A new coordinate system for Longitudinal Train Dynamics simulations

Qing Wu*

Centre for Railway Engineering, Central Queensland University, Rockhampton Australia

(Manuscript Received: 12 October 2017; Revised: 13 November 2017; Accepted: 26 November 2017)

Abstract

This paper presents a new coordinate system for whole-trip Longitudinal Train Dynamics (LTD) simulations. The new coordinate system integrates the concepts of both inertial and non-inertial coordinate systems which are called the global coordinate system and the local coordinate system respectively in this paper. The global coordinate system is used to describe vehicle positions, velocities, and accelerations on the track. The local coordinate system is a movable coordinate system attached on the train; it is designed to determine draft gear deflections only. The purpose of the new coordinate system is to decrease the truncation errors for whole-trip LTD simulations that have very long simulated train trips, for example 600 km. Results in this paper have demonstrated the effectiveness of the new non-inertial coordinate system. Simulation results provide the confidence to study very long train trips. The new coordinate system can maintain the computational precision and reduce the computational time by 8.6%.

Keywords: Coordinate system; Longitudinal train dynamics; Whole-trip simulation; Computing speed; Truncation errors

1. Introduction

Longitudinal Train Dynamics (LTD) is an important area of Vehicle System Dynamics. In LTD, vehicles are assumed without lateral and vertical motions [1-8], i.e., only one degree of freedom (longitudinal) is considered for each vehicle. Force elements considered are: (1) in-train forces, (2) air brake forces, (3) traction forces, (4) dynamic brake (DB) forces, (5) propulsion resistance, (6) gravitational components, and (7) curving resistance [9-18].

The idea of a new coordinate system was derived from the studies of whole-trip LTD simulations for heavy haul trains. A whole-trip LTD simulation [13] simulates a train running loaded from a departure station to a terminal station, then running empty from the terminal station back to the departure station. Whole-trip simulations were proposed on the grounds that [13]: (1) railways have fixed routes and slowly changing track conditions; (2) fixed train configurations are common for heavy haul operations; and (3) Automatic Train Operation systems and driving guidance make the driving commands similar from trip to trip. Compared with sectional or short-trip simulations, a whole-trip simulation covers all stages of a trip which enables thorough examinations of relevant issues and provides more detailed assessments. Whole-trip LTD simulations face the same challenges that are faced by short-trip simulations: high nonlinearities and large numbers of vehicles [19]. In addition, whole-trip LTD simulations have much longer operational time. A whole-trip simulation usually covers hundreds kilometres of track (e.g., 600 km) and significant hours of operation (e.g., 10 hours).

Due to the long simulated distances of whole-trip LTD simulations, computational precision and computational speed requirements for whole-trip LTD simulations are higher than those for short trip LTD simulations. This paper studies the issues of computational precision and computational speed from two perspectives: data types and coordinate systems. Section 2 reviews two types of conventional coordinate systems that have been used for LTD simulations. Section 3 studies the fundamental issue of programming data types and truncation errors. In Section 4, a new coordinate system for LTD is presented. The effectiveness and computational efficacy of the new coordinate system are also studied in this paper.

2. Conventional coordinate systems

Equation of Motions (EOMs) for LTD can be found in many references [4, 7-9, 11, 20-24]. Most of them are based on the conventional inertial coordinate system as shown in Figure 1. The inertial coordinate system (e.g., [24]) defines all vehicles on the basis of one single global coordinate system, i.e., positions ($x_i$), velocities ($v_i$), and accelerations ($a_i$) of all vehicles are described with respect to the origin of the global system.

