Heuristic Algorithms for Train Station Parking Using Information of Transponders

Kaicheng Li¹, Jiateng Yin²
¹National Engineering Research Center of Rail Transportation Operation and Control System, Beijing Jiaotong University, Beijing, China
²State Key laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University
*Corresponding author, email: kchli@bjtu.edu.cn

Abstract
Train Station Parking (TSP) has received increasing concentration as Platform Screen Doors (PSDs) are widely used in Urban Rail Transit. Aiming to enhance the accuracy and robustness of TSP, we proposed three algorithms which are Newton Dynamics based Algorithm (NDA), Heuristic Learning based Algorithm (HLA) and Heuristic Algorithm based on deceleration deviations Sequences (HAS) by using the information of transponders, essential locating equipments in subway. Then we verify the three algorithms on time-delay of the braking system and the initial speed of the train in TSP simulation platform. The result indicates that HLA and HAS can keep parking errors in 30cm while NDA can't. Furthermore HAS achieves the best performance compared with NDA and HLA.

Keywords: transponders, train station parking (TSP), heuristic algorithms.

1. Introduction
Subway is playing an increasingly significant role in urban public transportation system [1] in recent years for its high efficiency, convenience and safety. Many new metro stations are equipped with platform screen doors to prevent passengers from falling down the platform. Hence, higher precision for TSP is required to open the PSDs. However inaccurate train station parking has occurred now and then. For instance, for the station with installation of PSDs, inaccurate parking will cause that the screen door cannot correct alignment; then, lead to passengers cannot exchange effectively [2].

Some intelligent control methods [3] have already been proposed in TSP. The predictive fuzzy control system was proposed by K. Yoshimoto, where parking was divided into multiple sections so as to apply different control rules from a set of control rules [4, 5]. But it is difficult to build integral fuzzy rules. Hou employed terminal iterative learning control for TSP and obtain good results [6]. Chen made two simplifications about the train dynamics equation based on the braking characteristics of TSP in urban rail transit [7], and presented the soft computing techniques to parking control [8]. Zhou explored some machine learning methods in TSP [9]. Nevertheless, these works need a large amount of historical field data to optimize the control parameters, which is not easy to obtain before designing control rules. Furthermore, all of the above methods are assumed that train positioning information is accurate in simulation, and neglected the disturbances in train braking systems and external factors. Transponders are key locating equipments in the European Train Control System (ETCS) and Chinese Train Control System (CTCS), installed in subway station to provide precise location data for the train passing them [10]. The accurate positioning information is of great value for TSP [11, 12], which will be employed to design online control algorithms for TSP.

Considering of all above factors, this paper presents three algorithms by using information of transponders to improve the accuracy and robustness of TSP. The three algorithms don’t need the precise parameters of the train model trying to keep the train parking errors in 30cm. It is known that accuracy of TSP is affected by many factors, and the two main factors are different initial speed when the train is entering the stopping area and the time delay (T_d) of train braking system which will be stressed in the rest of this paper.
2. Train Station Parking Algorithms

The TSP system in this paper includes the following three parts: braking system, core algorithms (NDA, HLA and HAS), and information of transponders (as shown in Figure 1). The algorithms use the velocity and position of the train and the location information of transponders to output different deceleration rate. In order to facilitate our subsequent discussions, we defined a parking area L (as shown in Figure 2). In this section, there are n transponders which are marked T1, T2, ..., Tn. We denote the first transponder as initial point, and the last transponder as a parking point. \( v_i \) is the speed of a train when passing by the transponder \( T_i \).

The maximum value what parking controller outputs is \( d_{\text{MAX}} \) because of the limit of train brake system.

![Figure 1. The Block Diagram of TSP System](image1)

![Figure 2. TSP Area](image2)

TSP is a control process which takes train operation data for input and is subjected to a time-varying disturbance as well. In general, representing as following control model:

\[
y = f(v_{\text{lim}}, v, s, G, \varphi)
\]  

(1)

Where \( v_{\text{lim}} \) is the limited speed, \( v \) is the speed of a train, \( s \) represents train location, \( G \) is track curvature information, \( \varphi \) are all kinds of disturbances which include the weather, the number of passengers and so on, and \( y \) is the output.

