Bulletin of Mathematics
Vol. 04, No. 02 (2012), pp. 143–160.

AN OPTIMAL CONTROL SOLUTION USING MULTIPLE SHOOTING METHOD

Said Munzir, Vera Halfiani and Marwan

Abstract. This paper concerns with the solution of optimal control. Optimal control is an optimization model which generally consists of differential equations system, where the system is built on the state variables representing the system's condition over time and the control variables which is the decision variables to be determined, so the optimal value of the objective criterion is obtained. Optimal control problem mostly has nonlinear system that is difficult to solve analytically. Therefore, the numerical method could be an alternative solution that can be implemented to the problem. One of the numerical methods to solve optimal control problem is Multiple Shooting Method. This method is essentially used to solve the Boundary Value Problem (BVP) on the discussion of differential equations. This method will transform the optimal control problem into numerical formulation in Nonlinear Programming (NLP) form. Furthermore, the NLP can be solved by using the Lagrange Multiplier method, so that the nonlinear algebraic system will be obtained. The final solution of the problem is obtained by implementing the Newton Method to the nonlinear equations system. Afterwards, the second order sufficient condition is applied to guarantee that the final solution is the desired optimal solution.

1. INTRODUCTION

Optimal control is a part of dynamic optimization which aims to determine control variables so that an optimal value of the objective function is obtained while satisfying some constraints related to the problem system [1].

Received 22-05-2012, Accepted 13-07-2012.
2010 Mathematics Subject Classification: 49J20,
Key words and Phrases: Optimal control, shooting method, Lagrange Multiplier.

Optimal control problem is formulated in the form of differential equations. It consists of state variables which represent condition of the system over time and control variables which is the decision variables to be determined in order to obtain the optimal value of the objective function.

This paper explains the use of multiple shooting method to numerically solve an optimal control problem. The implementation of this method is supported by some other methods. Multiple shooting method is one of numerical method which was firstly found to solve optimal control problem. Essentially, this method is used to solve boundary value problem in differential equation scope. Therefore, this method is sufficient to be applied in optimal control problem which consists of differential equations system and often formulated in boundary value problem. By applying multiple shooting method, the system of differential equations will be transformed into numerical formulation.

Multiple shooting will transform an optimal control problem into a non-linear programming problem. The constraints which are differential equations will be turned into a system of ordinary non-linear algebraic equations. Subsequently, the non-linear equations constraints and the objective function which are now a problem of non-linear programming are combined using Lagrange Multiplier. Finally, a Lagrangian function that represent the original optimal control problem is obtained. This new function is an optimization problem without any constraints and its optimum values can be determined through finding its first derivative. The first derivative will be in the form of system of non-linear equations and the solution of this system are found using Newton method which is implemented in Matlab language programming.

Newton method will provide a stationary point which optimize the Lagrangian function. However, this point can be either the maxima, the minima, or the saddle point. To verify this point, second order sufficient condition is required. If the point do not satisfy the sufficient condition, then the Newton iteration is repeated to produce a new point. Otherwise, the point is the optimum solution to the initial optimal control problem.

2. THEORIES AND METHODS

2.1 Optimal Control

Formally, optimal control problem consists of a time variable t, state variables $\boldsymbol{x}(t)$, control varibles $\boldsymbol{u}(t)$, state equations $\dot{\boldsymbol{x}}(t)$, boundary conditions, and an objective function J. Time t is a continous unit and defined in a certain interval with initial time t_i and boundary time t_f. Generally, optimal control problem is written in to the following system.

$$\min_{u(t)} J = \int_{t_i}^{t_f} I(\boldsymbol{x}(t), \boldsymbol{u}(t), t)dt + F(x_1, t_1),$$

subjects to

$$\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \\
\boldsymbol{x}(t_i) &= \boldsymbol{x}_i, \\
(\boldsymbol{x}(t), t) &\in E^{n+1}, \quad when \quad t = t_f \\
\boldsymbol{u}(t) &\in E^r
\end{aligned}$$

The objective function of optimal control problem is a mapping from control functions to a real number value which will be maximized or minimized [2].

2.2 Multiple Shooting Method

Shooting method is a method to solve a boundary value problem in differential equations problem. To ilustrate the concept of this method, it is given the following equation.

$$\dot{x}(t) = x(t), t_i \leq t \leq t_f.$$

The analytical solution of above equation is

$$x(t) = x(t_i)e^{t-t_i},$$

with $e = 2.71$. Then $x(t_i) = x_i$ will be determined such that it will satisfy $x(t_f) = b$ for given value b. Therefore, equation $x(t_f) - b = 0$ or $x_i e^{t-t_i} - b = 0$ is obtained. This derivation is called as shooting method. Generally, the shooting method can be summarized as follow.

- Give an initial value $x_i = x(t_i)$.
- Form a solution of the differential equation from t_i to t_f.
- Evaluate an error of the boundary condition $x(t_f) - b$.
- Find the value of x_i that satisfies $x_i e^{t_f - t_i} - b = 0$.

