計算工学講演会論文集　Vol.28 (2023年5月)　　計算工学会

OpenModelica を使った速度履歴最適化と機構解析

Speed History Optimization and Multi Body Dynamics using OpenModelica

吉田史郎[1)]
Shiro Yoshida

1) 湘南技術開発株式会社 代表取締役（〒251-0035 神奈川県藤沢市片瀬海岸3-19-17湘南ホームズ402号, E-mail: Sir_Yoshida@nifty.com)

In the previous report of the author, the optimized operating procedure was obtained by converging the final solution of the state-space to be zero, which is composed of cart position, cart velocity, pendulum angle and cart driving force. And, this procedure was reproduced by using GEKKO Optimization Suite and OpenModelica. In this report, computational procedure of CasADi ,which is a combination of Runge-Kutta method and Direct Multiple Shooting is summarized. And finally, its calculation results are described.

Key Words : *State space equation, Optimization, OpenModelica, Python, CasADi, IPOPT*

1.　はじめに

前報[1,2]では外部ソフトGEKKO Optimization Suite[3]を使って天井クレーンと倒立振子の運動履歴を計算した後,その結果をOpenModelica[4]のLibrary接続機能を使って,再現した．これに対して,PythonでOpenModelicaを操作するという前提で,新たにCasADi[5]の最適化機能と連動する形でOpenModelicaを使って天井クレーンと倒立振子の運動履歴を計算する事を考えた．そこでRunge-Kutta法とDirect Multiple Shooting法を組み合わせたCasADiとOpenModelicaを連繋させる手順について報告する．

2.　天井クレーン走行最適化

(1) モデル化

吊り荷運搬中の天井走行クレーンの台車を急停止すると吊り荷が揺れる．一方,台車を増速～半減速～半増速～減速の手順で運転する事により，台車の停止とともに吊り荷を停止できる．その際，図-1～図-2に示すように,台車位置,台車速度,振子傾斜角,台車推進力から構成される状態空間方程式から求まる時刻歴応答のうち最終解をゼロに収束させる．

$$\dot{X} = AX + BU \quad (1)$$

$$\dot{y} = v \quad (2)$$

$$\dot{v} = \epsilon\theta + u \quad (3)$$

$$\dot{\theta} = q \quad (4)$$

$$\dot{q} = -\theta - u \quad (5)$$

u：外力（N）

ϵ：質量比　$m_2/(m_1+m_2)$

$m_1 = 10(kg)$　　$m_2 = 1(kg)$

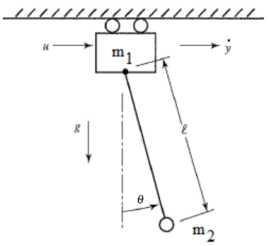
y：台車位置（m）

v：台車速度（m/s）

θ：振子傾斜角（rad）

q：振子傾斜角速度（rad/sec）

GEKKO Optimization Suite では動的応答の最適化に Interior Point Optimizer (ITOPT) を使っている．一方,CasADi では Runge-Kutta 法および direct multiple-shooting に基づいている．ここではその機能を概括するとともに,得られた計算結果について報告する．



$$\begin{bmatrix}\dot{y}\\ \dot{v}\\ \dot{\theta}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \epsilon & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -1 & 0\end{bmatrix}\begin{bmatrix}y\\ v\\ \theta\\ q\end{bmatrix} + \begin{bmatrix}0\\ 1\\ 0\\ -1\end{bmatrix} u$$

図-1 状態空間方程式：天井走行クレーン

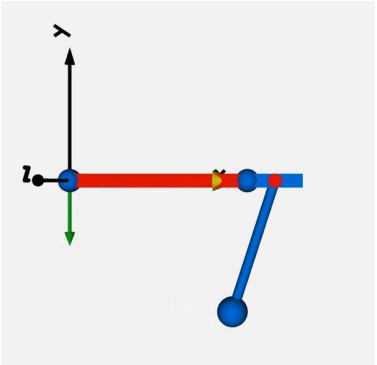


図-2 計算結果：天井走行クレーン

(2) **内点法** (Interior Point Optimizer : ITOPT)

内点法では最適化の対象となる目的関数(Objective Function)を障壁関数(Barrier Function : μ ln(x))と組み合わせて最適値を探索する．ここで言う障壁関数とは目的関数が最適値に近づくにつれて極小化し,最適値をわずかに外れたところで,急速に極大化する関数を指す.これを簡略化した事例を図-3～図-8 に示す.

