Solving a Five-linked Robotic Manipulator CNC Problem Using Direct Collocation with Nonlinear Programming

Cheng-cai Mei, Xiu-qian Pan, Jyun-jye Chen and Xiu-long Wu
College of Information and Communications
Zhejiang Industry and Trade Vocational College
Wenzhou, Zhejiang, China
gmmeichengcai@126.com

***Abstract*—Computer numerical control (CNC) and robotic welding have long been applied to industrial manufacturing in production lines. This paper introduces optimal control to layout the maneuvering sequence for a five-linked manipulator arm. Two- point boundary-value problem (TPBVP) is inevitable in most of the dynamic optimal control problem. Direct collocation with Nonlinear Programming (DCNLP) converts a TPBVP into a nonlinear programming problem. DCNLP has been extensively applied in solving the space and aircraft control problems but is not much adopted in solving robotic optimization problems. The paper requires a manipulator to weld up two cylinders which are intersecting and perpendicular to each other. A least energy maneuvering sequence is expected.**

Keywords-direct collocation; robotic manipulator; optimal control

I. INTRODUCTION

One of the major uses of robotic manipulators is CNC. However many manipulators installed in the production lines only perform repetitive work once they are programmed. DCNLP is extensively applied in aerospace applications and orbital mechanics [1]. It has not yet become popular in the field of robotics using optimal control [1]. This study would like to take a deeper look into optimal control and see how a manipulator can go beyond with the aid of optimization technology.

When two cylinders intersect each other with a right angle, the equations of intersection of the two bodies become transcendental. This study combines robotic control and optimal control and requests the five-linked manipulator to weld up the intersection of two cylinders. The manipulator is modeled according to the Denavit-Hartenberg (D-H) convention [2]. The Lagrange-Euler equation formulates the dynamics and serves as the equations of motion (EOM). The necessary conditions (N.C.s) of optimal control describe the state variables, Lagrange multipliers, control elements and final time when the optimality occurs. However N.C.s inevitably end up with a TPBVP which complicates the procedures for finding numerical solution [3].

DCNLP converts a TPBVP into a nonlinear programming problem [4]. It divides the trajectory into segments and approximates the solution of EOM by cubic polynomials within each segment. This study seeks the least-energy maneuvering sequence for the five joint actuators when the manipulator welds up the two bodies. Although DCNLP requires larger memory space and faster CPU, it guarantees robust convergence when it locates the numerical solution.

II. DYNAMICS AND EQUATIONS OF MOTION

This work follows the D-H convention to formulate the geometry of the five-linked manipulator arm. The manipulator at its parking position is shown in Fig. 1.

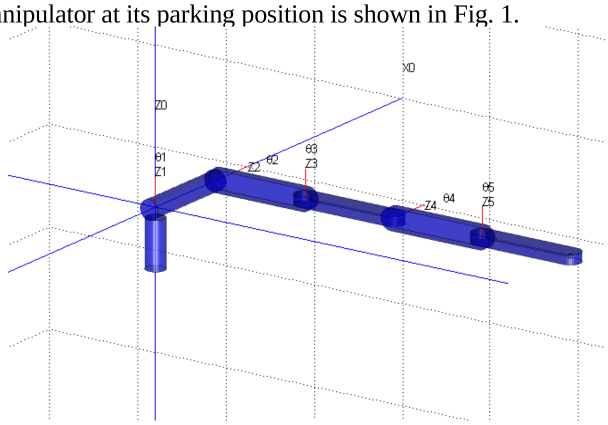

Figure 1. Configuration of the five-linked manipulator at its parking position.

The geometry of the manipulator is described by the five homogeneous transformation matrices.

