

## Method for synchronizing a system by determining a common local time interval

The present invention relates to a method for synchronizing a system. The present invention also relates to an associated controller, system and railway installation.

5       The invention applies to the railroad field, the method and the system being intended for a RAILWAY installation.

Traditionally, railway installations comprise stations provided with platforms, for example intended for passengers to leave and enter trains.

10       It is known to equip the platforms with platform barriers, also called platform screen doors (PSD). The platform barriers are situated on the edge of the platforms and are essentially intended to prevent passenger suicides or accidents.

The platform barriers define an open position and a closed position. Such platform barriers are controlled automatically so as for example to be in the open position only when a vehicle is parked at a platform.

15       Furthermore, some railway installations are provided with an automatic railway traffic control system, also called Communication Based Train Control (CBTC). This is a system based on a communication between vehicles and traffic management devices.

20       When the platforms have platform barriers, the CBTC system provides the articulation between the open position or closed position of the platform barriers and the movement of vehicles. As an illustration, a vehicle approaching a platform provided with platform barriers will only be authorized to enter each station if the platform barriers are locked in the closed position.

Such a CTBC system is therefore a distributed system that must therefore guarantee good safety while allowing the fastest possible response time.

25       There is therefore a need for such a method guaranteeing good safety in a distributed system while allowing the fastest possible response time.

30       The present description describes a method for synchronizing a system including a plurality of redundant processing chains, the system including a global clock pacing the system, each processing chain including a first specific local clock pacing the respective processing chain, and a second local clock able to measure the time, the method including providing a first instant via the global clock for each processing chain, converting the first instant into a first local instant for each processing chain, carrying out the same series of operations on the same input data on each processing chain from the first local instant, incrementing the second specific local clock of each processing chain as long as  
35       the series of operations is implemented, to obtain a second local instant, calculating the local time interval, defined as the difference between the second local instant and the first

local instant, comparing the calculated local time intervals to determine a local time interval shared by a majority of the processing chain.

According to specific embodiments, the synchronizing method includes one or more of the following features, considered alone or according to any technically possible combinations:

- each processing chain includes a local clock able to date the events, the method further including synchronizing the third local clocks of each processing chain on the common local time interval.

- each processing chain includes a third local clock able to date the events, the method further including providing a first time margin, and providing a local instant corresponding to the performance of the same operation on each of the processing chains, dating of said local instant in the third local clock of each processing chain, and setting a lower bound for the set of third local clocks at the performance instant of the operation considered to be the minimum of the local datings from which the first time margin is subtracted.

- the method further includes providing a second time margin, and providing a local instant corresponding to the performance of the same operation on each of the processing chains, dating of said local instant in the third local clock of each processing chain, and setting an upper bound for the set of third local clocks at the performance instant of the operation considered to be the maximum of the local datings to which the second time margin is added.

- the resolution of each clock is greater than or equal to 100 milliseconds.

The present description describes a controller including a plurality of redundant calculating chains, each processing chain including a first specific local clock pacing the respective processing chain, and a second specific local clock measuring the time, the controller being intended to be part of a system including a global clock pacing the system, the controller being able to receive a first instant via the global clock, send the first instant for each processing chain, order each processing chain to convert the first instant into a first local instant, carry out the same series of operations on the same input data on each processing chain from the first local instant, increment the second specific local clock of the processing chain in question as long as the series of operations is implemented, to obtain a second local instant, calculate the local time interval, defined as the difference between the second local instant and the first local instant, compare the calculated local time intervals to determine a local time interval shared by a majority of the processing chain.

According to one specific embodiment, each processing chain includes a local clock able to date the events, the controller being able to synchronize the third local clocks of each processing chain on the common local time interval.

The present description describes a system including a controller as previously described.

According to one particular embodiment, the system is a system for managing the movement of at least one vehicle in a railway installation, the railway installation comprising tracks, at least one vehicle suitable for moving on the tracks and at least one station, the controller being able to ensure the arrival of the vehicle at each station and the departure of the vehicle from each station.

