Computation of Two-Point Contact between Wheelset and Rail Exploiting the Structure of Non-Penetrating Contact Equations for Straight Track

Behrooz Fallahi* and Chao Pan#

* Faculty of Mechanical Engineering
Northern Illinois University
590 Garden Road, Dekalb 60115
bfallahi@niu.edu

# Institute for Mechatronics
Northern Illinois University
590 Garden Road, Dekalb 60115
z1647714@students.niu.edu

Abstract

Computation of the contact points between wheel and rail is a fundamental problem in railroad vehicle modeling. In solving the governing equations for the contact points, two questions should be answered [1]. They are: 1) what is the number of contact points? And 2) where are their locations? To answer these questions, the non-penetrating contact equations are derived in track coordinate system [2]. It is shown that [3] five equations per contact should be constructed. Four of these equations locate common normal and the fifth equation makes the length of common normal to become zero. For two-contact-points problem [3], the Jacobian matrix per set of contact point can be partitioned as:

\[
\begin{bmatrix}
c_{11} & 0 & c_{13} \\
0 & c_{23} \\
c_{31} & c_{32} & c_{33}
\end{bmatrix}
\]

where, \( C \) is the Jacobean matrix and, the bloc matrices are:

\[
c_{11} = \begin{bmatrix}
\frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial y_{wp}} & 0 \\
\frac{\partial f_2}{\partial \alpha} & 0 & \frac{\partial f_2}{\partial y_{rp}} \\
\frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial y_{wp}} & \frac{\partial f_4}{\partial y_{rp}}
\end{bmatrix}
\]

\[
c_{13} = \begin{bmatrix}
0 & \frac{\partial f_1}{\partial \phi_{ws}} \\
0 & \frac{\partial f_2}{\partial \phi_{ws}} \\
\frac{\partial f_4}{\partial \phi_{ws}} & \frac{\partial f_4}{\partial \phi_{ws}}
\end{bmatrix}
\]

\[
c_{21} = \begin{bmatrix}
\frac{\partial g}{\partial \alpha} & \frac{\partial g}{\partial y_{wp}} & \frac{\partial g}{\partial y_{rp}}
\end{bmatrix}
\]

\[
c_{23} = \begin{bmatrix}
\frac{\partial g}{\partial \phi_{ws}} \\
\frac{\partial g}{\partial \phi_{ws}}
\end{bmatrix}
\]

Let \( \Delta x \) and \( V \) be the vector of correction terms and functions whose zeros are sought. That is:

\[
\Delta x = [\Delta \alpha, \Delta y_{wp}, \Delta y_{rp}, \Delta s_{tr}, \Delta z_{ctr}, \Delta \phi_{ws}]^T, \quad V = [f_1, f_2, f_4, g, f_3]^T
\]

Partition \( \Delta x \) and \( V \) to match the partitions of the Jacobean matrix \( C \). That is:

\[
\Delta x = [\Delta x_1, \Delta x_2, \Delta x_3]^T, \quad V = [V_1, V_2, V_3]^T, \quad \Delta x_1 = [\Delta \alpha, \Delta y_{wp}, \Delta y_{rp}]^T, \quad \Delta x_2 = \Delta s_{tr}, \quad \Delta x_3 = [\Delta z_{ctr}, \Delta \phi_{ws}]^T \text{ and, } V_1 = [f_1, f_2, f_4]^T, \quad V_2 = g, \quad V_3 = f_3
\]

Then we have,

\[
C_{11} \Delta x_1 + C_{13} \Delta x_3 = V_1 \quad (1)
\]

\[
C_{21} \Delta x_1 + C_{23} \Delta x_3 = V_2 \quad (2)
\]

\[
C_{31} \Delta x_1 + C_{32} \Delta x_2 + C_{33} \Delta x_3 = V_1 \quad (3)
\]

To compute the correction terms, use Equation (2) to solve for \( \Delta x_1 \) to get:

\[
\Delta x_1 = C_{11}^{-1} \cdot (V_1 - C_{13} \cdot \Delta x_3) \quad (4)
\]

Substitute Equation (5) into Equation (3) to get:
\[
[C_{23} - C_{21}(C_{11})^{-1}C_{13}]\Delta x_3 = V_2 - C_{21}(C_{11})^{-1}V_1
\] (5)

Note that Equation (6) can be constructed for each contact point. Therefore, for two-contact-point problem, we have:

\[
\begin{bmatrix}
C_3^1 - C_{21}^1(C_{11}^1)^{-1}C_{13}^1 \\
C_3^2 - C_{21}^2(C_{11}^2)^{-1}C_{13}^2
\end{bmatrix}\Delta x_3 = \begin{bmatrix} V_2^1 - C_{21}^1(C_{11}^1)^{-1}V_1^1 \\
V_2^2 - C_{21}^2(C_{11}^2)^{-1}V_1^2 \end{bmatrix}
\] (6)

Note that the superscripts denote the contact point number. Equation (7) can be used to solve for \(\Delta x_3\). Once \(\Delta x_3\) is computed, use Equation (5) to compute \(\Delta x_2\). To compute \(\Delta x_2\), use Equation (4). That is:

\[C_{32}\Delta x_2 = V_3 - C_{31}\Delta x_1 - C_{33}\Delta x_3\] (7)

Based on the method described, an algorithm is implemented to find the contact points. The results for one case are reported in Table 1. To verify the solution, the plot of the function \(F\) as described in [2] is reported in Figure 1. The zero of \(F\) is the \(y_{wp}\) in Table 1. As can be seen the \(x\)-intercept of the graph matches the value of \(y_{wp}\) of the Table 1.

<table>
<thead>
<tr>
<th>Left</th>
<th>Initial</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>179.822</td>
<td>179.946</td>
</tr>
<tr>
<td>(y_{wp})</td>
<td>-0.007</td>
<td>-0.004</td>
</tr>
<tr>
<td>(y_{rp})</td>
<td>-0.004</td>
<td>-0.001</td>
</tr>
<tr>
<td>(m)</td>
<td>0.11</td>
<td>0.109</td>
</tr>
</tbody>
</table>

**Figure 1:** Plot of left track at \(y=0.01\) \(\phi =7^\circ\)

In summary, the advantage of the method presented in this work is that, it uses block matrices whose dimensions are much smaller than the full Jacobean matrix. Hence it is more efficient than the method based on using the full Jacobean matrix.

**References**

