The use of pattern recognition algorithms in an Automatic Vehicle Identification system

J. F. W. Pretorius & B. M. Steyn
Railway Engineering, Spoornet, South Africa

Abstract

Apart from reading vehicle numbers passed at a given point the Automatic Vehicle Identification (AVI) system employed by Spoornet is required to build a record of the train consist and requires identification of the vehicle types such as wagons and locomotives. This must be accurate even if electronic tags are missing or damaged on some of the vehicles, or when the train speed or direction changes while being measured. In many instances clearance between tracks on a double line section prohibit the installation of reader antennas between the lines. Under these conditions it is required that the train consist shall not be corrupted while two trains pass on a double line at the same site. In order to ensure reliable performance in accordance with these requirements different pattern recognition algorithms had to be implemented on the system. This paper will present the algorithms employed, some of the operational problems experienced as well as unique developments undertaken to solve these problems.

1 Background of the Automatic Vehicle Identification system

The main purpose of the AVI system is to determine train consist at predetermined sites along the railway track [1]. The AVI system consists of field sites as well as an AVI server. The field sites read consist data and send this data to the server which makes the data available in the operational processes of Spoornet. In addition to the RF tag reader’s wheel detectors are employed to determine the speed and axle spacing when a train traverses the site. A wheel sensor interface is used to couple the wheel transducers to the site processing unit. Figure 1 shows the layout of a typical AVI field site. Tags are mounted on each side of the vehicles in order to identify front and rear ends of a vehicle [2].
2 Development of a new wheel sensor interface

The original wheel sensor interface was used to detect train wheels passing the AVI field site, time stamp the wheels detected and send the information to the field station processor via RS232. The initial design made use of proprietary wheel sensors and was integrated as part of the wheel sensor interface. Detection problems were experienced under various field conditions. It was not possible to replace the problematic wheel sensors because of the integrated design. Since wheel sensor data form an important part of the reliability and correctness of AVI consist data, it was necessary to upgrade this part of the initial design. It was decided to specify a new wheel sensor interface with a simple and open interface to accommodate a variety of wheel sensors. This will enable the use of any wheel sensor with a switched output. The upgrade also created the opportunity to implement various algorithms which could withstand real life situations much better. In addition to this other algorithms which was developed as interim solutions were retained, thus making the system upgrade even more robust.

3 Algorithms implemented to improve reliability of data

This section will describe some of the algorithms implemented in order to improve reliability and accuracy of the AVI data. Some of the algorithms were
developed as a backup for the unreliable operation of the wheel sensors while others were implemented as part of the wheel transducer upgrade.

3.1 Determine the direction which the train was travelling

As mentioned the AVI system uses two proximity sensors to detect the wheels of the train. The sequence in which the sensors detect the first wheel was used to determine the direction of travel. In practice a fault conditions occurred when the first sensor failed to detect the train wheel and resulted in reporting the wrong direction of travel. This could create havoc in systems relying on the AVI data.

A new algorithm was implemented to detect the direction of travel even if the systems failed to detect some of the wheels. Figure 2 shows the pulses produced by the wheel sensors as train passes the site. Pulses marked 1 originates at sensor 1, and so on. The principle employed in the new algorithm, is to use the fact that the distances between bogies are much bigger than the distance between axles of a bogie. This can thus be used to identify the middle part of a vehicle. A count associated with first sensor triggered after the middle of a vehicle is accumulated for the whole train. The sensor with the highest count will then indicate the direction of movement. This algorithm will work even if some wheels were skipped or inserted.

Mathematical implementation of the algorithm:

1. If \( \frac{D[i+1]}{D[i]} > 4 \) Check which sensor was triggered first

\( D[i] \) – Time between wheel sensor pulses (see Figure 2).
2. Add 1 to the counter for the sensor which was triggered first.
3. Repeat steps 1 and 2 for all wheel pulses.
4. The sensor with the highest count indicates the direction of movement.