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

$$^{1}A_2 = \begin{bmatrix} 0 & 0 & 1 & L_1 \\ -\cos\theta_2(t) & \sin\theta_2(t) & 0 & 0 \\ -\sin\theta_2(t) & -\cos\theta_2(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$^{2}A_3 = \begin{bmatrix} \cos\theta_3(t) & -\sin\theta_3(t) & 0 & L_2 \\ 0 & 0 & -1 & 0 \\ \sin\theta_3(t) & \cos\theta_3(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$^{3}A_4 = \begin{bmatrix} \cos\theta_4(t) & -\sin\theta_4(t) & 0 & L_3 \\ 0 & 0 & -1 & 0 \\ -\sin\theta_4(t) & -\cos\theta_4(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$^{4}A_5 = \begin{bmatrix} \cos\theta_5(t) & -\sin\theta_5(t) & 0 & L_4 \\ 0 & 0 & -1 & 0 \\ -\sin\theta_5(t) & -\cos\theta_5(t) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$, $\theta_4(t)$ and $\theta_5(t)$ denote the joint angles. L_1, L_2, L_3, L_4 and L_5 denote the length of each link and $L_i = 1$ meter. The Lagrange-Euler equation and the Lagrangian function are defined as (6) and (7).

$$\vec{\tau} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\vec{\theta}}}\right) - \frac{\partial L}{\partial \vec{\theta}} \text{ and} \quad (6)$$

$$L = \frac{1}{2}\sum_{i=1}^{5}\sum_{j=1}^{i}\sum_{k=1}^{i}\left[Tr\left(U_{ij} J_i U_{ik}^T\right)\cdot \dot{\theta}_j \dot{\theta}_k\right] + \sum_{i=1}^{5} m_i \vec{g}\left({}^{0}A_i \cdot {}^{i}\vec{r}_i\right) \quad (7)$$

where

$$J_i = \begin{bmatrix} \frac{1}{3}m_i L_i^2 & 0 & 0 & \frac{1}{2}m_i L_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}m_i L_i & 0 & 0 & m_i \end{bmatrix}, \quad (8)$$

$$U_{ij} = \begin{cases} {}^{0}A_{j-1} Q_j \, {}^{j-1}A_i & 0 \le j < i \\ 0 & j > i \end{cases}, \ 1 \le i, j \le 5 \text{, and} \quad (9)$$

$$\frac{d\,{}^{j-1}A_i}{d\theta_i} = Q_i \cdot {}^{j-1}A_i . \quad (10)$$

The gravitational acceleration is $\vec{g} = [0 \quad 0 \quad -9.81 \quad 0]$. $\vec{\tau}$ denotes the control element vector at each joint and m_i denotes the mass of the i-th link. $m_i = 1$ Kg.

III. Direct Collocation with Nonlinear Programming

A. *Discretizing the trajectory*

DCNLP discretizes the time history of EOM into 100 segments in this study. What it actually does is that DCNLP approximates (6), which is a set of 10 first-order differential equation, into a hundred sets of difference equations. Within each segment, cubic polynomials are formed to approximate the solution by using Hermite-Simpson's interpolation [5][6]. The slope of the cubic at the center of the segment is defined as x_c'. In the meantime, the time derivative of any state variable at any time, denoted as f_c, is available from (6). Once the defect functions $\Delta_i = f_c - x_c' = f_c + \frac{3}{2T}(x_i - x_{i+1}) + \frac{1}{4}(f_i + f_{i+1})$ are defined [7], they serve as a group of 10×100 constraint functions. As soon as DCNLP drives all the defect functions to zero, the numerical iteration converges and the optimal solution is found.

