by tweaking the state variables and the control elements. The core technology of crunching these numbers is nonlinear programming.

4.3 Impact clearance constraints

Joint 1 of the manipulator is adequately placed at $\vec{P}(0, y_0, z_0)$ within the cylinder so that Joint 2 does not impact the cylinder. However, Joints 3, 4 and 5 might scratch or slam into the enclosing

cylinder when the manipulator is in operation within the tunnel space. Any damage to the tunnel must be prevented. An inequality constrain can help achieve this goal.

The coordinates of Joints 3, 4 and 5 can be obtained as follows:

$$\vec{R}_3(t) = {}^0A_1 \bullet {}^1A_2 \bullet {}^2\vec{r}_3 \quad (14)$$

$$\vec{R}_4(t) = {}^0A_1 \bullet {}^1A_2 \bullet {}^2A_3 \bullet {}^3\vec{r}_4 \quad (15)$$

$$\vec{R}_5(t) = {}^0A_1 \bullet {}^1A_2 \bullet {}^2A_3 \bullet {}^3A_4 \bullet {}^4\vec{r}_5 \quad (16)$$

where ${}^i\vec{r}_{i+1} = [L_i \quad 0.0 \quad 0.0 \quad 1.0]^T$.(17) defines the interior space of the cylinder:

$$C(x,y,z):(y-y_0)^2+(z-z_0)^2-r^2<0 \text{ or}$$
$$\forall C_k(\vec{R}_i(t))<0, k=1,...,100 \quad (17)$$

As long as (17) holds, Joint *i* remains inside of the cylinder for all *t*.It guarantees that the manipulator mechanism does not damage the cylindrical interior tunnel surface at any time. Substituting (14), (15) and (16) into (17), it produces another300 inequality constraints. In total, there are 1000+300+300 constraints. All the possible solutions found must comply to the 1600 constraints. This study intends to identify the solution which consumes the least energy among all the possible solutions which satisfy all the constraints. Hence the cost function, a.k.a. objective function,*J*has to be specified.

$$J=\frac{1}{2}\sum_{i=1}^{100}(\tau_1^2[i]+\tau_2^2[i]+\tau_3^2[i]+\tau_4^2[i]+\tau_5^2[i])\triangle t \quad (18)$$

(18) is appended as the 1601st entryin the list of constraint.

The first thousand entries cover the dynamics of the manipulator. The next 300 entries ensure that welding task is accurately done. The following next 300 entries prohibit the mechanism of manipulator injuring the interior surface of the tunnel. The last entry defines the cost function which is what we are minimizing about. These 1601 equations bring the first part of the DCNLP into being and set up a ready-to-go stage for numerical iteration.

4.4 Jacobian matrix

Once the time history is discretized into 100 intervals, the state variables $\vec{x}(t)$ and control elements $\vec{\tau}(t)$ are transformed into a set of 1511 parameters:

$$\bar{\mathbf{x}}=[\theta_1(1),\omega_1(1)...\theta_5(1),\omega_5(1),\tau_1(1)...\tau_5(1),\ldots,$$
$$\theta_1(100),\omega_1(100)...\theta_5(100),\omega_5(100),\tau_1(100)...\tau_5(100),$$
$$\theta_1(101),\omega_1(101)...\theta_5(101),\omega_5(101),t_f]^T.$$

Note that there are no $\tau_1(101)...\tau_5(105)$. All the control inputs only exist at the center of each segment. On the contrary the state variables $\theta_1(i),\omega_1(i)...\theta_5(i),\omega_5(i)$ exist at the left and right boundaries of each segment and $1 \leqslant i \leqslant 101$.

The second part of DCNLP picks up from here. In order to calculate the 1601×1511Jacobian matrix, user is requested to provide a subroutine which takes the partial derivatives of the 1601 constraint equations against the 1511 parameters, i.e. $\bar{\mathbf{x}}$. The software SNOPT then sniffs around the Jacobian, tweaks about individual values of the parameters in $\bar{\mathbf{x}}$, and tries its best not only to diminish (10) and (13) but also to satisfy (17) with nonlinear programming techniques. The iteration halts as soon as (10), (13) and (17) are appeased.