The present description describes a railway installation comprising tracks, at least one vehicle suitable for moving on the tracks, and at least one system as previously described.

Other features and advantages of the invention will appear upon reading the following description of embodiments of the invention, the description being provided as an example only and in reference to the drawings, which are:

- figure 1, a schematic illustration of a railway installation,
- figure 2, a flowchart of an example embodiment of the railway installation,
- figure 3, a flowchart of an example implementation of another example synchronization method,
- figures 4 and 5, flowcharts of an example embodiment of another example synchronization method, and
- figures 6 and 7, a flowchart of an example implementation of another example synchronization method.

A railway installation 10 is shown schematically in figure 1.

The railway installation 10 is for example an urban rail transport installation of the subway or tram type.

The railway installation 10 comprises tracks 12, at least one station, at least one vehicle 16 and a management system 18.

The tracks 12 are able to see to the movement of vehicles 16.

The stations define places where the vehicles 16 can park.

Each station comprises at least one platform.

Generally, a vehicle 16 parks at a platform of a station to allow passengers to move between the vehicle 16 and the platform.

At least some platforms have platform barriers.

The platform barriers are generally installed at the edge of the tracks 12 and are for example formed by doors and barriers separating the platform from the tracks 12. The barriers and the doors are for example glass and have a sufficient height to dissuade an individual from straddling them.

5           The platform barriers are intended to improve safety, in particular by preventing users from stepping onto the tracks 12.

A closed position and an open position are defined for the platform barriers.

In the closed position, the doors are closed and locked.

In the open position, the doors are open and/or unlocked.

10           Furthermore, the platform barriers installed on a platform comprise a platform barrier management module, not shown in figures.

The platform barrier management module is suitable for receiving control signals and sending at least one state signal.

15           In the example of figure 1, the control signals are an open signal and a close signal.

Upon receiving the open signal, the close signal, respectively, the platform barrier management module is suitable for moving the platform barriers into the open position, the closed position, respectively.

20           The state signal is a signal indicating that the doors of the platform barriers are locked in the closed session.

The vehicle 16 is suitable for moving on the tracks 12 and is for example a rail vehicle such as a subway or tram.

The management system 18 is intended to see to the movement of a vehicle 16 in the railway installation 10 independently of the presence or absence of platform barriers.

25           The management system 18 comprises at least one controller 20 installed in each vehicle 16.

The controller 20 includes input modules 22, a computer 24 and output modules 26.

Alternatively, the controller 20 includes a plurality of computers 24.

30           The management system 18 also includes a communication interface 28.

The communication interface 28 is in particular able to see to a two-way communication with the controller 20 on the one hand and the pieces of equipment installed on the platforms on the other hand.

35           The communication interface 28 is also able to ensure communication between each input module 22 and each output module 26.

For example, the communications between the barrier management module and the controller 20 or an element of the controller 20 are carried out by radio via the communication interface 28, and the communications between the input 22 and output 26 modules are carried out by wired connection.

5       The controller 20 is for example the onboard part of a CBTC system.

In this case, the controller is for example the onboard ATC (Automatic Train Control).

The controller 20 comprises an antenna and is suitable for exchanging messages with the barrier management module by using the communication interface 28.

10       Furthermore, the controller 20 comprises a memory comprising software instructions and a processor able to execute at least part of the instructions.

Alternatively, the controller 20 includes a plurality of memories and/or a plurality of processors, each memory comprising software instructions and each processor being able to execute at least part of the instructions.

15       Each input module 22 is able to receive data coming from at least one sensor and produce an input signal corresponding to at least part of the received data.

Each input module 22 includes a plurality of redundant processing chains, the redundancy ensuring security of the obtained result.

20       Each processing chain of the input module 22 includes a first specific local clock 30A.

The first local clock 30A is able to pace the processing chain. This means that each processing chain of the input module 22 performs its operations at the rhythm given by its first local clock 30A.