$(x-3)^2$の極小値を探索する場合,パラメータ μ が小さくなるにつれて,目的関数の最小値に近づく.

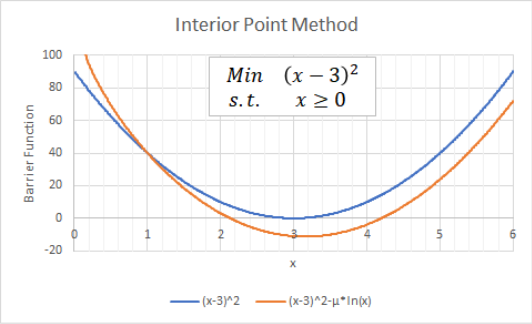


図-3 Min $(x-3)^2$: μ=10

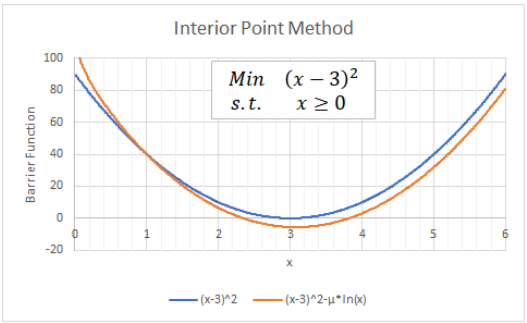


図-4 Min $(x-3)^2$: μ=5

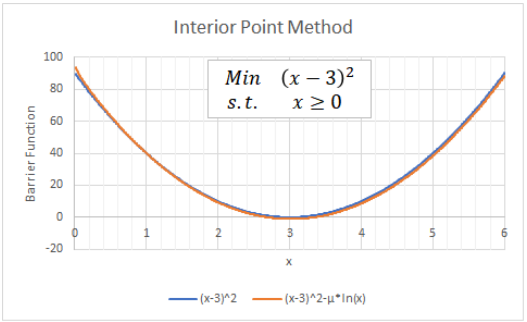


図-5 Min $(x-3)^2$: μ=1

$(x+1)^2$の極小値を探索する場合,パラメータ μ が小さくなるにつれて,目的関数の最小値に近づく.

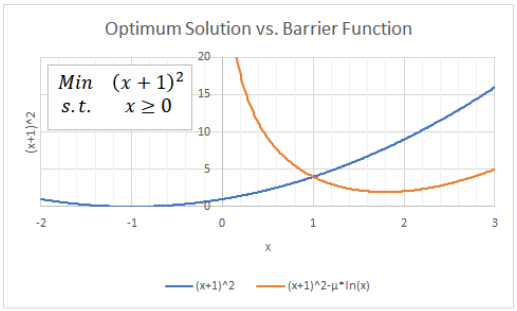


図-6 Min $(x+1)^2$: μ=10

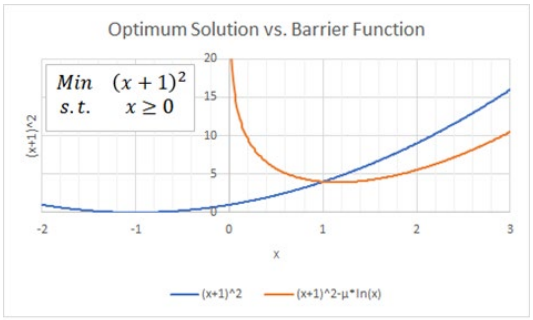


図-7 Min $(x+1)^2$: μ=5

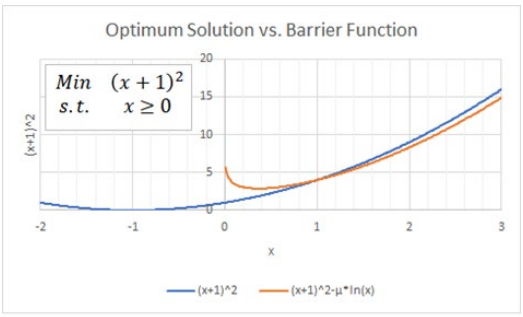


図-8 Min $(x+1)^2$: μ=1

(3) **ルンゲ・クッタ法** (Runge–Kutta method)

