Design of the Optimal Thruster Combinations Table for the Real Time Control Allocation of Spacecraft Thrusters

Min Wang, Yongchun Xie

Abstract—The purpose of onboard thruster control allocation is to select specific thrusters and calculate their firing durations in order to realize force and torque commands derived from the control system of a spacecraft. This study is conducted to design an optimal thruster combinations table for the use of onboard real time control allocation problems. The algebraic method for generating the optimal thruster combinations table (OTCT) with respect to thruster configuration is explained in detail based on linear programming (LP). With the table pre-stored in the onboard computer, a simple but optimal algorithm to solve the thruster control allocation problem in real time is implemented. The proposed new algorithm, which has the same optimal solutions as LP algorithm, is much faster than LP in real time control allocation. Its effectiveness is demonstrated in the simulation.

I. INTRODUCTION

THE reaction control system, which uses reaction thrusters as actuators, is one of the most important subsystems in spacecraft control system for its adaptability to a variety of space missions. Since thrusters are fixed to the vehicle frame, the deliverable force/torque produced by a single thruster is definite. However, force and torque commands derived from the control law can be realized through the combination of thrusters distributed on the spacecraft with different locations and orientations. The main task of thruster control allocation is to select specific thrusters and calculate their firing durations in order to realize force and torque commands derived from the control system of a spacecraft. It has a direct effect on the control accuracy and fuel consumption of the complicated space missions [1].

So far, there are three main approaches for the thruster control allocation problem: the decoupling method, the linear programming (LP) method and the optimal catalogue method. The decoupling method divides thrusters into combinations according to predefined maneuvers and control modes of a spacecraft. Control systems of many spacecraft are based on this method for its simplicity in onboard computations. However, this method is only applicable for regular thruster configurations (axis-symmetric without coupling effects [2]), which results in a limitation on the thruster configuration design. Besides, it can not minimize propellant consumption, which is important for extending the mission-life time. Thereupon, the LP method was proposed based on the simplex method [2], [5]-[7]. Although it can find the optimum solution to the thruster control allocation problem, the processing power needed by this algorithm was too high for the on board processing budget and is thus not often considered for real time application in space. Recently, some new algorithm related to spacecraft thruster control allocation problems have been proposed [3]-[6]. The concept of optimal thruster combinations was introduced. However, the method of generating the optimal thruster combinations, which is crucial, was not given.

This paper mainly concentrates on the design of the optimal thruster combinations table with respect to thruster configuration, as presented in section II. An algebraic method of generating all the optimal thruster combinations is detailed and its geometrical interpretation is given as well. Based on the optimal thruster combinations table, an effective real time control allocation algorithm for spacecraft thrusters is described in section III. Related simulation results were implemented in section IV, to allow a comparison of the performance of different control allocation algorithms. Section V summarizes the results and performance achieved.

II. DESIGN OF THE OPTIMAL THRUSTER COMBINATIONS TABLE

A. General Statement of the Problem

Generally, the thruster control allocation problem is formulated as follows: $At = u$ (1)

where $t = [t_1 \cdots t_n]^T \in R^n$ represents the firing duration for each thruster ($n$ is the total number of thrusters). It is the sought-for variable; $u = [u_1 \cdots u_m]^T \in R^m$ is the vector of input control command derived from the controller and $m$ is the dimension of vector $u$; $A$ is the $m \times n$ thruster configuration matrix. Its columns represent deliverable force and torque produced by each thruster.

Obviously, if $A$ is a nonsingular square matrix ($m=n$), this problem will have a unique solution $t = A^{-1}u$. However, the reaction control system of a spacecraft usually contains a large number of thrusters (about 30), which makes $n \gg m$. Consequently, there exist a large number of feasible solutions.
for the given problem. Among them, we hope to find a solution that can minimize fuel consumption or some other cost function. Therefore, the problem can be formulated to a linear programming model as follow:

\[
\begin{align*}
\min & \quad \sum_{j=1}^{m} c_j t_j \\
\text{s.t.} & \quad At = u \\
& \quad t \geq 0
\end{align*}
\]  

(2)

where \( c_j > 0 \) is the efficiency factor of the \( j \)-th thruster. The constraint equation \( At = u \) here indicates the purpose of implementing the required control command \( u \). And \( t \geq 0 \) is the nonnegative constraint, which means that all thrusters are unidirectional actuators.