25       Each processing chain of the input module 22 is thus able to perform operations at a rhythm based on local dates provided by its local clock 30A.

Each processing chain of the input module 22 also includes a second specific local clock 30B.

The second local clock 30B is able date events.

In this way, the second local clock 30B can be described as a safety clock.

30       According to one embodiment, the second local clock 30B is a dedicated physical component.

According to another embodiment, the second local clock 30B is a simple counter incremented by the first local clock 30A.

35       The computer 24 includes a plurality of redundant processing chains, the redundancy ensuring security of the obtained result.

Each processing chain of the computer 24 includes a first specific local clock 32A pacing the respective processing chain.

Thus, each processing chain is thus able to perform operations at a rhythm based on local dates provided by its local clock 32A.

5 Each processing chain of the computer 24 also includes a second specific local clock 32B.

The second local clock 32B is able date events.

In this way, the second local clock 32B can be described as a safety clock.

10 According to one embodiment, the second local clock 32B is a dedicated physical component.

According to another embodiment, the second local clock 32B is a simple counter incremented by the first local clock 32A.

Each output module 26 is able to send orders to an actuator, these orders being an output signal.

15 Each output module 26 includes a plurality of redundant processing chains, the redundancy ensuring security of the obtained result.

Each processing chain of the output module 26 includes a first specific local clock 34A.

20 The first local clock 34A is able to pace the processing chains of the output module 26. This means that each processing chain of the output module 26 performs its operations at the rhythm given by its first local clock 34A.

Each processing chain of the output module 26 is thus able to perform operations at a rhythm based on local dates provided by its local clock 34A.

25 Each processing chain of the output module 26 also includes a second specific local clock 34B.

The second local clock 34B is able date events.

In this way, the second local clock 34B can be described as a safety clock.

According to one embodiment, the second local clock 34B is a dedicated physical component.

30 According to another embodiment, the second local clock 34B is a simple counter incremented by the first local clock 34A.

Unless otherwise indicated:

- the local clocks 30, 32 and 34 are not synchronized with one another.
  - the first local clocks A (used for pacing) of the set of processing chains making up
- 35 a given module are synchronized by a method that is not described. Thus, at any moment, the chains perform the same processing operations.

- the first and second clocks A and B of a given processing chain are synchronized.

Thus, the local clock B (used for dating) is updated using the period of the cycles.

In this context, a local clock refers indifferently to the first local clock 30A, 32A, 34A or the second local clock 30B, 32B, 34B. Thus, the local clock 30, 32 and 34 is able to  
5 pace a processing chain and date events.

Furthermore, the resolution of each clock 30, 32 and 34 is for example greater than or equal to 100 milliseconds.

In some embodiments, the system 18 includes a global clock pacing the system  
18.

10 The controller 20 is able to provide synchronization between its various elements by implementing a synchronization method of the system 18.

Before providing a detailed description of embodiments of the synchronization method of the system 18, we will quickly describe one operating mode of the railway installation 10 corresponding to the state of the art to better highlight the advantages of  
15 the synchronization method embodiments of the system 18.

The case of figure 2 is thus described below.

The boxes identified using reference signs 100, 102, 104, 106, 108, 110, 112, 114, 116, 118 and 120 illustrate the main elements of this example embodiment.

More specifically, the box 100 shows the acquisition of a common signal; the box  
20 102 shows the worst-case response time of an input module 22; the box 104 shows the worst-case lag in sending data between an input module 22 and the computer 24; the box 106 shows the worst-case response time of the computer 24; the box 108 shows the worst-case lag in sending data between the computer 24 and an output module 26; the boxes 110 and 112 schematically shows the asynchrony; the box 114 shows the worst-  
25 case response time of an output module 26; the box 116 shows the actual response time of the controller 20; the box 120 shows the safety margin, the box 118 corresponding to the safety response time of the controller 20, i.e., the sum of the actual response time and the safety margin.

