

(19)



(11)

EP 2 196 972 A1

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:

16.06.2010 Bulletin 2010/24

(51) Int Cl.:

G08G 1/01 (2006.01)(21) Application number: **08171580.7**(22) Date of filing: **12.12.2008**

(84) Designated Contracting States:

**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT
RO SE SI SK TR**

Designated Extension States:

AL BA MK RS(71) Applicant: **Nederlandse Organisatie voor
toegepast****-natuurwetenschappelijk onderzoek TNO
2628 VK Delft (NL)**

(72) Inventors:

- **Papp, Zoltan**
2625 LL Delft (NL)

- **Doodeman, Gerardus Johannes Nicolaas**
5501 DM Veldhoven (NL)

- **Nelisse, Martin Willem**
2624 JS Delft (NL)

- **Sijs, Joris**
2595 GM Den Haag (NL)

- **Theeuwes, Johannes Adrianus Cornelis**
5612 NE Eindhoven (NL)

- **Driessen, Bart**
2628 VK Delft (NL)

(74) Representative: **Hatzmann, Martin****Vereenigde
Johan de Wittlaan 7
2517 JR Den Haag (NL)**(54) **Object tracking system, object infrastructure provided with object tracking system and method for tracking objects**

(57) An object tracking system is described comprising

- a plurality of sensor nodes (10) that each provide a message (D) indicative for an occupancy status of a detection area of an object infrastructure monitored by said sensor node,

- a map-integrator with

- an object database facility comprising state information of objects present at the object infrastructure, and a database updating facility (DBU) for updating the database facility (32, 34) on the basis of messages (D) provided by the sensor nodes.

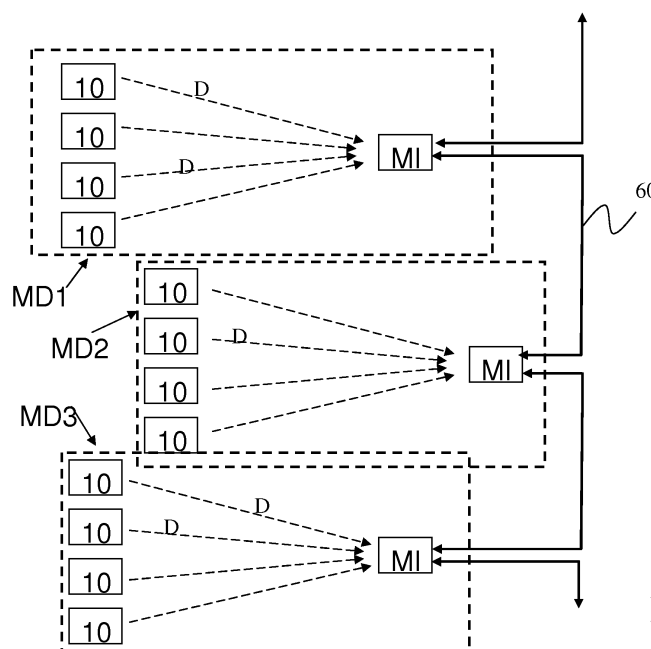


Figure 3

Description

BACKGROUND OF THE INVENTION

Field of the invention

[0001] The present invention relates to an object tracking system.
The present invention relates to an object infrastructure provided with an object tracking system.
The present invention further relates to a method for tracking objects.

Related Art

[0002] In "Measuring Traffic", Statistical Science, 2007, Vol. 22, No. 4, pp. 581-597, Peter J. Bickel et al. describe a traffic performance measurement system, PeMS, that functions as a statewide repository for traffic data gathered by thousands of automatic sensors. It has integrated data collection, processing and communications infrastructure with data storage and analytical tools. This system provides for global information relating to the average traffic speed and the traffic density.

[0003] There is a need for a system that provides information relating to the state of individual traffic participants. This information can be used in advanced cruise controllers for traffic management or for access control for example.

SUMMARY OF THE INVENTION

[0004] The object tracking system according to the present invention comprises

- a plurality of sensor nodes that each provide a message indicative for an occupancy status of a detection area of an object infrastructure monitored by said sensor node,
- a message interpretator including an object database facility with state information of objects present at the object infrastructure, and a database updating facility for updating the database facility on the basis of messages provided by the sensor nodes.

[0005] In the object tracking system according to the present invention objects can be tracked with relatively simple and cheap means. It is sufficient that the sensor nodes sense an occupancy state, i.e. whether a detection area associated with the sensor node is occupied by an object or not and that they merely provide a message that indicates whether the occupancy state is changed. The relatively cheap and simple construction of the sensor nodes contributes to an economically feasible application in object tracking systems for large object infrastructures. Optionally, the message may additionally include the value of the occupancy state after the change was detected.

[0006] In an embodiment the database update facility comprises

- an association facility for associating the messages provided by the sensor nodes with the state information present in the object data base facility,
- a state updating facility for updating the state information on the basis of the messages associated therewith.

The association facility selects for which state information the received messages are relevant, and provides the selected messages to the state updating facility. In this way the state updating facility can operate more efficiently, than in case no selection takes place.

[0007] The present invention is in particular suitable for tracking vehicles. For these objects, suitable sensor elements to be used in the sensor nodes are for example magneto restrictive sensors. These sensors determine whether their associated detection area is occupied by detection of a perturbation of the earth magnetic field. Alternatively, magnetic loop sensors, may be used, which detect a change of inductance caused by the presence of ferromagnetic material.

[0008] Preferably each sensor node is provided with a wireless transmission facility that transmits the preprocessed data, e.g. the occupance status or an indication of a change thereof to a data to a receiver facility coupled to the message interpreter. The absence of wiring towards the message interpreter makes the installation easier and cost effective. Furthermore it is attractive if the sensor nodes provide their message at an event basis, e.g. if a perturbation of the earth magnetic field exceeds a threshold value. This reduces communication load of the message interpreter and minimizes power consumption of the sensor nodes.

In an embodiment a set of sensor elements may be clustered in a sensor module. For example the sensor module may be a camera that monitors a part of the object infrastructure, wherein each photosensitive element of the camera serves as a sensor element of the object tracking system. A camera may be used for example if a perturbation of the earth

magnetic field can not be measured. This is the case for example if (parts of) the infra structure comprises metal components e.g. a bridge. It is not necessary that the detection areas of the sensor elements are complementary. The detection areas may overlap, or spaces may exist between the detection areas, but it is required that the detection area of the sensor be smaller than the object to be tracked. Ideally the sensor elements are point detectors.

[0009] The sensor nodes can be either randomly distributed over the object infrastructure or placed in a pattern optimized for the object tracking problem in hand.

[0010] In an embodiment the object tracking system comprises a plurality of system modules, each module comprising a respective subset of the plurality of sensor nodes for monitoring a respective section of the object infrastructure and a respective message interpreter, the object tracking system has a communication facility for enabling system modules of mutually neighboring sections to exchange state and detection information. In this way the object tracking system can be easily expanded if required. A new system module need only to communicate with the system modules arranged for neighboring sections. Neighboring sections may be arranged in one dimensional scheme, e.g. in case of a narrow road. For example if a certain road is already provided with an object tracking system, it is sufficient to provide for a communication facility between the system module for the last section of said object tracking system and the new system module for the appended section. In case of infrastructures for unconstrained object movements the new module may communicate with other modules neighboring in various directions. As the system modules merely need to exchange state information and object-detection information (i.e. the unprocessed sensor signals) in a limited subarea of the respective sections, the amount of communication between the system modules is modest resulting in a scalable object tracking system.

[0011] The association facility associates the messages provided by the sensor nodes or neighboring system modules with the state information present in the object data base facility. In other words the association facility determines the probability that the detections are caused by a particular object for which state information is present in the object data base facility. If the messages cannot be associated with state information of an already identified object here or in the neighboring system module, a new entry may be added to the database. Alternatively, the entry for the new object may be added by a separate procedure. For example the object infrastructure may have an access with an object identification facility that provides for an identification of every object that enters the infrastructure.

[0012] The individual sensor nodes do not need to provide other information than an occupancy status of their associated detection area. However, optionally the sensor node may associate its own signal with a color, shape, or other signature of the tracked objects to facilitate or obviate association by the message interpreter. An association facility for associating the detection signals obtained on asynchronous basis with state information of a particular object may be based on one of the following methods.

- Gating,
- Nearest Neighbor (NN),
- (Joint) Probabilistic Data Association ((J)DPA),
- Multiple Hypothesis Tracker (MHT), and
- MCMCDA.

[0013] Gating comprises forming a gate around the predicted measurement of an object. The size and shape of the gate are chosen in such a way that unlikely messages are precluded to be associated with this particular object-track.

The method determines a statistical, quadratic distance d_{oi}^2 from object i. A measurement y is associated with the

state-vector x^{oi} of object i if $d_{oi}^2 \leq G$, with G some constant threshold and d_{oi}^2 equal to:

Various methods can be used for finding the Threshold G. However, this data association method is not suitable for associating event based messages. Moreover problems arise when two gates overlap.

[0014] The Nearest Neighbor method also uses a gate, but it can handle overlapping gates. The sum of all possible combinations to associate a certain measurement to a certain track is analyzed. The chosen combination associates the most measurements to a track for a minimum sum of distances.

[0015] A (Joint) Probabilistic Data Association method is described in Multitarget-Multisensor Tracking: Principles and Techniques. YBS, 1995, by Y. Bar-Shalom and R. Li, for example. The (Joint) Probabilistic Data Association methods (J)PDA is unsuitable for event-based sampling because it assumes that one target can give rise to at most one measurement and one measurement is a result of at most one object. This cannot be assumed with event based sampling. An extension to the JPDA can be found in O. Songhwai, S. Sastry, and L. Schenato, "A Hierarchical Multiple-Target Tracking Algorithm for Sensor Networks," in Proc. of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005. Where the original JPDA assumes that a probability density function (PDF) of state-vector $x_{k+1|k}$ is a Gaussian function, in the extended JPDA the PDF can be a non-Gaussian function.

[0016] A Multiple Hypothesis Tracker (MHT) allows that the state-vector of a single object can has multiple tracks. This method resembles to the Particle filter as described in B. Ristic, S. arulampalam, and N. Gordon, "Beyond the Kalman filter: Particle filter for tracking applications", 2002. Therein, each state is estimated by simulating N states with each a different probability. It is a drawback of this method that it requires a high computational power.

[0017] A further data association method is the Markov chain Monte Carlo data association (MCMCDA). All observations are used to classify and cluster them. To that end the whole set of observations is divided into a number of partitions represented by the set w . This is done n_{mc} times resulting in n_{mc} sets of w , i.e. possible partitions. The set of w with the highest probability, given the number of objects in the previous sample instant, is chosen and the state-vectors of the tracks a are updated according the partitioned observation. The computational time can be decreased by not using the total history of observations, but by using a moving horizon. A downside of this method is that each observation can belong to at most one object and, making this method unsuitable for event-based state-estimation.

[0018] Although these data association methods can be applied to associate messages obtained by (a) synchronous sampling, they are not suitable for association messages obtained by event-based sampling. For then the main issue with present association methods is that they assume either that one target results in at most one observation or that one observation comes from at most one object. According to a preferred embodiment of the present invention messages received from the sensor nodes are first associated with a particular object. In other words it is determined which object most probably caused the observation by the sensor resulting in the message. Subsequently the state of said object is (re-)estimated using that message. The estimation is based on the latest observation and the current state of the said object. The state of the object that is most probable in view of all available observations is calculated. The state of an object can also be estimated at a point in time later than the last message, but before a new message has arrived. In that case the error covariance matrix is bounded, as it is known that the state change of the object must be within the detection boundaries of the sensor node.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] These and other aspects are described in more detail with reference to the drawing. Therein:

Figure 1 shows a first view of an embodiment of an object infrastructure provided with an object tracking system according to the invention,

Figure 2 shows a second view of an embodiment of an object infrastructure provided with an object tracking system according to the invention,

Figure 3 shows another view of an embodiment of an object tracking system according to the invention,

Figure 4 schematically shows a part of the object infrastructure that is provided with a plurality of sensor nodes,

Figure 5 schematically shows a signal flow in a sensor node,

Figure 6 schematically shows a possible hardware implementation of a sensor node,

Figure 7 shows a possible method carried out by a sensor node,

Figure 8 shows a signal flow in a message interpreter,

Figure 9 shows a possible hardware implementation of a message interpreter,

Figure 10 shows neighboring infrastructure regions, with specially handled subregions marked ,

Figure 11 shows an overview of a method carried out by the message interpreter,

Figure 12 shows a first detail of the method of Figure 11,

Figure 13 shows a second detail of the method of Figure 11,

Figure 14 shows an example of an object to be detected at a reference position and orientation and at a different position and orientation,

Figure 15 shows a definition of a set S and the equidistant sampled set Λ ,

Figure 16 shows detection of an object at multiple detection points,

Figure 17 shows a definition of the set O_n of possible positions o_k^i for a single detection point,

Figure 18 shows a definition of the set ON of possible positions o_k^i for multiple detection points,

Figure 19 shows a derivation of $ON(q)$ given 2 detections and 2 different samples of q ,

Figure 20 shows (Left) determination of $-\Lambda$, (right) the object's possible position set $^{\wedge}On$ given d_n and q ,

Figure 21 shows (left) the mean of all Gaussians from $f(o|z_1, \theta)$ and $f(o|z_2, \theta)$; (right) The selection of means of the Gaussians from $f(o|z_1, \theta)$ and $f(o|z_2, \theta)$, of which their mean $\hat{o}_{ni}$ is close or in $C_N(\theta)$,

Figure 22 shows an association result with event-based data-association,

Figure 23 shows an association result with Nearest Neighbor data-association,

Figure 24 shows time sampling of a signal $y(t)$,

Figure 25 shows event sampling of a signal $y(t)$,

Figure 26 shows event sampling: Send-on-Delta,

Figure 27 shows the Gaussian function,

Figure 28 shows a top view of the Gaussian function,
 Figure 29 shows an approximation of $\Lambda_{Hke}(y_n)$ as a sum of Gaussian functions,
 Figure 30 shows position, speed and acceleration of a simulated object,
 Figure 31 shows a position estimation error for various methods,
 Figure 32 shows a speed estimation speed for various methods,
 Figure 33 shows a factor of increase in estimation error after z_{ke} , or $\bar{y}_{ka}$.

DETAILED DESCRIPTION OF EMBODIMENTS

[0020] In the following detailed description numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be understood by one skilled in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, and components have not been described in detail so as not to obscure aspects of the present invention.

[0021] The invention is described more fully hereinafter with reference to the accompanying drawings, in which embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. In the drawings, the size and relative sizes of layers and regions may be exaggerated for clarity.

[0022] It will be understood that when an element or layer is referred to as being "on", "connected to" or "coupled to" another element or layer, it can be directly on, connected or coupled to the other element or layer or intervening elements or layers may be present. In contrast, when an element is referred to as being "directly on", "directly connected to" or "directly coupled to" another element or layer, there are no intervening elements or layers present. Like numbers refer to like elements throughout. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

[0023] It will be understood that, although the terms first, second, third etc. may be used herein to describe various elements, components, and/or sections, these elements, components, and/or sections should not be limited by these terms. These terms are only used to distinguish one element, component or section from another element, component, and/or section. Thus, a first element, component, and/or section discussed below could be termed a second element, component, and/or section without departing from the teachings of the present invention.

[0024] Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. It will be further understood that terms, such as those defined in commonly used dictionaries, should be interpreted as having a meaning that is consistent with their meaning in the context of the relevant art and will not be interpreted in an idealized or overly formal sense unless expressly so defined herein. All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety. In case of conflict, the present specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

[0025] Figure 1 and 2 show a first and a second view of an embodiment of an object infrastructure 80 provided with an object tracking system. The object infrastructure is intended to allow stationary and/or moving objects 70 thereon, e.g. a road or a parking place. The object infrastructure may be part of a public or private space, e.g. a recreational park.

[0026] The object tracking system comprises a plurality of sensor nodes 10 that each provide a message indicative for an occupancy status of a detection area of the object infrastructure monitored by said sensor node 10. As shown therein the sensor nodes are randomly distributed over the object infrastructure.

[0027] The object tracking system comprises a message interpretator MI, each comprising an object database facility, an association facility and a state updating facility. Each message interpretator is responsible for handling messages D from a respective section 80A, 80B, 80C, 80D of the object infrastructure 80.

[0028] Figure 3 is another schematic view of the object tracking system. Figure 3 shows how sensor nodes 10 transmit (detection) messages to a message interpreter MI in their neighborhood. The message interpreters MI may also communicate to each other via a communication channel 60 to indicate that an object crosses a boundary between their respective sections and to exchange a status of such an object. As shown in Figure 3, the object tracking comprises a plurality of system modules MD1, MD2, MD3. Although three modules are shown in this example, any number of system modules is possible, dependent on the application. For example for an isolated object infrastructure, e.g. an intersection of roads a single module may be applicable, while on a long road thousands of modules may be present. Each module MD 1, MD2, MD3 comprises a respective subset of the plurality of sensor nodes 10 for monitoring a respective section of the object infrastructure and a respective message interpreter MI. The object tracking system further has a communication facility 60 for enabling system modules MD1, MD2, MD3 of mutually neighboring sections to exchange state information. In the embodiment shown, messages from the sensor nodes are directly transmitted to a message interpreter. Alternatively, the sensor nodes may form a network that routes the messages to the message interpreters. In that case the transmitters may have a short transmission range.

[0029] Figure 4 schematically shows a part of the object infrastructure that is provided with a plurality of sensor nodes j having position c_j . The sensor nodes have a detection area with radius R . An object i is present at the infrastructure having a position (v_x^i, v_y^i) . In this case if the object substantially covers the detection area the sensor node indicates that the detection area is occupied as indicated in gray. Otherwise the sensor node indicates that the detection area is not occupied (white). In practice the fraction of the detection area that should be covered before an occupied status is detected may deviate from the above-mentioned 50% depending on the type of object.

[0030] Figure 5 schematically illustrates the signal flow for the sensor node 10, having sensor element 12, a processing unit 14 (with memory), and a radio link 16.

[0031] The sensor element 12 is capable of sensing the proximity of the objects to be tracked. The processing unit 14 determines if an object is present or absent on the basis of the signals from the sensor element 12. If an occupancy status of the detection area of the sensor changes, the processing unit 14 initiates a transmission of a message D indicating the new occupancy status. In an alternative embodiment the message may include a time stamp indicative of the time t at which the new occupancy status occurred. Alternatively the sensor nodes may transmit occupancy status information on a periodical basis for example. However, an event-based transmission enables a lower power use. The message D sent should reach at least one message interpreter MI . A concrete implementation of the sensor node 10 is used for road object tracking: in this case the sensor element 12 is a magnetoresistive component, which measures the disturbance on the earth magnetic field induced by the objects. Alternatively, a magnetic rod or loop antenna may be used to detect the occupancy by an object.

[0032] Figure 6 shows a possible implementation of the hardware involved for the sensor node 10 of Figure 5. The sensor element 12 is coupled via an A/D converter 13 to a microcontroller 14 that has access to a memory 15, and that further controls a radio transmitter 16 coupled to an antenna 17.

[0033] Figure 7 schematically shows a method performed by a sensor node to generate a message indicative for occupancy status of a detection area of the sensor node.

[0034] Starting (Step S1: initialization) from an off-state of the sensor node, input from the A/D converter is received (Step S2). In a next step S3, offset is removed from the sensed value.

[0035] In step S4 it is determined whether the occupancy state of the detection area as reported by the last message transmitted by the sensor node was ON (object was alpresent in the detection range) or OFF (no object present in the detection range). This occupancy state is internally stored in the sensor node.

[0036] In the first case, program flow continues with step S5. In the second case processing flow continues with step S9. In step S5 it is determined whether a signal value v obtained from the A/D converter, and indicative for an occupied status of the detection area is below a first predetermined value T_L . If this is not the case program flow continues with step S2. If however the value is lower than said first predetermined value then program flow continues with step S6. In step S6 it is verified whether the signal value v remains below the first predetermined value T_L for a first predetermined time period. During step S6 the retrieval of input from the A/D convertor is continued. If the signal value v returns to a value higher then said predetermined value T_L before the end of said predetermined time-period then processing flow continues with step S2. Otherwise the value for the occupancy state is internally saved as unoccupied in step S7, and a message signaling this is transmitted in step S8.

[0037] In step S9 it is determined whether the signal value v obtained from the A/D converter, and indicative for an occupied status of the detection area is above a second predetermined value T_H . The second predetermined value T_H is higher than the first predetermined value T_L . If this is not the case program flow continues with step S2. If however the value is higher than said second predetermined value T_H then program flow continues with step S10. In step S 10 it is verified whether the signal value v remains above the second predetermined value T_H for a second predetermined time period, which may be equal to the first predetermined time period. During step S10 the retrieval of input from the A/D convertor is continued. If the signal value v returns to a value lower then said predetermined value T_H before the end of said predetermined time-period then processing flow continues with step S2. Otherwise the value for the occupancy state is internally saved as occupied in step S11, and a message signaling this is transmitted in step S 12.

[0038] Figure 8 illustrates the signal flow in a message interpreter MI . As shown in more detail in Figure 8, a radio receiver 20 receives the binary "object present" signals D (optionally with timestamp) from the sensor nodes 10 via the radio link and runs a model based state estimator algorithm to calculate the motion states of the objects individually (i.e. each real world object is represented in the message interpreter). The sensor density may be chosen dependent on the required accuracy of the estimation. If a very accurate object tracking is desired multiple sensors per object area may be present.

[0039] The message interpreter MI has an object database facility 32, 34 that comprises state information of objects present at the object infrastructure.

[0040] The message interpreter MI further has a sensor map 45describing the spatial location of the sensor nodes 10. Alternatively, the sensor nodes may transmit their location, or their position could even be derived by a localization method for wireless sensor networks.

[0041] The message interpreter MI further has an association facility 40 for associating the messages D provided by

the sensor nodes 10 with the state information present in the object data base facility 32, 34. The association facility 40 may associate the messages received with state information for example with one of the methods Gating, Nearest Neighbor (NN), (Joint) Probabilistic Data Association ((J)DPA), Multiple Hypothesis Tracker (MHT) and the MCMCDA.