B. *The cost function and bead constraint functions*

This study requires the manipulator to weld up the two cylinders along the intersection. Since least-energy maneuvering control is the major tone of this study, the cost function is laid out as (11).

$$J = \frac{1}{2}\sum_{k=1}^{100}\left(\tau_1^2[k] + \tau_2^2[k] + \tau_3^2[k] + \tau_4^2[k] + \tau_5^2[k]\right)\cdot \frac{t_f}{100} . \quad (11)$$

The equations of the cylinders are $x^2 + (y+a)^2 = 1.21$ and $(y+a)^2 + z^2 = 1.21$. a is the offset distance between the cross-point of the cylinder centerlines and the origin. The intersection is comprised of eight segments of bead. See (12).

$$\begin{pmatrix} x = \pm\sqrt{1.21 - t^2} \\ y = t - a \\ z = \pm\sqrt{1.21 - t^2} \end{pmatrix} \text{and} \begin{pmatrix} x = \pm\sqrt{1.21 + t^2} \\ y = -t - a \\ z = \pm\sqrt{1.21 + t^2} \end{pmatrix}. \quad (12)$$

The exact position of end-effectors is obtained by the five homogeneous transformation matrices.
$\vec{r}_5 = [x_R \quad y_R \quad z_R \quad 1]^T = {}^{0}A_1 \cdot {}^{1}A_2 \cdot {}^{2}A_3 \cdot {}^{3}A_4 \cdot {}^{4}A_5 \cdot {}^{5}\vec{r}_5$.
Write it out explicitly and yield

$$\begin{aligned} x_R &= c_1\left(L_1 + L_3 s_3 + c_4\left(L_4 + c_5 L_5\right)s_3 + c_3 L_5 s_5\right) + \\ & s_1\left(-\left(L_4 + c_5 L_5\right)s_2 s_4 + \right) + \\ & s_1\left(c_2\left(L_2 + c_3\left(L_3 + c_4\left(L_4 + c_5 L_5\right)\right)L_5 s_3 s_5\right)\right), \end{aligned} \quad (13)$$

$$y_R = s_1\left(L_1 + L_3 s_3 + c_4\left(L_4 + c_5 L_5\right)s_3 + c_3 L_5 s_5\right) - c_1\left(-\left(L_4 + c_5 L_5\right)s_2 s_4\right) + c_1\left(c_2\left(L_2 + c_3\left(L_3 + c_4\left(L_4 + c_5 L_5\right)\right) - L_5 s_3 s_5\right)\right), \tag{14}$$

$$z_R = -L_2 s_2 - c_3\left(L_3 + c_4\left(L_4 + c_5 L_5\right)\right)s_2 - c_2\left(L_4 + c_5 L_5\right)s_4 + L_5 s_2 s_3 s_5 \tag{15}$$

where $c_i = \cos(\theta_i(t))$ and $s_i = \sin(\theta_i(t))$.

IV. Simulation and Results

For the sake of clearness of visual effect, only one of the eight beads will be looked into in this work. The offset distance a is set to be 2 meters. The manipulator is requested to finish the welding task on each bead exactly within t_f seconds. The specified t_f is 1.1 seconds which corresponds to the radius of the cylinders in (12).

Case 1: The end-effectors track along the bead from the upper left shoulder to the central front center of the body. The motion-still diagram of the manipulator is shown in Fig. 2. The cost J_1 of Case 1 is 2897.3255 which is found after the iteration.

Case 2: The end-effectors track along the bead from the upper right shoulder to the central front center. See Fig. 3. The simulation indicates J_2 is 1537.8889. It is to the authors' surprise that Case 2 takes less J than Case 1 does with a magnitude of half. Since Case 1 and Case 2 are nothing but symmetric operations with respect to $\vec{y}$ axis, the significant gap in J brings up curiosity to the authors. Thus Case 1 is investigated again.

Case 3: This case repeats what Case 1 does but two additional actions need to be added first in order to coach DCNLP how to work out a better iteration.

- Since Case 3 may be anticipated to be a mirror image operation of Case 2, the initial guess of $\theta_1(0)$ of Case 3 is set to be $180° - 6.4453°$, i.e. $173.5547°$, where $6.4453°$ is the initial value of $\theta_1(0)$ of Case 2 previously determined.
- Since Case 2 and Case 3 should ideally be symmetric operations with respect to $\vec{y}$, authors suggest that the torque outputs $\tau_1(t)$ of actuator #1 in Case 2 may flip the signs first and then be borrowed into Case 3 to be used as initial guess of $\tau_1(t)$ before kicking off the iteration.