In multiple shooting method, the "shoot" interval is partitioned into some short intervals. It is given the system of differential equations below.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t), \quad t_i < t < t_f. \tag{1}$$

The initial values $\boldsymbol{x_i} = \boldsymbol{x}(t_i)$ are then determined such that the boundary value

$$\varphi(\boldsymbol{x}(t_f), t_f) = \mathbf{0}$$

will give values that satisfy equation (1). The idea of multiple shooting method is splitting the time domain t into some shorter intervals. let $\boldsymbol{x}_k$ for $k = 0, 1, 2, ..., m-1$ be the initial value for the dynamic variables at starting point of every short interval segment $k+1$. At each segment $k+1$, the solution of the differential equations from t_k to t_{k+1} is established, and let the solution be $\widehat{\boldsymbol{x}}_k$. From every interval segment, it can be defined the set of non-linear programming variables as $(\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{m-1})$. To ensure that the solution function at each segment forms a single continuous function, the following constraint is formed [3].

$$[\boldsymbol{x}_1 - \widehat{\boldsymbol{x}}_0, \boldsymbol{x}_2 - \widehat{\boldsymbol{x}}_1, ..., \varphi(\boldsymbol{x}(t_m), t_m)] = \mathbf{0}.$$

2.3 Euler Method

To implement the multiple shooting method, a solution equation of the state equation is needed. The solution can be found either analytically or numerically. However, there is always a case in the initial value problem that the analytical solution can not be determined. Hence, Euler method can be used to form the approximate solution. Let

$$\dot{x}(t) = f(t, x) \qquad x(t_i) = x_i \qquad t_i \leq t \leq t_f$$

The interval $[t_i, t_f]$ is divided into m subintervals that are bounded by mesh point $t_k = t_i + k\Delta t$ for $k = 1, 2, \ldots, m$ where $\Delta t = (t_f - t_i)/m$ and then the approximate value $x_0, x_1, \ldots, x_m$ of $x(t_0), x(t_1), \ldots, x(t_m)$ is determined [4]. The parameter Δt is called step size [5].

The gradient at (t_0, x_0) is $f(t_0, x_0)$ where x_0 is the initial condition which is given. Hence, the tangent line which passes point (t_0, x_0) is $x - x_0 = s(t_0, x_0)(t - t_0)$ [6]. Generally, the equation used in Euler method can be written as follow.

$$t_{k+1} = t_k + \Delta t, \qquad x_{k+1} = x_k + \Delta t f(t_k, x_k), \qquad for \quad k = 0, 1, \ldots, m-1$$

.

2.4 Lagrange Multiplier

Lagrange multiplier method is a formulation to get a solution of optimization problem with constraints. Using this method, a non-linear optimization is transformed into a new function with new additional variables, this function is called as Lagrangian. Lagrangian is defined as the sum of an objective function and the linear combination of its constraints [7].

Let $\boldsymbol{z}^*$ is the local minimizer of a function f which has constraints $\boldsymbol{h(z) = 0}$ and assume that the gradients of the constraints which are $\nabla h_1(\boldsymbol{z}^*), \dots, \nabla h_m(\boldsymbol{z}^*)$ are linearly independent. There is a unique vector $\boldsymbol{\lambda}^* = [\lambda_1^*, \dots, \lambda_m^*]$ which is called Lagrange multiplier such that

$$\nabla f(\boldsymbol{z}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\boldsymbol{z}^*) = 0$$

[8].

The Lagrangian can be defined as

$$L(\boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{z}) + \boldsymbol{\lambda}^T \boldsymbol{h(z)}$$

with necessary condition

$$\nabla_{\boldsymbol{z}} L(\boldsymbol{z}, \boldsymbol{\lambda}) = \boldsymbol{0} \quad (2)$$

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{z}, \boldsymbol{\lambda}) = \boldsymbol{0} \quad (3)$$

[9].

2.4.1 First Order Necessary Condition

From the derivation of Lagrange multiplier method, it can be conclude the first order necessary condition as follow [9].

Theorem 1 *Let $\boldsymbol{z}^*$ be the local extreme value of a function f whose constraints $\boldsymbol{h(z) = 0}$ and assume that $\boldsymbol{z}^*$ is the regular point of the constraints, then there exists $\boldsymbol{\lambda} \in E^m$ such that*

$$\nabla f(\boldsymbol{z}^*) + \boldsymbol{\lambda}^T \nabla \boldsymbol{h(z^*) = 0}.$$

2.4.2 Second Order Condition

Equation (2) and (3) define the necessary condition of the Lagrange multiplier method where the solution of the equations can give the maximum value as well. The second order condition provides the following two things [11].

a. A condition which a point should satisfy in order to be local minimizer (necessary condition)

b. A condition which ensure that the point is the local minimizers (sufficient condition)

Second order conditions are stated as follow [9].