[0042] The message interpreter further has a state updating facility 50 for updating the state information on the basis of the messages D associated therewith by the association facility 40. Once the messages D are associated with a particular object the state of that object in a local object data base is updated by the state updating facility 50.

[0043] The association facility 40 and the state updating facility 50 together form a database updating facility DBU.

[0044] In the embodiment shown a global map builder 65 may exchange this updated information with global map builders of neighboring message interpreters via network interface 60 (wired or wireless) and to receive close to border detections. Other uses are also possible (e.g. to calculate system level features like object density and average velocity, but these are independent from the motion state estimation). , to exchange the motion state of crossing objects.

[0045] A message interpreter MI, shown in Figure 9, consists of a radio receiver 20, coupled to antenna 22, a processing unit 24 (with memory 28) and a network interface 65, as well as a real-time clock 26. In other embodiments a real-time clock may be part of the sensor node, and the sensor node may embed a time-stamp indicative for time at which an event was detected in the message. However, generally a message interpreter will have a more reliable clock, as it can be more reliably synchronized with a reference clock. The network interface 65 couples the message interpreter MI via the communication channel 60 to other message interpreters.

[0046] In the embodiment shown the microcontroller 24 of Figure 9 processes the received messages D. The memory 28 stores the local and global object map and the sensor map as well as the software for carrying out the data association and state estimation tasks. In an alternative embodiment separate memories may be present for storing each of these maps and for storing the software. Likewise dedicated hardware may be present to perform one or more of these tasks.

[0047] There is no communication or any other interaction between the objects tracked. The result of the processing (i.e. the estimation of the motion states of all sensed objects) is present in the memory of the message interpreters in a distributed way.

Message interpreters may run additional (cooperative) algorithms to deduct higher level motion characteristics and/or estimate additional object characteristics (e.g. geometry).

[0048] For applications in relative small area, e.g. a parking place, the object tracking system may comprise only a single message interpreter MI. In that case the global map builder is superfluous, and local object map is identical to the global object map.

[0049] In the embodiment shown in Figure 3, each message interpreter MI for a respective module comprises hardware as described with reference to Figure 8 and 9.

[0050] Operation of the message interpreter is further illustrated with respect to Figures 10-13

[0051] Figure 10 schematically shows a part of an object infrastructure having sections R_{j-1} , R_j , R_{j+1} . By way of example it is presumed that an object moves in a direction indicated by arrow X from R_{j-1} , via R_j , to R_{j+1} .

[0052] Figure 11 shows an overview of a method for detecting the object performed by the message interpreter for section R_j , using the messages obtained from the sensor nodes.

[0053] In step S20 the method waits for a message D from a sensor node. At the moment that a message D is received, program flow continues with step S21, where the time t associated with the message is registered. The registered time t associated with the message may be a time-stamp embedded in the message or a time read from an internal clock of the message interpreter.

[0054] In step S22, it is verified whether the detection is made by a sensor node in a location of section R_j that neighbors one of the neighboring sections R_{j-1} or R_{j+1} . If that is the case, then in step S23 the event is communicated via the communication network interface to the message interpreter for that neighboring section. In step S24 it is determined which object O in the object data base facility is responsible for the detected event. An embodiment of a method used to carry out step S24 is described in more detail in Figure 12. After the responsible object O is identified in Step 25, i.e. an association is made with existing object state information, it is determined in Step 26 whether it is present in the section R_j . If that is the case, control flow continues with Step S27. Otherwise control flow returns to step S20, where the state of object O is estimated. A procedure for estimating the state is described in more detail with reference to Figure 13. In step S28 it is determined whether the state information implies that the object O has a position in a neighboring section R_{j-1} or R_{j+1} . In that case the updated state information is transmitted in step S29 to the message interpreter for the neighboring section and control flow returns to step S20. Otherwise the control flow returns immediately to Step S20.

[0055] A method to associate a message D at time t, with an object O is now described in more detail with reference to Figure 12.

[0056] In a first step S40, an object index i is initialized (e.g. $i=1$). In a next step S41, the current state known for the object with that index i is retrieved from the object database facility. In the next step S42 a probability is determined that the object O caused the detection reported by the message D at time t. The object index i is incremented in step S43 and if it is determined in step S44 that i is less than the number of objects, the steps S41 to S43 are repeated. Otherwise

in step S45 it is determined which object caused the detection reported by the message D at time t with the highest probability. In step S46 the index of that object is returned as the result of the method.

[0057] A method to estimate (update the present estimation of) the state of an object is now described in more detail with reference to Figure 13.

[0058] In step S60 the messages $D_1, \dots, D_n$ associated with object O are selected.

[0059] In step S61 a probability density function is constructed on the basis of the associated messages $D_1, \dots, D_n$.

[0060] In step S62 the current state S_0 and time to for object O is retrieved from the object database.

[0061] In step S63 it is determined whether the time for which the state S of the object O has to be determined is greater than the time to associated with the current state S_0 .

[0062] If that is the case, the state S (determined by the estimation method) is the state update of S_0 to t, performed in step S65. If that is not the case, then the message D relates to a detection preceding the detection that resulted in the earlier estimation for state S_0 . In that case the state S_0 is updated using the detection D by the state estimation method in step S64.

[0063] In the claims the word "comprising" does not exclude other elements or steps, and the indefinite article "a" or "an" does not exclude a plurality. A single component or other unit may fulfill the functions of several items recited in the claims. The mere fact that certain measures are recited in mutually different claims does not indicate that a combination of these measures cannot be used to advantage. Any reference signs in the claims should not be construed as limiting the scope. Further, unless expressly stated to the contrary, "or" refers to an inclusive or and not to an exclusive or. For example, a condition A or B is satisfied by any one of the following: A is true (or present) and B is false (or not present), A is false (or not present) and B is true (or present), and both A and B are true (or present).

[0064] More details relevant for the present invention are described in the following Annexes:

A1: Estimation and association for multiple target tracking based on spatially, distributed detections

A2: On Event Based State Estimation

A1: Estimation and association for multiple target tracking based on spatially, distributed detections

[0065] **Summary.** In this paper we consider the multiple object tracking problem with event-based observations. For that we predefine a number detection points which are spatially distributed along the road. Whenever the edge of an object crosses one of the detection points, the position of that detection point together with the time of the event are received by our tracking algorithm. We assume that objects can cover multiple detection points and propose a method to estimate the object's position and orientation from these detections using the shape of the object. Beside that another method is designed which associates newly received detections with a known object. The objects are tracked with an event-based state-estimator that uses the estimated position and orientation, although its design is out of the scope of this paper. Finally our tracking algorithm is critically assessed in a simulation of a parking lot.

1 Introduction

[0066] In multiple target tracking [1-3] one aims to track all the objects/targets, which are moving in a certain area. Three basic problems arise from tracking objects. The first one is how to measure the object's position. The second one is to associate a certain measurement with its correct object and the third one is a state-estimator to keep track of all the objects. This paper considers the first 2 issues when objects are not measured but detected.

[0067] Consider a system in which objects are detected when they cross a predefined 'detection' point. These detectors are triggered by the event that the object's edge crosses its position. However, they cannot distinguish between the objects. This paper describes a method in which a new detection is associated with the object that most probable generated it. Also, a method is described which estimates the position and orientation of the object given the observations in position and time due to the detections. Other examples in which sensor-data is generated due to an event can be found in [4, 5].

[0068] This paper is organized as follows. Section 2 defines background knowledge such as the notation of (object) variables and functions that are used throughout this paper. After that the problem is formulated in section 3 together with existing methods. Section 4 describes the approach which is taken in the design. A more detailed description of the estimation and associated is presented in Section 5 and 6 respectively. Finally both methods are tested in a small application example presented in Section 6 and conclusions are drawn in section 7. But let's start with the background information.

2 Background

[0069] In order to be clear about notations and variables this section describes those that can be found throughout this paper.

2.1 Variables

[0070] $\mathbb{R}$ defines the set of real numbers whereas the set $\mathbb{R}_+$ defines the non-negative real numbers. $\mathbb{R}_{x,y}$ defines the set spanned by the vectors e_x and e_y , the point $p := x \cdot e_x + y \cdot e_y$ is shortly denoted as $p = (x, y)^T$. The set $\mathbb{Z}$ defines the integer values and $\mathbb{Z}_+$ defines the set of non-negative integer numbers. The variable 0 is used either as null, the null-vector or the null-matrix. Its size will become clear from the context.

[0071] Vector $x(t) \in \mathbb{R}^n$ is defined as a vector depending on time t and is sampled using some sampling method. The time t at sampling instant $k \in \mathbb{Z}_+$ is defined as $t_k \in \mathbb{R}$. The variables $\tau_k \in \mathbb{R}$, $x_k \in \mathbb{R}^n$ and $x_{0:k} \in \mathbb{R}^{n \times k+1}$ are defined as:

$$\tau_k := t_k - t_{k-1}, \quad (1a)$$

$$x_k := x(t_k), \quad (1b)$$

$$x_{0:k} := (x(t_0) \ x(t_1) \ \dots \ x(t_k)). \quad (1c)$$

[0072] The matrix $A(t_2 - t_1) \in \mathbb{R}^{a \times b}$ depends on the difference between two time instants $t_2 > t_1$ and is shortly denoted as $A_{t_2-t_1}$.

2.2 Functions

[0073] The transpose, inverse and determinant of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted as A^T , A^{-1} and $|A|$ respectively.

[0074] Let us define the probability of the random vector $x \in \mathbb{R}^n$ as the scalar $Pr(x) \in \{0,1\}$ and the conditional probability of x given the vector $u \in \mathbb{R}^m$ as the scalar $Pr(x|u) \in \{0,1\}$. The probability density function (PDF), as defined in [6] section B2, of the vector $x \in \mathbb{R}^n$ is denoted as $p(x)$ and the conditional PDF of x given $u \in \mathbb{R}^q$ is denoted as $p(x|u)$. The expectation and covariance of x are denoted as $E[x]$ and $cov(x)$ respectively. The conditional expectation of x given a vector u is denoted as $E[x|u]$. The definitions of $E[x]$, $E[x|u]$ and $cov(x)$ can be found in [6] sections B4 and B7.

[0075] The Gaussian function, shortly noted as Gaussian, depending on vectors $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ and on matrix $P \in \mathbb{R}^{n \times n}$ is defined as:

$$\begin{aligned} G(x, u, P) &: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}, \\ &= \frac{1}{(2\pi)^{n/2} |P|} e^{-0.5(x-u)^T P^{-1} (x-u)}. \end{aligned} \quad (2)$$

[0076] If $p(x) = G(x, u, P)$, then by definition it holds that $E[x] = u$ and $cov(x) = P$.

[0077] Assume we have the set $\mathbb{C} \subset \mathbb{R}^q$ and the vectors $x \in \mathbb{R}^q$ and $y \in \mathbb{R}^q$. Then the function $\|x - y\| \in \mathbb{R}$ is defined as the distance between vectors x and y . The function $|(x - \mathbb{C})| \in \mathbb{R}$ is defined as the shortest distance from vector x to set $\mathbb{C}$:

$$\begin{aligned} |\langle x - \mathbb{C} \rangle| : \mathbb{R}^q &\rightarrow \mathbb{R}, \\ &:= \min(\|x - c\|), \quad \forall c \in \mathbb{C}. \end{aligned} \quad (3)$$

2.3 Object variables

[0078] Assume there exist an object which is moving in the 3D world space. This object is observed with, for example, a camera or sensors in the road. Meaning that the object is projected to a 2D space, i.e. $\mathbb{R}_{xy}$. If we assume that the shape of the projected object is constant and known, then we can draw a smallest, rectangular box around the object.

[0079] For the box we define a position-vector $o = (x, y)^T \in \mathbb{R}_{xy}$, equal to the center of the box, and an orientation-vector $\theta \in \mathbb{R}$. In the case of $o = 0$ and $\theta = 0$ the corners of this box, as shown in Figure 14, are defined in the set $\mathbb{C}_0$:

$$\mathbb{C}_0 := \{c_1, c_2, c_3, c_4\}, \quad \text{with } c_i \in \mathbb{R}_{xy}. \quad (4)$$

[0080] Notice that for an object having a certain o and θ the new corner-positions of the object's box are calculated with $\mathbb{C}_0$. For that a rotation matrix $T \in \mathbb{R}^{q \times q}$ is used as defined in (5). An example of the object's box for a certain o and θ is graphically depicted in Figure 14.

$$T_k = \begin{pmatrix} \cos(\theta_k) & \sin(\theta_k) \\ -\sin(\theta_k) & \cos(\theta_k) \end{pmatrix}. \quad (5)$$

[0081] Beside the positions o and θ each object also has a certain shape or geometry which covers a certain set of positions in $\mathbb{R}_{xy}$, i.e. the grey area of Figure 14. This closed set is denoted with $S \subset \mathbb{R}_{xy}$ and is defined as the union of the open set of the object's body $S_B \subset \mathbb{R}_{xy}$ and the closed set of the object's edge $S_E \subset \mathbb{R}_{xy}$, i.e. $S := S_B \cup S_E$. The set S is approximated by a set of sampled position-vectors $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_K]$, with $\lambda_i \in \mathbb{R}_{xy}$. To define the vectors λ_i we equidistant sample the rectangular box defined by $\mathbb{C}_0$ using a grid with a distance r . Each λ_i is a grid point within the set S as graphically depicted in Figure 15.

[0082] The aim is to estimate position, speed and rotation of the object in the case that its acceleration and rotational speed are unknown. Therefore the object's state-vector $s(t) \in \mathbb{R}^5$ and process-noise $w(t) \in \mathbb{R}^2$ are defined as:

$$s(t) := \left(o(t) \quad \theta(t) \quad \frac{\delta o(t)}{\delta t} \right)^T, \quad w(t) := \left(\frac{\delta o^2(t)}{\delta t^2} \quad \frac{\delta \theta(t)}{\delta t} \right)^T. \quad (6)$$

[0083] Next the problem is formulated using this background knowledge.

3 Problem formulation

[0084] A total of E objects are observed within the set $\mathbb{R}_{xy}$. The vectors $o^i = (x^i, y^i)^T$ and θ^i are the i^{th} object's position- and rotation-vector respectively. T^i represents the i^{th} object's rotation-matrix dependent on θ^i . The dynamical process of object i with state-vector s^i , process-noise w^i and measurement-vector m^i is defined with the following state-space model:

$$s_k^i := A_{\tau_k} s_{k-1}^i + B_{\tau_k} w_{k-1}^i, \quad (7a)$$

$$m_k^i := \begin{pmatrix} o_k^i \\ \theta_k^i \end{pmatrix} = C s_k^i + v_k^i, \quad (7b)$$

5 with

$$C := \begin{pmatrix} I^{3 \times 3} & 0 \end{pmatrix}, \quad p(w_k^i) := G(w_k^i, 0, Q^i), \quad p(v_k) := f(v_k). \quad (7c)$$

10

The definition of the elements of state-vector $s^i(t)$, also shown in Figure 16, are:

$$s^i(t) := \left(x^i(t) \ y^i(t) \ \theta^i(t) \ \frac{dx^i(t)}{dt} \ \frac{dy^i(t)}{dt} \right)^T. \quad (8)$$

15

[0085] The objects are observed in $\mathbb{R}_{xy}$ by a camera or a network of sensors. For that M 'detection' points are marked within $\mathbb{R}_{xy}$ and collected in the set $D \subset \mathbb{R}_{xy}$. The position of a detection point is denoted as $d \in D$. The k^{th} taction of the system generates the observation vector $z_k^i \in \{\mathbb{R}_{xy}, \mathbb{R}\}$ if the edge of the i^{th} object covers one of the detection points $d_k \in D$ at time t_k :

20

25

$$z_k^i := (d_k, t_k), \quad \text{if} \quad (T_k^i)^{-1} (d_k - o_k^i) \in S_E^i. \quad (9)$$

[0086] However, the system does not know which object was detected for it can be any object. As a result the system will not generate z_k^i but a general observation vector $z_k \in \{\mathbb{R}_{xy}, \mathbb{R}\}$, which is yet to be associated with an object. Therefore, due to the k^{th} detection, the observation vector z_k is generated whenever one of the E object covers a detection point $d_k \in D$ at time t_k :

30

35

$$z_k := (d_k, t_k), \quad \text{if} \left\{ \exists i : (T_k^i)^{-1} (d_k - o_k^i) \in S_E^i \right\}. \quad (10)$$

[0087] From equations (9) and (10) we conclude that z_k^i of (9) is the result after the received observation vector z_k (10) is associated with object i . Notice that both definitions of z_k and z_k^i assume that the object's edge is detected exactly at a detection point d . In reality the detection will be affected by noise. The object therefore has some probability to be detected at a position $v \in \mathbb{R}_{xy}$ which is close to d . This is modeled by defining that the object's position at the instant of the detection, i.e. v , is a random vector with mean d and covariance $\varepsilon \in \mathbb{R}$:

40

45

$$p(v) := G(v, d, \varepsilon I). \quad (11)$$

50

[0088] Figure 16 shows an example of object i which is detected by multiple detection points. The covariance ε of each detection point is also indicated.

[0089] The sampling method of the observation vectors $z_{0:k}$ is a form of event sampling [4, 5, 7]. For a new observation vector is sampled whenever an event, i.e. object detection, takes place. With these event samples all N objects are to be tracked. To accomplish that three methods are needed. The first one is the association of the new observation-vector z_k to an object i and therefore denote it with z_k^i . Suppose that all associated observation-vectors z_n^i are collected in

55

the set $Z_k^i \in \{z_{0:k}\}$. Then the second method is to estimate m_k^i from the observation-set Z_k^i . This is used in the third method, which is a state-estimator.

[0090] Present association methods are: Gating and Nearest Neighbor (NN) [2], (Joint) Probabilistic Data Association ((J)DPA) [2,8], Multiple Hypothesis Tracker (MHT) [9] and the MCMCDA [10]. Although these can be transformed for associating the event samples $z_{0:k}$, this paper will show that the estimation of m_k^i results in a probability that z_k is in fact z_k^i , i.e. $Pr(z_k = z_k^i)$. Therefore the problem which is covered in this paper is the estimation of m_k^i from the set Z_k^i , which also results in the probability $Pr(z_k = z_k^i)$. For that we assume that the shape of the object is known and that it is samples as shown in Section 2.3. The state-estimation is not covered in this paper, although it is used in the application example. Before going into the mathematical details of the estimation we will first describe the approach that is taken.

4 Approach measurement estimation

[0091] In the problem formulation we stated that Z_k^i is defined as the set with all observation-vectors from $z_{0:k}$ that were associated with object i . We will first redefine this set before continuing with the approach for estimating m_k^i .

[0092] The set $Z_k^i \in [z_{0:k}]$ is defined as the set of all observation-vectors z_n which were associated with object i , from which their detection point is still covered by the object. We will first show how this is done. At time step k we have the observation-set Z_{k-1}^i and the observation z_k was associated to object i , i.e. z_k^i . Now if the object's edge is detected at d_k for the first time, then z_k^i is added to the set Z_{k-1}^i . However, if the object's edge is detected at d_k for the second time, then z_k^i is not added to the set Z_{k-1}^i and the vector z_n^i , for which holds that $d_n = d_k$, is removed from Z_{k-1}^i . This because in the second case, it means that object i drove off the detection point positioned at $d_n = d_k$. Therefore Z_k^i is defined as:

$$Z_k^i := \begin{cases} Z_{k-1}^i \cap z_k, & \text{if } d_k \neq d_n, \forall z_n^i \in Z_{k-1}^i, \\ Z_{k-1}^i \setminus z_n, & \text{if } \{\exists n | d_k = d_n \wedge z_n^i = (d_n, t_n) \in Z_{k-1}^i\}. \end{cases} \quad (12)$$

[0093] With this definition of Z_k^i the approach for estimating m_k^i , i.e. $p(m_k^i | Z_k^i)$, is given. For clarity we assume that the object's shape is rectangular and that all its detection points are denoted with d_n , with $n \in N \subset [0, k]$.

1. The first step is to position the object on each detection point d_n and mirror its set S into the set O_n , as shown in Figure 17 for a single detection. This way we transform the points that are covered by the object, into possible vectors of the object's position $o_k^i \in O_n$ given that it is detected at the detection point d_n .

2. The second step, graphically depicted in Figure 18, is to turn all sets O_n simultaneously around their detection point d_n . This way, each possible orientation θ_k^i of the object results in a corresponding possible object's position o_k^i . For o_k^i must be inside all the sets O_n , $\forall n \in N$, and therefore thus inside the intersection of all sets O_n , $\forall n \in N$, which is denoted as O_N .

[0094] Therefore if we apply these two steps for a number of orientations θ_k^i , then at each orientation we have a set O_N which has to contain the object's position o_k^i . From all these orientations we can calculate $p(m_k^i | Z_k^i)$ as shown in the next section.

5 Measurement estimation

[0095] Estimation of the measurement-vector m_k^i given the observation set Z_k^i results in calculating $p(m_k^i | Z_k^i)$.

Because both m_k^i and Z_k^i always belong to the same object and at sample instant k throughout this section we will remove the sub- and superscripts i and k in the rest of this section. Therefore we have; $m_k^i \rightarrow m$ and $Z_k^i \rightarrow Z$. The set Z consists of the observation vectors z_n , for all $n \in N \subset [0, k]$, that were associated to the same object.