5 Numerical results

The manipulator is placed at the center of the horizontal tunnel whose radius is r and its linkages are initially at rest. The manipulator end–effectors is required to weld along a helix $\vec{R}(t)_{helix}$ as given in (19).

$$\vec{R}(t)_{\text{helix}}=\begin{bmatrix}x(t)\\y(t)\\z(t)\end{bmatrix}=\begin{bmatrix}\frac{2t}{t_f}\\r\sin 2\pi t\\r\cos 2\pi t\end{bmatrix} \quad (19)$$

where $0 \leqslant t \leqslant t_f$ .Note that this is a final–time–free optimal control case. The final time t_f is hand–picked by DCNLP and it varies from case to case.

The nomenclature information of the manipulator is shown as follows.

$m_i = 1kg, \mathrm{L}_i = 1m$.

$x_0 = 0m, y_0 = 0m, z_0 = 0m$.

t_f unspecified.

$\vec{\theta}$ (0)and $\vec{\theta}(t_f)$: to be determined by SNOPT.

$\vec{\omega}(0) = \vec{0}$, $\vec{\omega}(t_f)$ is free.

Eight sets of simulation are performed. Each set contains two scenarios: one with interior tunnel surface constrain, and the other one without. The radius r of the tunnel is varied from 2m to 1.4m with 0.1mdecrement each time. The results are tabulated in Tables 1 and 1. The cost values and the final times spent are plotted against radius in Figs. 4 and 5.

Table 1 Cost *J* against radius r

No.	r	$J_{constrained}$	J_{free}
1	2.00	1809.3879	1809.3878
2	1.90	1587.1589	1766.6396
3	1.80	1520.0822	1517.7042
4	1.70	1305.1617	1321.1801
5	1.60	1133.5740	1131.4906
6	1.50	986.3006	985.6486
7	1.45	921.2707	921.2706
8	1.40	Not Available	866.2620

Table 2 Final time t_f against radius r

No.	r	$t_{f_{constrained}}$	$t_{f_{free}}$
1	2.00	2.5270	2.5270
2	1.90	2.4610	2.5031
3	1.80	2.3990	2.3885
4	1.70	2.2537	2.2525
5	1.60	2.1354	2.1370
6	1.50	2.0884	2.1021
7	1.45	2.0119	2.0119
8	1.40	Not Available	1.9374

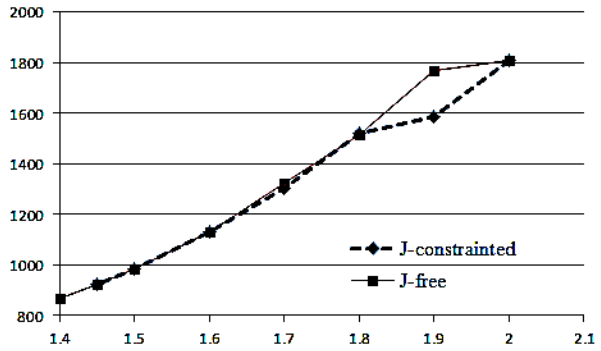

Fig. 4 Cost values of the constrained and unconstrained cases

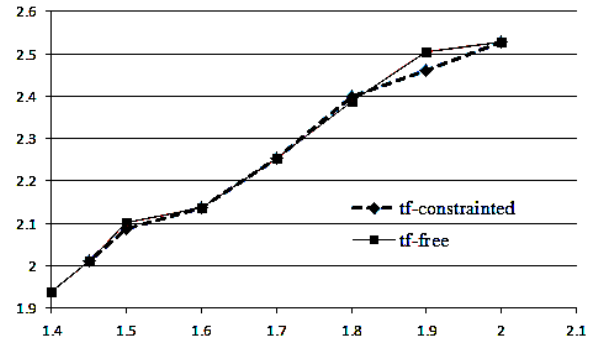

Fig. 5 Final time of the constrained and unconstrained cases

The following phenomena are observed from Tables I and II.

A. The larger the tunnel radius is, the longer it takes to complete the welding task.

B. The larger the tunnel radius is, the more cost, or energy, it takes to complete the job.

C. In general, cost J and final time t_f of the constrained cases are subtly smaller in magnitude than those of the constrain-freed cases.

D. Once the radius is smaller than 1.45 m, the manipulator fails to operate in the over-congested tunnel.

Three motion-stilled snapshots of the manipulator are manifested. Fig. 6 shows the motion of the manipulator which should be adequately constrained. According Fig. 6, at some moments, Joint 5 pokes out of the tunnel surface and would certainly cause damage to the inner surface of the tunnel. Figs. 7~10 show the constrained motions of

the manipulator with radii being 1.45 m, 1.5 m, 1.6 m and 1.90 m. In Figs. 7~10, the joints of the manipulator obviously stay clear of the interior tunnel surface due to the designated constrains mentioned in C in this section.