Theorem 2 (Second order necessary condition) *Let $\boldsymbol{z}^*$ be the local minimizer of a function f whose constraints $\boldsymbol{h(z)} = \mathbf{0}$ and $\boldsymbol{z}^*$ is a regular point of the constraints, then there exists $\boldsymbol{\lambda} \in E^m$ such that*

$$\nabla f(\boldsymbol{z}^*) + \boldsymbol{\lambda}^T \nabla \boldsymbol{h(z^*)} = \mathbf{0}.$$

If $M = \{\boldsymbol{v} | \nabla \boldsymbol{h(z^*)v} = \mathbf{0}\}$ is the tangent plane, then matrix

$$\nabla_{\boldsymbol{zz}} L(\boldsymbol{z}^*) = \nabla_{\boldsymbol{zz}} f(\boldsymbol{z}^*) + \boldsymbol{\lambda}^T \nabla_{\boldsymbol{zz}} (\boldsymbol{h(z^*)}$$

is positive semidefinite on M, which is $\boldsymbol{v}^T \nabla_{\boldsymbol{zz}} L(\boldsymbol{z}^*)\boldsymbol{v} \geq \mathbf{0}$ for every $\boldsymbol{v} \in M$.

Theorem 3 (Second order sufficient condition) *If there exists a point $\boldsymbol{z}^*$ that satisfies $\boldsymbol{h(z^*)} = \mathbf{0}$ and $\boldsymbol{\lambda} \in E^m$ such that*

$$\nabla f(\boldsymbol{z}^*) + \boldsymbol{\lambda}^T \nabla \boldsymbol{h(z^*)} = \mathbf{0},$$

and matrix

$$\nabla_{\boldsymbol{zz}} L(\boldsymbol{z}^*) = \nabla_{\boldsymbol{zz}} f(\boldsymbol{z}^*) + \boldsymbol{\lambda}^T \nabla_{\boldsymbol{zz}} (\boldsymbol{h(z^*)}$$

is positive definite on $M = \{\boldsymbol{v} | \nabla \boldsymbol{h(z^*)v} = \mathbf{0}\}$, which is that for every $\boldsymbol{v} \in M, \boldsymbol{v} \neq \mathbf{0}$, it applies $\boldsymbol{v}^T \nabla_{\boldsymbol{zz}} L(\boldsymbol{z}^*)\boldsymbol{v} > \mathbf{0}$, then $\boldsymbol{z}^*$ is the strict local minimizer of the function f with constraints $\boldsymbol{h(z^*)} = \mathbf{0}$.

The symbol $\nabla_{\boldsymbol{zz}} f(\boldsymbol{z})$ represents the Hessian matrix of the function f with respect to $\boldsymbol{z}$, that is a $m \times m$ matrix, $\nabla_{\boldsymbol{zz}} f(\boldsymbol{z}) = [a_{kl}] = \left[\dfrac{\partial^2 f(\boldsymbol{z})}{\partial z_k \partial z_l}\right]$ for $k, l = 1, 2, \ldots, m$.

2.4.3 Some Properties of Matrix

The determination of the properties of matrix $\nabla_{\boldsymbol{zz}}L(\boldsymbol{z})$ (positive/negative definite or semidefinite, or indefinite) in the Theorem 3 will determine the type of extreme point $\boldsymbol{z}^*$. Therefore, some explanation about this properties is necessary. Let A be a symmetrical matrix, then A is positive definite if and only if

$$\boldsymbol{w^T A w > 0}$$

for every non-zero vector $\boldsymbol{w}$. However, this property is not always easy to be determined. There is another simple equivalent property which involves the eigen values of matrix A. A matrix A may have the following properties [10].

a. Positive definite, if $\boldsymbol{w^T A w > 0}$ for every $\boldsymbol{w}$ or if all the eigen values of A are positive.

b. Positive semidefinite, if $\boldsymbol{w^T A w \geq 0}$ for every $\boldsymbol{w}$ or if all the eigen values of A are non-negative.

c. Negative definite, if $\boldsymbol{w^T A w < 0}$ for every $\boldsymbol{w}$ or if all the eigen values of A are negative.

d. Negative semidefinite, if $\boldsymbol{w^T A w \leq 0}$ for every $\boldsymbol{w}$ or if all the eigen values of A are non-positive.

e. Indefinite, if $\boldsymbol{w^T A w}$ is both positive and negative or if all eigen values of A are both positive and negative.

Another approach to determine these properties of a symmetrical matrix is to find the determinant of its leading principal minor [11].

Theorem 4 (Properties of a symmetrical matrix) *It is given that a symmetrical $m \times m$ matrix A, the following hold.*

a. If A is positive semidefinite or positive definite then $det(A) \geq 0$ or $det(A) > 0$ respectively.

b. A is positive definite if and only if all its leading principal minors are positive, that is $A_k > 0$ for $k = 1, 2, \dots, m$.

c. A is positive semidefinite if and only if all its principal minors are non-negative, that is $A_k^{(q)} \geq 0$ for all possible selection of $\{q_1, q_2, \dots, q_k\}$ for $k = 1, 2, \dots, m$.

d. A is negative definite if and only if all the leading principal minors of $-A$ are positive, that is $-A_k > 0$ for $k = 1, 2, \ldots, m$.

e. A is negative semidefinite if and only if all the principal minors of $-A$ are non-negative, that is $-A_k^{(q)} \geq 0$ for all possible selection of $\{q_1, q_2, \ldots, q_k\}$ for $k = 1, 2, \ldots, m$.

f. A is indefinite if (c) and (e) are not satisfied.