[0096] Although the measurement vector is defined as $m = (o, \theta)^T$, with $o \in \mathbb{R}_{xy}$ and $\theta \in \mathbb{R}$, the detection point at time-step n are defined as $d_n \in \mathbb{R}_{xy}$. Meaning that the objects orientation is not directly. However, because every observation vector $z_n \in Z$ detects the object for one and the same θ , the PDF $p(m|Z)$ is approximated by sampling in θ , i.e.:

$$p(m|Z) \approx \left(\sum_{l=1}^L \alpha_l \right)^{-1} \sum_{l=1}^L \alpha_l p(o|Z, \theta = l\Delta), \quad (13a)$$

with

$$\Delta := \frac{2\pi}{L} \quad \text{and} \quad \alpha_l := Pr(\theta = l\Delta|Z). \quad (13b)$$

[0097] The main aspect of equation (13a) is to determine $p(o|z, \theta)$. To do that we define the set $O_n(\theta) \in \mathbb{R}_{xy}$ to be equal to all possible object positions o , given that the object is detected at position $d_n \in z_n(\in Z)$ and that the object's rotation is equal to θ . The determination of $O_n(\theta) \in \mathbb{R}_{xy}$ is presented in the next section. Therefore, if one object is detected at multiple detection points $d_n, \forall n \in N$, then the set of all possible object positions o given a certain θ equals $O_N(\theta)$:

$$O_N(\theta) := \bigcap_{n \in N} O_n(\theta). \quad (14)$$

[0098] Equation (14) is graphically explained in Figure 19 for two different values of θ and $N = \{1, 2\}$.

[0099] Both $p(o|Z, \theta)$ and α_l are related to the set $O_N(\theta)$ due to the fact that it defines the set of possible object positions o for a given θ . To calculate $p(o|Z, \theta)$ and α_l we define the functions $f(o|Z, \theta)$ and $g(o|Z, \theta)$:

$$\begin{aligned} f(o|z_n, \theta) &:= \begin{cases} 0 & \text{if } o \notin O_n(\theta), \\ 1 & \text{if } o \in O_n(\theta), \end{cases} \\ g(o|Z, \theta) &:= \prod_{n \in N} f(o|z_n, \theta) = \begin{cases} 0 & \text{if } o \notin O_N(\theta), \\ 1 & \text{if } o \in O_N(\theta), \end{cases} \end{aligned} \quad (15)$$

[0100] Therefore the PDF $p(o|Z, \theta)$ and probability α_l are:

$$p(o|Z, \theta) := \frac{g(o|Z, \theta)}{\int_{-\infty}^{\infty} g(o|Z, \theta) do}, \quad \alpha_l := \frac{\int_{-\infty}^{\infty} g(o|Z, \theta) do}{\int_{-\infty}^{\infty} f(o|z_n, \theta) do}. \quad (16)$$

[0101] With (16) both $p(m|Z)$ is calculated according to (13). The rest of this section is divided into two parts. The first part derives the probability function based on a single detection, i.e. $f(o|z_n, \theta)$. While the second part derives the probability function based on a multiple detections, i.e. $g(o|Z, \theta)$.

5.1 Single event detection

[0102] In order to derive $f(o|z_n, \theta)$ we will use the set Λ , defined in 2.3, which contains the sampled positions λ_i that are covered by the object if $o = 0$ and $\theta = 0$. Notice that if the object covers the origin, i.e. $(x, y)^T = 0$, then the possible values of the object position o are given by the set $-\Lambda$. This is graphically depicted in Figure 20 (left). From that we can conclude that if the object covers the detection point d_n , given a certain orientation θ and rotation-matrix T , the sampled set Λ can be transformed into a sampled set of O_n , denoted with $\hat{O}_n$:

$$\hat{O}_n := [\hat{o}_1, \hat{o}_2, \dots, \hat{o}_K], \quad \text{with} \quad \hat{o}_i := d - T\lambda_i \in O_n. \quad (17)$$

[0103] Figure 20 (right) graphically depicts the determination of $\hat{O}_n$ from the set Λ for a given θ and detection point d_n .

[0104] The function $f(o|z_n, \theta)$, as defined in (15), is approximated by placing a Gaussian function at each sampled position $\hat{o}_i \in \hat{O}_n$ with a certain covariance dependent on the grid-size r :

$$f(o|z_n, \theta) \approx 2\pi\gamma^2 \sum_{i=1}^K G(o, \hat{o}_i, \gamma I^{2 \times 2}), \quad (18a)$$

with,

$$\gamma := \left(\frac{2\Delta}{K} \right)^2 \left(0.25 - 0.05e^{-\frac{4(K-1)}{13}} - 0.08e^{-\frac{4(K-1)}{180}} \right). \quad (18b)$$

[0105] The approximation of (18) assumes that the object is detected exactly at d_n . In Section 3 we stated that the detection can be a bit of a detection point. The PDF that the object is detected at position $v \in \mathbb{R}_{xy}$ given the detection point d_n is defined in (11). Inserting this uncertainty into (18) results in the final $f(o|z_n, \theta)$:

$$f(o|z_n, \theta) \approx 2\pi\gamma^2 \sum_{i=1}^K \int_{-\infty}^{\infty} G(v, d, \epsilon I) G(o, v - T\lambda_i, \gamma I) dv. \quad (19)$$

[0106] Equation (19) is solved with the following Proposition and the fact that $G(x, a + b, C) = G(x - b, a, C)$:

Proposition 1. Let there exist two Gaussian functions of the random vectors $x \in \mathbb{R}^n$ and $m \in \mathbb{R}^q$ and the matrix $\Gamma \in \mathbb{R}^{q \times n}$; $G(x, u, U)$ and $G(m, \Gamma x, M)$. Then they have the following property:

[0107]

$$\int_{-\infty}^{\infty} G(x, u, U) G(m, \Gamma x, M) dx = G(\Gamma u, m, \Gamma U \Gamma^T + M). \quad (20)$$

Proof. The proof can be found in Section 9.

[0108] Applying Proposition 1 to (19) results in:

$$f(o|z_n, \theta) \approx 2\pi\gamma^2 \sum_{i=1}^K G(o, \hat{o}_i, R), \quad \text{with} \quad R := (\epsilon + \gamma) I^{2 \times 2}. \quad (21)$$

[0109] From $f(o|z_n, \theta)$ based on a single detection, the next step to multiple detections, i.e. $g(o|Z, \theta)$, is taken.

5.2 Multiple event detections

[0110] The aim of this section is to calculate the function $g(o|Z, \theta)$ by substituting equation (21) in the definition of $g(o|Z, \theta)$ as shown in (15):

$$g(o|Z, \theta) \approx \prod_{n \in N} 2\pi\gamma^2 \sum_{i=1}^K G(o, \hat{\sigma}_i^n, R), \quad \text{with} \quad \hat{\sigma}_i^n := d_n - T\lambda_i. \quad (22)$$

[0111] If N contains m elements, then calculating equation (22) would result in K^m products of m Gaussian functions and sum them afterwards. This would take too much processing power if m is large. That is why equation (22) is calculated differently.

[0112] Instead of using all detection points d_n we will use a subset of them. The derivation of this subset is graphically depicted in Figure 21 for $N = \{1, 2\}$. For that consider the rectangular set $\mathbb{C}_0 \in \mathbb{R}_{xy}$ of Section 2.3 defined by its corners $[c_1, c_2, c_3, c_4]$. For each detection point d_n we define the set $\mathbb{C}_n(\theta) \subset \mathbb{R}_{xy}$ with corner-points $[c_{n1}(\theta), c_{n2}(\theta), c_{n3}(\theta), c_{n4}(\theta)]$ defined as:

$$c_{ni}(\theta) := Tc_i + d_n. \quad (23)$$

[0113] Let us define the rectangular set $\mathbb{C}_N(\theta) \subset \mathbb{R}_{xy}$ as the intersection of the sets $\mathbb{C}_n(\theta)$, $\forall n \in N$, i.e.:

$$\mathbb{C}_N(\theta) := \bigcap_{n \in N} \mathbb{C}_n(\theta). \quad (24)$$

[0114] Meaning that each detection point d_n defines a rectangular set denoted with $\mathbb{C}_n(\theta)$ dependent on rotation θ . The intersection of all these rectangular sets is defined with the set $\mathbb{C}_N(\theta)$.

[0115] In the beginning of this section we defined two different sets shown in Figure 17 and 18. The first set, $O_n(\theta)$, shown in Figure 17 defines all possible object positions o based on a single detection at d_n . The second set, i.e. $O_N(\theta)$, shown in Figure 18, defines all possible object positions o based on all detections at d_n , $\forall n \in N$. Notice that as a result $O_n(\theta) \subset \mathbb{C}_n(\theta)$ and $O_N(\theta) \subset \mathbb{C}_N(\theta)$. Meaning that only within the set $\mathbb{C}_N(\theta)$ all the functions $f(o|z_n, \theta)$ have an overlapping area in which they are 1. Outside $\mathbb{C}_N(\theta)$ there is always at least one $f(o|z_n, \theta)$ which is 0 and therefore makes $g(o|Z, \theta)$ outside $\mathbb{C}_N(\theta)$ equal to 0. Therefore $g(o|Z, \theta)$ of (22) can be approximated by taking only those Gaussians of the functions $f(o|z_n, \theta)$ into account of which their mean, i.e. $\hat{\sigma}_i^n$, is close or in the set $\mathbb{C}_N(\theta)$. We define that close to $\mathbb{C}_N(\theta)$ means a distance of at most $\gamma + \varepsilon$, which defined R in (21). The function $g(o|Z, \theta)$ of (22) is therefore approximated as:

$$g(o|Z, \theta) \approx \prod_{n \in N} 2\pi\gamma^2 \sum_{i \in \mathbb{I}^n} G(o, \hat{\sigma}_i^n, R), \quad (25a)$$

with

$$\{\mathbb{I}^n : |\langle \hat{\sigma}_i^n - \mathbb{C}_N(\theta) \rangle| \leq (\gamma + \varepsilon), \forall i \in \mathbb{I}^n\}. \quad (25b)$$

We can even decrease the number of Gaussians of (25) even further. This because if for a certain detection point d_n it holds $\mathbb{C}_N(\theta) \subset O_n(\theta)$, it means that when we remove the detection point d_n it will not affect the set $\mathbb{C}_N(\theta)$. Therefore

equation (25) is reduced to:

$$g(o|Z, \theta) \approx \prod_{n \in \mathbb{N}} 2\pi\gamma^2 \sum_{i \in \mathbb{I}^\theta} G(o, \delta_i^n, R), \quad (26a)$$

with

$$\{\mathbb{N} \subset N : \mathbb{C}_N(\theta) \subset O_n(\theta), \forall n \in \{N \setminus \mathbb{N}\}\}. \quad (26b)$$

[0116] The calculation of (26) is done by applying the following two propositions. The first one, i.e. Proposition 2, shows how to rewrite a product of a summation of Gaussians into a summation of a product of Gaussians. The second one, i.e. Proposition 3, proves that a product of Gaussians results in a single Gaussian.

Proposition 2. *The product of a summation of Gaussians can be written into a summation of a product of Gaussian:*

[0117]

$$\prod_{j=1}^C \rho_j \sum_{i=1}^{C_j} G(x, x_j^i, R) = \left(\prod_{j=1}^C \rho_j C_j \right) \sum_{m=1}^{\prod_{j=1}^C C_j} \prod_{j=1}^C G(x, x_j^{f(j,m)}, R), \quad \text{with} \quad (27a)$$

$$f(j, m) := \left\lceil \frac{m - \sum_{r=1}^j (f(r-1, m) - 1) (\prod_{r=1}^{C+1} \rho_r)}{\prod_{r=j+1}^{C+1} \rho_r} \right\rceil, \quad (27b)$$

$$f(0, m) := 1, \quad \rho_{j+1} := 1.$$

The proof is given by writing out the left hand side of (27a) and restructuring it.

Proposition 3. *The product of Gaussians is again a Gaussian:*

[0118]

$$\prod_{j=1}^C G(x, x_j, R) = \beta G\left(x, \frac{\sum_{j=1}^C x_j}{C}, \frac{R}{C}\right) \quad \text{and} \quad \beta = \prod_{j=2}^C G\left(x_j, \frac{\sum_{n=1}^{j-1} x_n}{j-1}, \frac{jR}{j-1}\right). \quad (28)$$

The proof is given in Section 10.

[0119] Now applying Propositions 2 and 3 on (26) results in a solution of $g(o|Z, \theta)$ as a summation of Gaussians of the form:

$$g(o|Z, \theta) = \sum_{i=1}^H \beta^i(\theta) G(o, o^i(\theta), R^i(\theta)), \quad (29)$$

Equation (29) is approximated as a single Gaussian function:

$$g(o|Z, \theta) \approx \bar{\beta}(\theta) G(o, \bar{o}(\theta), \bar{R}(\theta)), \quad \text{with} \quad (30a)$$

$$\begin{aligned}
\bar{\beta}(\theta) &:= \sum_{i=1}^H \beta^i(\theta), \quad \bar{o}(\theta) := \sum_{i=1}^H \frac{\beta^i(\theta)}{\bar{\beta}} o^i(\theta), \\
\bar{R}(\theta) &:= \sum_{i=1}^H \frac{\beta^i(\theta)}{\bar{\beta}} \left(R^i(\theta) + (\bar{o}(\theta) - o^i(\theta)) (\bar{o}(\theta) - o^i(\theta))^T \right).
\end{aligned} \tag{30b}$$

[0120] With the result of (30) we can approximate $g(o|Z, \theta)$. In order to calculate the PDF $p(m|Z)$, equation (30) is substituted into equation (16) together with $g(o|z_n, \theta)$ of (21) to calculate $p(o|Z, \theta)$ and α_i . Substituted these results into (13) gives:

$$p(m|Z) = \sum_{i=1}^L \frac{\bar{\beta}(l\Delta)}{\sum_{l=1}^L \bar{\beta}(l\Delta)} G(o, \bar{o}(l\Delta), \bar{R}(l\Delta)). \tag{31}$$

[0121] As was mentioned in the problem formulation, the PDF $p(m|Z)$ also gives us the probability that a new observation vector is generated by an certain object i . This is discussed in the next section.

6 Detection association

[0122] The total probability that a new observation vector z_k is generated by object i is equal to the total probability of the measurement-vector m_k^i given the observation set $[Z_{k-1}^i, z_k]$. For this probability we can use $p(m_k^i | Z_{k-1}^i, z_k)$ which is equal to equation (31). The definition of a PDF is that its total probability, i.e. its integral from $-\infty$ to ∞ , is equal to 1. To make sure that $p(m_k^i | Z_{k-1}^i, z_k)$ of equation (31) has a total probability of 1, it is divided by its true probability $Pr(m_k^i | Z_{k-1}^i, z_k)$. In order to be able to compare these different measurement-vector per object, we normalize each probability with the surface covered by the object. As a result, $Pr(z_k = z_k^i)$ is equal to:

$$Pr(z_k = z_k^i) = \frac{1}{2\pi(\gamma^i)^2 K^i} \sum_{l=1}^L \bar{\beta}(l\Delta). \tag{32}$$

[0123] The variables γ^i and K^i are equal to γ and K respectively, which define the approximation of the function $f(m_k^i | z_n^i, \theta_k^i)$ as shown in (18). With the probability of (32) one can design a method which associates an observation-vector due to a new detection, to its most probable object i . Although the estimation method requires a certain amount of processing power, one can reduce this by reducing the number of samples in the set Λ . Meaning that association and estimation can be done with different sizes of Λ . Moreover, if the objects have a rectangular shape, then with some tricks one can reduce the amount of processing power to a level at which both association as well as estimation can run real-time.

[0124] Now that both the measurement estimation as well as the detection association are designed, both are tested in a multiple object tracking application.

7 Application example

[0125] As an application example we take a parking lot of 50 by 50 meters with a network of wireless sensors distributed randomly along the road's surface. Each sensor can detect a crossing vehicle. A total of 2500 sensors was used resulting in a density of one sensor per square meter. The vehicles are all assumed rectangular shaped objects with a length of 5 meters and a width of 2 meters. A total of 4 vehicles manoeuvre within the parking lot and are tracked using a data-associator followed with an event state-estimator.

[0126] The simulation case is made such that it contains two interesting situation. One in which two vehicles cross each other in parallel and one where two vehicles cross perpendicular. For comparison the objects are tracked using two different association methods. The first one is a combination of Gating and detection association of 6. The second one is a combination of Gating and Nearest Neighbor.

[0127] The result of the detection associator (DA) for both crossings is shown in Figure 22 while the result of the Nearest Neighbor (NN) associator is shown in Figure 23. In both results the real object is plotted in a thick, solid line while its estimated one is plotted in a thin, solid line. The associated detections of each object are given with a symbol which is different for each object; '□' if associated with vehicle 1, '○' if associated with vehicle 2, '▽' if associated with vehicle 3 and '*' if associated with vehicle 4. Figure 22 shows with the DA all detections were correctly associated to the one object, while If NN is used as an association method, we see that a lot of incorrect associated detections. Therefore we can concluded that using the detection association of 6 results in less estimation-error compared to NN.

[0128] A second simulation is done to compare the percentage of incorrect associated detections. Again for the both DA as well as NN only now 4 different amount of detection points were used: 3000, 2500, 2000 and 1500. This table shows that the detection association has a better performance compared to Nearest Neighbor.

Table 1. Percentage of incorrect association

amount of detection points	DA	NN
3000	0 %	4.5 %
2500	0 %	5.6 %
2000	0 %	7.8 %
1500	0 %	2.2 %

8 Conclusions

[0129] This paper presents a method for estimating the position- and rotation-vector of objects from spatially, distributed detections of that object. Each detection is generated at the event that the edge of an object crosses a detection point. From the estimation method a detection associator is also designed. This association method calculates the probability that a new detection was generated by an object i .

[0130] An example of a parking lot shows that the detection association method has no incorrect associated detections in the case that two vehicles cross each other both in parallel as well as orthogonal. If the association method of Nearest Neighbor was used, a large amount of incorrect associated detections were noticed, resulting in a higher state-estimation error.

[0131] The data-assimilation can be further improved with two adjustments. The first one is replacing the set S with S_E only at the time-instants that the observation vector is received. The second improvement is to take the detection points that have not detected anything also in account.

References

[0132]

1. A. Poore and S. Gadaleta, "Some assignment problems arising from multiple target tracking," Mathematical and Computer Modelling, vol. 43, pp. 1074-1091, 2006.
2. Y Bar-Shalom and R. Li, Multitarget-Multisensor Tracking: Principles and Techniques. YBS, 1995.
3. S. Blackman and R. Popoli, Design and Analysis of Modern Tracking Systems. Artech House, Norwood, MA, 1999.
4. V. Nguyen and Y Suh, "Improving estimation performance in Networked Control Systems applying the Send-on-delta transmission method," Sensors, vol. 7, pp. 2128-2138, 2007.
5. K. Åström and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in 41st IEEE Conference on Decision and Control, Las Vegas, Nevada, USA, 2002.
6. N. Johnson, S. Kotz, and A. Kemp, Univariate discrete distributions. JOHN WILEY and SONS, Inc, 1992.
7. M. Miskowicz, "Send-on-delta concept: an event-based data-reporting strategy" Sensors, vol. 6, pp. 49-63, 2006.
8. R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," in IEE International Seminar on Target Tracking: Algorithms and Applications, 2001.
9. B. Ristic, S. arulampalam, and N. Gordon, Beyond the Kalman filter: Particle filter for tracking applications, 2002.
10. O. Songhwa, S. Sastry, and L. Schenato, "A Hierarchical Multiple-Target Tracking Algorithm for Sensor Networks," in Proc. of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.
11. K. Mardia, J. Kent, and J. Bibby, Multivariate analysis. Academic press, Inc. London, 1979.
12. D. Kelly, Introduction to probability. Macmillan Publishing Company, 1994.
13. H. Durant-Whyte, B. Rao, and H. Hu, "Towards a fully decentralized architecture for multisensor data fusion," in 1990 IEEE Int. Conf. on Robotics and Automation, Cincinnati, Ohio, USA, 1990, pp. 1331-1336.

9 Proof of Proposition 1

[0133] *Proof.* Defined are two Gaussian functions with the vectors $x \in \mathbb{R}^n$, $u \in \mathbb{R}^n$, $m \in \mathbb{R}^q$ and matrices $U \in \mathbb{R}^{n \times n}$, $M \in \mathbb{R}^{q \times q}$, $\Gamma \in \mathbb{R}^{q \times n}$: $G(x, u, U)$ and $G(m, \Gamma, M)$. Suppose we define the following PDFs and relation of m with some $c \in \mathbb{R}^q$:

$$m = \Gamma x + c, \quad \text{with} \quad (33a)$$

$$p(c) := G(m, 0, M) \quad \text{and} \quad p(x) := G(x, u, U). \quad (33b)$$

Then from probability theory [6] $p(m)$ is equal to:

$$p(m) := \int_{-\infty}^{\infty} p(m|x) p(x) dx \quad (34a)$$

$$= \int_{-\infty}^{\infty} G(m, \Gamma x, M) G(x, u, U) dx. \quad (34b)$$

[0134] Applying theorem 3.2.1 of [11] on (33b) we have that $p(\Gamma x) = G(\Gamma x, \Gamma u, \Gamma U \Gamma^T)$. Now if we have the random vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ with $p(a) = G(a, u_1, U_1)$ and $p(b) = G(b, u_2, U_2)$ then they have the property $p(a + b) = G(a + b, u_1 + u_2, U_1 + U_2)$ as proven in [12]. Applying this on (33a) results in:

$$p(m) = G(m, \Gamma u, \Gamma U \Gamma^T + M), \quad (35a)$$

$$\Rightarrow \int_{-\infty}^{\infty} G(m, \Gamma x, M) G(x, u, U) dx = G(m, \Gamma u, \Gamma U \Gamma^T + M). \quad (35b)$$

10 Proof of Proposition 3

[0135] *Proof.* A product of Gaussians can be written as:

$$\prod_{j=1}^N G(x, x_j, R) = G(x, x_N, R) \prod_{j=1}^{N-1} G(x, x_j, R), \quad (36a)$$

$$= \beta_{N-1} G(x, x_N, R) G\left(x, \frac{\sum_{j=1}^{N-1} x_j}{N-1}, \frac{R}{N-1}\right). \quad (36b)$$

[0136] From Proposition 1 and the Kalman filter in Information form [13], a product of 2 Gaussians equals:

$$G(x, u, U) G(m, x, M) = G(m, u, U + M) G(x, d, D), \quad (37a)$$

with

$$D^{-1} = U^{-1} + M^{-1}, \quad d = DU^{-1}u + DM^{-1}m. \quad (37b)$$

[0137] Applying (37) on (36b), together with the fact that $G(x, y, Z) = G(y, x, Z)$ we have:

$$\prod_{j=1}^N G(x, x_j, R) = \beta_{N-1} G\left(x_N, \frac{\sum_{j=1}^{N-1} x_j}{N-1}, \frac{NR}{N-1}\right) G\left(x, \frac{\sum_{j=1}^N x_j}{N}, \frac{R}{N}\right). \quad (38)$$

[0138] Equation (38) is equal to (28) for:

$$\beta_N = \beta_{N-1} G\left(x_N, \frac{\sum_{j=1}^{N-1} x_j}{N-1}, \frac{NR}{N-1}\right), \quad (39a)$$

$$= \prod_{i=2}^N G\left(x_i, \frac{\sum_{j=1}^{i-1} x_j}{i-1}, \frac{iR}{i-1}\right). \quad (39b)$$

A2: On Event Based State Estimation

[0139] **Summary.** To reduce the amount of data transfer in networked control systems and wireless sensor networks, measurements are usually taken only when an event occurs, rather than at each synchronous sampling instant. However, this complicates estimation and control problems considerably. The goal of this paper is to develop a state estimation algorithm that can successfully cope with event based measurements. Firstly, we propose a general methodology for defining event based sampling. Secondly, we develop a state estimator with a hybrid update, i.e., when an event occurs the estimated state is updated using measurements; otherwise the update is based on the knowledge that the monitored variable is within a bounded set used to define the event. A sum of Gaussians approach is employed to obtain a computationally tractable algorithm.

