REVISTA COLOMBIANA DE COMPUTACIÓN
Volumen 8, número 2
Págs 26 - 38

Stochastic Algorithm for Search an Optimal Trajectory of an Automatic Manipulator

Fedossova A.V. * Santoyo J.S. †

Abstract

Is presented the stochastic outer approximations algorithm for robot trajectory planning. Initially, it is had an automatic manipulator of three degrees of freedom. The proposed problem consists of finding total optimal time of displacements which must adjust to the trajectory using cubic splines constrained by the speed, acceleration and jerk. It's a problem of optimization subject to an infinite set of constraints. In order to solve we used the stochastic outer approximations algorithm.

Keywords: outer approximations methods, semi-infinite programming.

1 Problem Definition

We considered a robot arm (manipulator) of *j* degrees of freedom $\theta 1$, $\theta 2$ and θj (j links). Each link has a length and mass associated. Since the robot position varies with time we can define a robot trajectory as a parametric curve:

$$\theta(T) = (\theta_1(T), \theta_2(T), ..., \theta_l(T))^T, \quad T \in [\ 0, T_f\ ],$$

where l is the number of degrees of freedom and T_f is the total travel time. Let " . " indicate the derivative with respect to the time t and " ' " the one with respect to T.

We formulated the optimization problem of optimal trajectory follows [8]. Suppose that we know n points in the trajectory of the manipulator. Are

$$[\theta_1(T_1), \theta_1(T_2), ..., \theta_1(T_n)], [\theta_2(T_1), \theta_2(T_2), ..., \theta_2(T_n)], ..., [\theta_l(T_1), \theta_l(T_2), ..., \theta_l(T_n)]$$

* National University of Colombia - Campus Bogotá, Avenue Street 30 N° 45-03 - Bogotá, Colombia, afedosova@unal.edu.co
† Autonomous University of Bucaramanga, Street 48 No. 39-234 Bucaramanga, Colombia, jsdiaz@unab.edu.co

vector points (nodes) which passes the trajectory of automatic manipulator. The proposed problem consists of finding total optimal time of displacements which must adjust to the trajectory using cubic splines constrained by the speed, acceleration and jerk. We get $t_1 < t_2 < \ldots < t_n$ a sequence at time where t_i this is the time where the robot is in the position [θ1 (T_i), θ2 (T_i), ..., θj (T_i)]. Natural constraints applied to the parametric curve are:

$$\sum_{i=1}^{t} \left(\frac{d\theta_i}{dT}\right)^2 > 0, \quad T \in (0, T_f)$$

$$\frac{d\theta}{dT}(0) = \frac{d\theta}{dT}(T_f) = 0$$

$$\frac{d^2\theta}{dT^2}(0) = \frac{d^2\theta}{dT^2}(T_f) \neq 0$$

Are $d_1 = t_2 - t_1$, $d_2 = t_3 - t_2$, ..., $d_{n-1} = t_n - t_{n-1}$ the displacement times. Is Q_{ij} cubic spline for the link i the robotic arm that approximates to $\theta_i(t)$ in [t_j , t_{j+1}].

Problem can then be formulated as SIP:

$$\min \sum_{j=1}^{n-1} d_j$$

$$\text{subject to } |Q'_{ij}(t)| \leq C_{i,1}$$

$$|Q''_{ij}(t)| \leq C_{i,2}$$

$$|Q'''_{ij}(t)| \leq C_{i,3}$$

$$i = 1, \ldots, l$$

where $C_{j,1}, C_{j,2}$ and $C_{j,3}$ are the bounds for the velocity, acceleration and jerk, respectively, on joint *i*.

We introduce $t = r(T), \ T \in [0, 1],$

and is $g(T) = r'(T) > 0, \ T \in [\ 0, 1\ ].$

So the problem is reformulated as

$$r(0) = 0$$

$$r(1) \quad \text{is minimum}$$

$$r'(T) > 0 \quad T \in [\ 0, 1\ ]$$

$$|\ \dot{\theta}_i^* \ | \leq C_{i,1}$$

$$|\ \ddot{\theta}_i^* \ | \leq C_{i,2}$$

$$|\ \dddot{\theta}_i^* \ | \leq C_{i,3} \ \ i = 1, ..., l$$

and $$\dot{\theta}^* \ (0) = \dot{\theta}^* \ (1) = 0$$

Where $C_{i,1}$, $C_{i,2}$ and $C_{i,3}$ are the bounds for the velocity, acceleration and jerk, respectively, on joint *i*, and

$$\theta^*(t) = \theta(r^{-1}(T))$$

is the parametric curve.