2.5 Newton Method

Newton method can be used to find the root of an equation. This method uses iteration and requires the guessed initial value for the roots. The formula used in this iteration requires the first derivative of the function of the equation,

$$\widehat{z} = z_0 - \frac{f(z_0)}{f'(z_0)} \tag{4}$$

where z_0 is the guessed initial value, and $\widehat{z}$ is the root of the function $f(z)$ [12].

In multi-variables equations system, the the formula used in Newton iteration is similar to the equation (4). For instance, it will be determined the $l-$vector $z = (z_1, \ldots, z_l)$ such that

$$\boldsymbol{f}(\boldsymbol{z}) = \begin{bmatrix} f_1(\boldsymbol{z}) \\ \vdots \\ f_m(\boldsymbol{z}) \end{bmatrix} = \mathbf{0}$$

It is assume that the number of functions is the same as the number of varables, $m = l$. Hence, the linear approach of newton method is

$$\boldsymbol{f}(\widehat{\boldsymbol{z}}) = \boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{G}(\widehat{\boldsymbol{z}} - \boldsymbol{z})$$

where $\boldsymbol{G}$ is Jacobian matrix which is defined as

$$G = \nabla_{\boldsymbol{z}} \boldsymbol{f} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \cdots & \frac{\partial f_1}{\partial z_l} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \cdots & \frac{\partial f_2}{\partial z_l} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial z_1} & \frac{\partial f_m}{\partial z_2} & \cdots & \frac{\partial f_m}{\partial z_l} \end{bmatrix}$$

Applying $\boldsymbol{f}(\widehat{\boldsymbol{z}}) = \mathbf{0}$ gives the following linear system

$$G\boldsymbol{p} = -\boldsymbol{f}(\boldsymbol{z}) \tag{5}$$

After finding $\boldsymbol{p}$ from the equation (5), the solution $\widehat{\boldsymbol{z}}$ is determined by looping the following equation

$$\widehat{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{p}$$

To solve the equation (5), it is important to make sure that the Jacobian G is a non-singular matrix. This is the same condition to the problem a function with a single variable in equation (4) where $f'(z) \neq 0$ [3].

2.6 MATLAB

MATLAB (stands for MATrix LABoratory) is a computer software which is developed by Math Works Inc. This software has beed widely used in many fields such as sciences and engineering. MATLAB is a programming language with interactive environment to develop algorithm, data visualization, data analysis, and numerical calculation. MATLAB can solve the technical calculation problem faster compared to other traditional language programming software such as C, C++, and Fortran [13].

With its optimal calculation in processing matrices and vectors, MATLAB can provide intuitive language to express problems and their solutions mathematically or visually. MATLAB is used for various purposes, some of them are stated as follow [14].

a. Developing algorithm ans numerical calculation.

b. Symbolic calculation.

c. Modelling, Simulations, and prototype development.

d. Data analysis and image processing.

e. Science and engineering visualization.

3. RESULT

3.1 Numerical Solution of Optimal Control Problem

It is given the following optimal control problem.

$$\min J = \int_{t_i}^{t_f} I(x(t), u(t), t)dt, \tag{6}$$

subjects to

$$\begin{align} \dot{x}(t) &= f(x(t), u(t), t), \tag{7}\\ x(t_f) &= x_f, \tag{8}\\ (x(t), t) &\in R^2, \tag{9}\\ u(t) &\in R \tag{10}\end{align}$$

The state equation (7) is an ordinary differential equation with boundary condition (8). By implementing multiple shooting method, the domain interval t is partitioned into some subintervals. Let

$$t_i = t_0 < t_1 < \ldots < t_m = t_f.$$

then, the solution of state equation (7) is determined at every subinterval $[t_k, t_{k+1}$ for $k = 0, 1, 2, \ldots, m-1$.

For $k = 0, 1, 2, \ldots, m-1$, let $x(t_k) = x_k$ be the given initial value for the state equation's solution and $u(t_k) = u_k$ at every interval $[t_k, t_{k+1}]$.By using Euler method, the value $x(t_{k+1})$ is obtained as the following formulation.

$$x(t_{k+1)} = x_k + \Delta t_{k+1} f(x_k, u_k, t_k)$$

where $\Delta t_{k+1} = t_{k+1} - t_k$.