2.1. Newton Dynamics based Algorithm (NDA)

When the train passes by the transponder \( T_i \), it can receive accurate positioning information \( s_i \). The current theory of braking ratio of \( y_i \) can be calculated by the formula (1) according to the current speed where \( v_i \) is equal to 0. As the train passes by the next transponder, we can get the output of the controller based on formula (1).

\[
v_n^2 - v^2 = 2v_i \cdot s_i
\]

It can be derived that by using the information of transponders placed on the station, the output sequence of TSP can be described as \( Y = (y_1, y_2, \cdots, y_{n-1}, y_n) \) and \( n \) presents the amount of the transponders. Based on Newton Dynamics, if not considering the interference factors, it can be regarded that:

\[
Y = (y_1, y_2, \cdots, y_{n-1}, y_n) = (\frac{v_1^2}{2s_1}, \frac{v_2^2}{2s_2}, \ldots, \frac{v_{n-1}^2}{2s_{n-1}}, \frac{-v_n^2}{2s_n})
\]

(2)

Thus, theoretical braking rate can be calculated and outputted by (2) every time passing by the transponder.

*Heuristic Algorithms for Train Station Parking Using Information of Transponders (Kaicheng Li)*
2.2. Heuristic Learning based Algorithm (HLA)

Due to the influence of interference factors, as is shown in formula (3), (4), (5) we attain the actual average braking ratio $d_i$ based on formula(4), where $D_i$ represents the distance between two transponder that is $D_i=S_i - S_{i+1}$, and $v_i$ is the train speed when passing by transponder $T_i$. Due to the influence of these factors, $d_{Ei}$ and $d_i$ exists certain deviation marked in (5).

$$v_n^2 - v_i^2 = 2d_{Ei} \cdot s_i$$  \hspace{1cm} (3)

$$v_{i+1}^2 - v_i^2 = 2d_i \cdot D_i$$  \hspace{1cm} (4)

$$\Delta d_i = d_i - d_{Ei}$$  \hspace{1cm} (5)

$\Delta d_i$ is always not equal to zero. According to the deviation between output value and expect value, we can postulate that interfering factors effecting parking average dispersion to the stopping progress [13], and the average value is $\Delta d_i$. Based on the above assumptions we proposed a fixed learning rate to dynamically adjust train braking controller output, and the model as shown below.

$$y_{i+1} = d_{E(i+1)} - k\Delta d_i$$  \hspace{1cm} (7)

In formula (7), $k$ represents the learning rate and $y_{i+1}$ represents the output of the controller. The model can dynamically adjusts the output according to the previous control information on a real-time basis, and offer a certain compensation caused by the interference.

2.3. Heuristic Algorithm Based on Deceleration Deviations Sequence (HAS)

It can be seen that HLA can compensate part of the errors derived by uncertain factors while HLA can only learn from $\Delta d_i$. It needs to be stressed that $\Delta d_1, \Delta d_2, \Delta d_3, \ldots, \Delta d_{i-1}$, $\Delta d_i$ are all important information we can get from the transponders. Aiming to take advantage all of these data, HAS is proposed to learning from the past, not only the last transponder information but also the information from previous transponders. The learning rate is no longer a constant but several sequences $\xi_i$. It is shown in formula (8) that if the train gets through transponder $T_i$, the output of the controller will learn from $\Delta d_1, \Delta d_2, \Delta d_3, \ldots, \Delta d_{i-1}, \Delta d_i$ to get the optimal output $y_{i+1}$.

$$y_{i+1} = d_{E(i+1)} - \sum_{j=1}^{i} \xi_j \Delta d_j$$  \hspace{1cm} (8)

The other is that how to obtain the learning rate matrix $\xi_i$. Unique learning rate is no longer effective because the information from the transponder $v_{i-1}$ is more important than the transponders earlier. So heuristic adaptive learning rate is proposed in formula (9).

$$\xi = \begin{bmatrix}
\omega & 0 & 0 & \cdots & 0 \\
\omega^2 & \omega & 0 & \cdots & 0 \\
\omega^3 & \omega^2 & \omega & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\omega^n & \omega^{n-1} & \cdots & \omega & 0
\end{bmatrix}, \quad \omega < 1$$  \hspace{1cm} (9)
In the matrix, $\omega$ represents a basic learning constant which is always smaller than 1 and it's obvious that if the train get through transponder $T_i$, the learning array is $(\omega_1^i, \omega_2^i, \cdots, \omega)$. HAS can learn from deceleration deviations sequences from previous information in which aspect HAS is different from HLA.