Thus, in the operating mode illustrated by figure 2, it appears that each element of  
30 the controller 20 implements its cycle by using its local clock. The cycles of each element of the controller 20 are therefore carried out asynchronously. The safety response time of the controller 20 (i.e., for which the operation of the railway installation 10 is ensured) is the sum of four elements, which are:

- the actual response time of the controller 20, the response time in particular depending  
35 on the response time of the input modules 22, the response time of the computer 24,

the response time of the output modules 26, the lag of the communication interface 28 and the asynchronies between these different elements of the controller 20,

- the margin to be taken into account corresponding to the sum of the differences between the actual response time and the worst-case response time of one of the input modules 22, the computer 24 and one of the output modules 26,
- the margin due to the difference between the actual transmission time and the worst-case transmission time between an input module 22 and the computer 24, and between the computer 24 and the output module 26, and
- the margin due to the resolution of the clocks 30, 32, 34 used to guarantee the worst-case response times and the worst-case transmission times.

All of the synchronization method embodiments seek to reduce at least one of the four above elements.

One example embodiment of a synchronization method is now described in reference to figure 3.

Figure 3 is a flowchart in which the boxes identified using reference signs 200, 202, 204, 206 and 208 illustrate the main elements of a first example embodiment of a synchronization method.

Such a method makes it possible to reduce the actual response time of the controller 20 by eliminating the asynchrony between the local clocks 30, 32 and 34 while synchronizing them with the global clock.

The synchronization method includes sending a synchronization signal via a dedicated module, for example belonging to the controller. This sending is symbolized by the box 208.

The sending is done several times in order to ensure high redundancy in the reception of the synchronization signal.

The synchronization signal propagates with a low lag time.

According to the described example, the synchronization signal includes an identifier.

Upon receiving the synchronization signal having the expected identifier, each element of the controller 20 carries out the operations to be done.

To guarantee the synchronous operation of the controller 20, each element of the controller 20 finishes its operations and sends the messages before the next synchronization signal arrives.

If the synchronization signal is unavailable, the element of the controller 20 carries out the operations to be done at a slower pace than the pace of the global clock illustrated

by box 202. As a result, synchronization of the system 18 is easy to carry out once the element of the controller 20 is again able to receive the synchronization signal.

In a first such example, the method allows faster synchronization and implementation of the operations to be done by the system 18. This is shown by boxes 204 and 206, which show that the asynchrony is eliminated and the actual response time is more restricted than three cycles.

Furthermore, box 200 illustrates a constant delay making it possible to optimize the response time of the system 18.

A second example embodiment of a synchronization method is now described in reference to figure 4.

Figure 4 is a flowchart in which the boxes identified using reference signs 300, 302, 304, 306, 308, 310, 312, 314, 316 and 318 illustrate the main elements of a second example embodiment of a synchronization method.

It should be noted that the reference signs MVD and square brackets also appear, the reference sign MVD referring to the term "Maximum Validity Duration", and the square brackets indicating either the minimum value on the local basis of the second clock at a given instant, or the maximum value on the local basis of the second clock at a given instant.

The method includes, for each input signal produced by each processing chain of an input module 22, associating an input date expressed in the local clock of the processing chain of the input module 22 having produced the signal (see box 306).

The method also includes sending each input signal from an input module 22 to a computer 24.

As illustrated by box 308, the method also includes generating output signals intended for one or several output modules 26.

An output signal is the result of the application of a first function  $f_1$  specific to the output signal to at least one received input signal.

Thus, the first function  $f_1$  determines the value of the output signal.

For example, the first function  $f_1$  is a logic operation applied to a plurality of received input signals. The "AND" operation or the "OR" operation are specific cases of logic operations.

If a single received input signal is involved in the first function  $f_1$ , another signal separate from a received input signal can be involved in calculating the output signal. Typically, a signal corresponding to another received input signal, but phase-shifted by a known value, can be used. The preceding logic operations are then applied to the

received input signal and to the signal corresponding to another received input signal, but phase-shifted by a known value.