The above equation is the solution of the state equation whose value will be found at every subinterval simultaneously. However, to make the final solution function continuous at interval $[t_i, t_f]$, it is defined that the initial value x_{k+1} at every subinterval is the same as the boundary value $x(t_{k+1})$ at the preceding subinterval.

$$\begin{align} x(t_{k+1}) &= x_{k+1}\\ x(t_{k+1}) - x_{k+1} &= 0\\ x_k + \Delta t_{k+1} f(x_k, u_k, t_k) - x_{k+1} &= 0\end{align}$$

Hence, a new optimization system replacing the optimal control problem (6)

- (10) is obtained.

$$\min J = \sum_{k=0}^{m-1} I(x_k, u_k, t_k)\Delta t_{k+1}$$

subject to

$$\begin{aligned} x_0 + \Delta t_1 f(x_0, u_0, t_0) - x_1 &= 0 \\ x_1 + \Delta t_2 f(x_1, u_1, t_1) - x_2 &= 0 \\ &\vdots \\ x_{m-2} + \Delta t_{m-1} f(x_{m-2}, u_{m-2}, t_{m-2}) - x_{m-1} &= 0 \\ x_{m-1} + \Delta t_m f(x_{m-1}, u_{m-1}, t_{m-1}) - x_f &= 0 \end{aligned}$$

The above optimization system is now a constrained non-linear programming problem which can be solve by applying Lagrange multiplier method.

Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ be a vector of the Lagrange multiplier, then the previous constrained non-linear programming system can be reformulated as follow.

$$L = \sum_{k=0}^{m-1} I(x_k, u_k, t_k) + \boldsymbol{\lambda}^T \begin{bmatrix} x_0 + \Delta t_1 f(x_0, u_0, t_0) - x_1 \\ x_1 + \Delta t_2 f(x_1, u_1, t_1) - x_2 \\ \vdots \\ x_{m-1} + \Delta t_m f(x_{m-1}, u_{m-1}, t_{m-1}) - x_f \end{bmatrix}$$

and the first order condition are

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = \mathbf{0} \tag{11}$$

$$\nabla_{\boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = \mathbf{0} \tag{12}$$

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = \mathbf{0} \tag{13}$$

where $\boldsymbol{x} = (x_0, x_1, \dots, x_m)$ and $\boldsymbol{u} = (u_0, u_1, \dots, u_{m-1})$.

The first order conditions (11) - (13) give a system of non-linear algebraic equations. The final solution of the this problem which is originally the optimal control can be obtained by solving the system of equations (11) - (13). Finally, the Newton method is applied to solve this system of non-linear algebraic equations.

To see how the Newton method which is implemented in a code program works to this problem, it is given the example below.

$$\min J = \int_0^1 x^2 + u^2 dt, \tag{14}$$

subjects to

$$\dot{x}(t) = (x - tu)^2 \tag{15}$$
$$x(1) = 2, (x(t), t) \in R^2 \tag{16}$$
$$u(t) \in R \tag{17}$$

Here, the domain interval t is divided into four partitions by step size $\Delta t = 0.25$. Hence, the optimal control problem (14) - (17) can be transformed into the system below.

$$\min J = \sum_{k=0}^{3} (x_k^2 + u_k^2) 0.25 \tag{18}$$

subject to

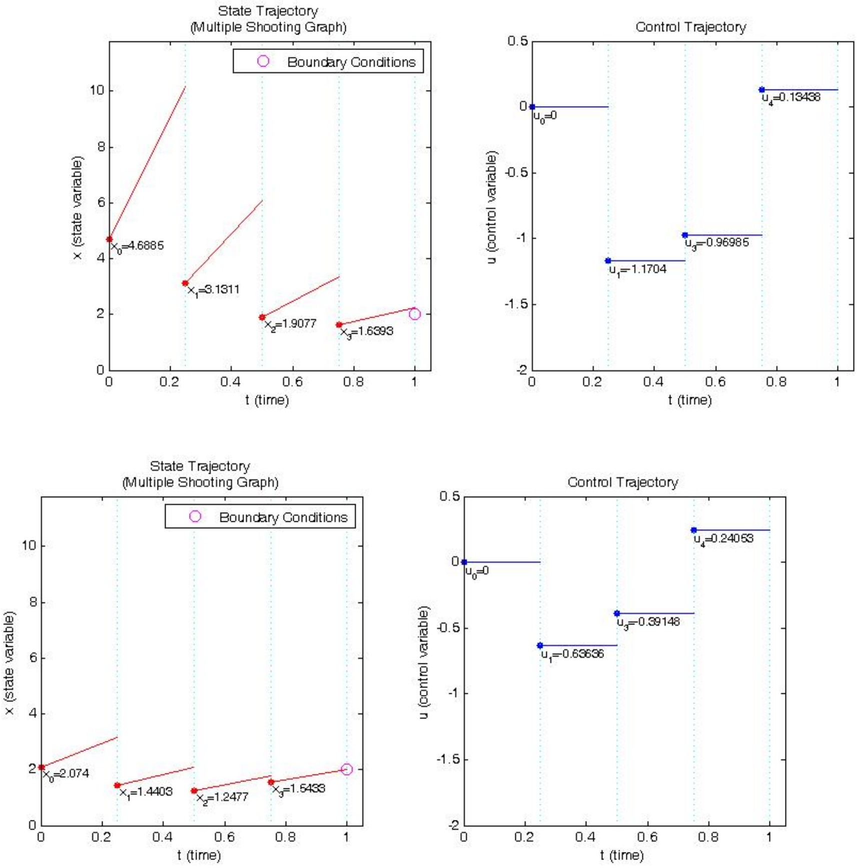
$$x_0 + 0.25x_0^2 - x_1 = 0 \tag{19}$$
$$x_1 + 0.25(x_1 - 0.25u_1)^2 - x_2 = 0 \tag{20}$$
$$x_2 + 0.25(x_2 - 0.5u_2)^2 - x_3 = 0 \tag{21}$$
$$x_3 + 0.25(x_3 - 0.75u_3)^2 - 2 = 0 \tag{22}$$