1 Introduction

[0140] Different methods for state estimation have been introduced during the last decades. Each method is specialized in the type of process, the type of noise or the type of system architecture. In this paper we focus on the design of a state estimator that can efficiently cope with event based sampling. By event sampling we mean that measurements are generated only when an a priori defined event occurs in the data monitored by sensors. Such an effective estimator is very much needed in both networked control systems and wireless sensor networks (WSNs) [1]. Especially in WSNs, where the limiting resource is energy, data transfer and processing power must be minimized. The existing estimators that could be used in this framework are discussed in Section 4. For related research on event based control, the interested reader is referred to the recent works [2], [3].

[0141] The contribution of this paper is twofold. Firstly, we introduce a general mathematical description of event based sampling. We assume that the estimator does not know when new measurements are available, which usually results in unbounded eigenvalues of its error-covariance matrix. To obtain an estimator with a bounded error-covariance matrix, we develop an estimation algorithm with hybrid update, which is the second main contribution. The developed event based estimator is updated both when an event occurs, with received measurements, as well as at sampling instants synchronous in time. Then the update is based on the knowledge that the monitored variable is within a bounded set used to define the event. In order to meet the requirements of a low processing power, the proposed state estimator is based on the Gaussian sum filter [4, 5], which is known for its computational tractability.

2 Background notions and notation

[0142] $\mathbb{R}$ defines the set of real numbers whereas the set $\mathbb{R}_+$ defines the non-negative real numbers. The set $\mathbb{Z}$ defines the integer numbers and $\mathbb{Z}_+$ defines the set of non-negative integer numbers. The notation $\underline{0}$ is used to denote either the null-vector or the null-matrix. Its size will become clear from the context.

[0143] A vector $x(t) \in \mathbb{R}^n$ is defined to depend on time $t \in \mathbb{R}$ and is sampled using some sampling method. Two different sampling methods are discussed. The first one is time sampling in which samples are generated whenever time t equals some predefined value. This is either synchronous in time or asynchronous. In the synchronous case the time between two samples is constant and defined as $t_s \in \mathbb{R}_+$. If the time t at sampling instant $k_a \in \mathbb{Z}_+$ is defined as t_{k_a} , with $t_{0_a} := 0$, we define:

$$x_{k_a} := x(t_{k_a}) \quad \text{and} \quad x_{0_a:k_a} := (x(t_{0_a}), x(t_{1_a}), \dots, x(t_{k_a})).$$

[0144] The second sampling method is event sampling, in which samples are taken when an event occurred. If t at event instant $k_e \in \mathbb{Z}_+$ is defined as t_{k_e} , with $t_{0_e} := 0$, we define:

$$x_{k_e} := x(t_{k_e}) \quad \text{and} \quad x_{0_e:k_e} := (x(t_{0_e}), x(t_{1_e}), \dots, x(t_{k_e})).$$

[0145] A transition-matrix $A_{t_2-t_1} \in \mathbb{R}^{a \times b}$ is defined to relate the vector $u(t_1) \in \mathbb{R}^b$ to a vector $x(t_2) \in \mathbb{R}^a$ as follows:

$$x(t_2) = A_{t_2-t_1} u(t_1).$$

[0146] The transpose, inverse and determinant of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted as A^T , A^{-1} and $|A|$ respectively. The i^{th} and maximum eigenvalue of a square matrix A are denoted as $\lambda_i(A)$ and $\lambda_{\max}(A)$ respectively. Given that $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$ are positive definite, denoted with $A > 0$ and $B > 0$, then $A > B$ denotes $A - B > 0$. $A \geq 0$ denotes A is positive semi-definite.

[0147] The probability density function (PDF), as defined in [6] section B2, of the vector $x \in \mathbb{R}^n$ is denoted with $p(x)$ and the conditional PDF of x given $u \in \mathbb{R}^q$ is denoted as $p(x|u)$. The expectation and covariance of x are denoted as $E[x]$ and $\text{cov}(x)$ respectively. The conditional expectation of x given u is denoted as $E[x|u]$. The definitions of $E[x]$, $E[x|u]$ and $\text{cov}(x)$ can be found in [6] sections B4 and B7.

[0148] The Gaussian function (shortly noted as Gaussian) of vectors $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ and matrix $P \in \mathbb{R}^{n \times n}$ is defined as $G(x, u, P) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$, i.e.:

$$G(x, u, P) = \frac{1}{\sqrt{(2\pi)^n |P|}} e^{-0.5(x-u)^T P^{-1}(x-u)}. \quad (1)$$

[0149] If $p(x) = G(x, u, P)$, then by definition it holds that $E[x] = u$ and $\text{cov}(x) = P$.

[0150] The element-wise Dirac-function of vector $x \in \mathbb{R}^n$, denoted as $\delta(x) : \mathbb{R}^n \rightarrow \{0, 1\}$, satisfies:

$$\delta(x) = \begin{cases} 0 & \text{if } x \neq \underline{0}, \\ 1 & \text{if } x \equiv \underline{0}, \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x) dx = 1. \quad (2)$$

[0151] For a vector $x \in \mathbb{R}^n$ and a bounded Borel set $Y \subset \mathbb{R}^n$, the set PDF is defined as $A_Y(x) : \mathbb{R}^n \rightarrow \{0, 1\}$ with $1 \in \mathbb{R}$ defined as the Lebesgue measure [8] of the set Y , i.e.:

$$\Lambda_Y(x) = \begin{cases} 0 & \text{if } x \notin Y, \\ v^{-1} & \text{if } x \in Y. \end{cases} \quad (3)$$

3 Event sampling

[0152] Many different methods for sampling a vector $y(t) \in \mathbb{R}^q$ can be found in literature. The one mostly used is time sampling in which the k_a^{th} sampling instant is defined at time $t_{ka} = t_{ka-1} + \tau_{ka-1}$ for some $\tau_{ka-1} \in \mathbb{R}_+$. Recall that if $y(t)$ is sampled at t_a it is denoted as y_{ka} . This method is formalized by defining the observation vector $z_{ka-1} = (y_{ka-1}^T, t_{ka-1})^T \in \mathbb{R}^{q+1}$ at sampling instant k_{a-1} . Let us define the set $H_{ka}(z_{ka-1}) \subset \mathbb{R}$ containing all the values that t can take between t_{ka-1} and $t_{ka-1} + \tau_{ka-1}$, i.e.:

$$H_{ka}(z_{ka-1}) := \{t \in \mathbb{R} \mid t_{ka-1} \leq t < t_{ka-1} + \tau_{ka-1}\}. \quad (4)$$

[0153] Then time sampling defines that the next sampling instant, i.e. k_a , takes place whenever present time t exceeds the set $H_{ka}(z_{ka-1})$. Therefore Z_{ka} is defined as:

$$z_{ka} := (y_{ka}, t_{ka}) \quad \text{if } t \notin H_{ka}(z_{ka-1}). \quad (5)$$

[0154] In the case of synchronous time sampling $\tau_{ka} = t_s, \forall_{ka} \in \mathbb{Z}_+$, which is graphically depicted in Figure 24. Notice that with time sampling, the present time t specifies when samples of $y(t)$ are taken, but time t itself is independent of $y(t)$. As a result $y(t)$ in between the two samples can have any value within $\mathbb{R}^q$. Recently, asynchronous sampling methods have emerged, such as, for example "Send-on-Delta" [9,10] and "Integral sampling" [11]. Opposed to time sampling, these sampling methods are not controlled by time t , but by $y(t)$ itself.

[0155] Next, we present a general definition of event based sampling, which recovers the above mentioned asynchronous methods, for a particular choice of ingredients. Let us define the observation vector at sampling instant $k_e - 1$ as

$k_e - 1$ as $z_{ke-1} := (y_{ke-1}^T, t_{ke-1})^T \in \mathbb{R}^{q+1}$. With that we define the following bounded Borel set in *time-measurement-space*, i.e. $H_{ke}(Z_{ke-1}, t) \subset \mathbb{R}^{q+1}$, which depends on both Z_{ke-1} and t . In line with time sampling the next event instant, i.e. k_e , takes place whenever $y(t)$ leaves the set $H_{ke}(Z_{ke-1}, t)$ as shown in Figure 25 for $q = 2$. Therefore z_{ke} is defined as:

$$z_{ke} := (y_{ke}, t_{ke}) \quad \text{if } y(t) \notin H_{ke}(z_{ke-1}, t). \quad (6)$$

The exact description of the set $H_{ke}(Z_{ke-1}, t)$ depends on the actual sampling method. As an example $H_{ke}(z_{ke-1}, t)$ is derived for the method "Send-on-Delta", with $y(t) \in \mathbb{R}$. In this case the event instant k_e occurs whenever $|y(t) - y_{ke-1}|$ exceeds a predefined level Δ , see Figure 26, which results in $H_{ke}(z_{ke-1}, t) = \{y \in \mathbb{R} \mid -\Delta < y - y_{ke-1} < \Delta\}$.

[0156] In event sampling, a well designed $H_{ke}(z_{ke-1}, t)$ should contain the set of all possible values that $y(t)$ can take in between the event instants $k_e - 1$ and k_e . Meaning that if $t_{ke-1} \leq t < t_{ke}$, then $y(t) \in H_{ke}(z_{ke-1}, t)$. A sufficient condition is that $y_{ke-1} \in H_{ke}(z_{ke-1}, t)$, which for "Send-on-Delta" results in $y(t) \in [y_{ke-1} - \Delta, y_{ke-1} + \Delta]$ for all $t_{ke-1} \leq t < t_{ke}$.

4 Problem formulation: State estimation based on event sampling

[0157] Assume a perturbed, dynamical system with state-vector $x(t) \in \mathbb{R}^n$, process-noise $w(t) \in \mathbb{R}^m$, measurement-vector $y(t) \in \mathbb{R}^q$ and measurement-noise $v(t) \in \mathbb{R}^q$. This process is described by a state-space model with $A_t \in \mathbb{R}^{n \times n}$, $B_t \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{q \times n}$. An event sampling method is used to sample $y(t)$. The model of this process becomes:

$$x(t+\tau) = A_\tau x(t) + B_\tau w(t), \quad (7a)$$

$$y(t) = Cx(t) + v(t), \quad (7b)$$

$$z_{k_e} = (y_{k_e}^T, t_{k_e})^T \quad \text{if } y(t) \notin H_{k_e}(z_{k_e-1}, t), \quad (7c)$$

with

$$p(w(t)) := G(w(t), 0, Q) \quad \text{and} \quad p(v(t)) := G(v(t), 0, V). \quad (7d)$$

[0158] The state vector $x(t)$ of this system is to be estimated from the observation vectors $z_{0e:k_e}$. Notice that the estimated states are usually required at all synchronous time samples k_a , with $t_s = t_{k_a} - t_{k_a-1}$, e.g., as input to a controller that runs synchronously in time. As such our goal is to construct an event-based state-estimator (EBSE) that provides an estimate of $x(t)$ not only at the event instants t_{k_e} but also at the sampling instants t_{k_a} . Therefore, we define a new set of sampling instants t_n as the combination of sampling instants due to event sampling, i.e. k_e , and time sampling, i.e. k_a :

$$\{t_{0:n-1}\} := \{t_{0a:k_a-1}\} \cup \{t_{0e:k_e-1}\} \quad \text{and} \quad t_n := \begin{cases} t_{k_a} & \text{if } t_{k_a} < t_{k_e}, \\ t_{k_e} & \text{if } t_{k_a} \geq t_{k_e}. \end{cases} \quad (8a)$$

and

$$t_0 < t_1 < \dots < t_n, \quad x_n := x(t_n), \quad y_n := y(t_n). \quad (8b)$$

[0159] The estimator calculates the PDF of the state-vector x_n given all the observations until t_n . This results in a hybrid state-estimator, for at time t_n an event can either occur or not, which further implies that measurement data is received or not, respectively. In both cases the estimated state must be updated (not predicted) with all information until t_n . Therefore, depending on t_n a different PDF must be calculated, i.e.:

$$\text{if } t_n = t_{k_a} \Rightarrow p(x_n | z_{0e:k_e-1}) \quad \text{with } t_{k_e-1} < t_{k_a} < t_{k_e}, \quad (9a)$$

$$\text{if } t_n = t_{k_e} \Rightarrow p(x_n | z_{0e:k_e}). \quad (9b)$$

The important parameters for the performance of any state-estimator are the expectation and error-covariance matrix of its calculated PDF. Therefore, from (9) we define:

$$x_{n|n} := \begin{cases} E[x_n | z_{0e:k_e-1}] & \text{if } t_n = t_{k_a} \\ E[x_n | z_{0e:k_e}] & \text{if } t_n = t_{k_e} \end{cases} \quad \text{and} \quad P_{n|n} := \text{cov}(x_n - x_{n|n}). \quad (10)$$

The PDFs of (9) can be described as the Gaussian $G(x_n, x_{n|n}, P_{n|n})$. The square root of the eigenvalues of $P_{n|n}$, i.e.

$\sqrt{\lambda_i(P_{n|n})}$, define the shape of this Gaussian function. Together with $x_{n|n}$ they indicate the bound which surrounds 63% of the possible values for x_n . This is graphically depicted in Figure 27 for the 1D case and Figure 29 for a 2D case, in a top view. The smaller the eigenvalues $\lambda_i(P_{n|n})$ are, the smaller the estimation-error is.

[0160] As such, the problem of interest in this paper is to construct a state-estimator suitable for the general event

sampling method introduced in Section 3 and which is computationally tractable. Furthermore, it is desirable to guarantee that $P_{n|n}$ has bounded eigenvalues for all n .

[0161] Existing state estimators can be divided into two categories. The first one contains estimators based on time sampling: the (a)synchronous Kalman filter [12, 13] (linear process, Gaussian PDF), the Particle filter [14] and the Gaussian sum filter [4,5] (nonlinear process, non-Gaussian PDF). These estimators cannot be directly employed in event based sampling as if no new observation vector z_{ke} is received, then $t_n - t_{ke} \rightarrow \infty$ and $\lambda_i(P_{n|ke-1}) \rightarrow \infty$. The second category contains estimators based on event sampling. In fact, to the best of our knowledge, only the method proposed in [15] fits this category. However, this EBSE is only applicable in the case of "Send-on-Delta" event sampling and it requires that any PDF is approximated as a single Gaussian function. Moreover, the asymptotic property of $P_{n|n}$ is not investigated in [15].

[0162] In the next section we propose a novel event-based state-estimator, suitable for any event sampling method, together with a preliminary result on asymptotic analysis.

5 An event-based state estimator

[0163] The EBSE estimates x_n given the received observation vectors until time t_n . Notice that due to the definition of event sampling we can extract information of all the measurement vectors $y_{0:n}$. For with $t_i \in \{t_{0:n}\}$ and $t_{je} \in [t_{0e:ke}]$ it follows that:

$$\begin{cases} y_i \in H_{j_e}(z_{j_e-1}, t_i) & \text{if } t_{j_e-1} \leq t_i < t_{j_e}, \\ y_i = y_{j_e} & \text{if } t_i = t_{j_e}. \end{cases} \quad (11)$$

[0164] Therefore, from the observation vectors $z_{0e:ke}$ and (11) the PDFs of the hybrid state-estimation of (9), with the bounded, Borel set $Y_i \subset \mathbb{R}^q$, results in:

$$p(x_n | y_0 \in Y_0, y_1 \in Y_1, \dots, y_n \in Y_n) \quad \text{with} \quad (12a)$$

$$Y_i := \begin{cases} H_{j_e}(z_{j_e-1}, t_i) & \text{if } t_{j_e-1} < t_i < t_{j_e}, \\ \{y_{j_e}\} & \text{if } t_i = t_{j_e}. \end{cases} \quad (12b)$$

[0165] For brevity (12a) is denoted as $p(x_n | y_{0:n} \in Y_{0:n})$ and with Bayes-rule [16] yields:

$$p(x_n | y_{0:n} \in Y_{0:n}) := \frac{p(x_n | y_{0:n-1} \in Y_{0:n-1}) p(y_n \in Y_n | x_n)}{p(y_n \in Y_n | y_{0:n-1} \in Y_{0:n-1})}. \quad (13)$$

[0166] To have an EBSE with low processing demand, multivariate probability theory [17] is used to make (13) recursive:

$$p(a|b) := \int_{-\infty}^{\infty} p(a|c)p(c|b)dc \quad \Rightarrow \quad (14a)$$

$$p(x_n | y_{0:n-1} \in Y_{0:n-1}) = \int_{-\infty}^{\infty} p(x_n | x_{n-1}) p(x_{n-1} | y_{0:n-1} \in Y_{0:n-1}) dx_{n-1}, \quad (14b)$$

$$p(y_n \in Y_n | y_{0:n-1} \in Y_{0:n-1}) = \int_{-\infty}^{\infty} p(x_n | y_{0:n-1} \in Y_{0:n-1}) p(y_n \in Y_n | x_n) dx_n. \quad (14c)$$

[0167] The calculation of $p(x_n|y_{0:n} \in Y_{0:n})$ is done in three steps: 1. Assimilate $p(y_n \in Y_n|x_n)$ for both $t_n = t_{ke}$ and $t_n = t_{ka}$. 2. Calculate $p(x_n|y_{0:n} \in Y_{0:n})$ as a summation of N Gaussians. 3. Approximate $p(x_n|y_{0:n} \in Y_{0:n})$ as a single Gaussian function. The reason for this last step is to design an algorithm in which $p(x_n|y_{0:n} \in Y_{0:n})$ is described by a finite set of Gaussians and therefore attain computational tractability. Notice that (13) gives a unified description of the hybrid state-estimator, which makes an asymptotic analysis of the EBSE possible, as it will be shown later in this section.

5.1 Step 1: measurement assimilation

[0168] This section gives a unified formula of the PDF $p(y_n \in Y_n|x_n)$ valid for both $t_n = t_{ke}$ and $t_n = t_{ka}$. From multivariate probability theory [17] and (7b) we have:

$$p(y_n \in Y_n|x_n) := \int_{-\infty}^{\infty} p(y_n|x_n)p(y_n \in Y_n)dy_n \quad \text{and} \quad p(y_n|x_n) = G(y_n, Cx_n, V). \quad (15)$$

[0169] The PDF $p(y_n \in Y_n)$ is modeled as a uniform distribution for all $y_n \in Y_n$. Therefore, depending on the type of instant, i.e. event or not, we have:

$$p(y_n \in Y_n) := \begin{cases} \Lambda_{H_{ke}}(y_n) & \text{if } t_{ke-1} < t_n < t_{ke}, \\ \delta(y_n - y_{ke}) & \text{if } t_n = t_{ke}. \end{cases} \quad (16)$$

[0170] Substitution of (16) into (15) gives that $p(y_n \in Y_n|x_n) = G(y_{ke}, Cx_n, V)$ if $t_n = t_{ke}$. However, if $t_n = t_{ka}$ then $p(y_n \in Y_n|x_n)$ equals $\Lambda_{H_{ke}}(y_n)$, which is not necessarily Gaussian. Moreover, it depends on the set H_{ke} and therefore on the actual event sampling method that is employed. In order to have a unified expression of $p(y_n \in Y_n|x_n)$ for both types of t_n , independent of the event sampling method, $\Lambda_{H_{ke}}(y_n)$ can be approximated as a summation of N Gaussians, i.e.

$$\Lambda_{H_{ke}}(y_n) \approx \sum_{i=1}^N \alpha_n^i G(y_n, y_n^i, V_n^i) \quad \text{and} \quad \sum_{i=1}^N \alpha_n^i := 1. \quad (17)$$

This is graphically depicted in Figure 29 for $y_n \in \mathbb{R}^2$. The interested reader is referred to [4] for more details.

[0171] Substituting (17) into (16) yields the following $p(y_n \in Y_n|x_n)$ if $t_n = t_{ka}$:

$$p(y_n \in Y_n|x_n) \approx \sum_{i=1}^N \alpha_n^i \int_{-\infty}^{\infty} G(y_n, Cx_n, V) G(y_n, y_n^i, V_n^i) dy_n. \quad (18)$$

Proposition 1. [12, 14] Let there exist two Gaussians of random vectors $x \in \mathbb{R}^n$ and $m \in \mathbb{R}^q$, with $\Gamma \in \mathbb{R}^{q \times n} : G(m, \Gamma x, M)$ and $G(x, u, U)$. Then they satisfy:

$$\int_{-\infty}^{\infty} G(x, u, U) G(m, \Gamma x, M) dx = G(\Gamma^T u, m, \Gamma U \Gamma^T + M), \quad (19)$$

$$G(x, u, U) G(m, \Gamma x, M) = G(x, d, D) G(m, \Gamma u, \Gamma U \Gamma^T + M), \quad (20)$$

with $D := (U^{-1} + \Gamma^T M^{-1} \Gamma)^{-1}$ and $d := D U^{-1} u + D \Gamma^T M^{-1} m$.