These two additional edifications efficiently escort the numerical iteration to an eye-brushing new result. The new J_3 is no longer 2897.3255 but is 1527.8424. However, $\theta_1(0)$ of Case 3 is optimally selected to be $-10.7948°$ rather than $173.5547°$ by computer. See Fig. 4 and Table I.

TABLE I. Values of the Three Cost Functions

	Three Welding Cases		
	Case 1	***Case 2***	***Case 3***
J	J_1=2897.3255	J_2 =1537.8889	J_3 =1527.8424

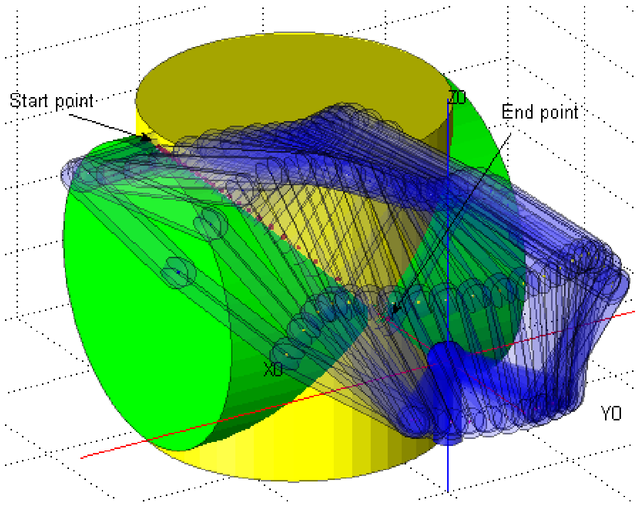

Figure 2. Motion-still diagram of Case 1.

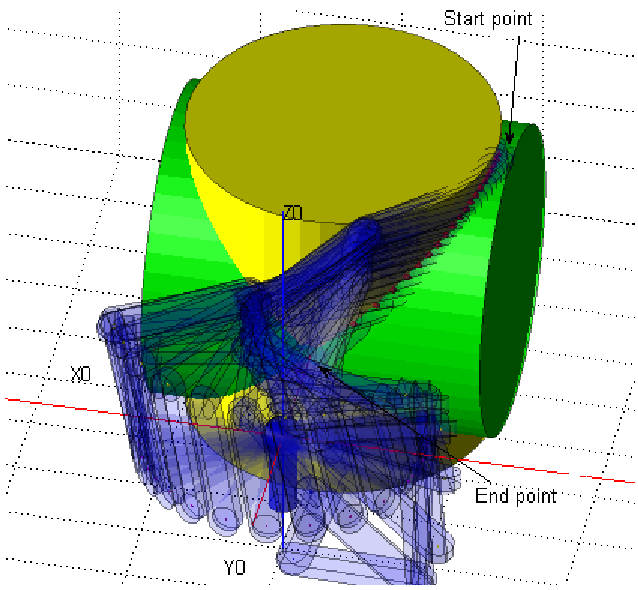

Figure 3. Motion-still diagram of Case 2.