We get $$T = r^{-1}(t), \ \ t \in [\ 0, t_f\ ].$$

Then, $$\frac{dT}{dt} = \frac{1}{\frac{dr}{dT}} = \frac{1}{r'(T)} = \frac{1}{g(T)}.$$

By the chain rule we get,

$$\dot{\theta}_j = \frac{d}{dt}\{\theta_j(r^{-1}(t))\} = \frac{d}{dt}\{\theta_j(T)\}$$

$$\dot{\theta}_j = \frac{\theta'_j(T)}{g(T)}$$

$$\ddot{\theta}_j = \frac{g(T).\theta''_j(T).\frac{1}{g(T)} - \theta'_j(T).g'(T).\frac{1}{g(T)}}{g^2(T)}$$

$$\ddot{\theta}_j = \frac{\theta''_j(T) - \theta'_j(T).\frac{g'(T)}{g(T)}}{g^2(T)}$$

$$\dddot{\theta}_j = \frac{\theta'''_j(T) - 3\theta''_j(T).\frac{g'(T)}{g(T)} + \theta'_j(T).\frac{3(g'(T))^2 - g(T).g''(T)}{g^2(T)}}{g^3(T)} \quad .$$

Rewriting the constraints of previous model we have:

$$(-1).|\,\dot{\theta}_j\ (T)| \;\; \geq \;\; (-1)\cdot C_{j,1},$$

If

$$\frac{(-1)}{C_{j,1}}.|\,\dot{\theta}_j\ (T)| \geq -1$$

Then

$$|\,\dot{\theta}_j\ (T)| = \frac{\theta'_j(T)}{g(T)}.$$

Same way,

$$1 - \frac{(-1)^p}{C_{j,1}}.\frac{\theta'_j(T)}{g(T)} \geq 0.$$

we obtain

$$(-1).|\,\ddot{\theta}_j\ (T)| \;\; \geq \;\; (-1)\cdot C_{j,2}$$

And

$$\frac{(-1)}{C_{j,2}}.|\,\ddot{\theta}_j\ (T)| \geq -1.$$

After

$$1 - \frac{(-1)^p}{C_{j,2}}.\frac{\theta''_j(T) - \theta'_j(T).\frac{g'(T)}{g(T)}}{g^2(T)} \geq 0.$$

And

$$(-1).|\dddot{\theta}_j\ (T)|\ \geq\ (-1)\cdot C_{j,3}$$
$$\frac{(-1)}{C_{j,3}}.|\dddot{\theta}_j\ (T)| \geq -1.$$

Where we obtain

$$1 - \frac{(-1)^p}{C_{j,3}}.\frac{\theta_j'''(T) - 3.\theta_j''(T).\frac{g'(T)}{g(T)} + \theta_j'(T).\frac{3.(g'(T))^2 - g(T).g''(T)}{g^2(T)}}{g^3(T)} \geq 0,$$

Where

$$p \in \{\ 0,1\ \}, T \in [\ 0,1\ ]$$

And

$$j = 1,2,3.$$

Problem can then be rewrite as SIP:

$$\min_{x \in R^n} \int_0^1 g(T)\, dT, \qquad (1)$$

$$\text{subject to} \quad g(T) > 0$$
$$|\ \dot{\theta}_{j(T)}\ |\ \leq\ C_{j,1}$$
$$|\ \ddot{\theta}_j\ (T)|\ \leq\ C_{j,2}$$
$$|\ \dddot{\theta}_j\ (T)|\ \leq\ C_{j,3}$$
$$j = 1,2,3$$
$$\forall T \in [\ 0,1\ ],$$

Where $C_{j,1}$,$C_{j,2}$ and $C_{j,3}$ are the bounds for the velocity, acceleration and jerk, respectively, on joint *i*. The end conditions velocity equal to zero, that means

$$\dot{\theta}_j = 0,$$

and eventually also acceleration equal to zero.

$$\ddot{\theta}_j = 0,$$

are satisfied for

$$T \in [\ 0, \bar{T}\ ].$$

We consider with more detail the problem (1).