There is no need to attain the train model in the learning algorithm mentioned above, and the algorithm can adaptively modified through the way of learning and the actual operating parameters, which is conducive to reduce the impact of these factors. Furthermore, it is significantly efficient to the train braking system affected by the responses of different train braking system and the environment. We will verify the efficiency of the three algorithms for error correction which is based on the transponder information in following section.

3. Simulation Model and Performance Indices

3.1. Train Station Parking Model

In order to simulate the factors which would affect TSP, we use Simulink Toolbox of Matlab to build the simulation model of train operational system. The model comprises five components which is input module, generation module, control module, execution module, and output-display module respectively. Input module is used for inputting variant information of TSP process, and generation module mainly simulates basic resistances and random disturbances when a train is in motion. The resistance formula employed in this paper is considered as formula (10) [14].

\[
R = \alpha \nu^2 + \beta \nu + \gamma
\]  

(10)

Control module is the core component in the simulation model, which includes all controllers of different algorithms. Execution module simulates the actual train braking system. Finally, the simulation results (e.g. parking error and controller output) are exported in the output-display module.

We use the following formula (11) as transfer function of the braking system, which was introduced in previous work [15, 16].

\[
G(s) = \frac{d_0}{1 + T_d s} e^{-T_p s}
\]

(11)

Where $d_0$ represents the basic braking force of the train, $T_d$ and $T_p$ represent the time-delay and the time-constant of the braking system respectively.

3.2. Performance Indices

After constructing the simulation platform and programming the three algorithms, the performance of different need to be evaluated. The simulation is carried out for $n$ times in this paper and the parking error for each time is $e_i$. We proposed the three performance indices to verify the effectiveness and robustness of the algorithms.

I. Mean of absolute stopping errors

\[
E_{\text{avg}} = \frac{\sum_{i=1}^{n} |e_i|}{n}
\]

(12)

II. The maximum parking error

\[
E_{\text{max}} = \max(|e_i|, i = 1, 2, 3, \cdots, n)
\]

(13)

III. In subway operation, if the parking error is higher than 30cm, it can be considered that the control algorithm is not robust. So parameter $p$ is proposed to test that probability.
\[ P_{ex} = \frac{\sum \theta_i}{n} \times 100\% , \quad \theta_i = \begin{cases} 0 & \text{if } e_i > 30 \text{cm} \\ 1 & \text{if } e_i \leq 30 \text{cm} \end{cases} \] (14)

4. Simulation Evaluation
4.1. Simulation Parameters

Chapter 3 describes the construction of simulation model with which mass of TSP Simulation is done. The parameters which are calculated by using data from Beijing Subway Yizhuang Line are in Table 1. Note that the parameters are only used in simulation but not algorithm design and the three algorithms don’t need the precise train braking model parameters. In addition, it can be seen that the initial speed of the train and the time delay of the braking system are not certain but kept in a certain range which will affect the accuracy of TSP. In the following sections 4.2 and 4.3 the two factors will be stressed and discussed.

<table>
<thead>
<tr>
<th>Parameters of locomotive and railway</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistance (m/s²)</td>
<td>((\alpha, \beta, \gamma) = (1.36 \times 10^{-4}, 1.45 \times 10^{-2}, 1.244))</td>
</tr>
<tr>
<td>Position of Transponders (m)</td>
<td>((100, 64, 36, 16, 4, 0))</td>
</tr>
<tr>
<td>Initial speed of a train entering the stopping area (m/s)</td>
<td>(v_i \in (9, 11))</td>
</tr>
<tr>
<td>Time delay of braking system</td>
<td>(T_d \in (0.5, 0.7))</td>
</tr>
<tr>
<td>Time constant of braking system</td>
<td>(T_p = 0.4)</td>
</tr>
<tr>
<td>Learning rate of HLA</td>
<td>(k = 0.778)</td>
</tr>
<tr>
<td>Basic learning constant of HAS</td>
<td>(\omega = 0.74)</td>
</tr>
</tbody>
</table>

4.2. The Changes of the Train’s Initial Speed

When a train’s initial speed \(v_1\) which a train attempts to park a station varies between 9m/s to 11.5m/s, and all other parameters are kept constant, the interval of each change is 0.025m/s after 100 times simulation. The simulation results are shown in Figure 3 and the data is summarized in Table 2.