Applying Proposition 1 ((19) to be precise) and $G(x,y,Z) = G(y,x,Z)$ on (18) yields:

$$p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i G(y_n^i, Cx_n, V + V_n^i), \quad \text{if } t_n = t_{k_d}. \quad (21)$$

In conclusion we can state that the unified expression of the PDF $p(y_n \in Y_n | x_n)$, at both $t_n = t_{k_e}$ and $t_n = t_{k_d}$, for any event sampling method results in:

$$p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i G(y_n^i, Cx_n, R_n^i) \quad \text{with } R_n^i := V + V_n^i. \quad (22)$$

If $t_n = t_{k_e}$ the variables of (22) are: $N = 1$, $\alpha_n^1 = 1$, $y_n^1 = y_{k_e}$ and $V_n^1 = 0$. If $t_n = t_{k_d}$ the variables depend on $\Lambda_{Hke}(y_n)$ and its approximation. As an example these variables are calculated for the method "Send-on-Delta" with $y \in \mathbb{R}$.

Example 1. In "Send-on-Delta", for certain N , the approximation of $\Lambda_{Hke}(y_n)$, as presented in (17), is obtained with $i \in \{1, 2, \dots, N\}$ and:

$$y_n^i = y_{k_e-1} - \left(\frac{N-2(i-1)-1}{2N} \right) 2\Delta \quad \text{and} \quad (23)$$

$$\alpha_n^i = 1/N, \quad V_n^i = \left(\frac{2\Delta}{N} \right)^2 \left(0.25 - 0.05e^{-\frac{4(N-1)}{15}} - 0.08e^{-\frac{4(N-1)}{160}} \right), \quad \forall i.$$

With the result of (22), $p(x_n | y_{0:n} \in Y_{0:n})$ can also be expressed as a sum of N Gaussians.

5.2 Step 2: state estimation

[0172] First the PDF $p(x_n | y_{0:n-1} \in Y_{0:n-1})$ of (14b) is calculated. From the EBSE we have $p(x_{n-1} | y_{0:n-1} \in Y_{0:n-1}) := G(x_{n-1}, x_{n-1|n-1}, P_{n-1,n-1})$ and from (7a) with $\tau_n := t_n - t_{n-1}$ we have $p(x_n | x_{n-1}) := G(x_n, A_{\tau_n} x_{n-1}, B_{\tau_n} Q B_{\tau_n}^T)$. Therefore using (19) in (14b) yields:

$$p(x_n | y_{0:n-1} \in Y_{0:n-1}) = G(x_n, x_{n|n-1}, P_{n,n-1}) \quad \text{with} \quad (24)$$

$$x_{n|n-1} := A_{\tau_n} x_{n-1|n-1} \quad \text{and} \quad P_{n|n-1} := A_{\tau_n} P_{n-1|n-1} A_{\tau_n}^T + B_{\tau_n} Q B_{\tau_n}^T.$$

Next $P(x_n | y_{0:n} \in Y_{0:n})$, defined in (13), is calculated after multiplying (22) and (24):

$$p(x_n | y_{n-1} \in Y_{0:n-1}) p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i G(x_n, x_{n|n-1}, P_{n|n-1}) G(y_n^i, Cx_n, R_n^i). \quad (25)$$

Equation (25) is explicitly solved by applying Proposition 1:

$$p(x_n | y_{0:n-1} \in Y_{0:n-1}) p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i \beta_n^i G(x_n, x_n^i, P_n^i) \quad \text{with} \quad (26a)$$

$$x_n^i := P_n^i \left(P_{n|n-1}^{-1} x_{n|n-1} + C^T (R_n^i)^{-1} y_n^i \right), \quad P_n^i := \left(P_{n|n-1}^{-1} + C^T (R_n^i)^{-1} C \right)^{-1} \quad (26b)$$

and $\beta_n^i := G(y_n^i, Cx_{n|n-1}, CP_{n|n-1}C^T + R_n^i).$

The expression of $p(x_n|y_{0:n} \in Y_{0:n})$ as a sum of N Gaussians is the result of the following substitutions: (26) into (13), (26) into (14c) to obtain $p(y_n \in Y_n|y_{0:n-1} \in Y_{0:n-1})$ and the latter into (13) again. This yields

$$p(x_n|y_{0:n} \in Y_{0:n}) \approx \sum_{i=1}^N \frac{\alpha_n^i \beta_n^i}{\sum_{i=1}^N \alpha_n^i \beta_n^i} G(x_n, x_n^i, P_n^i). \quad (27)$$

The third step is to approximate (27) as a single Gaussian to retrieve a computationally tractable algorithm. For if both $p(x_{n-1}|y_{0:n-1} \in Y_{0:n-1})$ and $p(y_n \in Y_n|x_n)$ are approximated using N Gaussians, the estimate of x_n in (27) is described with M_n Gaussians. The value of M_n equals $M_{n-1}N$, meaning that M_n increases after each sample instant and with it also the processing demand of the EBSE increases.

5.3 Step 3: state approximation

[0173] $p(x_n|y_{0:n} \in Y_{0:n})$ of (27) is approximated as a single Gaussian with an equal expectation and covariance matrix, i.e.:

$$p(x_n|y_{0:n} \in Y_{0:n}) \approx G(x_n, x_{n|n}, P_{n|n}) \quad \text{with} \quad (28a)$$

$$x_{n|n} := \sum_{i=1}^N \frac{\alpha_n^i \beta_n^i x_n^i}{\sum_{i=1}^N \alpha_n^i \beta_n^i}, \quad P_{n|n} := \sum_{i=1}^N \frac{\alpha_n^i \beta_n^i}{\sum_{i=1}^N \alpha_n^i \beta_n^i} \left(P_n^i + (x_{n|n} - x_n^i)(x_{n|n} - x_n^i)^T \right). \quad (28b)$$

The expectation and covariance of (27), equal to $x_{n|n}$ and $P_{n|n}$ of (28), can be derived from the corresponding definitions. Notice that because the designed EBSE is based on the equations of the Kalman filter, the condition of computational tractability is met.

5.4 Asymptotic analysis of the error-covariance matrix

[0174] In this section we investigate the asymptotic analysis of the error-covariance matrix of the developed EBSE. By this we mean that we analyze $\lim_{n \rightarrow \infty} P_{n|n}$ which for convenience is denoted as P_∞ . Note that for the classical Kalman filter (KF) [12] such an analysis is already available. However, for any other type of estimator asymptotic analysis remains a very challenging problem, which is why in most cases it is not even considered.

[0175] Let us first recall the result on the asymptotic analysis of the Kalman filter. If $x(t)$ of (7) is estimated, directly from $y(t)$, with the KF at synchronous sampling times $t_n := n \cdot t_s$, then $P_{n|n}$ is updated as follows:

$$P_{n|n} = \left((A_{t_s} P_{n-1|n-1} A_{t_s}^T + B_{t_s} Q B_{t_s}^T)^{-1} + C^T V^{-1} C \right)^{-1}. \quad (29)$$

[0176] In [18,19] it is proven that if the eigenvalues of A_{t_s} are within the unit circle and (A_{t_s}, C) is observable, then $P_\infty = P_K$. The matrix P_K equals the solution of:

$$P_K = \left((A_{i_s} P_K A_{i_s}^T + B_{i_s} Q B_{i_s}^T)^{-1} + V^{-1} \right)^{-1}. \quad (30)$$

5

For the EBSE however, we cannot prove that P_∞ equals a constant matrix, Instead we will prove that all the eigenvalues of P_∞ are bounded, i.e. that $\lambda_{\max}(P_\infty) < \infty$. As described in Section 4 this is a valid indication of an estimator's performance. **[0177]** The main result of this section is obtained under the standing assumption that Λ_{Hke} is approximated using a single Gaussian. Note that the result then also applies to the estimator presented in [15], as a particular case. We assume that the eigenvalues of the $A_{\tau n}$ -matrix are within the unit-circle and $(A_{\tau n}, C)$ is an observable pair. The following technical Lemmas will be of use.

10

Lemma 1. *Given the process model (7) and covariance matrices $P > 0$ and $Q > 0$, then for any $0 < \tau_1 \leq \tau_2$ we have that*

15

$$A_{\tau_1} P A_{\tau_1}^T \preceq A_{\tau_2} P A_{\tau_2}^T$$

and

20

$$B_{\tau_1} Q B_{\tau_1}^T \preceq B_{\tau_2} Q B_{\tau_2}^T.$$

See the Appendix for the proof.

25

Lemma 2. *Let any square matrices $V_1 \leq V_2$ and $W_1 \leq W_2$ with $V_1 \geq 0$ and $W_1 \geq 0$ be given. Suppose that the matrices U_1 and U_2 are defined as $U_1 := (V_1^{-1} + C^T W_1^{-1} C)^{-1}$ and $U_2 := (V_2^{-1} + C^T W_2^{-1} C)^{-1}$, for any C of suitable size. Then it holds that $U_1 \leq U_2$.*

Proof. From [20] we have that and

30

$$C^T W_1^{-1} C \succeq C^T W_2^{-1} C.$$

Hence, it follows that

35

$$V_1^{-1} + C^T W_1^{-1} C \succeq V_2^{-1} + C^T W_2^{-1} C,$$

which yields

40

$$U_1^{-1} \succeq U_2^{-1}.$$

Thus, $U_1 \leq U_2$, which concludes the proof.

45

Next, recall that $H_{ke}(y_n)$ is assumed to be a bounded set. Therefore, it is reasonable to further assume that Λ_{Hke} can be approximated using the formula (17), for $N = 1$, and that there exists a constant matrix $\bar{V}$ such that

$$V_n^1 \preceq \bar{V}$$

50

for all n .

Theorem 1. *Suppose that the EBSE, as presented in Section 5, approximates Λ_{Hke} according to (17) with $N = 1$ and the above assumptions hold. Then $\lambda_{\max}(P_\infty) < \lambda_{\max}(\tilde{P}_K)$ where $\tilde{P}_K$ is equal to the solution of*

55

$$\tilde{P}_K = \left((A_{i_s} \tilde{P}_K A_{i_s}^T + B_{i_s} Q B_{i_s}^T)^{-1} + (V + \bar{V})^{-1} \right)^{-1}.$$

See the Appendix for the proof.

6 Illustrative example

[0178] In this section we illustrate the effectiveness of the developed EBSE in terms of state-estimation error, sampling efficiency and computational tractability. The case study is a 1D object-tracking system. The states $x(t)$ of the object are position and speed while the measurement vector $y(t)$ is position. The process-noise $w(t)$ represents the object's acceleration. Then given a maximum acceleration of $0.5[m/s^2]$ its corresponding Q , according to [21], equals 0.02. Therefore the model as presented in (7) yields $A = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} \frac{\tau^2}{2} \\ \tau \end{pmatrix}$, $C = (10)$ and $D = 0$, which is in fact a discrete-time double integrator. The acceleration in time is shown in Figure 30 together with the object's position and speed. The sampling time is $t_s = 0.1$ and the measurement-noise covariance is $V = 0.1 \cdot 10^{-3}$.

[0179] Three different estimators are tested. The first two estimators are the EBSE and the asynchronous Kalman filter (AKF) of [13]. For simplicity, in both estimators we used the "Send-on-Delta" method with $\Delta = 0.1 [m]$. For the EBSE we approximated $\Lambda_{Hke}(y_n)$ using (23) with $N = 5$. The AKF estimates the states only at the event instants t_{ke} . The states at t_{ka} are calculated by applying the prediction-step of (14b). The third estimator is based on the quantized Kalman filter (QKF) introduced in [21] that uses synchronous time sampling of y_{ka} . The QKF can deal with quantized data, which also results in less data transfer, and therefore can be considered as an alternative to EBSE. In the QKF $\bar{y}_{ka}$ is the quantized version of y_{ka} with quantization level 0.1, which corresponds to the "Send-on-Delta" method. Hence, a comparison can be made.

[0180] In Figure 31 and Figure 32 the state estimation-error of the three estimators is plotted. They show that the QKF estimates the position of the object with the least error. However, its error in speed is worse compared to the EBSE. Further, the plot of the AKF clearly shows that prediction of the state-estimates gives a significant growth in estimation-error when the time between the event sampling-instants increases ($t > 4$).

[0181] Beside estimation error, sampling efficiency η is also important due to the increased interest in WSNs. For these systems communication is expensive and one aims to have the least data transfer. We define

$\eta \in \mathbb{R}_+$ as $\eta := \frac{(x_i - x_{i|l})^T (x_i - x_{i|l})}{(x_i - x_{i|l-1})^T (x_i - x_{i|l-1})}$, which is a measure of the change in the estimation-error after the measurement update with either z_{ke} or $\bar{y}_{ka}$ was done. Notice that if $\eta < 1$ the estimation error decreased after an update, if $\eta > 1$ the error increased and if $\eta = 1$ the error remained the same. For the EB SE $i = k_e$ with $i - 1$ equal to $k_e - 1$ or $k_a - 1$. For the AKF $i = k_e$ with $i - 1 = k_e - 1$. For the QKF $i = k_a$ and $i - 1 = k_a - 1$. Figure 33 shows that for the EBSE $\eta < 1$ at all instants n . The AKF has one instant, $t = 3.4$, at which $\eta > 1$. In case of the QKF the error sometimes decreases but it can also increase considerably after an update. Also notice that η of the QKF converges to 1. Meaning that for $t > 5.5$ the estimation error does not change after an update and new samples are mostly used to bound $\lambda_i(P_{ka|ka})$. The EBSE has the same property, although for this method the last sample was received at $t = 4.9$.

[0182] The last aspect on which the three estimators are compared is the total amount of processing time which was needed to estimate all state-vectors. For the EBSE, both x_{ke} and x_{ka} were estimated and it took 0.094 seconds. The AKF estimated x_{ke} and predicted x_{ka} in a total time of 0.016 seconds and the QKF estimated x_{ka} and its total processing time equaled 0.022 seconds. This means that although the EBSE results in the most processing time, it is computationally comparable to the AKF and QKF, while it provides an estimation-error similar to the QKF, but with significantly less data transmission. As such, it is most suited for usage in networks in general and WSNs in particular.

7 Conclusions

[0183] In this paper a general event-based state-estimator was presented. The distinguishing feature of the proposed EBSE is that *estimation* of the states is performed at two different type of time instants, i.e. at event instants t_{ke} , when measurement data is used for update, and at synchronous time sampling t_{ka} , when no measurement is received, but an update is performed based on the knowledge that the monitored variable lies within a set used to define the event. As a result, it could be proven that, under certain assumptions, for the error-covariance matrix of the EBSE it holds that $\lambda_{max}(P_\infty) < \infty$, even in the situation when no new observation z_{ke} is received anymore. Its effectiveness for usage in WSNs has been demonstrated on an application example.

References

[0184]

1. I. F. Akyildiz, W. Su, Y Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a survey," Elsevier, Computer Networks, vol. 38, pp. 393-422, 2002.

2. E. Johansson, T. Henningsson, and A. Cervin, "Sporadic control of first-order linear stochastic systems," in Hybrid Systems: Computation and Control, ser Lecture Notes in Computer Science, vol. 4416. Pisa, Italy: Springer Verlag, 2007, pp. 301-314.
3. W. P. M. H. Heemels, J. H. Sandee, and P. P. J. van den Bosch, "Analysis of event-driven controllers for linear systems," International Journal of Control, vol. 81, no. 4, 2008.
4. H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," Automatica, vol. 7, pp. 465-479, 1971.
5. J. H. Kotecha and P. M. Djurić, "Gaussian sum particle filtering," IEEE Transaction Signal Processing, vol. 51, no. 10, pp. 2602-2612, 2003.
6. N. L. Johnson, S. Kotz, and A. W. Kemp, Univariate discrete distributions. John Wiley and Sons, 1992.
7. L. Aggoun and R. Elliot, Measure Theory and Filtering. Cambridge University Press, 2004.
8. H. L. Lebesgue, "Integrale, longueur, aire," Ph.D. dissertation, University of Nancy, 1902.
9. K. J. Åström and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems in 41st IEEE Conf, on Dec, and Contr., Las Vegas, USA, 2002.
10. M. Miskowicz, "Send-on-delta concept: an event-based data-reporting strategy," Sensors, vol. 6, pp. 49-63, 2006.
11. , "Asymptotic Effectiveness of the Event-Based Sampling according to the Integral Criterion," Sensors, vol. 7, pp. 16-37, 2007.
12. R. E. Kalman, "A new approach to linear filtering and prediction problems," Transaction of the ASME Journal of Basic Engineering, vol. 82, no. D, pp. 35-42, 1960.
13. M. Mallick, S. Coraluppi, and C. Carthel, "Advances in Asynchronous and Decentralized Estimation," in In Proceeding of the 2001 Aerospace Conference, Big Sky, MT, USA, 2001.
14. B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman filter: Particle filter for tracking applications. Boston, Massachusetts: Artech House, 2004.
15. V. H. Nguyen and Y. S. Suh, "Improving estimation performance in Networked Control Systems applying the Send-on-delta transmission method," Sensors, vol. 7, pp. 2128-2138, 2007.
16. K. V Mardia, J. T. Kent, and J. M. Bibby, Multivariate analysis. Academic press, London, 1979.
17. D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers. John Wiley and Sons, 2007.
18. M. L. J. Hautus, "Controllability and observability conditions of linear autonomous systems," in Indagationes Mathematicae, vol. 32, 1972, pp. 448-455.
19. A. V. Balakrishnan, Kalman Filtering Theory. Optimization Software, New York, 1987.
20. D. S. Bernstein, Matrix Mathematics. Princeton University Press, 2005.
21. R. E. Curry, Estimation and control with quantized measurements. Boston, Massachusetts: MIT Press, 1970.
22. G. F. Franklin, J. D. Powel, and A. Emami-Naeini, Feedback control of dynamic systems. Addison-Wesley, 1995.

A Proof of Lemma 1

[0185] Suppose that $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are defined as the state-space matrices for the time-continuous counterpart of (7). Then it is known [22] that for any sampling period $\tau > 0$, A_τ and B_τ of (7a) are obtained from their corresponding continuous-time matrices A and B as follows:

$$A_\tau := e^{A\tau} := \sum_{i=0}^{\infty} A^i \frac{\tau^i}{i!} \quad \text{and} \quad B_\tau := \int_0^\tau e^{A\eta} d\eta B := \sum_{i=0}^{\infty} A^i B \frac{\tau^{i+1}}{(i+1)!}. \quad (31)$$

Using (31) one obtains:

$$\begin{aligned} A_{\tau_2} P A_{\tau_2}^T - A_{\tau_1} P A_{\tau_1}^T &= \left(\sum_{i=0}^{\infty} A^i \frac{\tau_2^i}{i!} \right) P \left(\sum_{j=0}^{\infty} (A^T)^j \frac{\tau_2^j}{j!} \right) - \left(\sum_{i=0}^{\infty} A^i \frac{\tau_1^i}{i!} \right) P \left(\sum_{j=0}^{\infty} (A^T)^j \frac{\tau_1^j}{j!} \right) = \\ &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} A^i P (A^T)^j \left(\frac{\tau_2^i}{i!} \frac{\tau_2^j}{j!} \right) - \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} A^i P (A^T)^j \left(\frac{\tau_1^i}{i!} \frac{\tau_1^j}{j!} \right) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} A^i P (A^T)^j \left(\frac{\tau_2^{i+j} - \tau_1^{i+j}}{i! j!} \right) \end{aligned}$$

As for any $\tau > 0$ the series $e^{A\tau}$ converges [22], then $A_{\tau_2} P A_{\tau_2}^T - A_{\tau_1} P A_{\tau_1}^T$ also converges. Then, since $0 < \tau_1 \leq \tau_2$ and P

> 0 , for any fixed i, j , we have

$$A^i P (A^T)^j \left(\frac{\tau_2^{i+j} - \tau_1^{i+j}}{i!j!} \right) \succeq 0$$

for any matrix A and thus, it follows that

$$A_{\tau_2} P A_{\tau_2}^T \succeq A_{\tau_1} P A_{\tau_1}^T.$$

The same reasoning can be used to prove that

$$B_{\tau_1} Q B_{\tau_1}^T \preceq B_{\tau_2} Q B_{\tau_2}^T.$$

B Proof of Theorem 1

[0186] Under the hypothesis, for the proposed EBSE, $P_{n|n}$ of (28), with $\tau_n := t_n - t_{n-1}$ and $R_n := V + V_n^1$, becomes:

$$P_{n|n} = \left((A_{\tau_n} P_{n-1|n-1} A_{\tau_n}^T + B_{\tau_n} Q B_{\tau_n}^T)^{-1} + C^T R_n^{-1} C \right)^{-1}. \quad (32)$$

The upper bound on $\lambda_{\max}(P_{\infty})$ is proven by induction, considering the asymptotic behavior of a KF that runs in parallel with the EBSE, as follows. The EBSE calculates $P_{n|n}^{(1)}$ as (32) and the KF calculates $P_{n|n}^{(2)}$ as (29) in which V is replaced with $R := V + \bar{V}$. Notice that for these estimators we have that $\tau_n \leq t_s$ and $R_n \leq R$, for all n . Let the EBSE and the KF start with the same initial covariance matrix P_0 .

[0187] The first step of induction is to prove that

$$P_{1|1}^{(1)} \preceq P_{1|1}^{(2)}.$$

From the definition of $P_{1|1}^{(1)}$ in (32) and $P_{1|1}^{(2)}$ in (29) we have that $P_{1|1}^{(1)} = \left((A_{\tau_1} P_0 A_{\tau_1}^T + B_{\tau_1} Q B_{\tau_1}^T)^{-1} + C^T R_1^{-1} C \right)^{-1}$ and $P_{1|1}^{(2)} = \left((A_{t_s} P_0 A_{t_s}^T + B_{t_s} Q B_{t_s}^T)^{-1} + C^T R^{-1} C \right)^{-1}$.