ルンゲ・クッタ法は初期値問題に対して常微分方程式に近似解を与える数値解法である．初期値y_0から任意の時刻t_nにおける近似値y_nを求めるにあたって,通常,下記手順にしたがって4次精度の近似解を計算する：

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (6)$$

$$t_{n+1} = t_n + h \quad (7)$$

但し,

$k_1 = f(t_n, y_n)$：区間の最初t_nにおける勾配

$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$：区間の中央$t_n + \frac{h}{2}$における勾配の近似値

$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$：区間の中央$t_n + \frac{h}{2}$における別の勾配の近似値

$k_4 = f(t_n + h, y_n + hk_3)$：区間の最初$t_n + h$における勾配の近似値

(4) Direct Multiple-Shooting Method

Direct Multiple Shooting 法は時刻方向に離散化して最適化する数値解法である．そして「ある時間幅の終端におけるシステムの状態と,次の時間幅の始端におけるシステムの状態が等しい」という制約を導入することで,次の時間幅との連続性を担保する．初期値x_0から任意の時刻t_fにおける近似値x_nを下記手順にしたがって最適解を計算する：

$$\dot{x} = f\big(x(t), u(t)\big) \tag{8}$$

$$\min_{x(t),u(t)} \int_{t_0}^{t_f} l\big(x(t), u(t)\big)dt + m\Big(x(t_f)\Big) \tag{9}$$

Subject to $\dot{x}(t) = f\big(x(t), u(t)\big)$

$g\big(x(t), u(t)\big) = 0$

$h\big(x(t), u(t)\big) > 0$

但し,

t_0, t_f：制御の開始時刻,終了時刻

$l\big(x(t), u(t)\big)$：t_0～t_fにおける評価関数

$m\Big(x(t_f)\Big)$：終了時刻における状態$x(t)$に対する評価関数

$g\big(x(t), u(t)\big), h\big(x(t), u(t)\big)$：システムの等式制約,不等式制約

3. CasADiに基づく計算手順

(1) 最適制御と状態空間方程式

動的なシステムとは, ある時刻のシステムの状態が,以前の時刻の状態に依存する動的なシステムである． このシステムの性能評価するには, 何らかの評価関数を設定する． その評価関数と目標値の差を最小化するように,システムの入力を決定する制御方法である．

第 2 章で記述した状態空間方程式を, ここで述べた「動的なシステム」に適用する．そして以下の手順により, 任意の時刻におけるシステム応答の近似解を得るとともに,内点法(ipopt)により最適解を求める：

```
dt = T/N # length of a control interval
for k in range(N): # loop over control intervals
    # Runge-Kutta 4 integration
    k1 = f(X[:,k],         U[:,k])
    k2 = f(X[:,k]+dt/2*k1, U[:,k])
    k3 = f(X[:,k]+dt/2*k2, U[:,k])
    k4 = f(X[:,k]+dt*k3,   U[:,k])
    x_next = X[:,k] + dt/6*(k1+2*k2+2*k3+k4)
    opti.subject_to(X[:,k+1]==x_next) # close the gaps

# ---- solve NLP              ------
opti.solver("ipopt") # set numerical backend
sol = opti.solve()   # actual solve
```

(2) 計算結果

前節で述べた計算手順にしたがって得られた計算結果を図-9～図-10 に示す.