<table>
<thead>
<tr>
<th>Items</th>
<th>(E_{avg}) (cm)</th>
<th>(E_{max}) (cm)</th>
<th>(P_{ex}) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDA</td>
<td>22.78</td>
<td>49.26</td>
<td>80</td>
</tr>
<tr>
<td>HLA</td>
<td>9.71</td>
<td>23.1</td>
<td>100</td>
</tr>
<tr>
<td>HAS</td>
<td>4.24</td>
<td>10.5</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 3. Parking Errors as the Initial Speed Changes
In Figure 3 and Table 2, NDA algorithm can't keep its parking error in 30cm in the change of initial speed as it does not have adaptivity. HLA and HAS achieve better average parking error than NDA, they can keep the parking error in 30cm. Furthermore, HAS gets the best performance in the three algorithms as it utilize more information than HLA. The maximal parking error by HAS is only 10.5cm which embodies the effectiveness of the algorithm.

4.3. The Variations of Train Brake System Delay

When the braking system delay $T_d$ changes 100 times simulation intervals of 0.0036s between 0.42s to 0.78s. In Figure 4, denote the parking errors of the four parking control algorithms. The results are shown in Figure 4 and Table 3.

![Figure 4. Parking Errors as $T_d$ Changes](image)

<table>
<thead>
<tr>
<th>Items</th>
<th>$E_{avg}$ (cm)</th>
<th>$E_{max}$ (cm)</th>
<th>$P_{ex}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDA</td>
<td>22.1</td>
<td>28.5</td>
<td>100</td>
</tr>
<tr>
<td>HLA</td>
<td>9.02</td>
<td>23.3</td>
<td>100</td>
</tr>
<tr>
<td>HAS</td>
<td>2.55</td>
<td>4.88</td>
<td>100</td>
</tr>
</tbody>
</table>

From the data in Table 3 and Figure 4 we can see that the three algorithms can all achieve 30cm parking error while $T_d$ of the braking system varies from 0.5 to 0.7. Also HAS gets far better performance than the other two methods and HAS controls the parking errors in 5cm while most traditional control methods can’t.

4.5. Summary

According to the simulation experiments and the performance indices we have made in 4.1 and 4.2, we calculate and contrast the following indexes in conventional case under three algorithms: the changes of initial speed $v_1$, the system delays $T_d$. As can be seen in Table 2 and Table 3, mass of simulation data is summarized and compared. The results clearly indicates that HLA and HAS can keep the parking errors in 30cm when the initial speed of train and the $T_d$ of the braking system changes. Also when the train speed changes NDA has a chance to fail to accurate parking. Another result is that HAS performs much better than NDA and HLA not only for universal performance but also every indicators which means that learning from every transponder the train has get through makes the results of heuristic algorithm much better.

5. Conclusion

By using the information of the transponders placed on the subway station, NDA, HLA and HAS, the three model-free control algorithms are proposed in this paper to achieve TSP.
problem. NDA is based on Newton Dynamics and can’t learn from previous transponders. HLA can learn from the former transponder while HAS can learn from deceleration deviation sequences. We take the three parking algorithms proposed in analog simulation when initial speed and system delay changes, and the final results show that the HLA and HAS can meet the accuracy requirements of the train parking. At the same time, learning from not only one transponder enhances the accuracy and robustness of the algorithm. Moreover, the amount of its computation is small, which makes it easy to put into practical engineering applications and has a certain practical value to the development of rail transport.

Acknowledgement
This work is partially supported by the National High Technology Research and Development Program (“863” Program) of China under grant 2012AA112800, by New Scientific Star Program of Beijing under grant 2010B015, by the Fundamental Research Funds for the Central Universities under grant 2012JBM016, by State Key Laboratory of Rail Traffic Control and Safety and Beijing Key Laboratory of Urban Rail Transit Automation and Control under grant RCS2011ZT001.

References