---

*Corresponding author: Qing Wu (*)
Bld 70, Central Queensland University, Rockhampton, Queensland, Australia, QLD4701

e-mail: q.wu@cqu.edu.au

161
Assuming that there are \( n \geq 3 \) vehicles in the train, then the EOMs can be generalized as:

\[
\begin{align*}
    m_1a_1 &= F_1 - F_{e,1} \\
    m_i a_i &= F_i + F_{e,i} - F_{e,i} \quad (i = 2, n - 1) \\
    m_n a_n &= F_n + F_{e,n-1} - F_{e,n-1} \quad (3)
\end{align*}
\]

where the \( m_i (i = 1, n) \) is the mass of the \( i^{th} \) vehicle; \( F_{e,i} (i = 1, n - 1) \) is the sum of all the other forces (except in-train forces) that the \( i^{th} \) vehicle could be subjected to.

To demonstrate the truncation error in addition operations, some basic add operations were programmed using the Intel\textsuperscript{TM} Fortran language. The addition processes and corresponding results are listed in Table 2. The results show that truncation errors were found in both of the single precision cases. However, the second case had better precision than the first case as less significant figures were occupied by the integer part; more were available for the decimal part. Table 2 shows that there was no truncation error for the double precision cases as sufficient significant figures were provided.

The computing precision can also be compromised when subtraction operations are performed between two large numbers to determine a small difference. The subtraction error can be a significant issue for the precision of draft gear deflections. Draft gear deflections are of small values (e.g., 5mm) and they are determined by the relative locations of adjacent vehicles which are usually large numbers (e.g., 100 km). To demonstrate the truncation error in subtraction operations, numerical experiments were carried out as listed in Table 3. The same conclusions as those reached in addition operations can be drawn for the subtraction operations. One conclusion worth mentioning again is that, by decreasing the number of significant figures occupied by the integer part, the precision of simulations can be improved. Therefore, computing operations involving two small numbers have better precision than that of two large numbers.
### Table 1. Floating-point type of data

<table>
<thead>
<tr>
<th>Precision</th>
<th>Statement in Fortran</th>
<th>Statement in C/C++</th>
<th>Bit width</th>
<th>Significant figures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>Real (Kind=4)</td>
<td>Float</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Double</td>
<td>Real (Kind=8)</td>
<td>Double</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>Quadruple</td>
<td>Real (Kind=16)</td>
<td>Long double</td>
<td>16</td>
<td>33~36</td>
</tr>
</tbody>
</table>

### Table 2. Truncation errors in addition operations

<table>
<thead>
<tr>
<th>Case</th>
<th>Data precision</th>
<th>Theoretical results</th>
<th>Computing results</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Single</td>
<td>100000.0+0.123456789 =100000.123456789</td>
<td>A(KIND=4)= 100000.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B(KIND=4)= 0.123456789</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A+B=C(KIND=4)= 100000.123456789</td>
</tr>
</tbody>
</table>

### Table 3. Truncation errors in subtraction operations

<table>
<thead>
<tr>
<th>Case</th>
<th>Data precision</th>
<th>Theoretical results</th>
<th>Computing results</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Single</td>
<td>100000.0-0.123456789 =100000.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A(KIND=4)= 100000.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B(KIND=4)= -0.123456789</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A-B=C(KIND=4)= 100000.0-0.123456789</td>
</tr>
</tbody>
</table>