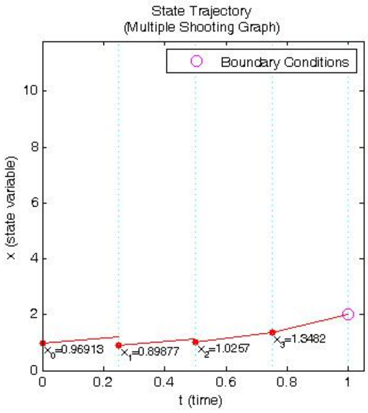
By setting initial value 10 to every variable $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}$ in Newton iteration implementation, The solutions are found at sixth iteration. The solution list below is produced by the code program implemented using Matlab.

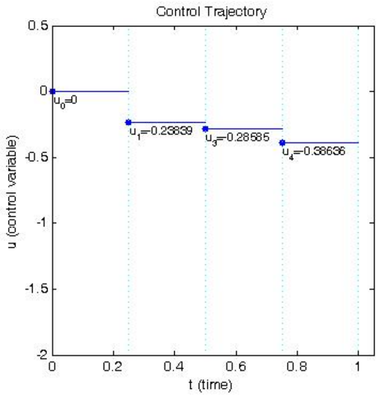
$$\begin{aligned} x_0 &= 0.64720, & u_0 &= 0.00000, & \lambda_1 &= -0.24448 \\ x_1 &= 0.75191, & u_1 &= -0.08652, & \lambda_2 &= -0.044740 \\ x_2 &= 0.90150, & u_2 &= -0.31050, & \lambda_3 &= -0.58765 \\ x_3 &= 1.18069, & u_3 &= -0.83951, & \lambda_4 &= -0.61832 \end{aligned}$$

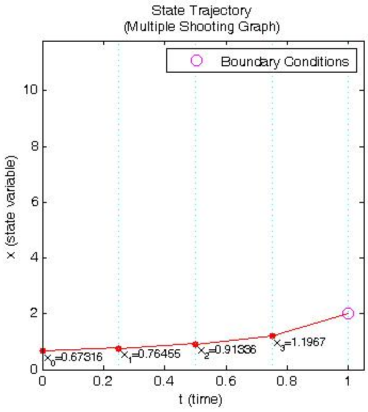
with objective function value $J = \sum_{k=0}^{3} (x_k^2 + u_k^2) 0.25 = 0.99991$.

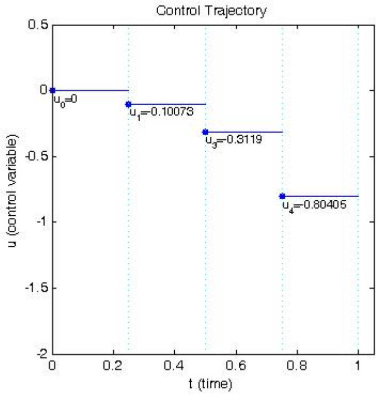
The second order condition test to the solution is done by using both the eigen values and leading principal minor of the Hessian matrix derived from the Lagrangian of the optimization system (18) - (22) that is bounded by the null space of the first derivative of constraints (19) - (22). The result shows that the eigen values $\{0.4308, 0.6716, 0.4944, 0.5000\}$ and the leading principal minor $A_1 = 0.50000, A_2 = 0.24619, A_3 = 0.11450, A_4 = 0.05023$ are all positive meaning that the solution is the strict local minimizer of the system (18) - (22). The Figure 1 shows the change of the state variable and control variable values at each iteration in Newton method and the final iteration shows that the state variable is eventually a continuous function in the given interval.