Such problem is generally solved by means of the simplex method [8]. For a better understanding of this paper, some important definitions in the simplex method, which will be frequently used in our new control allocation algorithm, should be explained at first:

1) nonbasic variables: The \( n-m \) variables which are set equal to zero in a set of \( m \times n \) equations (\( m < n \)).
2) basic variables: The remaining \( m \) variables, whose solution (referred to as basic solution) is obtained by solving the \( m \) equations.
3) \( z \)-row: A row in the simplex tableau [8] that contains coefficients of every variable in the objective function.
4) entering variable: The nonbasic variable having the most negative coefficient in the \( z \)-row of the simplex tableau.
5) leaving variable: The basic variable that will first reduce to zero with the growing of the entering variable in the next iteration step of the simplex method.

According to this method, there are \( n-m \) nonbasic variables and \( m \) basic variables in every basic feasible solution (BFS). In each iteration step, the simplex method selects an entering variable using the optimality condition [8] and then selects a leaving variable using the feasibility condition [8]. Movement from the present BFS to a better adjacent BFS and finally to the optimal solution is implemented by applying appropriate Gauss-Jordan computations. In the optimal solution, we denote the \( m \) nonzero basic variables the optimal basis.

That is to say, given a definite control command \( u \), if we can find its corresponding optimal basis in advance, the optimal solution can be obtained directly by the formulation below without any iteration steps.

\[
t = B^{-1} u
\]  

(3)

where \( B \) is a \( m \times m \) matrix consists of columns in matrix \( A \) associated with the optimal basis. This optimal basis is actually the optimal thruster combination with respect to \( u \). Hence, the method of generating the optimal thruster combinations is conceived. The baseline is to precompute all the optimal basis of matrix \( A \) with respect to all the possible vector of \( u \) in an offline mode. Here comes a question: how to test every possible \( u \) in order to find out all the optimal bases. This will be answered in the algebraic method of generating the optimal thruster combinations table.

### B. Algebraic Method of Generating the Optimal Thruster Combinations Table

There are two differences between the standard LP problem modeled as (2) and the problem we need to solve: 1) the uncertainty of the control command \( u \); 2) the solutions we seek for are not the values of \( t \) (the firing durations), but the optimal basis, i.e. thrusters with nonzero values in an optimal solution.

According to the simplex method, the uncertainty of \( u \) will affect the optimal value of \( t \). Since we only care about which variables are the basic variables in the optimal solution instead of their values, the uncertainty of \( u \) in this point may not be considered. Besides, the uncertainty of \( u \) also affects the determination of leaving variables. Since the nonnegative ratios of the right-hand side of the equations to the corresponding constraint coefficients under the entering variable in a simplex tableau cannot be computed without a definite value of \( u \), there may exist more than one possible leaving variable. However, it is this uncertainty that makes it possible to obtain all the optimal bases of matrix \( A \).

The algebraic algorithm for generating the optimal thruster combinations is summarized as follows:

**Step 1 Initialization**

Add artificial variables \( s = [s_1, \ldots, s_m] \) to the equations in problem (2) and penalize them in the objective function with \( M (M=\text{constant}) \) and \( M \) must be sufficiently large relative to the original objective coefficients). The resulting new model is given as

\[
\begin{align*}
\max & \quad -Z \\
\text{s.t.} & \quad -Z + \bar{c} \bar{t} = 0 \\
& \quad \bar{A} \bar{t} = u \\
& \quad \bar{t} \geq 0
\end{align*}
\]  

(4)

where \( \bar{t} = [t \ s]^T \), \( \bar{c} = [c_0 \ 0_{1,m}] \),

\[
\bar{A} = [A \ I_{m,n}], \quad c_{0j} = c_j - M \sum_{i=1}^{m} a_{ij}
\]

where \( a_{ij} \) is the element of row \( i \) and column \( j \) of matrix \( A \) \((i=1, \ldots, m; j=1, \ldots, n)\). Select the artificial variables as the initial basic variables \((t_b = s)\) and the original decision variable \( t \) as the initial nonbasic variable. The basic matrix \( B = I_{m,n} \).