Mathematical model use B-Spline function in target function, then, it lets bring a curve given by some nodes which should generate the trajectory. A B-Spline is a linear combination of blending functions. A B-Spline can be represented by:

$$B_{k,\varepsilon}(t) = \sum_{i=1}^{n} x_i B_{i,k,\varepsilon}(t).$$

Approximating function *g(T)* in:

$$F(x) = \int_0^1 g(T)\, dT,$$

the target function of problem is:

$$F(x) = \sum_{i=1}^{n} a_i x_i \;,$$

$$\text{con } a_i > 0,$$

Subject to constraints:

$$H_0 = g - \epsilon \geq 0$$

$$H_{j,1} = 1 - \frac{(-1)^p}{C_{j,1}} . \frac{\theta_j'(T)}{g(T)} \geq 0$$

$$H_{j,2} = 1 - \frac{(-1)^p}{C_{j,2}} \frac{\theta_j''(T) - \theta_j'(T) \frac{g'(T)}{g(T)}}{g^2(T)} \geq 0$$

$$H_{j,3} = 1 - \frac{(-1)^p}{C_{j,3}} \frac{\theta_j'''(T) - 3\,\theta_j''(T) \frac{g'(T)}{g(T)} + \theta_j'(T) \frac{3\,(g'(T))^2 - g(T).g''(T)}{g^2(T)}}{g^3(T)} \geq 0,$$

where $p \in \{ 0,1 \}, T \in [0,1]$ and $j = 1,2,3.$

The proposed problem consists of finding total optimal time of displacements which must adjust to the trajectory using cubic splines constrained by the speed, acceleration and jerk.

2 Stochastic Outer Approximations Algorithm

We are going to use the Stochastic Outer Approximating Method to solve the problem for robot trajectory planning. This approach can be considered as a developed Eaves-Zangwill method applying the multistart technique at each iteration for the search of relevant constraints parameters [1] [2] [3].

This method consists of replacing the original SIP problem (1) with the sequence of the approximating problems, where each one depends the finite set of constraints:

$$P(T_n): \qquad \min_{x \in R^n} f(x) = \int_0^1 g(T_n)\, dT,$$

$$\text{subject to} \quad g(T) > 0$$

$$|\, \dot{\theta}_j \,| \;\leq\; C_{j,1}$$

$$|\, \ddot{\theta}_j \,| \;\leq\; C_{j,2}$$

$$|\, \dddot{\theta}_j \,| \;\leq\; C_{j,3}$$

$$j = 1, 2, 3$$

$$\forall T \in [\ 0, 1\ ].$$

We denote $M_c(T)$ and $M_f(T)$ as the set of all compact subsets and the set of all finite subsets of

$$T \subset [0, 1]$$

Respectively. For the optimal criterion we choose the quasi-optimality function

$$Q(x', \bar{T_n}) : X^0 \times M_c(\bar{T}) \longrightarrow R^1_+$$

Such that for any compact

$$T_n \subset T \text{ y } x' \epsilon X^0$$

is used in order, to evaluate the quality of x' as a original problem solution. The resolution of additional problems is not needed in order to calculate the value quasi-optimality function in contrast to other semi-infinite programming problems ([1]). So, for our case, the quasi-optimality function is:

$$Q(x, \bar{T}) = \max(f(x) - \min_{\substack{x \in X^0: \\ g(x,T) \leq 0 \forall t \in \bar{T}}} f(x), \max_{T' \in \bar{T}} g(x, T')) \qquad (2)$$

and the quasi-optimal set of the problem is:

$$\aleph_{qopt}[\bar{T}] := \{x \in X^0 | Q(x, \bar{T}) = 0\}, T \in M_c(\bar{T})$$

And

$$\aleph^0_{qopt} = \aleph_{qopt}[\bar{T}] := \{x \in X^0 | Q(x, \bar{T}) = 0\}.$$

We used the inequality

$$Q(x, \bar{T}) \leq \alpha$$

as a stopping criterion for solving problem.