The method also comprises, for each output signal, calculating an expiry duration of the output signal by using a second function  $f_2$  (see box 316).

5        The second function  $f_2$  is deduced from the first function  $f_1$  specific to the output signal and applied on the local input dates associated with each received input signal on which the first specific function  $f_1$  has been applied.

10        In this case, the second function  $f_2$  makes it possible to determine the temporal validity of the output signal calculated by the first function  $f_1$ . The temporal validity is expressed in the form of an expiry date.

According to one specific case, the second function  $f_2$  is a set operation applied to the dates.

For a case with two signals having an acquisition date of  $d1$  and  $d2$ , respectively, the following operations can be considered for the second function  $f_2$ :

- 15            • union ( $d1$ ,  $d2$ );
- intersection ( $d1$ ,  $d2$ );
- min ( $d1$ ,  $d2$ );
- max ( $d1$ ,  $d2$ ), and
- $d2$ + predefined duration.

20        As a specific illustration, a first binary input signal  $E1$  and a second binary input signal  $E2$  are acquired.

The binary output signal denoted  $S$  is calculated by applying the first function  $f_1$  such that  $S = E1 \text{ AND } E2$ , the first function  $f_1$  then being the logic operation that associates the value of  $A \text{ AND } B$  with two binary properties  $A$  and  $B$ .

25        Furthermore, for this output signal, an expiry duration of the output signal is calculated by using a second function  $f_2$ .

In this particular case, the second function  $f_2$  is expressed as follows:

$$f_2 = \min (\text{acquisition date of } E1, \text{ acquisition date of } E2) + 100 \text{ ms.}$$

30        The method also comprises sending each output signal from the computer 24 to the output module 26.

The method also comprises estimating, via each processing chain of each output module 26, the shift between the local clock 34 of the processing chain of the output module 26 and the local clock 30 of at least one input module 22.

35        Furthermore, the estimate of the shifts is updated periodically, and extrapolated between two updates.

The estimate is, for example, done by synchronization between the output module 26 and at least one input module 22 (see box 300). For example, the output module 26 sends a signal containing the sending date according to its local clock 34 to determine the local time of the local clock 30 of the input module 22. The input module 22 in turn sends a signal containing, *inter alia*, the reception date according to its local clock for the signal previously sent by the output module 26. The shift corresponds to the difference between the reception date by the input module 22 of the signal sent by the output module 26 and the sending date by the output module 26 of the signal sent to the input module 22, compensated by the worst-case transmission duration of the signal sent by the output module 26.

In other words, the output modules 26 estimate the shift between their local time base and the time base of each input module 22. This operation requires ensuring a worst-case transmission time between the input module 22 and the output module 26.

As shown schematically by boxes 300, 310, 312, 316 and 318, the method then includes, for each expiry duration of the output signal, an operation for changing time base from the clock of the input module 22 (more specifically, see box 316) to the clock of the output module 26 (see box 318), the result of which is advantageously compensated by a safety margin by the output module 26 (see box 312).

In the illustrated example, the safety margin is the worst-case offset obtained in the estimating step. The safety margin is then specific to each output module 26.

According to another embodiment, the safety margin is less than or equal to the longest transmission duration of a datum between an input module 22 and an output module 26.

As shown schematically, figure 5 corresponding to the worst-case scenario, the safety margins related to the transmission time between an input module 22 and the computer 24, and transmission between the computer 24 and the output module 26, is reduced.

In figure 5, the box bearing reference 400 corresponds to a downgraded transfer time; the box 402 to a downgraded acquisition; the box 408 to stopping refreshing of the data when the expiry time of the output signal has expired; the box 410 shows the worst-case lag time of the output module 26; the box 404 to the actual response time of the controller 20; the box 412 shows the safety margin, the box 406 corresponding to the safety response time of the controller 20, i.e., the sum of the actual response time and the safety margin.

A third example embodiment of a synchronization method is now described in reference to figures 6 and 7.