[0188] Suppose we define $V_1 := A_{\tau_1} P_0 A_{\tau_1}^T + B_{\tau_1} Q B_{\tau_1}^T$, $V_2 := A_{t_s} P_0 A_{t_s}^T + B_{t_s} Q B_{t_s}^T$, $W_1 := R_1$ and $W_2 := R$. Then $W_1 \leq W_2$ and from Lemma 1 it follows that $V_1 \leq V_2$. Therefore applying Lemma 2, with $U_1 := P_{1|1}^{(1)}$ and $U_2 := P_{1|1}^{(2)}$, yields

$$P_{1|1}^{(1)} \preceq P_{1|1}^{(2)}.$$

[0189] The second and last step of induction is to show that if

$$P_{n-1|n-1}^{(1)} \preceq P_{n-1|n-1}^{(2)},$$

then

$$P_{n|n}^{(1)} \preceq P_{n|n}^{(2)}.$$

Let $V_1 := A_{\tau_n} P_{n-1|n-1}^{(1)} A_{\tau_n}^T + B_{\tau_n} Q B_{\tau_n}^T$, $V_2 := A_{t_s} P_{n-1|n-1}^{(2)} A_{t_s}^T + B_{t_s} Q B_{t_s}^T$, and let $W_1 := R_n$, $W_2 := R$. Notice that this yields $W_1 \leq W_2$. The second condition of Lemma 2, i.e. $V_1 \leq V_2$ also holds by applying Lemma 1, i.e.

$$A_{\tau_n} P_{n-1|n-1}^{(1)} A_{\tau_n}^T + B_{\tau_n} Q B_{\tau_n}^T \preceq A_{t_s} P_{n-1|n-1}^{(1)} A_{t_s}^T + B_{t_s} Q B_{t_s}^T \preceq A_{t_s} P_{n-1|n-1}^{(2)} A_{t_s}^T + B_{t_s} Q B_{t_s}^T.$$

Hence, applying Lemma 2, with $U_1 := P_{n|n}^{(1)}$ and $U_2 := P_{n|n}^{(2)}$ yields

$$P_{n|n}^{(1)} \preceq P_{n|n}^{(2)}.$$

[0190] This proves that

$$P_{\infty}^{(1)} \preceq P_{\infty}^{(2)},$$

which yields (see e.g., [20])

$$\lambda_{\max}(P_{\infty}^{(1)}) \preceq \lambda_{\max}(P_{\infty}^{(2)}).$$

As $P_{n|n}^{(2)}$ was calculated with the KF it follows from (30) that $P_{\infty}^{(2)} = \tilde{P}_K$, with $\tilde{P}_K$ as the solution of

$$\tilde{P}_K = \left(\left(A_{t_s} \tilde{P}_K A_{t_s}^T + B_{t_s} Q B_{t_s}^T \right)^{n-1} + R^{-1} \right)^{-1}, \text{ which completes the proof.}$$

Claims

1. Object tracking system comprising

- a plurality of sensor nodes (10) that each provide a message (D) indicative for an occupancy status of a detection area of an object infrastructure (80) monitored by said sensor node,
- a message interpreter (MI) including an object database facility (32, 34) with state information of objects present at the object infrastructure, and a database updating facility (DBU) for updating the object database facility (32, 34) on the basis of messages (D) provided by the sensor nodes.

2. The object tracking system according to claim 1, comprising a plurality of system modules (MD1, MD2, MD3), each module comprising a respective subset of the plurality of sensor nodes (10) for monitoring a respective section (80A, 80B, 80C, 80D) of the object infrastructure and a respective message interpreter (MI), the object tracking system further having a communication facility (60) for enabling system modules of mutually neighboring sections to exchange state information.

3. Object tracking system according to claim 1 or 2, wherein the database updating facility (DBU) comprises

- an association facility (40) for associating the messages (D) provided by the sensor nodes with the state information present in the object data base facility (32, 34),
- a state updating facility (50) for updating the state information on the basis of the messages (D) associated therewith.

4. Object tracking system according to claim 1, wherein the sensor nodes (10) provide the messages (D) at an event basis.

5. Object tracking system according to claim 1, wherein the sensor nodes (10) are provided with a wireless transmission facility (16) for wirelessly transmitting the message (D), and wherein the message interpreter (40) comprises a wireless reception facility (20) for receiving the message (D).

5 6. The object tracking system according to claim 1, wherein the sensor nodes (10) are randomly distributed over the object infrastructure (80).

7. The object tracking system according to claim 1, wherein the state updating facility (50) is arranged for updating the state information on the basis of event-based messages (D) and on the basis of messages sampled synchronous in time.

8. Object infrastructure (80) provided with an object tracking system according to one of the previous claims.

9. Method for tracking objects at an infrastructure, the method comprising

- a) providing a message indicative for an occupancy status of a detection area of an object infrastructure,
- b) storing state information of objects present at the object infrastructure,
- c) updating said stored information on the basis of the message.

10. Method according to claim 9, wherein the step of updating said stored information comprises,

- d) associating the message with state information present in the object data base facility,
- e) updating the state information associated in step d) on the basis of said message.

11. Method according to claim 9 or 10, wherein the steps a) to c) are independently performed for mutually non coinciding sections of the object infrastructure, the method further comprising the step of exchanging state information.

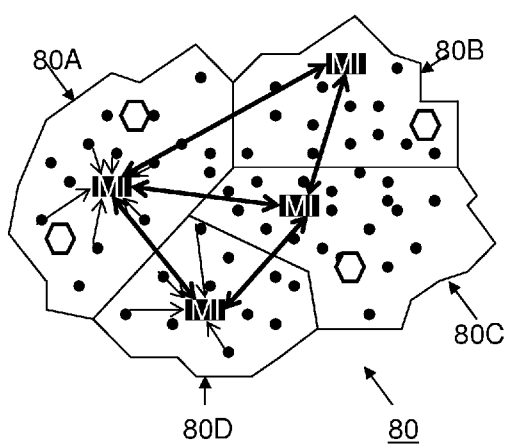


Figure 1

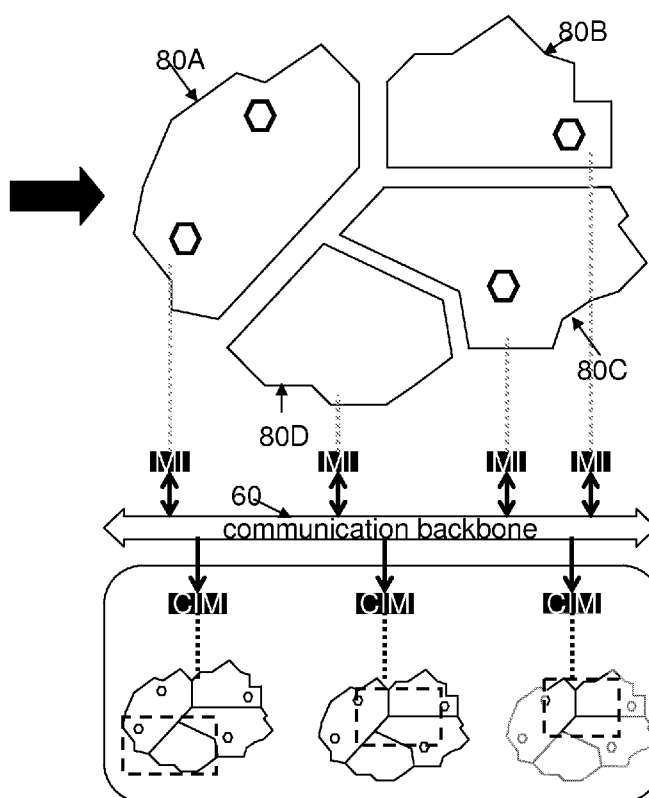
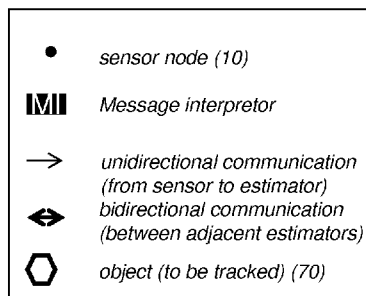