Next, we will propose the stochastic outer approximations algorithm used as the general approach for solving robot trajectory planning.

2.1 Algorithm SMETH.ACTIV.sip

Solve the problem of SIP (step 1).
Call the procedure SPROC.ACTIV.sip sending like parameter the obtained solution
to him (step 2);
Receive from SPROC.ACTIV.sip the new ones relevant constraints (step 2).
Form the new constraints set (step 3).

Step 0. Set: $n := 1, T_1 := \phi, x^1 \in X^0$.
Step 1. Find x^n - solution of problem $P(T_n)$.
Step 2. Call the procedure SPROC.ACTIV.sip with parameters x^n and T_n.
To obtain ΔT_n and Q_n
Step 3. Form the new constraints set:

$$T_{n+1} := \Delta T_n \cup \bigcup_{\substack{j: Q_j > \delta/n \\ 1 \leq j \leq n-1}} \Delta T_j.$$

Step 4. Set $n := n + 1$. and back to Step 1.

2.2 Procedure SPROC.ACTIV.sip

Use random search of a constraint. (step 1)
Apply a local algorithm search to find the constraints that not satisfied starting off of the random constraint. (step 1)
Verify the fulfillment of optimal criterion of algorithm. If one is not fulfilled it returns to main program taking the relevant new constraints. (step 2,4)
If one is fulfilled we continued looking for the constraints nonfulfilled. (step 2,3)

◇ Input: $x \in X^0, T \in M_f(T)$.
◇ Output: $Q \in \Re^1_+, \Delta T \in M_f(T)$.
◇ Parameter: $\delta > 0$.
Step 0. $i = 1$.
Step 1. Apply the algorithm of local search for solution of problem

$$\max_{T \in T^0} g(x, T)$$

beginning from a random point T_i in order to obtain the point $T_i^* \in T^0$ so:

$$T_i^* \in T_{stat}^{\varepsilon}(x), \qquad g(x, T_i) \leq g(x, T_i^*).$$

Step 2. $Q_i = max(g(x, T_1^*), ..., g(x, T_i^*))$
Step 3. (Control step). If

$$i.Q_i \leq \delta$$

then $i := i + 1$ and back to Step 1.
Step 4.

$$Q := Q_i;$$
$$\Delta Y := \{T_i, T_i^*\}.$$

Exit.

The proposed algorithm is a version of the general SMETH.ACTIV algorithm proposed by Zavriev, Volkov in [3] with our quasi-optimality function (2).

3 Numerical Experiments

Designing an optimization algorithm for solve a particular problem, besides the means of achieving efficiency in terms of computing runtime, analysis and selection of most appropriate techniques to achieve a level of generalization allowing implement it and extend it for solution other problems closely related.

3.1 Pc Characteristics

Processor: Intel Pentium (R) 4 CPU 2.80 Ghz.
RAM memory: 512 MB.
The algorithms that are mentioned in this document have been codified in MATLAB 7.1. We are using the B-Spline function (function bspline()), which allows approximate a curve given by some nodes which should generate trajectory. This function is a AMPL library and it was rewritten in MATLAB for use in main program to calculate its restrictions and calculations required.

3.2 Using Algorithms

Algorithm SMETH.ACTIV.sip has been used to solve a SIP problem, minimizing the objective function with respect to decision variables such as: B-spline coefficient used in objective function to approximate a curve by nodes for which they must generate the trajectory. In terms of procedure SPROC.ACTIV.sip, it was used to optimize with respect to the variable ”*t*”.

3.3 Parameters Description

Parameters used in model are:
”*n* (number of coefficients)” and it must be equal to 9.
”*k* (degree of spline)” and it's equal to 4.
”*nk* (number of knots)” it's equal to *n* plus *k*, equal to 11 ($nk = n + k$).
”*Knots* (knots vector)” it's equal to the number of knots, then it's 11.
”*x0* (coefficients)” is 9.
”*t* (blending parameter)” time of cubic spline.
”*c* (compute nonlinear inequalities at)” = 19 for considered problem.
”*ceq* (compute nonlinear equalities at *x*)” = 0 (not used)
”*T* (set of active constraints)”.