The method eliminates the synchronization of the first and second clocks A and B of a given processing chain. The clock used for pacing A is updated using the period of the cycle of the processing chain. The clock used for dating is updated at a much lower periodicity, the latter no longer being constrained by the duration of the processing operations to be done in each cycle.

The method defines an additional clock on each processing chain of each module, called clock of the module.

According to one embodiment, the clock of the module is a dedicated physical component.

According to another embodiment, the clock of the module is a simple counter incremented by the local clock B.

The synchronization method includes providing a first instant via the global clock for each processing chain.

Alternatively, the provision of the first instant is done by one of the processing chains.

In the illustrated example, only two processing chains are considered. The local clocks of each chain of the considered module are illustrated by box 504 and box 506, respectively.

The method also includes converting the first instant into a first local instant for each processing chain.

The method also comprises implementing the same series of operations on the same input data on each processing chain from the first local instant.

The local clock B specific to each processing chain is next incremented as long as the series of operations is carried out, to obtain a second local instant.

The method then includes calculating a local time interval defined as the difference between the second local instant and the first local instant.

The second local time intervals thus obtained are shown by box 500 and box 502, respectively.

The calculated local time intervals are next compared to determine a local time interval shared by a majority of the processing chains.

According to one embodiment, the comparison is carried out by a voting mechanism.

Alternatively, the common local time interval is the maximum of the local time intervals.

According to another embodiment, the common local time interval is the minimum of the local time intervals.

According to one alternative, the common local time interval is the median of the local time intervals.

According to another specific case, the common local time interval is the mean, in particular the arithmetic mean, of the local time intervals.

Each processing chain increments its clock of the module by a quantity equal to the common local time interval previously determined. The clock of the module is thus synchronized on both processing chains. The resulting clock of the module is represented by box 508.

The clock of the module is used by each processing chain to date the events.

In this case, it indeed appears that the temporal error in assigning a date to any instant T1 depends on the period between two incrementation instants of the clock of the module 510.

Such a method makes it possible to reduce the margin due to the resolution of the global clock, since the proposed method does not use the global clock to date an instant for all of the second local clocks 30B, 32B and 34B, the period between 2 incrementation instants of the clock of the module being able to be shorter than the period of the global clock.

According to another embodiment illustrated by figure 7 (see in particular boxes 600, 602, 604, 606, 608, 610), the method includes, in addition to the method previously described, calculating an estimate of the clock of the module between the two update instants of the clock of the module.

Between these two instants, at all instants, each chain increments its module clock.

At all instants, each processing chain can estimate the minimum value of the clock of the module, for example by subtracting a safety margin from the current value of the clock of the module. Likewise, each processing chain can estimate the maximum value of the clock of the module, for example by adding a second safety margin to the current value of the clock of the module. The resulting clock of the module is represented by box 608.

In other words, the method includes providing a first time margin. It should be noted that this margin is either a constant, or calculated as a function of the duration having elapsed between the moment of the operation in question and the time elapsed since the last resynchronization of the clocks of the module. The method also comprises providing a local instant corresponding to the performance of the same operation on each of the processing chains of the module in question, dating said local instant in the clock of the module of each processing chain of the module in question, and setting a lower bound

for the set of clocks of the module at the performance instant of the operation considered to be the minimum of the local datings from which the first time margin is subtracted.

According to another embodiment or additionally, the method includes providing a second time margin, providing a local instant corresponding to the performance of the same operation on each of the processing chains of the module in question, dating said local instant in the local clock of the module of each processing chain of the module in question, and setting an upper bound for the set of clocks of the module at the performance instant of the operation considered to be the maximum of the local datings to which the second time margin is added.

The safety margins are calculated based on the precision of the local instants on each of the chains and the resolution of the local clock.

In this case, it indeed appears that the temporal error in assigning a date to any instant is a function of the resolution of the local clock of each chain and resynchronization performances between each chain (see box 610).

All of the proposed synchronization method embodiments therefore make it possible to reduce at least one from among the actual response time of the controller 20, the margin related to the response time of the controller, the margin due to the transmission times and the margin due to the resolution of the global clock.