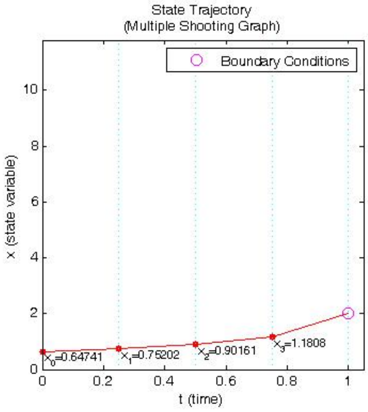


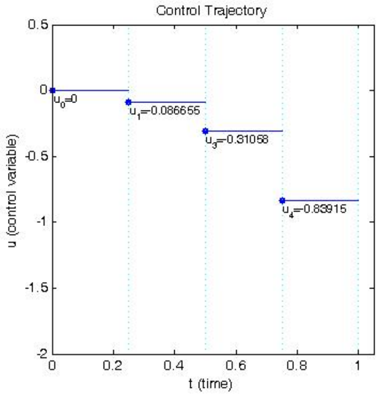

State Trajectory
(Multiple Shooting Graph)
Boundary Conditions
x (state variable)
t (time)
0
2
4
6
8
10
0.2
0.4
0.6
0.8
1
X_0=0.96913
X_1=0.89877
X_2=1.0257
X_3=1.3482


Control Trajectory
u (control variable)
t (time)
0.5
0
-0.5
-1
-1.5
-2
0.2
0.4
0.6
0.8
1
u_0=0
u_1=-0.23839
u_3=-0.28585
u_4=-0.38636


State Trajectory
(Multiple Shooting Graph)
Boundary Conditions
x (state variable)
t (time)
0
2
4
6
8
10
0.2
0.4
0.6
0.8
1
X_0=0.67316
X_1=0.76455
X_2=0.91336
X_3=1.1967


Control Trajectory
u (control variable)
t (time)
0.5
0
-0.5
-1
-1.5
-2
0.2
0.4
0.6
0.8
1
u_0=0
u_1=-0.10073
u_3=-0.3119
u_4=-0.80405


State Trajectory
(Multiple Shooting Graph)
Boundary Conditions
x (state variable)
t (time)
0
2
4
6
8
10
0.2
0.4
0.6
0.8
1
X_0=0.64741
X_1=0.75202
X_2=0.90161
X_3=1.1808


Control Trajectory
u (control variable)
t (time)
0.5
0
-0.5
-1
-1.5
-2
0.2
0.4
0.6
0.8
1
u_0=0
u_1=-0.086655
u_3=-0.31058
u_4=-0.83915

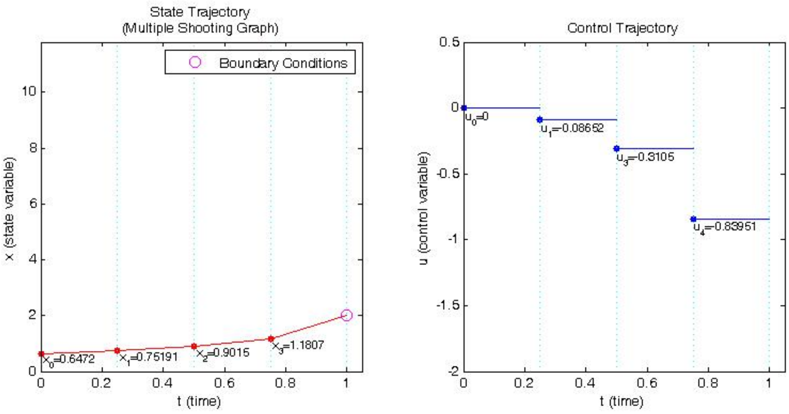


Figure 1: State Trajectory and Control Trajectory of Optimal Control Problem (14) - (17) at Iteration 1 - 6

3.2 Linear Optimal Control Problem and The guessed Initial Value for Newton Iteration

To solve the constrained non-linear programming formulation by using Lagrange multiplier, the objective function and the constraints have to be twice differentiable. In other words, if the system is linear on x and u then matrix $\nabla_{\boldsymbol{zz}} L(\boldsymbol{z}) = \mathbf{0}$ (where $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$) on all the domain points causing the local minimizer can not be determined. If the first order conditions (11) - (13) are satisfied but the Hessian matrix of the Lagrangian which is the second order condition is zero (or the determinant is zero), then the point that satisfies this condition is called singular point. Moreover, the final solution using Newton method requires the first derivative (the Jacobian matrix) of the system (11) - (13). This derivative is equivalent to the second derivative (the Hessian matrix) of the Lagrangian L with respect to $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}$. This matrix has to be invertible or a non-singular matrix so that the Newton iteration can work. This condition constrains that the Lagrangian has to be twice differentiable.

Another concern to this process is in giving the initial value to run the Newton iteration. The different initial values may cause different solutions or make an infinite iteration. The iteration results which are obtained by giving initial values $-5, -8, -9, -10$ are different each others and do not satisfy the second order condition such that the results are not the desired minimizer. Meanwhile, initial values -2 and -4 cause the iteration never ends and does not give any solution point. The graphs of the state variable for various guessed initial value are represented in Figure 2. The red line represents the state variable which is obtained from the final solution by giving the initial value 10 in the Newton iteration, while the yellow, blue, green, and black line (which is not the solution) represents the state variable which is obtained by setting the initial values $-5, -8, -9, -10$ respectively.