In table 1 you can find different values to limits of derivatives *Q1, Q2* and *Q3* used in problem (1).

Within the made tests were taken three variants that modify the constraints of objective function (denominated within the document as: A, B and C), these variants modify parameters like speed, acceleration and jerk; the variants A and C have similar constraints and obtained results reflect this similarity verifying that the stochastic outer approximations algorithm is applied of correct way. In tables 1 - 5 are show results using the A model.

In tables 2 - 5 are some results of numerical experiments. These tables are composed of a variable X Initial or *x0* that it represents the initial variable value used in the iteration, *Iterations number of Smeth* variable indicates the number of times it has used the algorithm Smeth, *Run number* variable represents the experiments number, *X optimal* variable is the optimal value of the variable *X*, *F(X optimal)* variable is the value of objective function at the end of iteration. Finally each table shows the total number of restrictions active is final amount of active constraints of problem (1).

For escape local solutions SMETH.ACTIV.sip algorithm was used several times after their solutions were used as initial points for the following runs.

These results were compared with values obtained by Ismael Vaz [4] (F.O. = 1.08289), and by Haaren-Retagne [5] (F.O. = 1.08351) and we have been noticed that we got good approximations (see F(X optimal) in tables). This indicates that actually has been made to optimize the trajectory of the robot arm using the stochastic outer approximations algorithm [3].

Table 1. Limits of model's derived

Limits of derived									
	Q_1			Q_2			Q_3		
	C11	**C12**	**C13**	**C21**	**C22**	**C23**	**C31**	**C32**	**C33**
A	2	8	250	3	18	650	4	50	1000
B	1	3	100	1	3	100	1	3	100
C	2	8	25	3	18	65	4	50	100

Each iteration of MATLAB has the following values:

Table 2. Run number 1 of MATLAB

Variables	Values
X initial	$[x0]=[0{,}0801587\ \ 0{,}6396061\ \ 0{,}1770871\ \ 0{,}7820946\ \ 0{,}5897483\ \ 0{,}5399054\ \ 0{,}7961875\ \ 0{,}4101524\ \ 0{,}6586666]$
Iterations number of Smeth	14
Run number	1
X optimal	$[x]=[0{,}0034723\ \ 0{,}006945\ \ 0{,}01042\ \ 0{,}013889\ \ 0{,}01389\ \ 0{,}01388928\ \ 0{,}01041696\ \ 0{,}006944634\ \ 0{,}00347231]$
$F(X$**optimal**$)$	$Fo=1{,}0995676$
Total of Active Constraints	1097

Table 3. Iteration number 2 of MATLAB

Variables	Values
X initial	$[x0]=[0{,}3761619\ \ 0{,}0745724\ \ 0{,}5870587\ \ 0{,}608147\ \ 0{,}989497\ \ 0{,}7930172\ \ 0{,}9233792\ \ 0{,}418342\ \ 0{,}257821];$
Iterations number of Smeth	14
Run Number	2
X optimal	$[x]=[0{,}0034725\ \ 0{,}006945\ \ 0{,}0104174\ \ 0{,}01389\ \ 0{,}01388\ \ 0{,}013889896\ \ 0{,}01041712\ \ 0{,}006949432\ \ 0{,}003472469]$
$F(X$**optimal**$)$	$Fo=1{,}0996165$
Total of Active Constraints	1582

Table 4. Iteration number 3 of MATLAB

Variables	Values
X initial	$[x0]=[0{,}546839\ \ 0{,}742002\ \ 0{,}027482\ \ 0{,}978773\ \ 0{,}556293\ \ 0{,}52204655\ \ 0{,}71687644\ \ 0{,}98795475\ \ 0{,}07273141];$
Iterations number of Smeth	19
Run number	3
X optimal	$[x]=[0{,}0034725\ \ 0{,}0069449\ \ 0{,}0104174\ \ 0{,}0139\ \ 0{,}01389\ \ 0{,}01389886\ \ 0{,}010417412\ \ 0{,}006949381\ \ 0{,}003472465]$
$F(X$**optimal**$)$	$Fo=1{,}0996158$
Total of Active Constraints	2540