### 4. A new non-inertial coordinate system

Computing precision of LTD can be improved by exploiting the conclusions reached in Section 3. The non-inertial coordinate system proposed in [20] and reviewed in Section 2 is a good example. This non-inertial coordinate system was designed to avoid differences of large numbers to determine small draft gear deflections, so the computing precision could be improved. However, this non-inertial coordinate system also had some disadvantages. As reported by Kerr and Blair [20], there were accumulative errors for vehicle positions along the train. The accumulative errors could be caused by the way the system accumulates accelerations. In this coordinate system, the acceleration of the $i^{th}$ vehicle was expressed as $\sum_{j=1}^{i} a_j$ which was the sum of the accelerations of all vehicles in front of the $i^{th}$ vehicle. Therefore, there were accumulative errors for the results of accelerations along the train. In other words, the errors concerning vehicles at the rear part of the train are larger than that at the front part of the train. Another disadvantage of this non-inertial coordinate system concerns its implementation. Also due to the dependent accelerations, the implementation of this coordinate system was awkward and could not be easily solved by conventional numerical integration methods. In [20], to be able to solve LTD, the draft gear model was simplified as three linear springs without damping; EOMs were solved using an approach called transition matrix approach. Comparatively, the implementation of the conventional inertial coordinate system is more straightforward and can be solved by a variety of numerical solvers.
In this paper, a new non-inertial coordinate system has been designed to alleviate the possible precision disadvantage of the conventional inertial coordinate system and to retain the straightforward implementation of it. The new coordinate system, as shown in Figure 3, integrates the concepts of both inertial and non-inertial coordinate systems. To differentiate the names, these two coordinate systems in Fig. 3 are called the global coordinate system and the local coordinate system respectively. The global coordinate system is the same as the one shown in Fig. 1 (the inertial coordinate system), which can be used to describe vehicle positions, velocities, and accelerations on the track. The local coordinate system is a movable coordinate system attached on the train; it is designed to determine draft gear deflections only. In Fig. 3, \( x_i \) is the location of the first vehicle of the train while \( ex_i \) is the sum of the deflections of all draft gears before the \( i \)th vehicle. Note that \( ex_i \) has removed the physical length of wagon bodies and connection systems. The computing process by using the new coordinate system is as follows:

- Replace \( x_i \) in Eqs. 1-3 by \( ex_i \); use a numerical solver to determine accelerations, velocities, and local locations (\( ex_i \)) of all vehicles. Determine the global location of the first vehicle:
  \[
  x_1 = x_0 + ex_1
  \]
  Where \( x_0 \) is the global location of the first vehicle in last time-step.

An important point related to this step is that both \( x_0 \) and \( x_1 \) have to be stored at least double precision type of data. Otherwise significant truncation error will be caused as demonstrated in Section 3.

- With reference to the local coordinate system, move the train as a whole system without any change to the relative positions and motions between individual vehicles:
  \[
  ex_i = ex_i - ex_1
  \]
  The purpose of this step is to keep \( ex_i \) at small values, so as to increase the precision for the difference operations.

- Determine the deflections of all draft gears:
  \[
  x_{f,d,i} = ex_i - ex_{i-1}
  \]

- Determine vehicle positions with reference to the global coordinate system:
  \[
  x_i = x_1 + (i-1)L_c + ex_i
  \]
  Where \( L_c \) is the coupler spacing.

The above steps are performed after the time integration, which means all velocities and accelerations with respect to the local coordinate system are the same with reference to the global coordinate system. Eq. 9 determines draft gear deflections for the calculations of draft gear forces. Eq.10 determines vehicle locations for calculations of other relevant forces. Note that, by using the new coordinate system, parameters updated by numerical solvers are \( v_i \) and \( ex_i \), i.e., the velocities and locations of vehicles with respect to the local coordinate system. The new coordinate system has avoided the differenting of large displacements to obtain small draft gear deflections, and has also retained the simple implementation of the conventional inertial coordinate system.

**Validation of the new coordinate system**

To validate the idea and ensure correct implementation of the new coordinate system, four train simulations were conducted for a Distributed Power (DP) train with two types of floating-point data (single precision and double precision) in two types of coordinate systems (the inertial system and the new non-inertial coordinate system). The DP train has the configuration of 2 locomotives + 105 wagons + 2 locomotives + 105 wagons. Knowing that the combination of a low speed, a small step-size, and long distance poses the biggest challenge for numerical computing, all four simulations were set as: the train started at the location of 500 km with a constant speed of 0.36 km/h (0.01 m/s); the step-size was 0.0001s (0.1ms) with the 4th Runge-Kutta numerical solver; the train operational time was 1800 seconds (half an hour). To be able to determine the theoretical locations of all vehicles, the resistance forces, traction forces, and brake forces were set as zero during the simulations, so the trains were running at a constant speed. Theoretical and computing results of the posi-