**Step 2 Iteration**

- Determine an entering variable: Select a nonbasic variable that has the most negative coefficient in the \( z \)-row;
- Determine leaving variables: Select all the basic variables associated with the positive constraint coefficients under the entering variable in the simplex tableau as possible leaving variables.
- Swap the entering variable with every possible
leaving variable respectively by the Gauss-Jordan row operations to produce a set of new basic feasible solutions $t_{ik}$ and the corresponding basic matrix $B_k$. As a result, the iteration process to seek for the optimal solution will branch. And the new objective coefficients and constraint coefficients associated with every new BFS can be obtained by

$$\bar{c} = \begin{bmatrix} c_B^{-1} A^T + c_o \end{bmatrix} \begin{bmatrix} c_B^{-1} A^T \end{bmatrix}$$

$$\bar{A} = \begin{bmatrix} B^{-1} A & B^{-1} \end{bmatrix}$$

where $c_B$ is the objective coefficients vector associated with the basic variables in new BFS.

**Step 3 Check the Optimality Condition**

If the z-row coefficients of the nonbasic variables are nonnegative, stop the iteration computation of the current branch, and the corresponding $t_{ik}$ is one of the optimal bases. Namely, thrusters associated with this $t_{ik}$ constitute an optimal thruster combination. Otherwise, go to step 2 to start a new iteration.

It must be noted that the right-hand side of the constraint equations is required to be nonnegative in a standard LP model. If the control command vector $u$ has negative elements, the constraint coefficients (elements of matrix $A$) will be standardized by multiplying -1 on both sides of the constraint equations. As a result, the objective coefficients vector $\bar{c}$ will be changed. That means all the standardized form of matrix $A$ should be found out before we start the algorithm proposed above. Different standardized forms of $A$ with respect to vector $u$ in different quadrant of the command space ensure that all the optimal thruster combinations could be obtained. Given the task dimension $m$, there are $2^m$ different standardized forms of $A$. Since $m$ is generally no more than 6 (considering translation and rotation) and the optimal thruster combinations are computed offline, the algorithm proposed above is feasible.

**C. Geometrical Interpretation of the Optimal Thruster Combinations**

Besides the cost function, all the information we need to know in order to generating the optimal thruster combinations table is the configuration matrix $A$. A 2-dimension example presented below will give some geometrical interpretation of the optimal thruster combinations, which can bring us a deeper understanding of the algebraic method proposed above.

Supposing that $u = \begin{bmatrix} u_x & u_y \end{bmatrix}^T$, and the configuration matrix $A$ with 4 thrusters is given as follow:

$$A = \begin{bmatrix} 200 & -200 & 60 & -60 \\ -100 & 100 & -150 & 150 \end{bmatrix}$$

Thrust vectors of the 4 thrusters $T_1$-$T_4$ in the 2-D command space is illustrated in Fig. 1. Suppose that the 4 thrusters have the same efficiency, which means the cost efficiency vector $c = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$.

The process of seeking for all the optimal thruster combinations according to the algorithm described in section B can be summarized to a branching tree displayed in Fig. 2. The BFS in every iteration step is represented by a bracket with thrusters number 1 to 4. And the initial basic variables are two artificial variables number 5 and 6, which are not real thrusters. According to different control command $u$, the algorithm branches to different node (thruster combination) as a result of selecting different entering and leaving variables at each stage based on the minimum nonnegative ratio test [8]. At the terminations of each branch, an optimal thruster combination is obtained. All the optimal thruster combinations in Fig. 2 can be summarized to 4 combinations below:

$$\begin{align*}
(T_2 & ~ T_3) \quad (T_1 & ~ T_3) \quad (T_1 & ~ T_4) \quad (T_2 & ~ T_4)
\end{align*}$$

Observe the above combinations in Fig. 1 we find that every two adjacent thrust vector form an optimal thruster combination, and these 4 combinations divide the whole command space into 4 independent area. If the control command $u$ is in a certain area, the optimal thruster combination w.r.t. $u$ is just the thrusters that formed this region. For example, if the vector $u$ is between the vector $T_1$ and $T_3$ as displayed in Fig. 1, its slope should between the slope of $T_1$ and $T_3$ as well, i.e.

$$\frac{60}{150} < \frac{u_x}{u_y} < \frac{200}{100} \quad (u_x > 0, ~ u_y < 0)$$

Search for the optimal combinations along the branch in Fig. 2 according to the conditions above, we can easily find out the optimal thruster combination with respect to vector $u$ is actually $(T_1 ~ T_3)$. This is the geometrical nature of the algebraic algorithm for generating the optimal thruster combinations by precomputing all the optimal bases. It also indicates the geometrical nature of the simplex method.

