21

Multi-Sensor Fusion for Mono and Multi-Vehicle Localization using Bayesian Network

C. Smaili[1], M. E. El Najjar[2], F. Charpillet[1] and C. Rose[1]
[1]LORIA-INRIA Lorraine- MAIA Team Campus Scientifique,
[2]LAGIS-CNRS UMR 8146 Polytech'Lille,
France

1. Introduction

Outdoor mobile robotised vehicles currently hold the attention of many researchers because they can bring solutions to many applications related to transport of passengers in urban environments. An example of robotised vehicle is the CyCab (Bom et al., 2005). Transport applications which can combine mono or multi vehicle operating mode are most efficient. For autonomous navigation application, the vehicle needs to know its position accurately (Dissanayake et al. 2001); (Thrun et al. 2001) and if possible on the road network. In this work, we propose to use the digital road map database for the geo-localization of the vehicle. The localization of a vehicle using or in respect to or on a road map is treated by several ways in the last ten years. This relatively recent research theme is known also as the map-matching or road-matching problem. It can be interesting to localize a vehicle using a road map because it can useful to recover the attributes associated with these data bases. Examples of attributes are the width of the road, the presence of landmarks for accurate localization, authorized maximum speed for advanced driver assistance system application etc. Unfortunately, the use of the road map to improve the localization is not a simple task. There are always errors on the estimate of the position and because the map can represents a deformed sight of the world.

Outdoor positioning systems often rely on GPS, because of its affordability and convenience. However, GPS suffers from satellite masks occurring in urban environments, under bridges, tunnels or in forests. GPS appears then as an intermittently-available positioning system that needs to be backed up by a dead-reckoning system (Zhao, 1997); (Abbott & Powell, 1999); (EL Najjar & Bonnifait, 2003). In this work, the proposed method of multi-sensors fusion for mono-vehicle localization is based on the use of encoders positioned at the rear wheel of the vehicle. We use these sensors to measure elementary rotations of the wheels and to estimate the displacement of the vehicle. Thus, a dead-reckoned estimated pose is obtained by integrating the elementary rotations of the wheels using a differential odometric model. The multisensor fusion of GPS and odometry is performed by a Bayesian Network (BN).

Afterwards, we extend the multi-sensor fusion method proposed in this work for mono-vehicle localization to be used for the localization of several vehicles moving in the same environment. We suppose in this extension that the vehicles evolve in a train configuration. In the literature, we found two ways to make moving a train of vehicles. The first one proposes to use an inter-vehicles communication (Bom et al., 2005). The second approach proposes to not use any communication support and to make moving vehicles near-by-near (Daviet & Parent). In our approach, we assume that the leader vehicle is equipped by accurate positioning sensors (GPS LRK, gyroscope LASER and road map database) and multi-sensors fusion approach in order to have continuous and accurate geo-position information. Followers' vehicles are equipped by relatively low cost proprioceptifs and exterioceptifs sensors (odometer, Lidar[1], DGPS) for localization task. The path of the leader vehicle is assumed to be propagated to followers using an inter-vehicle communication device.

The paper is organized as follows: section 2 describes the architecture of vehicle localization. In Section 3, we propose an overview of Bayesian networks formalism. Then, we describe a BN model for the localization of mono-vehicle and the extension of the method in same formalism for the localization for multi-vehicle in section 4. Finally, real data results and simulation are presented and analyzed.

2. Architecture of vehicle localization method

The vehicle localization method described in this section relies on Bayesian networks. The proposed approach can be described by Fig.1. Firstly, the algorithm combines the Anti-lock Braking System (ABS) measurements with a GPS position, if it is available. Then, using this estimate, segments around the estimation are selected in a radius of 30 meters by using a Geographical Information System (GIS-2D). Using these segments, map observations are built and merged with other data sensors using a method based on Bayesian network.