3.2 Identify missing tags by using the tags spacing pattern on the train

Another problem which occurred in the system with the unreliable wheel sensors was that the wheel sensor unit failed to detect wheels at high speeds. Since wheel sensor data was not available to check for any missing tags under these circumstances an algorithm was developed to determine missing tags using tag spacing as an input. The principle employed, is to determine a tag time slot in which a valid tag-read must appear. If no tag is found in the tag time slot, then a missing tag is reported for that wagon. The tag time slot (window) is determined by calculating a window related to the previous time spacing between tags (time prediction algorithm). A primary requirement, for this algorithm to work, is that the physical spacing between tags (and thus $T[i]$) must be more or less the same distance. Figure 3 shows the tag spacing and the tag present window.

![Diagram to show the tag spacing and the tag present window.](image)

3.3 Correction of skip wheel sensing data

Frequent miscounts from the wheel sensors create an error condition in the system where axles on a vehicle are not detected. This in turn will result in a failure to determine the train consist correctly. For a description of the algorithm developed to solve the problem at hand, the reader is referred to Figure 5 which shows wheel sensor data as time progresses.

- Determine the direction of movement of the train.
- Check the pattern sequencing for any missing pulses.
- From example in picture the following could be true
  - (No Faults: 1212121212121212)
  - (Faults: 1212212121212) Bold implies a wheel sensor pair being incomplete
- Calculate time T for all the wheel sensor pairs which was found complete.
If an axle is missing insert an axle into the sequence of pulses. If the first one in a pair is missing, insert the pulse by subtract the closest time \( T \) from the measured pulse and if the second pulse in a pair is missing insert a pulse by adding the closest time \( T \) to the measured pulse. By recognising these inherent patterns for the complete counting sequence, missing wheel sensor pulses can be reconstructed. In addition to this a faulty wheel sensor may be identified and reported.

### Field Setup to detect wheels on a train

The trains direction of movement

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
</table>

Wheel sensor 1       Wheel sensor 2

### Pulses produced from wheel sensors as time progress

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
</table>

1 Pulses from wheel sensor 1       2 Pulses from wheel sensor 2

**Figure 4:** Wheel sensor data as a function of time.

### 3.4 Vehicle identification using wheel patterns

Reading the tag on each vehicle identifies the individual vehicles forming part of a given train if the tags are present. If a tag is not present or faulty, an incorrect vehicle count/consist will result. To be able to accommodate this fault consist information is verified and corrected using the wheel sensor data. Identifying vehicles using the wheel sensor data will produce a vehicle list using wheel sensor data. This wagon list is then compared with the wagon list produced from reading the tag data. A vehicle with no tag will be inserted into the final vehicle list if wheels where sensed but no tag was read.

To identify vehicles using wheel sensor data the following algorithm was implemented:

- Gather all wheel sensor events and time-stamp it
- Determine the direction of movement
- Insert any missing wheel sensor events
- Calculate the speed of each axle of the train

\[
V(n) = \frac{T(2n) - T(2n - 1)}{D}
\]

Where
- \( n \) is the axle number,
- \( T \) is the time stamped wheel-sensing event and
- \( D \) is the distance between wheel sensors.
Pulses produced from wheel sensors as time progress for a 4 axle wagon.

![Diagram of wheel sensor pulses]

Figure 5: Wheel sensor pulses for a single wagon.

Table 1: Sample data from the implemented algorithm.

<table>
<thead>
<tr>
<th>Vehicle num.</th>
<th>Axle Num.</th>
<th>Time Stamp</th>
<th>Wheel sensor</th>
<th>Speed</th>
<th>Measured Distance</th>
<th>Expected Distance</th>
<th>Error</th>
<th>Error Sqr</th>
<th>Error Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>59273</td>
<td>0</td>
<td>1.832</td>
<td>1.825</td>
<td>1830</td>
<td>5</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>59768</td>
<td>1</td>
<td>1.834</td>
<td>6.482</td>
<td>6480</td>
<td>-2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>60741</td>
<td>0</td>
<td></td>
<td>1.826</td>
<td>1830</td>
<td>4</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>65945</td>
<td>0</td>
<td></td>
<td>1.832</td>
<td>1830</td>
<td>-2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>67409</td>
<td>0</td>
<td></td>
<td>1.979</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>68025</td>
<td>1</td>
<td>1.838</td>
<td>1.83</td>
<td>1830</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>69494</td>
<td>1</td>
<td>1.841</td>
<td>6.476</td>
<td>6480</td>
<td>4</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>70463</td>
<td>0</td>
<td></td>
<td>1.832</td>
<td>1830</td>
<td>-2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>75676</td>
<td>0</td>
<td></td>
<td>1.822</td>
<td>1830</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>77157</td>
<td>0</td>
<td></td>
<td>1.826</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>77794</td>
<td>1</td>
<td></td>
<td>1.826</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Calculate the distances between axles of the train.