**III. Onboard Real Time Optimal Control Allocation Algorithm for Spacecraft Thrusters**

Given a configuration matrix $A$, we can precompute all its optimal thruster combinations corresponding to the objective function by using the algebraic method proposed in section II. With these data, the onboard real time control allocation computation can be greatly simplified. Data that need to be
pre-stored in the onboard computer of the spacecraft are the optimal thruster combinations table and the inverse matrix table, which stores the inverse matrices that consists of columns in matrix $A$ with respect to each optimal combination, i.e. $B^{-1}_k$ in (3). With these tables, an effective real time control allocation algorithm can be arranged as follows:

- Compute the firing duration of each thruster in every optimal combination by (3), according to the control command $u$ derived from the controller;
- Select the optimal combination with nonnegative firing durations;
- Calculate the on and off time of each thruster in the selected combination.

It should be noted that if the optimal combination selected is not the correct one with respect to current control command $u$, the firing durations of thrusters will be infeasible (having negative elements). If a solution is feasible, it is also optimal. This is guaranteed by the feasible condition in simplex method [8]. We can also check this from Fig. 1. If other optimal combinations are selected other than $(T_i, T_j)$, at least one thrust vector in the combination will be required to have a negative direction in order to generate the command $u$ as shown in Fig. 1.

Since this real time control allocation algorithm only involves the multiplication between matrices and vectors, its computation speed is much faster than the simplex iterative algorithm. And it proves to be compatible with the processing power of on board computer.

IV. Simulations

Simulations are executed to test the performance of the control allocation algorithm based on the optimal thruster combinations table. The decoupling method, LP and optimal thruster combinations table method (OTCT) are applied to the final approaching phase of rendezvous and docking respectively.

A. Initial Conditions

The dimension of the control command $m$ is 6, considering translation and rotation in the body frame. The control cycle is limited to 1 second. The propulsion system includes 18 thrusters numbered 1 to 18, as listed in Table I which forms the $6 \times 18$ configuration matrix $A$ in model (2). All thrusters are supposed to have the same cost efficiency.

Using the algorithm proposed in section II, the 18 thrusters can be divided into 120 optimal combinations. Parts of them are listed below in Table II. The inverse matrix table composed of $B^{-1}_k$ ($k = 1, ..., 120$) can be obtained correspondingly. With these two tables pre-stored in the computer, the onboard real time control allocation algorithm can be implemented.
B. Simulation Results

First, we define the control error percentage so as to compare the accuracy of different control allocation algorithm. In each cycle, the control error is defined as 
\[ \Delta u = u^* - u, \]
where \( u^* \) is the actual command produced by thrusters and \( u \) is the expected command required by the controller. Define

\[ P = \sum_{i=1}^{3} |\Delta u_i|, \quad H = \sum_{i=4}^{6} |\Delta u_i| \]

as the momentum and angle momentum error in each control cycle.

Define

\[ P = \sum_{i=1}^{3} |u_i|, \quad H = \sum_{i=4}^{6} |u_i| \]

as the total momentum and angle momentum error in each control cycle. Hence, the control error percentage is defined as follow:

\[ P\% = \frac{\Delta P_{\text{total}}}{P_{\text{total}}} \times 100\% \]

\[ H\% = \frac{\Delta H_{\text{total}}}{H_{\text{total}}} \times 100\% \]

Performances of the three control allocation Algorithms applied to the final approaching phase of rendezvous and docking are listed in Table III.

As we can see, the control error percentage of LP and OTCT is much smaller than the decoupling method in that the level of maximum deliverable force and torque of these two methods is higher. Take the \( F_y \) command for example. The control errors of the three algorithms in each cycle are illustrated in Fig. 3. In the early phase of the simulation, the force command in \( y \) direction exceeded the force capability of the decoupling method (+90N produced by fully open thruster No. 7 and 14 simultaneously and -90N by thruster No. 8 and 12). As the OTCT is fixed to 6 thrusters per selection, its maximum force and torque level is a little lower than LP, which lead to the control error in cycle 3 to 5. However, this can be improved by adding 1 or 2 real time iteration steps according to the dual simplex algorithm [8].