Figure 2

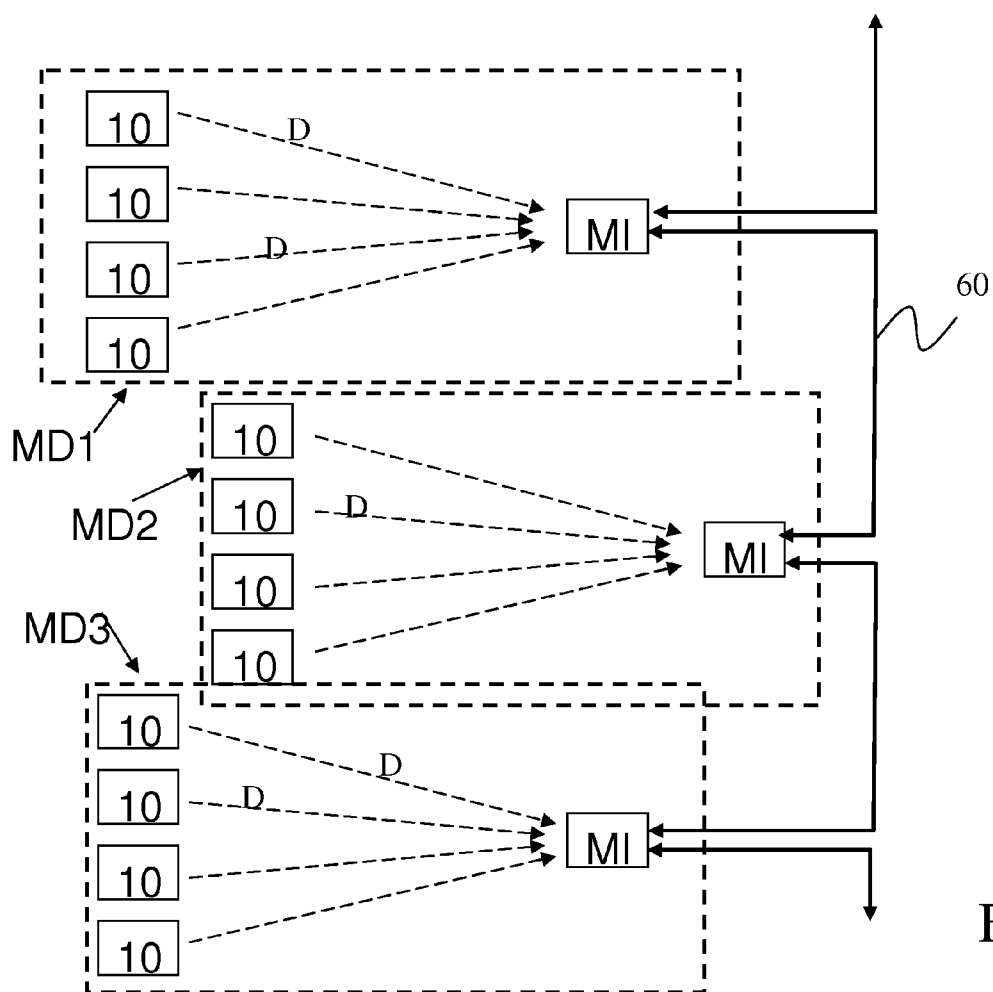


Figure 3

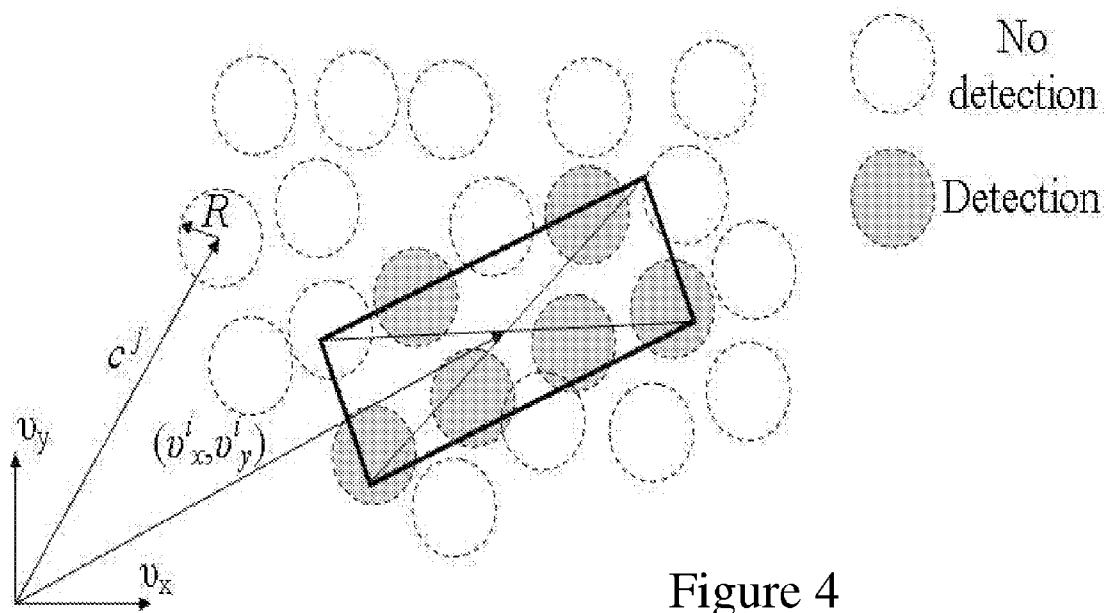


Figure 4

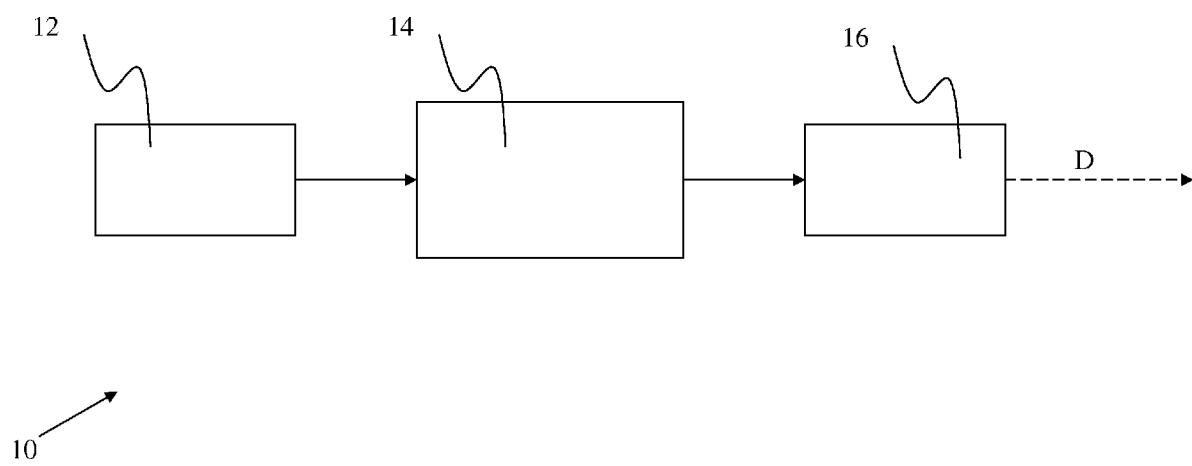


Figure 5

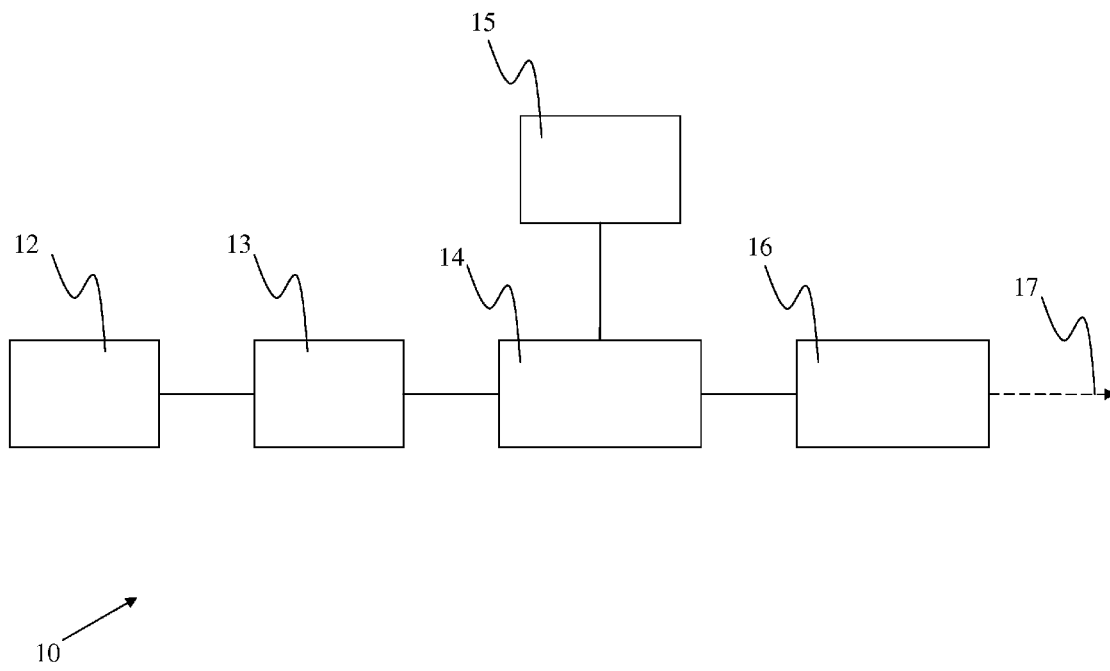


Figure 6

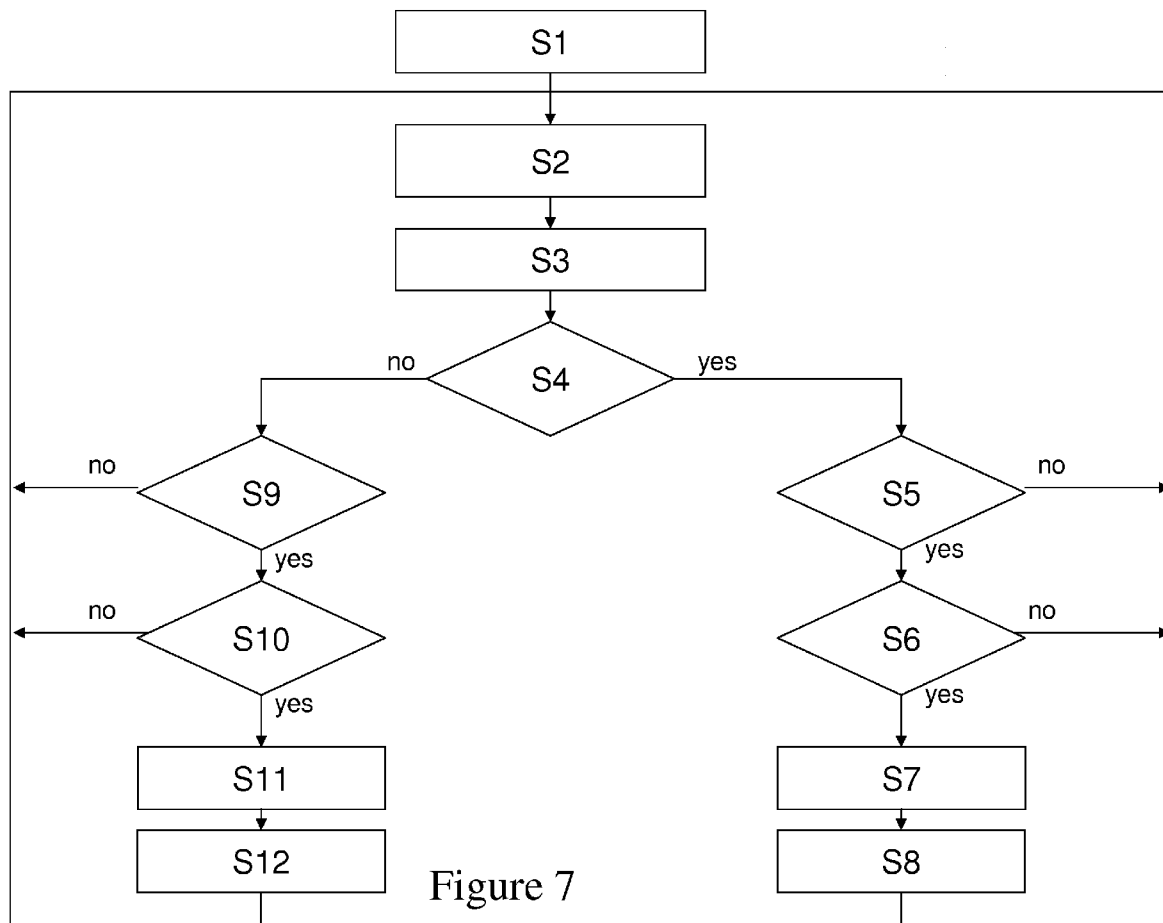


Figure 7

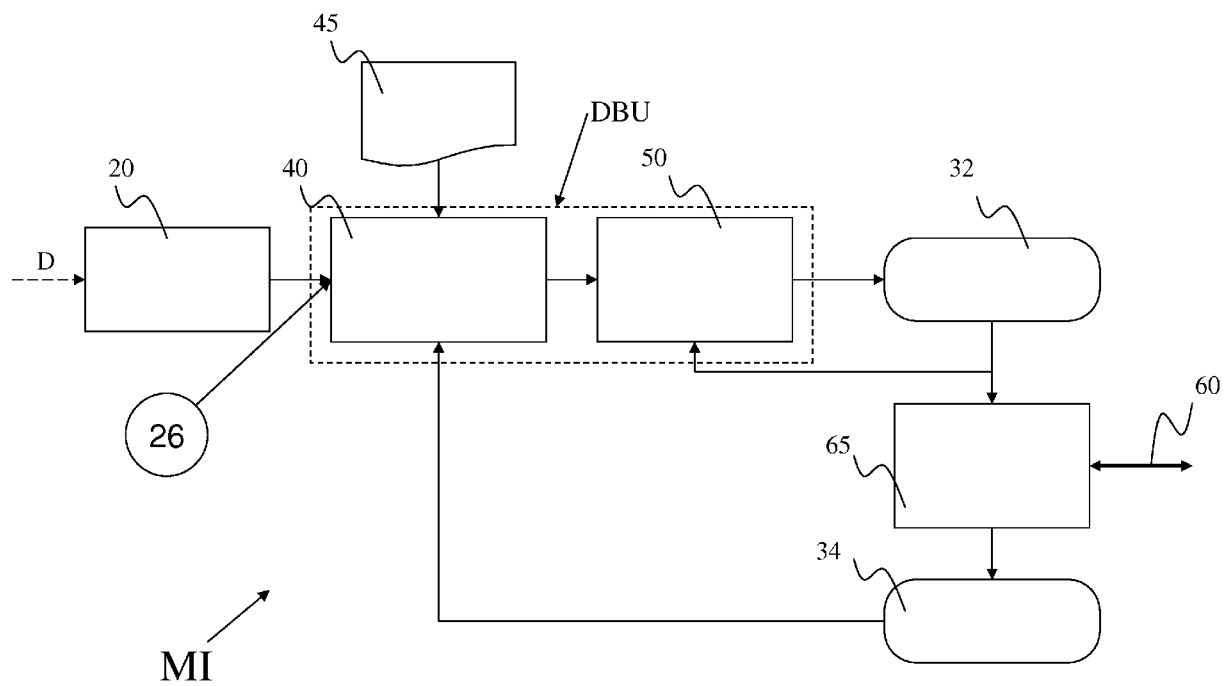


Figure 8

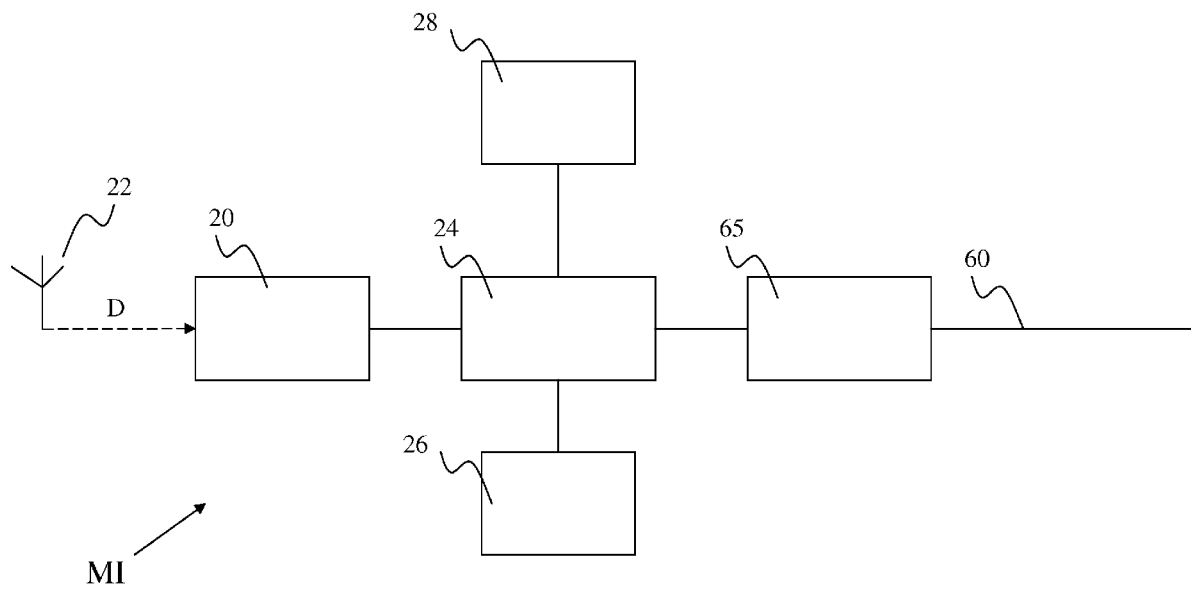


Figure 9

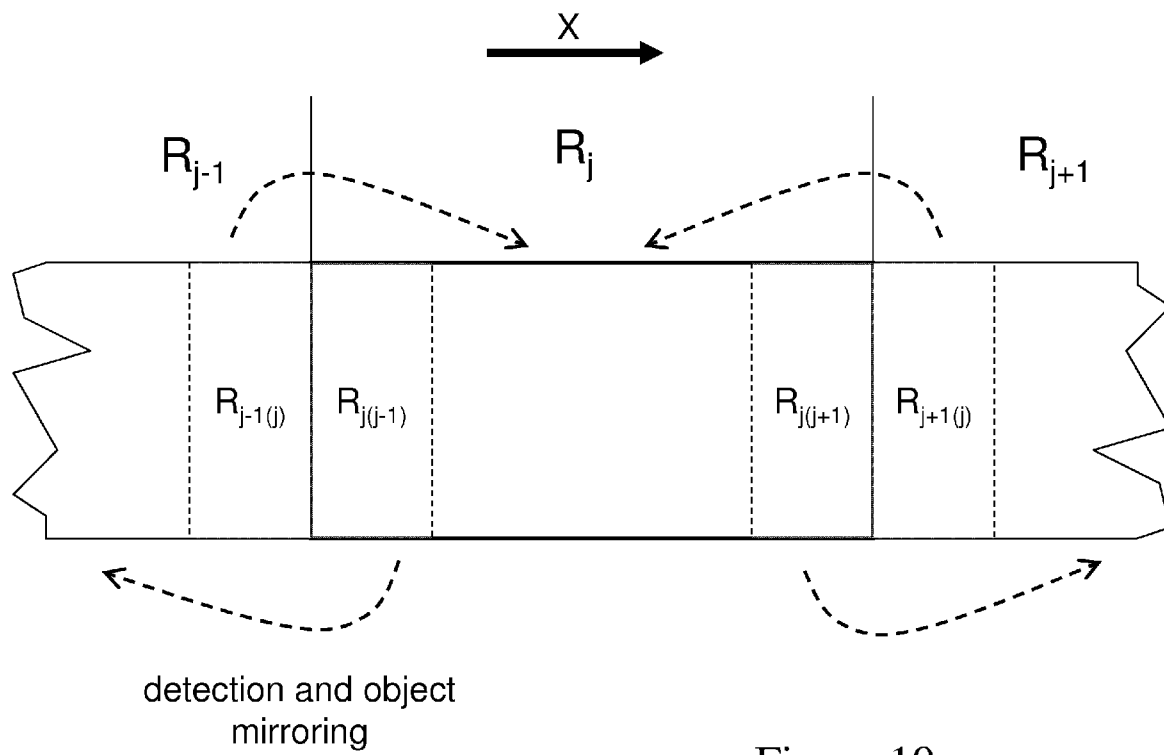


Figure 10

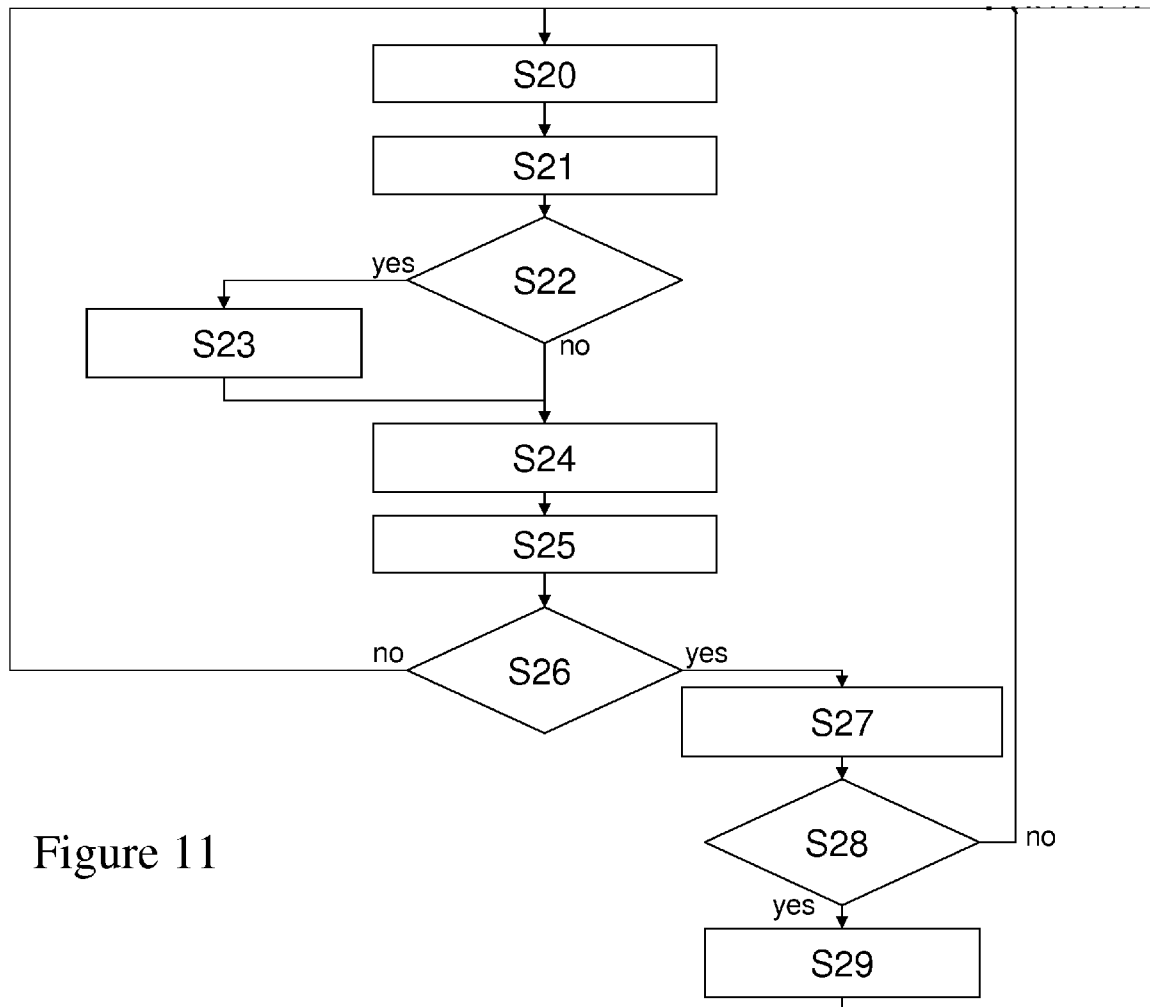


Figure 11

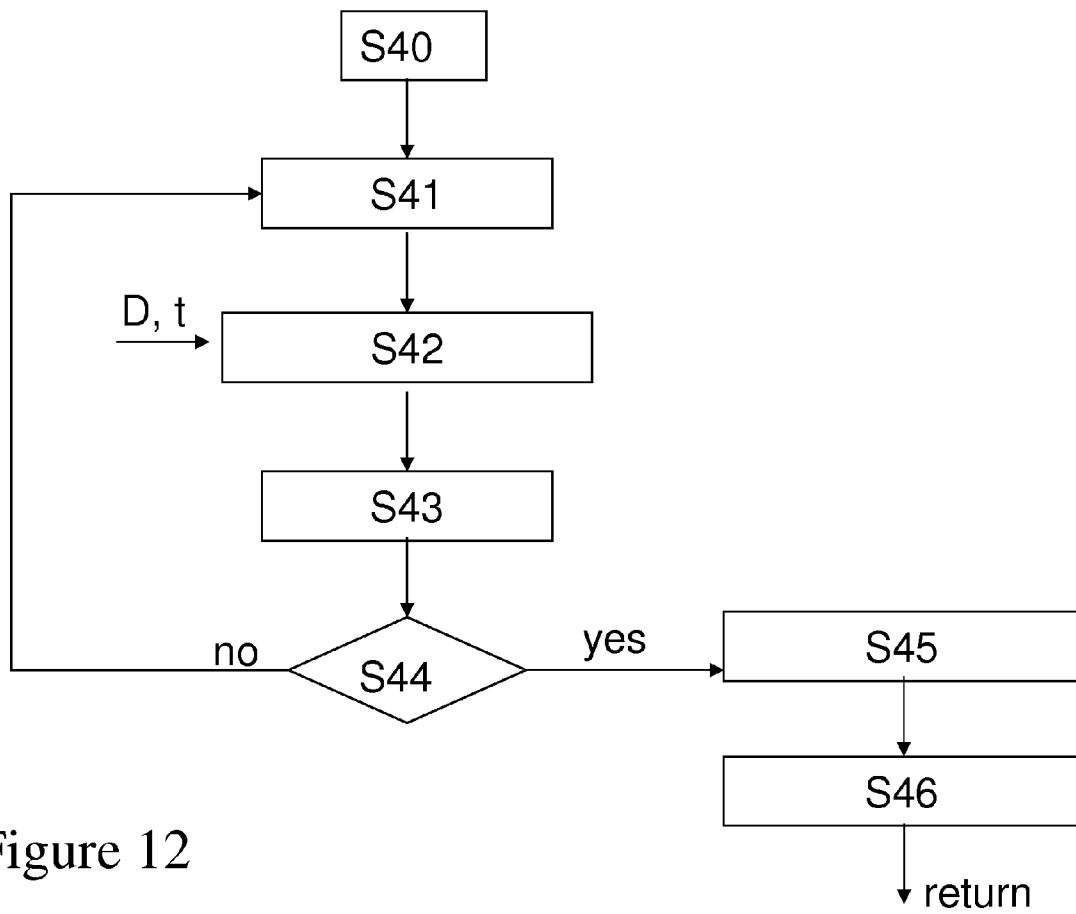


Figure 12

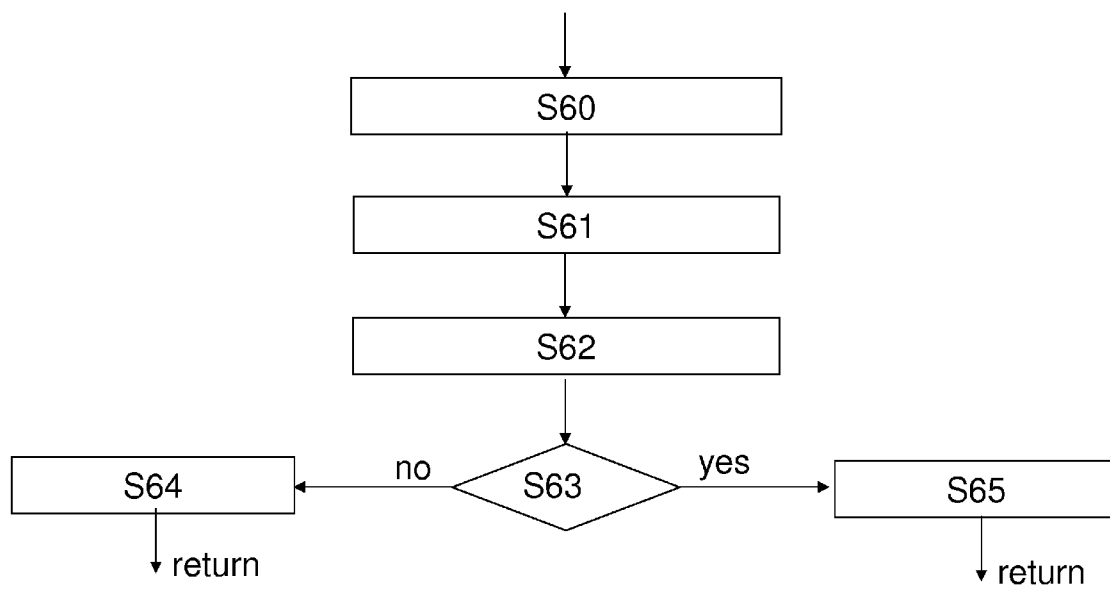


Figure 13

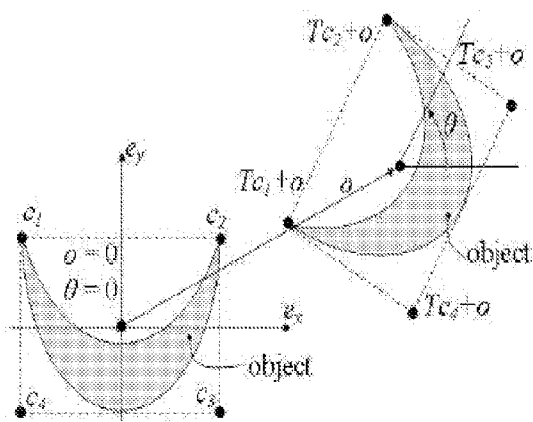


Figure 14

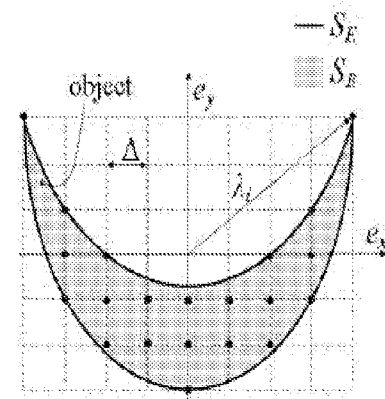


Figure 15

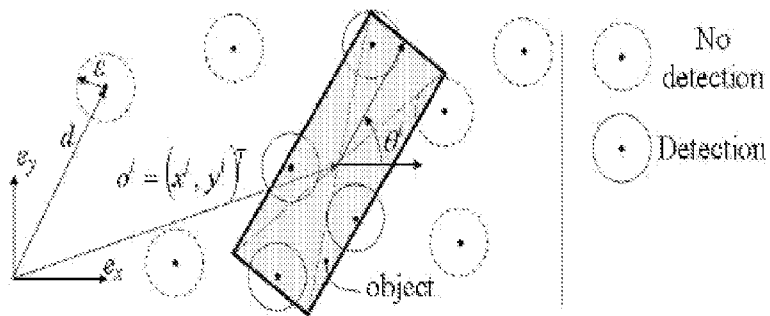


Figure 16

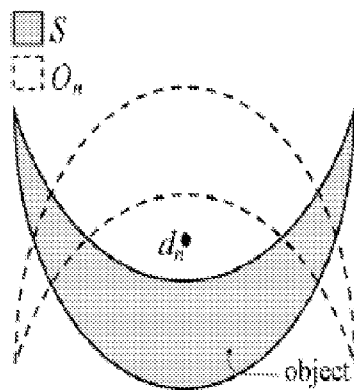


Figure 17

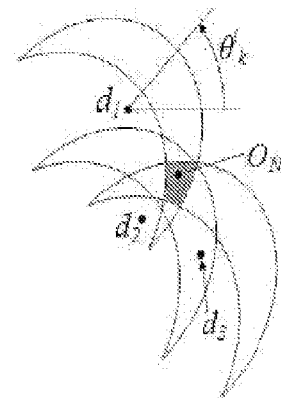


Figure 18

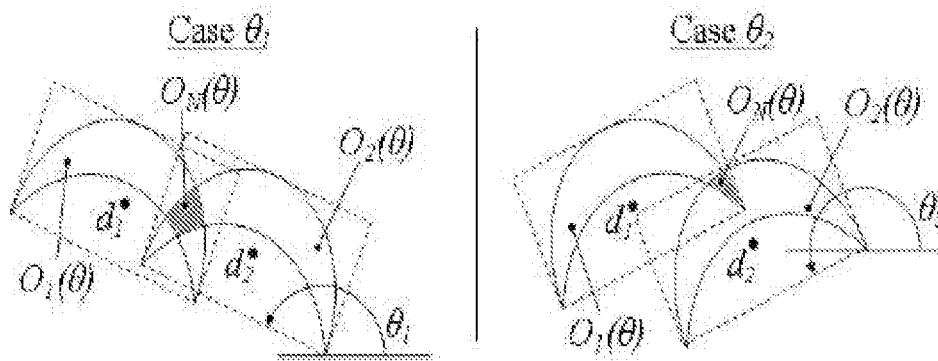


Figure 19

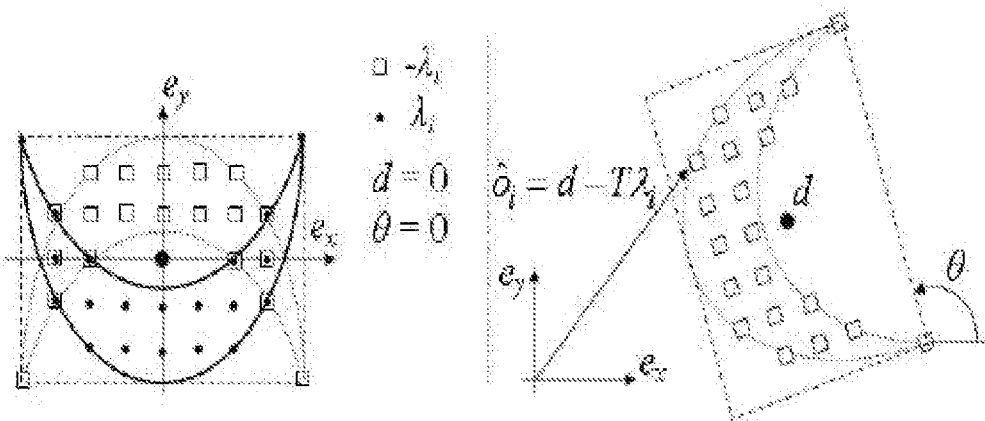


Figure 20

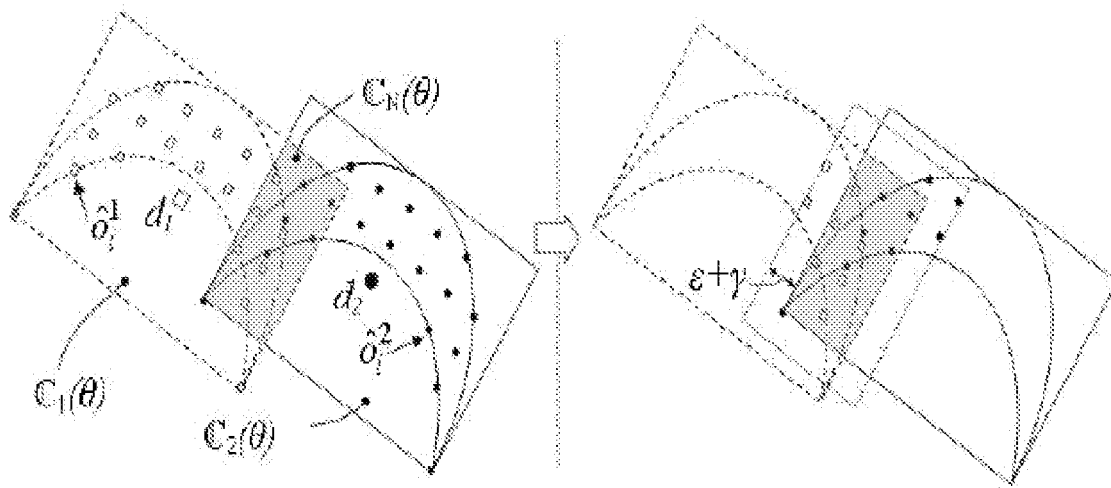


Figure 21

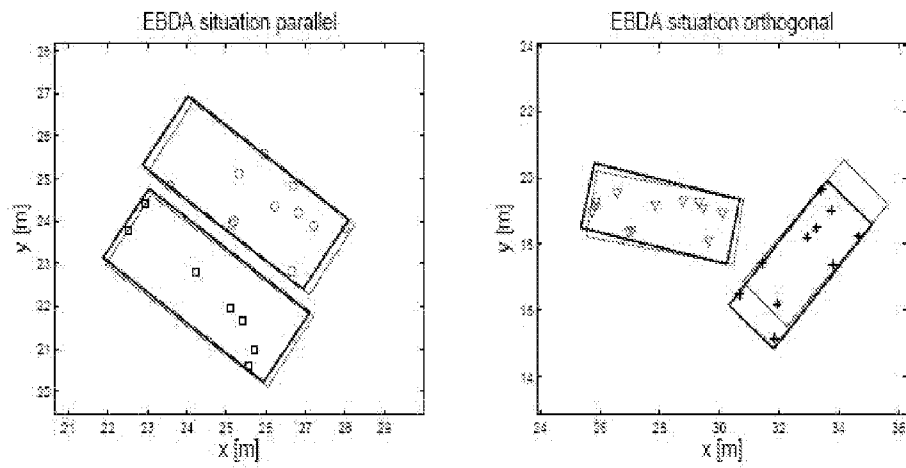


Figure 22

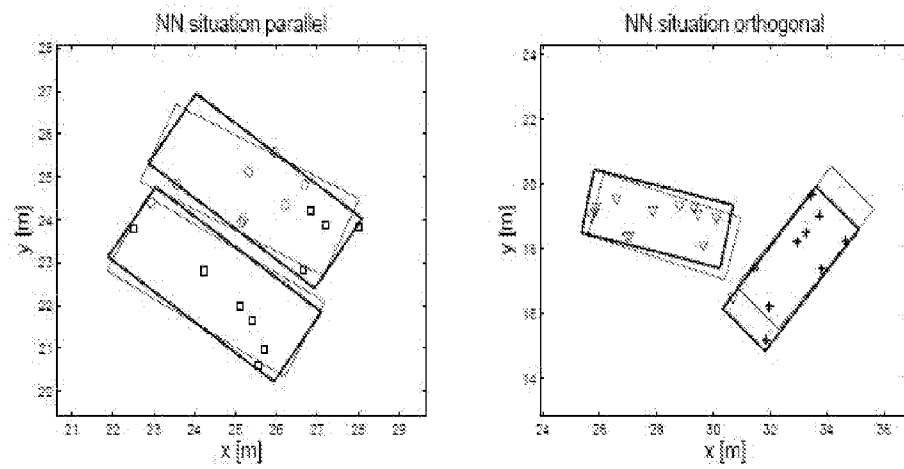


Figure 23

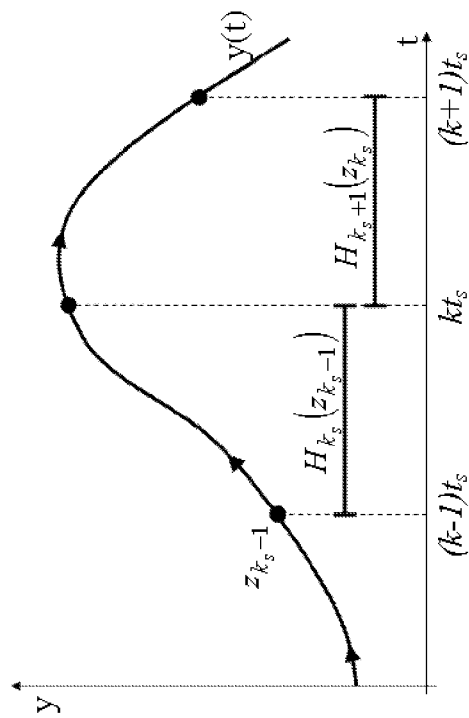


Figure 24

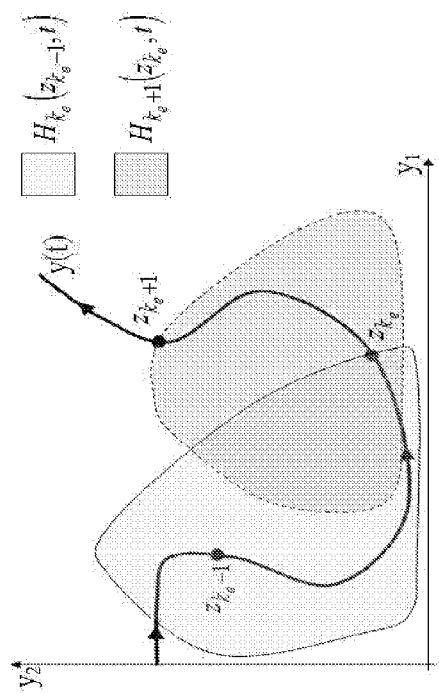


Figure 25

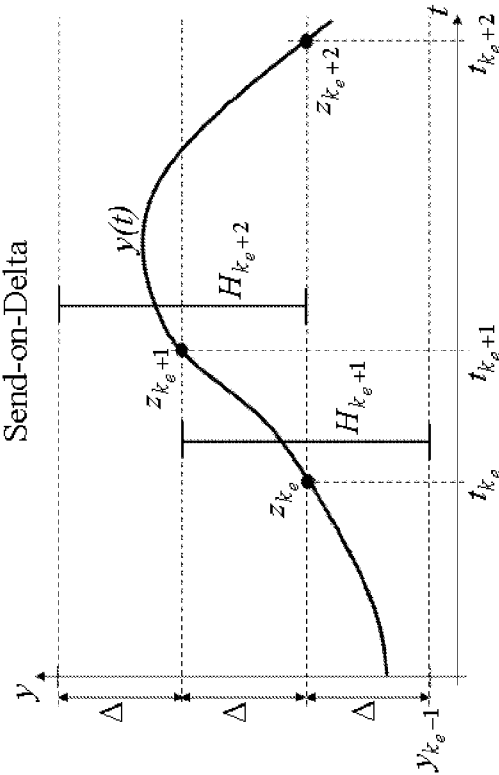


Figure 26

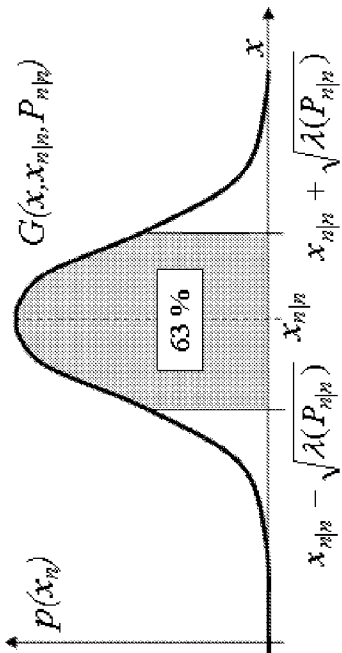


Figure 27

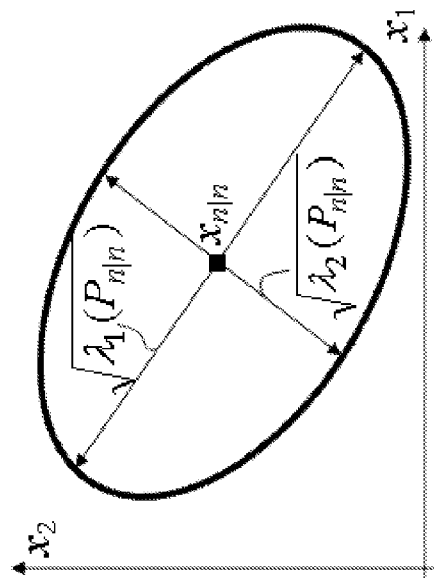


Figure 28

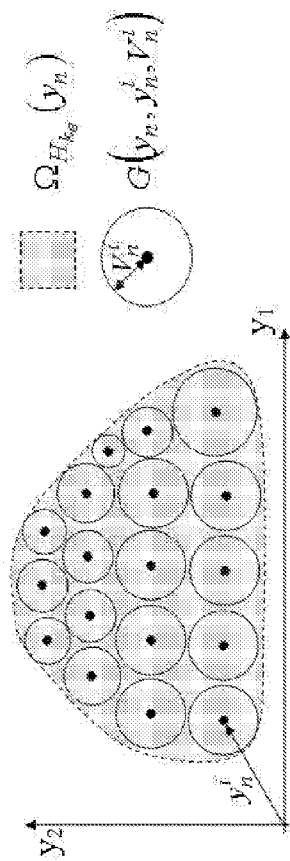


Figure 29

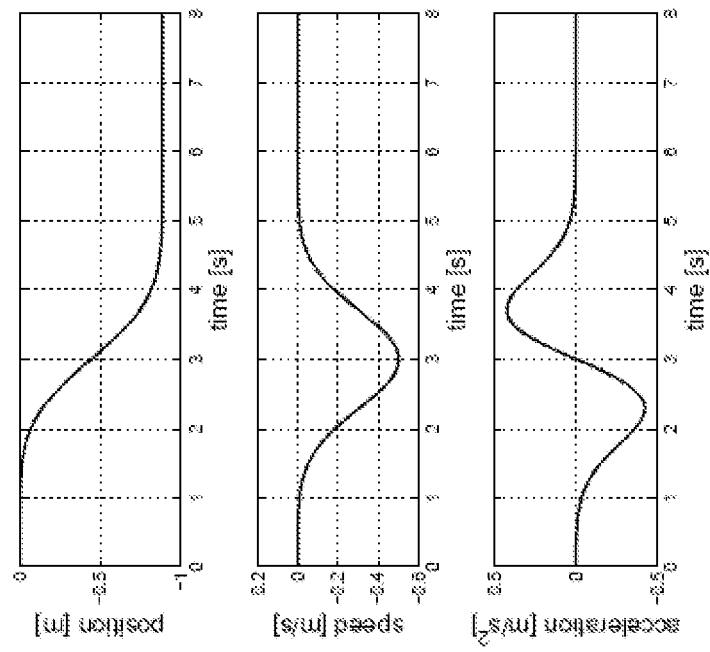


Figure 30

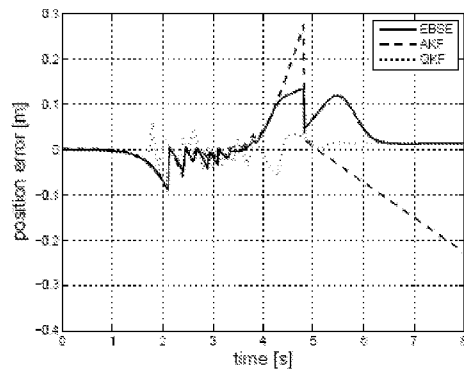


Figure 31

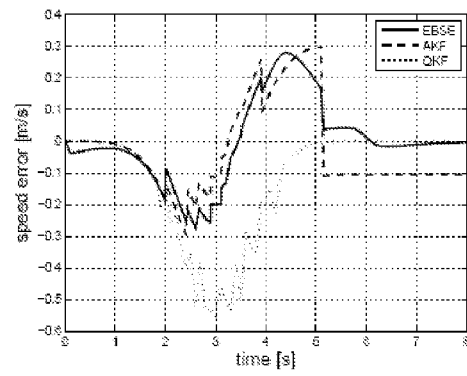


Figure 32

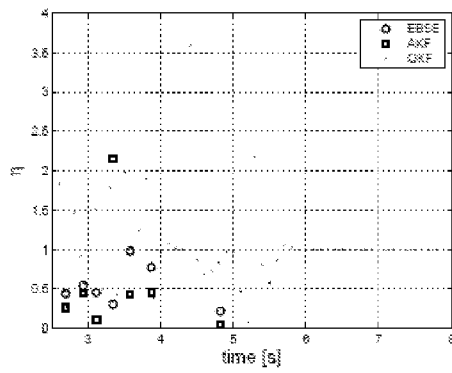


Figure 33



EUROPEAN SEARCH REPORT

 Application Number
EP 08 17 1580

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 2001/010541 A1 (FERNANDEZ DENNIS SUNGA [US] ET AL) 2 August 2001 (2001-08-02) * paragraphs [0001], [0003] - [0005], [0011], [0012], [0017], [0019], [0022] - [0024], [0026], [0030], [0031], [0036], [0037], [0041], [0044], [0053] * * paragraphs [0065] - [0067], [0069], [0071] - [0073], [0075], [0078], [0111] * * claims 1-4,6,12 * * figure 3 *	1-11	INV. G08G1/01
X	US 5 801 943 A (NASBURG ROBERT E [US]) 1 September 1998 (1998-09-01) * abstract * * figures 1,2,7,36 * * column 1, lines 11-14 * * column 5, lines 6-26 * * column 6, line 52 - column 7, line 25 * * column 8, lines 40-59 * * column 11, lines 5-26 * * column 12, line 50 - column 13, line 2 * * column 14, lines 3-25,39-45 * * column 15, lines 24-32 * * column 16, lines 25-58 * * column 18, lines 32-51 * * column 19, lines 3-21,41-45 * * column 20, line 37 - column 21, line 6 * * column 40, lines 58-63 * * claim 1 *	1-11	TECHNICAL FIELDS SEARCHED (IPC) G08G
The present search report has been drawn up for all claims			
Place of search		Date of completion of the search	Examiner
The Hague		30 June 2009	Quartier, Frank
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

2

EPO FORM 1503 03.82 (P04C01)



EUROPEAN SEARCH REPORT

Application Number
EP 08 17 1580

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	WO 01/78397 A (SAFEHOUSE INTERNAT LTD [AU]; WHELAN MICHAEL DAVID CLIVE [AU]; MCADELL) 18 October 2001 (2001-10-18) * abstract * * page 1, lines 3-6 * * page 2, lines 6-29 * * page 3, lines 15-20 * * page 4, lines 15-30 * * claims 1,2,22,23,34 * * figures 1,2 *	1-5,7-11	
A	WO 2006/068463 A (ULTRAWAVES DESIGN HOLDING B V [NL]; BLOEMENDAAL MARK [NL]) 29 June 2006 (2006-06-29) * page 1, lines 4-10 * * page 4, line 14 - page 5, line 10 * * page 8, lines 9-16 * * claims 1,5 * * figures 2,3 *	1-5,7-11	
The present search report has been drawn up for all claims			TECHNICAL FIELDS SEARCHED (IPC)
Place of search The Hague		Date of completion of the search 30 June 2009	Examiner Quartier, Frank
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

2

EPO FORM 1503 03.82 (P04C01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 08 17 1580

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

30-06-2009

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2001010541 A1	02-08-2001	US 6697103 B1	24-02-2004
		US 2009160939 A1	25-06-2009
		US 2002057340 A1	16-05-2002
		US 2001022615 A1	20-09-2001
		US 2001029613 A1	11-10-2001
US 5801943 A	01-09-1998	US 5696503 A	09-12-1997
WO 0178397 A	18-10-2001	CA 2406134 A1	18-10-2001
		EP 1279289 A1	29-01-2003
		US 2003163289 A1	28-08-2003
WO 2006068463 A	29-06-2006	EP 1834486 A1	19-09-2007

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- **Peter J. Bickel.** Measuring Traffic. *Statistical Science*, 2007, vol. 22 (4), 581-597 [0002]
- **Y. Bar-Shalom ; R. Li.** Multitarget-Multisensor Tracking: Principles and Techniques. YBS, 1995 [0015]