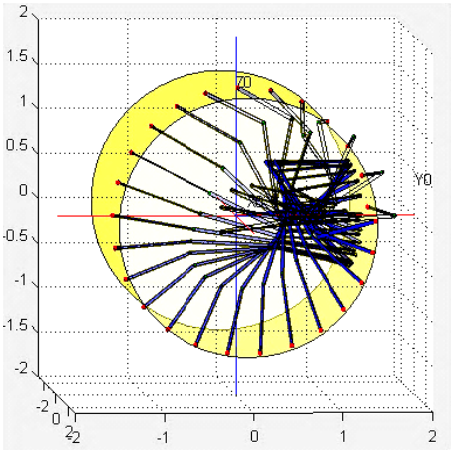

Fig. 6 Unconstrained motion of the manipulator withr=1.45m

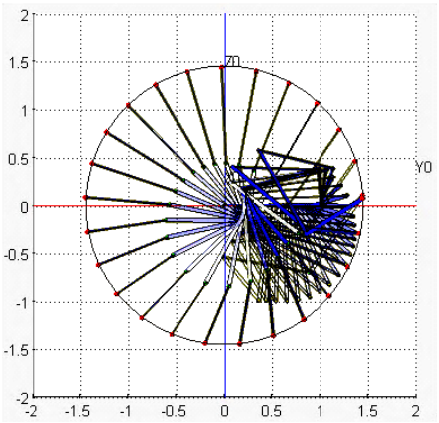

Fig. 7 Constrained motion of the arm with r=1.45m

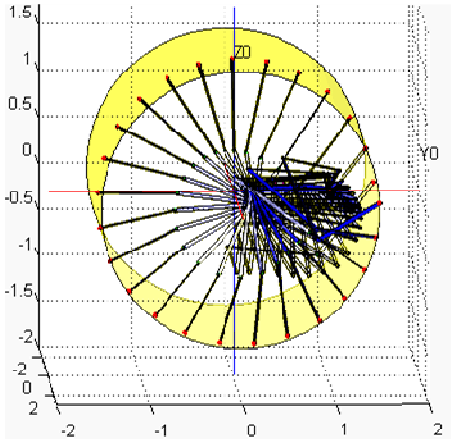

Fig. 8 Constrained motion of the arm with r=1.5m

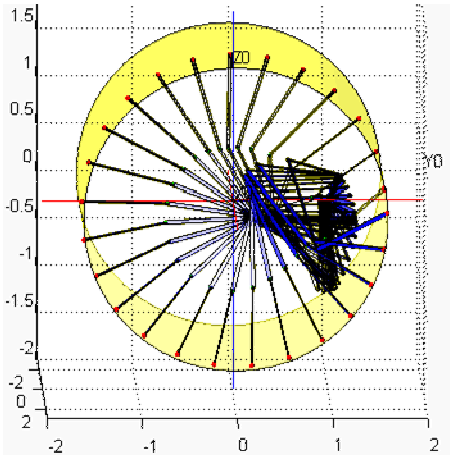

Fig. 9 Constrained motion of the arm with r=1.6

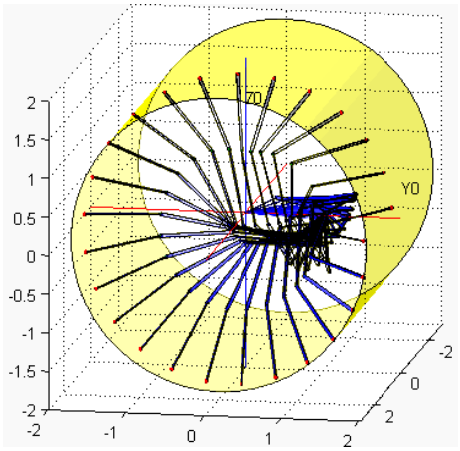

Fig. 10 Constrained motion of the arm with r=1.9m.

6 Conclusions

DCNLP is a straightforward and efficient approach to overrun TPBVP for many engineering applications. In this study, particular interest is laid on how a large–sized manipulator may efficiently work with least spent energy in a congested tunnel space without damaging the interior surface of the tunnel. Several points are drawn from the numerical results.