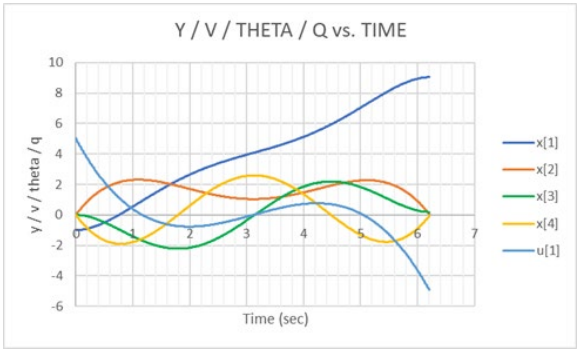


図-9 CasADi Output : Runge-Kutta Method

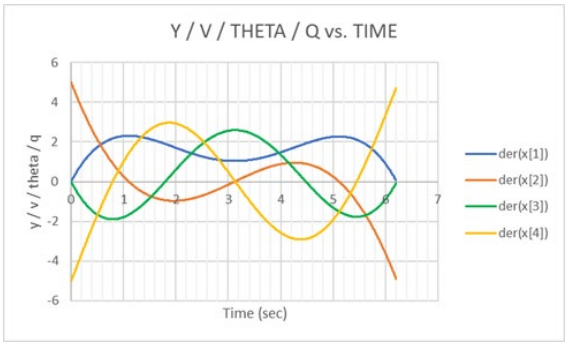


図-10 CasADi Output : Direct Multiple Shooting Method

u[1]：台車推力履歴

x[1]：台車位置履歴

x[2]=der(x[1])：台車速度履歴

x[3]：吊荷傾斜角履歴

x[4]=der(x[3])：吊荷傾斜角速度履歴

そして,CasADi から得られた台車推力履歴を Table 形式の入力データに整理した.

この入力データを図-11 に示す OpenModelica の Connection Editor で機構解析モデルに変換したところ,図-10 に示すような天井走行クレーンの出力が得られた.

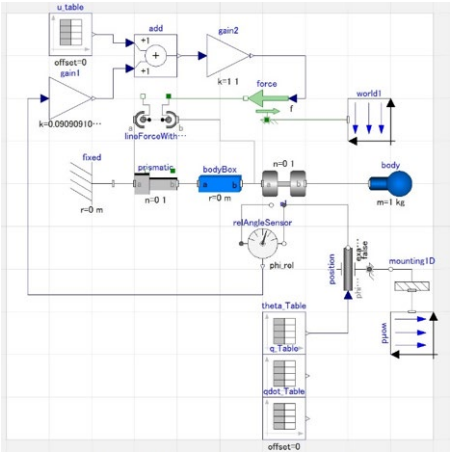


図-11 OpenModelica の ：Connection Editor

4. おわりに

(1) 天井走行クレーンの吊荷の揺れを最小化する制御システムを状態空間方程式で表現した.
(2) この吊荷の時刻歴応答を Runge-Kutta 法で計算した.
(3) さらにこの時刻歴応答を Direct Multiple-Shooting 法と内点法(IPOPT)で最適化した.
(4) 最後に CasADi から得られた台車推力履歴を使って,OpenModelica で機構解析を行った.
(5) 今回は CasADi で最適化した時刻歴応答を OpenModelica の Table 形式入力データに変換した.
(6) OpenModelicaから逐次CasADiを実行することを目指して,引き続き検討を続ける.

参考文献

[1] 吉田史郎,"OpenModelica を使った機構解析",1DCAE・MBD シンポジウム 2020, 2020,12, オンライン
[2] 吉田史郎," OpenModelica を使った機構解析 第 2 報",1DCAE・MBD シンポジウム 2021, 2021,12, オンライン
[3] Beal, L.D.R., Hill, D., Martin, R.A., and Hedengren, J. D., GEKKO Optimization Suite, Processes, Volume 6,Number 8, 201
[4] Open Source Modelica Consortium, "Modelica User's Guide" Available at <https://www.openmodelica.org/doc/OpenModelicaUserGuide/latest> [Accessed April 06 2020] (1986), pp. 65-69.
[5] Open Optimization in Engineering Center, "CasADi" Available at <http://labs.casadi.org/OCP> [Accessed 06 Sept 25 2021]