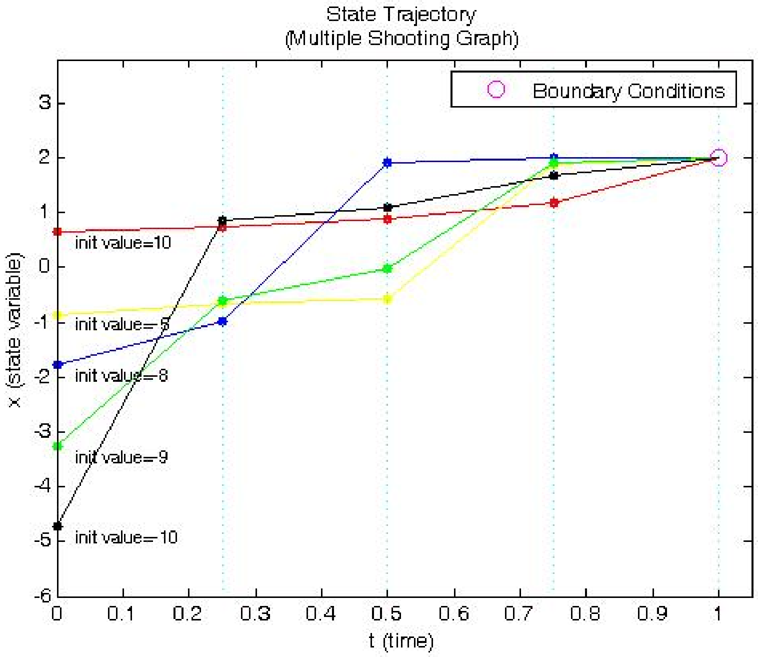


Figure 2: State Trajectory of Optimal Control Problem (14) - (17) obtained from some different initial values setting

4. CONCLUSION AND FUTURE RECOMMENDATION

4.1 Conclusion

a. By using multiple shooting method, an optimal control problem can be transformed into a numerical formulation in constrained non-linear programming form which can be solved by using Lagrange multiplier method.

b. The Newton method can be used to find the stationary points of non-linear equations which is obtained by deriving the Lagrangian function.

c. The solution obtained from the Newton method can be either maximum, minimum, or saddle point. Therefore, the second order condition is required to confirm that the solution point is the appropriate extreme point to the original problem.

4.2 Future Recommendation

a. In this study, the optimal control problem that is implemented consists of one state and control variable. The future study may consider the problem with multi variables.

b. The step size of the domain interval can be made smaller so that the more accurate solution can be obtained.

c. The code program in this study is only the implementation of Newton method. The future study may consider to code the whole methods used.

REFERENCES

1. Kirk, Donald E, *Optimal Control Theory, An Introduction*, Dover Publications Inc., New York, 1970.
2. Intriligator, Michael D, *Mathematical Optimization and Economics Theory*, Society for Industrial and Applied Mathematics, Philadelphia, 2001.
3. Betts, John T., *Practical Methods for Optimal Control Using Non-linear Programming*, Society for Industrial and Applied Mathematics, Philadelphia, 2001.
4. Dahlquist, Germund, and Åke Björck, Ned Anderson, *Numerical Methods*, Dover Publication, Inc., New York, 1974.
5. Mathews, John H. dan Kurtis D. Fink, *Numerical Methods Using MATLAB, Third Edition*, Prentice Hall. USA, 1999.
6. Buchanan, James L. and Peter R. Turner, *Numerical Method and Analysis*, McGraw-Hill,Inc.,Singapore, 1992.
7. Venkataraman, P, *Applied Optimization with MATLAB Programming*, A Wiley-Interscience Publication, New York, 2002.
8. Bertsekas, Dimitri p., *Non-linear Programming, Second Edition*, Athena Scientific, Massachusetts, 1999.

9. Luenberger, David G, *Linear and Nonlinear Programming*, Springer, New York, 2008.
10. Griva, Igor, dan Stephen G. Nash, Ariela Sofer, *Linear and Nonlinear Optimization, Second Edition*, Society for Industrial and Applied Mathematics. USA, 2009.
11. Antoniou, Andreas and Wu-Sheng Lu, *Practical Optimization; Algorithms and Engineering Applications*, Springer, New York, 2007.
12. Daniels, Richard W, *An Introduction to Numerical Methods and Optimization Techniques*, Elsevier North-Holland. New York, 1978.
13. The MathWorks™, *MATLAB® Getting Started Guide*, The MathWorks, Inc., United States, 2010.
14. Dukkipati, Rao V, *MATLAB An Introduction With Applications*, New Age International (P) Ltd., Publishers, New Delhi, 2010.

Said Munzir: Jurusan Matematika, FMIPA, Universitas Syiah Kuala, Jl. Tgk. Daud Beureuh No. 4 Darussalam-Banda Aceh 23111, Indonesia.
E-mail: marwan301@yahoo.com

Vera Halfiani: Jurusan Matematika, FMIPA, Universitas Syiah Kuala, Jl. Tgk. Daud Beureuh No. 4 Darussalam-Banda Aceh 23111,Indonesia.

Marwan: Jurusan Matematika, FMIPA, Universitas Syiah Kuala, Jl. Tgk. Daud Beureuh No. 4 Darussalam-Banda Aceh 23111, Indonesia.