By combining the embodiments, when technically possible, new embodiments of the synchronization method are obtained, these new embodiments having improved performance levels in terms of speed while guaranteeing a response of the system 18 fulfilling the safety criteria.

Furthermore, the proposed system 18 applies to other fields of automation for distributed systems, as long as a controller is present having characteristics similar to the controller 20 previously described.

15  
CLAIMS

1.- A method for synchronizing a system (18) including a plurality of redundant processing chains, the system (18) including a global clock pacing the system (18), each processing chain including a first specific local clock (30A, 32A, 34A) pacing the respective processing chain, and a second local clock (30B, 32B, 34B) able to measure the time, the method including:

- providing a first instant via the global clock for each processing chain,
- converting the first instant into a first local instant for each processing chain,
- implementing the same series of operations on the same input data on each processing chain from the first local instant,
- incrementing the second local clock (30B, 32B, 34B) specific to each processing chain is next incremented as long as the series of operations is carried out, to obtain a second local instant,
- calculating a local time interval defined as the difference between the second local instant and the first local instant,
- comparing the calculated local time intervals to determine a local time interval shared by a majority of the processing chains.

2.- The method according to claim 1, wherein each processing chain includes a local clock able to date the events, the method further including:

- synchronizing the third local clocks of each processing chain on the common local time interval.

3.- The method according to claim 1 or 2, wherein each processing chain includes a local clock able to date the events, the method further including:

- providing a first time margin, and
- providing a local instant corresponding to the performance of the same operation on each of the processing chains,
- dating said local instant in the third local clock of each processing chain, and
- setting a lower bound for the set of third local clocks at the performance instant of the operation considered to be the minimum of the local datings from which the first time margin is subtracted.

4.- The method according to any one of claims 1 to 3, wherein the method further includes:

- providing a second time margin, and
- providing a local instant corresponding to the performance of the same operation on each of the processing chains,
- dating said local instant in the third local clock of each processing chain,
- setting an upper bound for the set of third local clocks at the performance instant of the operation considered to be the maximum of the local datings to which the second time margin is added.

5.- The method according to any one of claims 1 to 4, wherein the resolution of each clock (30A, 32A, 34A; 30B, 32B, 34B) is greater than or equal to 100 milliseconds.

6.- A controller (20) including a plurality of redundant processing chains, each processing chain including a first specific local clock (30A, 32A, 34A) pacing the respective processing chain, and a second specific local clock (30B, 32B, 32C) measuring the time, the controller (20) being intended to be part of a system (18) including a global clock pacing the system (18), the controller (20) being able to:

- receive a first instant via the global clock,
- send the first instant for each processing chain,
- command each processing chain to:
  - - convert the first instant into a first local instant,
  - - implement the same series of operations on the same input data on each processing chain from the first local instant,
  - - increment the second local clock (30B, 32B, 34B) specific to the processing chain in question as long as the series of operations is carried out, to obtain a second local instant,
  - calculate a local time interval defined as the difference between the second local instant and the first local instant,
  - compare the calculated local time intervals to determine a local time interval shared by a majority of the processing chains.

7.- The controller (20) according to claim 6, wherein each processing chain includes a local clock able to date the events, the controller (20) being able to synchronize the third local clocks of each processing chain on the common local time interval.

8.- A system including a controller (20) according to claim 6 or 7.

5 9.- The system according to claim 8, wherein the system (18) is a system for managing the movement of at least one vehicle (16) in a railway installation (10), the railway installation (10) comprising tracks (12), at least one vehicle (16) suitable for moving on the tracks (12) and at least one station, the controller (20) being able to ensure the arrival of the vehicle (16) at each station and the departure of the vehicle (16) from each station.

10 10.- A railway installation (10), comprising:  
- tracks (12),  
- at least one vehicle (16) suitable for moving on the tracks (12), and  
- at least one system (18) according to claim 8 or 9.