- **O. Songhwa ; S. Sastry ; L. Schenato.** A Hierarchical Multiple-Target Tracking Algorithm for Sensor Networks. *Proc. of the 2005 IEEE International Conference on Robotics and Automation*, 2005 [0015] [0132]
- **B. Ristic ; S. arulampalam ; N. Gordon.** *Beyond the Kalman filter: Particle filter for tracking applications*, 2002 [0016]
- **A. Poore ; S. Gadaleta.** Some assignment problems arising from multiple target tracking. *Mathematical and Computer Modelling*, 2006, vol. 43, 1074-1091 [0132]
- **Y Bar-Shalom ; R. Li.** Multitarget-Multisensor Tracking: Principles and Techniques. YBS, 1995 [0132]
- **S. Blackman ; R. Popoli.** Design and Analysis of Modern Tracking Systems. Artech House, 1999 [0132]
- **V. Nguyen ; Y Suh.** Improving estimation performance in Networked Control Systems applying the Send-on-delta transmission method. *Sensors*, 2007, vol. 7, 2128-2138 [0132]
- **K. Åström ; B. Bernhardsson.** Comparison of Riemann and Lebesgue sampling for first order stochastic systems. *41st IEEE Conference on Decision and Control*, 2002 [0132]
- **N. Johnson ; S. Kotz ; A. Kemp.** Univariate discrete distributions. JOHN WILEY and SONS, Inc, 1992 [0132]
- **M. Miskowicz.** Send-on-delta concept: an event-based data-reporting strategy. *Sensors*, 2006, vol. 6, 49-63 [0132]
- **R. Karlsson ; F. Gustafsson.** Monte Carlo data association for multiple target tracking. *IEE International Seminar on Target Tracking: Algorithms and Applications*, 2001 [0132]
- **B. Ristic ; S. arulampalam ; N. Gordon.** *Beyond the Kalman filter: Particle filter for tracking applications*, 2002 [0132]
- **K. Mardia ; J. Kent ; J. Bibby.** Multivariate analysis. Academic press, Inc, 1979 [0132]
- **D. Kelly.** Introduction to probability. Macmillan Publishing Company, 1994 [0132]
- **H. Durant-Whyte ; B. Rao ; H. Hu.** Towards a fully decentralized architecture for multisensor data fusion. *1990 IEEE Int. Conf. on Robotics and Automation*, 1990, 1331-1336 [0132]
- **I. F. Akyildiz ; W. Su ; Y Sankarasubramaniam ; E. Cayirci.** Wireless Sensor Networks: a survey. *Elsevier*, 2002, vol. 38, 393-422 [0184]
- Sporadic control of first-order linear stochastic systems. **E. Johannesson ; T. Henningsson ; A. Cervin.** Hybrid Systems: Computation and Control. Springer Verlag, 2007, vol. 4416, 301-314 [0184]
- **W P. M. H. Heemels ; J. H. Sandee ; P. P. J. van den Bosch.** Analysis of event-driven controllers for linear systems. *International Journal of Control*, 2008, vol. 81 (4 [0184]
- **H. W Sorenson ; D. L. Alspach.** Recursive Bayesian estimation using Gaussian sums. *Automatica*, 1971, vol. 7, 465-479 [0184]
- **J. H. Kotecha ; P. M. Djurić.** Gaussian sum particle filtering. *IEEE Transaction Signal Processing*, 2003, vol. 51 (10), 2602-2612 [0184]
- **N. L. Johnson ; S. Kotz ; A. W. Kemp.** Univariate discrete distributions. John Wiley and Sons, 1992 [0184]
- **L. Aggoun ; R. Elliot.** Measure Theory and Filtering. Cambridge University Press, 2004 [0184]
- **Integrale, longueur, aire. H. L. Lebesgue.** Ph.D. dissertation. University of Nancy, 1902 [0184]
- **K. J. Åström ; B. M. Bernhardsson.** Comparison of Riemann and Lebesgue sampling for first order stochastic systems in 41st IEEE Conf. *Dec. and Contr.*, 2002 [0184]
- **M. Miskowicz.** Send-on-delta concept: an event-based data-reporting strategy. *Sensors*, 2006, vol. 6, 49-63 [0184]
- Asymptotic Effectiveness of the Event-Based Sampling according to the Integral Criterion. *Sensors*, 2007, vol. 7, 16-37 [0184]
- **R. E. Kalman.** A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, 1960, vol. 82 (D), 35-42 [0184]
- **M. Mallick ; S. Coraluppi ; C. Carthel.** Advances in Asynchronous and Decentralized Estimation. *Proceeding of the 2001 Aerospace Conference*, 2001 [0184]
- **B. Ristic ; S. Arulampalam ; N. Gordon.** *Beyond the Kalman filter: Particle filter for tracking applications*. Artech House, 2004 [0184]

- **V. H. Nguyen ; Y. S. Suh.** Improving estimation performance in Networked Control Systems applying the Send-on-delta transmission method. *Sensors*, 2007, vol. 7, 2128-2138 [0184]
- **K. V Mardia ; J. T. Kent ; J. M. Bibby.** Multivariate analysis. Academic press, 1979 [0184]
- **D. C. Montgomery ; G. C. Runger.** Applied Statistics and Probability for Engineers. John Wiley and Sons, 2007 [0184]
- **M. L. J. Hautus.** Controllability and observability conditions of linear autonomous systems. *Indagationes Mathematicae*, 1972, vol. 32, 448-455 [0184]
- **A. V. Balakrishnan.** Kalman Filtering Theory. *Optimization Software*, 1987 [0184]
- **D. S. Bernstein.** Matrix Mathematics. Princeton University Press, 2005 [0184]
- **R. E. Curry.** Estimation and control with quantized measurements. MIT Press, 1970 [0184]
- **G. F. Franklin ; J. D. Powel ; A. Emami-Naeini.** Feedback control of dynamic systems. Addison-Wesley, 1995 [0184]