The total firing duration is the sum of the firing duration of every thruster in each cycle, which indicates the fuel consumption level of the control allocation algorithm. From the data in Table III, we can see that the total firing duration has been reduced significantly by using OTCT method or LP method compared with the decoupling method. The reason for this observation is that, the efficiency of thrusters has been improved owing to the optimal thruster combinations table that divides thrusters into groups with respect to their own characteristics instead of predefined maneuvers. The total firing duration of the three algorithms in each control cycle is

\[
\begin{array}{cccc}
\text{Decoupling} & 15.75 & 48.75 & 147.00 & 0.018 \\
\text{LP} & 0.001 & 0.001 & 88.81 & 3.61 \\
\text{OTCT} & 0.59 & 2.75 & 87.97 & 0.031
\end{array}
\]

F represents force; M represents torque. x-roll, y-pitch, z-yaw.

<table>
<thead>
<tr>
<th>Combination Number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>...</th>
<th>...</th>
<th>114</th>
<th>115</th>
<th>116</th>
<th>117</th>
<th>118</th>
<th>119</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thruster Number</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>...</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE I**

**THRUSTS AND TORQUES PRODUCED BY THE 18 THRUSTERS IN BODY FRAME**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_x )</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-150</td>
<td>-150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( F_y )</td>
<td>0</td>
<td>-50</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>120</td>
<td>-120</td>
<td>100</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>-30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( F_z )</td>
<td>50</td>
<td>0</td>
<td>-120</td>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-30</td>
<td>0</td>
<td>0</td>
<td>-60</td>
<td>60</td>
<td>60</td>
<td>-60</td>
<td>60</td>
<td>-60</td>
</tr>
<tr>
<td>( M_x )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>-60</td>
<td>60</td>
<td>-60</td>
</tr>
<tr>
<td>( M_y )</td>
<td>350</td>
<td>0</td>
<td>-350</td>
<td>0</td>
<td>-120</td>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-120</td>
<td>0</td>
<td>120</td>
<td>0</td>
<td>-60</td>
<td>60</td>
<td>60</td>
<td>-60</td>
<td>60</td>
</tr>
<tr>
<td>( M_z )</td>
<td>0</td>
<td>350</td>
<td>0</td>
<td>350</td>
<td>0</td>
<td>0</td>
<td>-120</td>
<td>120</td>
<td>-150</td>
<td>150</td>
<td>0</td>
<td>-120</td>
<td>0</td>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**TABLE II**

**OPTIMAL THRUSTER COMBINATIONS TABLE**

<table>
<thead>
<tr>
<th>Combination Number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>...</th>
<th>...</th>
<th>114</th>
<th>115</th>
<th>116</th>
<th>117</th>
<th>118</th>
<th>119</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thruster Number</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>...</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE III**

**PERFORMANCES OF THE THREE CONTROL ALLOCATION ALGORITHMS**

<table>
<thead>
<tr>
<th>Control Error (%)</th>
<th>Total Firing Duration (s)</th>
<th>Total Computation Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P% )</td>
<td>( H% )</td>
<td>( 15.75 )</td>
</tr>
<tr>
<td>( H% )</td>
<td>( P% )</td>
<td>( 0.001 )</td>
</tr>
<tr>
<td>( 0.59 )</td>
<td>( 2.75 )</td>
<td>( 87.97 )</td>
</tr>
</tbody>
</table>
illustrated in Fig. 4. It reflects the general tendency of the fuel consumption. Curves of the OTCT and LP are very close to each other, which demonstrated the optimality of the OTCT solutions.

The total computation time the OTCT expend onboard is on the same level as the decoupling method, which is much faster than the LP, as listed in Table III. This is consistent with our analysis in section III.

V. CONCLUSION

The paper proposed an algebraic method to divide thrusters into groups (denominated the optimal thruster combinations table) according to thruster properties, so as to simplify the optimal onboard real time control allocation algorithm. The performance of the real time control allocation algorithm based on the optimal thruster combinations table in terms of fuel consumption is encouraging, as the new algorithm is able to get the same optimal solutions as the simplex iterative algorithm with a much faster computation speed, which proves to be compatible with the processing power of the onboard computer. Besides, it is also compatible with complex thruster configurations (non axis-symmetric with coupling effects). Its application in real time control allocation of spacecraft will provide the opportunity to reduce the fuel consumption and extend the mission-life time.

REFERENCES