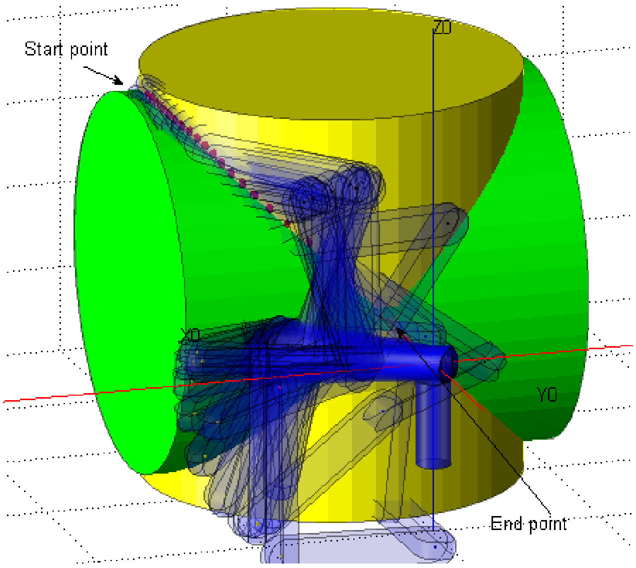

Figure 4. Motion-still diagram of Case 3.

V. CONCLUSIONS

It takes less energy to rotate a body if its moment of inertia is reduced. In Case 1, the manipulator extends its link length-wise while the motion is ongoing. However, in Case 2 it seems that the links are folding up when the manipulator tries to complete its task. See Figs. 2 and 3. That explains why J_2 is smaller than J_1.

According to Figs. 3 and 4, one can tell that the movements of links 1, 2 and 3 of Case 3 mainly stay in the first quadrant yet the movement of the same links of Case 2 span throughout the entire first and second quadrants. For this reason, J_3 is smaller than J_2. It again justifies that the numerical results found by DCNLP act in accordance with simple physics.

Comparing with indirect Runge-Kutta method, DCNLP does not discriminate how un-precisely the initial guesses are given for the iteration to start with. It may take some time for the iteration to converge but the convergence is always guaranteed. This is the forte of DCNLP. Cases 1 and 2 start with $\vec{x}(k)=\vec{0}$ plainly as the initial guesses where $1 \le k \le 101$. Both of them converge smoothly in their respective iterations.

.

As far as numerical method approach is concerned, it is a persistent nightmare that DCNLP might unearth a solution of local minimum particularly for a problem as complicate as (6) and (7). In this study, there are 1000 constraints subject to EOM and another 303 constraints subject to bead tracking. There are also 1510 state parameters waiting to be optimized. Case 1 and Case 3 are two existing extremities. DCNLP converges on both of them. However DCNLP fails to exclude Case 1 which is a local minimum. Conway suggests that Genetic Algorithm (GA) may be used to sniff out where the global or neo-global minimum solution is hiding in the solution domain [8]. The solution obtained from GA is then piped down to DCNLP serving as an initial guess, and then allow DCNLP to figure out the exact final optimal solution.

VI. FUTURE WORKS

The links of the manipulator arm often slew into the two cylinders in the process of welding. A path planning regarding obstacle avoidance and navigation is definitely needed to prevent the manipulator from damaging the welding pieces.

REFERENCES

[1] Brian R. Geiger, Joseph F. Horn, Anthony M. DeLullo, and Lyle N. Long, “Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming,” AIAA Paper No. 2006-6199, AIAA GNC Conference, Aug., 2006.

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

[3] A. E. Bryson Jr., Y. Ho, Applied Optimal Control, Hemisphere Publishing Corp, New York, 1975.

[4] C. R. Hargraves, and S. W. Paris, Optimal Trajectories by Implicit Simulation (OTIS), AFWAL-TR-88-3057, Nov. 1988.

[5] Jyun-jye F. Chen, Neighboring Optimal Feedback Control for Trajectories Found by Collocation and Nonlinear Programming, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1996.

[6] A. L. Herman and B. A. Conway, “Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules,” J. of Guidance, Control, and Dynamics, Vol. 19, No. 3, pp. 592-599, May-June 1996.

[7] Philip E. Gill, Walter Murray and Michael Sanders, User’s Guide for SNOPT 5.3: a Fortran Package for Large-scale Nonlinear Programming, Stanford Business Software, Inc., 1998.

[8] B. C. Conway, “A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems,” J Optim Theory Appl (2012) 152:271-306, doi 10.1007/s10957-011-9918-z