Table 5. Iteration number 4 of MATLAB

Variables	Valores
X initial	$[x0] = [0{,}5254609 \quad 0{,}700934 \quad 0{,}860602 \quad 0{,}940703 \quad 0{,}776184 \quad 0{,}10265818 \quad 0{,}05475759 \quad 0{,}47440722 \quad 0{,}2043938]$
Iterations number of Smeth	20
Run Number	4
X optimal	$[x] = [0{,}0034725 \quad 0{,}0069449 \quad 0{,}0104174 \quad 0{,}0130 \quad 0{,}01389 \quad 0{,}013889858 \quad 0{,}010417413 \quad 0{,}006944939 \quad 0{,}003472467]$
$F(X$optimal$)$	$Fo = 1{,}0996159$
Total of Active Constraints	1746

4 Conclusions

One of the main profits has been use of concepts of semi-infinite programming and especially the stochastic algorithm for solving of problems related with robotics.

Results obtained with stochastic outer approximating method by means of the computing tool prototype based on MATLAB, are appropriate for primary target of the research. Despite it is recommended for later studies, to use exactly initial and end points of trajectory in order to consider more appropriately, level of obtained improvement, since these comparisons were made starting off of itself model of trajectory and having like primary target minimize manipulator's time of route.

The use of MATLAB like programming tool, optimization, calculation and interface, facilitates the development remarkably and solution of mathematical problems of high complexity, since habitual development platforms such as C++, Java, among others, require of programming of mathematical tools that MATLAB already has implemented in his toolboxes, which allows to save effort of programming and lines of code.

Recent research opens the doors for development this thematic one in the future, based mainly in solution generalization to problem created, in search an optimal trajectory for different types from robots, in addition to its physical implementation in a real robot, where it is necessary to consider a sizing of all the mechanical parameters and control, applied in a specific solution within an objective function.

References

[1] Fedossova A., Kafarov V., Mahecha Bohórquez D.P., Outer Approximations Algorithm for the Air Pollution Control Problem. 2004.

[2] Fedossova A., Stochastic outer approximations algorithms for convex semi-infinite programming problems, Ph.D. Thesis, Moscow State University, Moscow, 2000, (in Russian).

[3] Volkov Y.V., Zavriev S.K., A general Stochastic Outer Approximations Method, SIAM J. Control Optim., 35 (4), 1997, pp. 1387-1421.

[4] A. Ismael F. Vaz, Applications, methods and tools y for SIP, Ph.D. Thesis. Minho University, 2003, (in Portuguese).

[5] Elke Haaren - Retagne. A semi-infinite programming algorithm for trajectory planning, Ph.D. Thesis. Trier University, 1992.

[6] Belén Melián, José A. Moreno Pérez, J. Marcos Moreno Vega. Metaheuristics: A global view, Spain - 2003.

[7] Barbolla, Rosa. Optimization: questions, exercises and applications, Prentice Hall - 2001 (in Spanish).

[8] Ismael Vaz, Robot trajectory planning with semi-infinite programming, SIAM Journal on Optimization, vol. 9, no. 2, 2001.

[9] Carrillo Esneider y Mora Gleimer. Outer Approximations Algorithm for minimizing costs of Air Pollution Control Problem. Bucaramanga, 2003, p.138-139. Autonomous University of Bucaramanga. School of Systems Engineering.

[10] Fedossova A., Kafarov V., Mahecha Bohrquez D.P., Outer Approximations Algorithm for the Air Pollution Control Problem. Colombian Journal of Computation. Vol. 4 No. 2, 2003.

[11] Chi,Mei-Hsiu, Linear semi-infinite and nondierentiable programming methods for robot trajectory planning problems. Ph.D. Thesis, Iowa University, 1991.

[12] Medvedev, V.C.; Potemkin, V.G., Neuronal Network, MATLAB 6.0. Dialog- MIFI, Moscow, 2002 (in Russian).