\[ S(n) = \frac{V(n) + V(n + 1) \times T(n + 2) - T(n)}{2} \]

Where \( n \) is the axle number and \( T \) is the time stamped wheel-sensing event and \( V \) is the axle speed.
• To identify the vehicles take the vehicle spacing from the database of known vehicles and match it against the spacing measured using the wheel sensing data. (Matching is done by subtracting the spacing found in the database for a vehicle from the measured spacing, this difference is the squared and summed to determine the matching value. If the matching value is less than a predefined limit, the vehicle is marked as a possible match. The vehicle with the smallest matching value from the possible vehicle match list, is then used as the identified vehicle).

• The wheel sensor data for this vehicle is removed from the wheel sensor data array and the next vehicle match is extracted from the remaining data.

3.5 Filtering of tags read on the second line (double line installations)

On double line installations it is possible to read tags on the second line when two trains are passing the reader site at the same time. Initially the RF signal was attenuated so that the reader can only read tags on the line closest to it. Although this solves the problem initially it was later found that the RF signal was temperature sensitive and the RF signal strength had to be adjusted frequently. An algorithm to solve this problem was implemented using a combination of time stamped wheel sensor data and tag data to identify the validity of the tag read. Figure 6 shows relationship between the tag and wheel sensor data for this case.

Figure 6: Tag and axle date as gathered by AVI field site.
The algorithm used to filter invalid tags from tag list is presented below:

- Gather all wheel sensor events on wheel sensors 1 and 2 and time stamp each event.
- Read all tags passing the reader and record the time that the tag was read the first (TB) and last time (TE).
- Determine the direction of movement of the train (see previous)
- Insert any missing wheel sensor events (see previous)
- Determine the vehicle consist using the axle data
- Calculate the axle positions of each axle on a vehicle by using the average between wheel sensors 1 and 2.
- Calculate position of the tags read by using the average between the first time that the tag was read and the last time that the tag was read.
- Filter any tags which were not present between the first and the last axle of a vehicle.
- Rebuild a vehicle list using the wheel sensor data and filtered tag data.

The implementation of this algorithm worked extremely well. During the tests all the attenuation on the RF signal was removed and a test tag was positioned opposite the antenna (far side of the railway line and easily read by the antenna) while a train would pass on the line. Even at very slow speeds where the tag on the other side of the line was clearly visible in the gaps between the wagons, it was not inserted into the wagon list once.

3.6 Find a missing tags number

Losing tags on vehicles is a real life situation and the process of managing these missing tags needs attention. It would be extremely helpful for the maintenance personnel to know the numbers of the vehicles with lost tags.

![Fixed Coupler Normal Coupler](image)

**Figure 7:** Orex wagon layout.

From the AVI system one could obtain information on vehicle number X in a train with a lost tag, but there is unfortunately no indication of which vehicle it is. On the Orex iron export line for example, a captive fleet with certain characteristics exist. This could be used to identify the wagon with a missing tag. On this fleet wagons are coupled in pairs with a permanent coupling. The
distance between a permanent coupling and a normal coupling differ. This is used to identify the wagons connected to one another. The partner wagons number is then used to lookup the wagon number for the wagon with the missing tag. Figure 7 shows the wagon layout on Orex.

4 Closing remarks

It was shown that data patterns embedded in the recorded data can be used to improve reliability of a system. Furthermore it can be used to identify faulty system components, rectify fault conditions which may occur and improve the overall data integrity. Most of these algorithms have been tested in service for more than 12 months now and have proved to perform very well, even in the presence of error prone components. Although the substandard wheel detectors were replaced these algorithms were retained to further ensure the fault tolerant operation of the system.

Acknowledgements

The authors would like to thank Mr. A. du Toit for his assistance during the tests and for his valuable input during the development of the algorithms. To the staff of Ansys, for their willingness to alter the configuration of the system and for the implementation of the code. Mr A. Surkov for his assistance during the manufacture of the upgrade wheel transducer units.

References