---

**Table 4. Comparison of two coordinate systems**

<table>
<thead>
<tr>
<th>Result types</th>
<th>Coordinate system</th>
<th>Data precision</th>
<th>First vehicle position (m)</th>
<th>Last vehicle position (m)</th>
<th>Computing time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing</td>
<td>Inertial</td>
<td>(1) Single</td>
<td>500000.0</td>
<td>497444.0</td>
<td>749.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) Double</td>
<td>500180.0002</td>
<td>497624.0002</td>
<td>879.12</td>
</tr>
<tr>
<td></td>
<td>New non-inertial</td>
<td>(3) Single</td>
<td>500180.0</td>
<td>497624.0</td>
<td>803.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4) Double</td>
<td>500179.9999</td>
<td>497623.9999</td>
<td>986.98</td>
</tr>
<tr>
<td>Theoretical</td>
<td></td>
<td></td>
<td>500179.9999</td>
<td>497623.9999</td>
<td></td>
</tr>
</tbody>
</table>
tions of the first vehicle and the last vehicle are listed in Table 4; the computing time for all simulation cases is also listed. Due to the intrinsic setting of the source code, the last outputs of the simulations were at time $t = 1799.9991s$. All results in this table correspond to $t = 1799.9991s$.

Table 4 shows that all cases except for the first one (inertial, single) gave accurate results with minor differences. The simulated train in the first case did not move at all. This was caused by the truncation error as previously discussed, because the displacement increment was smaller than the truncation error in each time-step, so the position of the train did not change. Comparing the results of the first case with that of the third case, it can be seen that the computing precision was significantly improved by using the new coordinate system. The new non-inertial coordinate system gave accurate results even with the single precision data type. Comparing the third and fourth cases, it can be seen that when using the new coordinate system, the single precision data case reached the same precision level as the double precision case.

In regard to the computational speed, it is understandable that computing using the double precision data takes longer than using the single precision data. Table 4 shows the computing time increased 17.3% and 18.6% for the inertial and the non-inertial coordinate systems respectively when the data type changed from single precision to double precision. LTD simulation results in Table 4 show that the non-inertial coordinate system is effective for whole-trip LTD simulations using single precision data. Comparing the second and third cases in Table 4, it can be seen that the non-inertial coordinate system with single precision data reduced the computing time by 8.6%. The improvement is smaller than the previously mentioned increases (17.3% and 18.6%) due to the additional cost from the coordinate shifting, i.e., computing of Eqs. 4 and 10.

The new coordinate system has been successfully implemented in the LTD simulator called TDEAS [3]. The simulator has been used for various studies including [25] and [26]; the simulator has achieved good results in the International Benchmarking of Longitudinal Train Dynamics Simulators [27].

Conclusion

Coordinate systems are the fundamental of LTD simulations. In LTD research and consulting projects, there is a demand for whole-trip simulations which need to simulate very long train trips, for example, 600 km. Due to the long simulated distances, computational precision and computational speed requirements are higher than those for short trip LTD simulations. This paper studies the issues of computational precision and computational speed from two perspectives: data types and coordinate systems. A new coordinate system for LTD simulations is presented, which combines a global coordinate system and a local coordinate system. The global coordinate system is used to describe vehicle positions, velocities, and accelerations on the track. The local coordinate system is a movable coordinate system attached on the train; it is designed to determine draft gear deflections only. The new coordinate system can decrease the truncation errors for whole-trip LTD simulations. The results of this paper have demonstrated the effectiveness of the new non-inertial coordinate system. Simulation results provide the confidence to study very long train trips. The new coordinate system can maintain the computational precision and reduce the computational time by 8.6%. The new coordinate system has been used for various studies.

Reference