(1) According to Figs. 7, 8, 9 and 10, once the tunnel surface is meticulously built into DCNLP as constrain, see (17), DCNLP will figure out the optimal solution which will guarantee all the joints stay clear of the interior surface of the tunnel, and in the meantime complete the welding task along the inner surface of the tunnel.

(2) The simulation helps engineers identify that the smallest possible radius of the tunnel is 1.45 meters for feasible operation of the manipulator. Once the radius gets smaller, the manipulator fails to operate due to not enough turnaround space. Since the solution is exploited and found based on optimal control theory, there are good reasons to believe that no feasible solution exists when radius is smaller than 1.45 m. This study avails in making predictions about the necessary working condition for the manipulator in advance.

(3) Comparing with the indirect method, direct collocation method takes much larger memory space. The 1601×1511 Jacobian matrix mentioned in 4.4 devours a great portion of computer memory even SNOPT has already treated the colossal matrix as a band matrix. However computer memory is easily available nowadays. This disadvantage is no longer a drawback.

(4) Both the direct collocation method and indirect method have thirst for CPU speed. Due to the large amount of calculation, DCNLP method is still not yet a candidate suitable for real–time operation.

(5) Once the Lagrange–Euler formulation is derived, the symbolic results appear simply too complex which is totally beyond human ability to code them into Fortran program. This study takes the advantage of Mathematica software and use the function Fortran Form to translate the results directly into Fortran program. The outcome is fault free because it is processed by computer. It adds tremendous confidence that the result is not flawed by any human mistake. Symbolic programming plays a decisive role in making this study real.

This paper brings together some important crucial principles and techniques of robotics including mechanics, optimization, nonlinear programming and direct collocation. Nonetheless this paper tries its best efforts to demonstrate how DCNLP method can be introduced to instruct a manipulator to work in a congested working space. DCNLP successfully accomplishes the goal and solves the problem with elegance.

References:

[1] Fu K S, Gonzalez R C, Lee C. S. G., Robotics: Control, Sensing, Vision, and Intelligence [M]. USA: McGraw-Hill Book Company, 1987.

[2] Bryson A. E. Jr, Ho Y., Applied Optimal Control [M]. New York, USA: Hemisphere Publishing Corp., 1975.

[3] Gill Philip E., Murray Walter, and Saunders Michael A., User's Guide for SNOPT 5.3: a Fortran Package for Large-scale Nonlinear Programming, Stanford Business Software, Inc. [EB/OL]. (1998-12-07). http://tomopt.com/ docs/ sndoc.pdf.

[4] Satoh Satoshi, Fujimoto Kenji, Hyon Sang-Ho, Gait generation via unified learning optimal control of Hamiltonian systems [J], Robotica(S0263-5747), 2013, 31(5): 717-732.

[5] Park Jong-keun. Convergence properties of gradient-based numerical motion-optimizations for manipulator arms amid static or moving obstacles [J]. Robotica (S0263-5747), 2004, 22(6): 649-659.

[6] RovidAndras, Szeidl Laszlo, Varlaki Peter, Integral operators in relation to the HOSVD-based canonical form [J]. Asian Journal of Control (S1561-8625), 2015, 17(2): 459-466.

[7] Yao Jianjun, Yang Qi, GaoShuang, Hu Shenghai. Optimization design for a jumping leg robot based on generalized inertia ellipsoid [J]. Robotica(S0263-5747), 2012, 30(7): 1213-1219.

[8] Korayem M. H., Azimirad V., Vatanjou H., and Lorayem A. H., Maximum load determination of nonholonomic mobile manipulator using hierarchical optimal control [J]. Robotica(S0263-5747), 2012, 30(1): 53-65.

[9] Wei Wang, Yan Zhuang, Wei-min Yun. Innovative control education using a low cost intelligent robot platform[J]. Robotica (S0263-5747), 2003, 21(3): 283-288.

[10] Lyle Noakes, Tomasz Popiel. Geometry for robot path planning [J]. Robotica (S0263-5747), 2007, 25(6): 691-701.

[11] Perkins, A. D., Waldron, K. J.,Csonka, P. J.. Heuristic control of bipedal running: steady-state and accelerated[J]. Robotic (S0263-5747), 2011, 29(6): 939-947.

[12] Pan X Q, Mei C C, Chen J J.Solving Five-linked Manipulator Arm Interception Problem with Optimal Control Techniques[J]. Journal of System Simulation (S1004-731X), 2012, 24(9): 1772-1776.