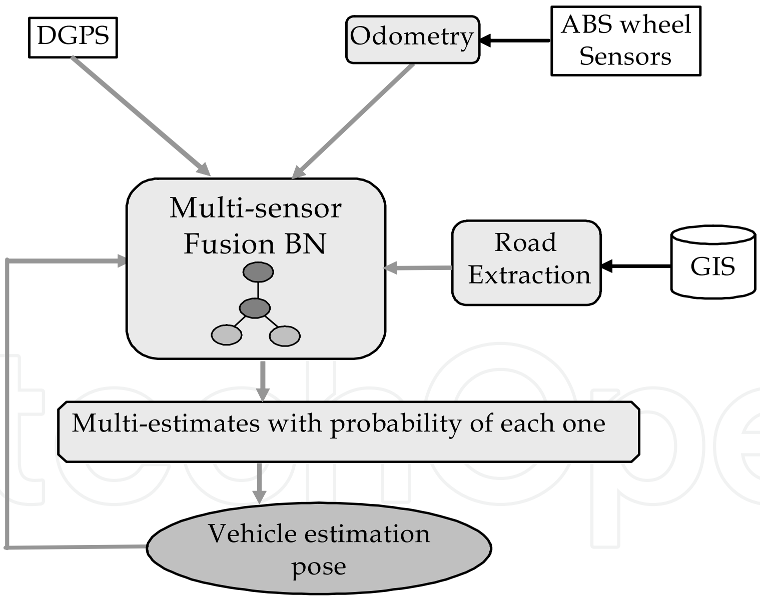


Fig.1. Synoptic of the Mono-vehicle localization method

2.1 Localization and heading estimation by combining odometry and GPS

Let us consider a car-like vehicle with front-wheel drive. The mobile frame is chosen with its origin M attached to the centre of the rear axle. The x-axis is aligned with the longitudinal

[1] Lidar is a SICK LASER telemeter.

axis of the car (see Fig.2). The vehicle's position is represented by the (x_k,y_k) Cartesian coordinates of M in a world frame. The heading angle is denoted θ_k. If the road is perfectly planar and horizontal, and if the motion is locally circular, the motion model can be expressed as (EL Najjar & Bonnifait, 2005):

$$X_{k+1} = \begin{cases} x_{k+1} = x_k + d_s.\cos(\theta_k + \frac{\omega_\theta}{2}) \\ y_{k+1} = y_k + d_s.\sin(\theta_k + \frac{\omega_\theta}{2}) \\ \theta_{k+1} = \theta_k + \omega_\theta \end{cases} \tag{1}$$

Where d_s is the length of the circular arc followed by M and ω_θ is the elementary rotation of the mobile frame. These values are computed using the (ABS) measurements of the rear wheels. Let denote X_{k+1} the state vector containing the pose.

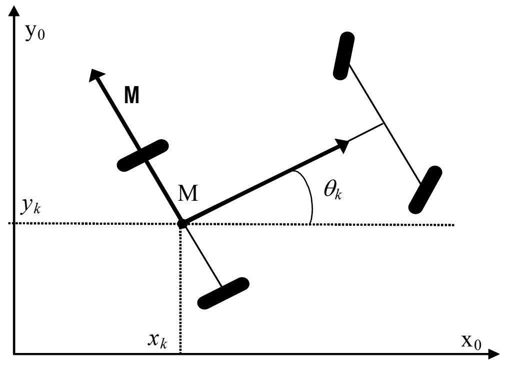


Fig.2. The mobile frame attached to the car

2.2 Cartographical and GPS observation equation

The selection of candidate roads is the first stage of the localization on a road map problem. Generally, this involves applying a first filter which selects all the segments close to the estimated position of the vehicle. The goal is then to select the most likely segment(s) from this subset. Nowadays, since the geometry of roadmaps is more and more detailed, the number of segments representing roads is increasing. The robustness and complexity of the localization depends mainly on the road selection module. In order to be focused on this point, an accurate map Géoroute V2 provided by the French National Institute of Geography (IGN) was used in this work. Our strategy is based on the fusion of several criteria using distance direction measurements within the framework of Bayesian Network. The pose obtained by GPS and odometry can be more accurately estimated by fusing the selected segment. The key idea is to model the fact that the true position of the vehicle is located around the centreline of the most likely road. This region depends mainly on the width of the road, which is an attribute also stored in the database. We suggest using the most likely road in order to build a new observation with its estimated associated error.

In practice, when the GPS satellites signal is blocked by buildings or tunnels, the odometric estimation is used to select the segments all around the estimation from the cartographical database. The cartographical observations can be obtained by projections onto the segments. If the orthogonal projection onto line does not make part of the segment, the closer extremity is used. When several segments are candidates, the cartographical observation function is a non-linear multi-modal observation. Considering a Gaussian distribution of noise to represent the uncertainty zone all around a segment, so the multi-modal observation is a multi-Gaussian observation one. The observation equation of the segment seg_i can be written:

$$Y_{carto}^{segi} = \begin{bmatrix} x_{carto} \\ y_{carto} \\ cap_{carto} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \beta_{carto} \tag{2}$$

Where (x_{carto}, y_{carto}) is the projection onto each segments and cap_{carto} is the segment heading. To represent the error of the cartographical observation, we choose a Gaussian distribution of the uncertainty zone all around the segment. So this error can be represented with an ellipsoid which encloses the road. This ellipsoid has its semi-major axis in the length of the segment and its semi-minor axis equals to the width of the road (EL Najjar & Bonnifait, 2005) (see Fig.3).

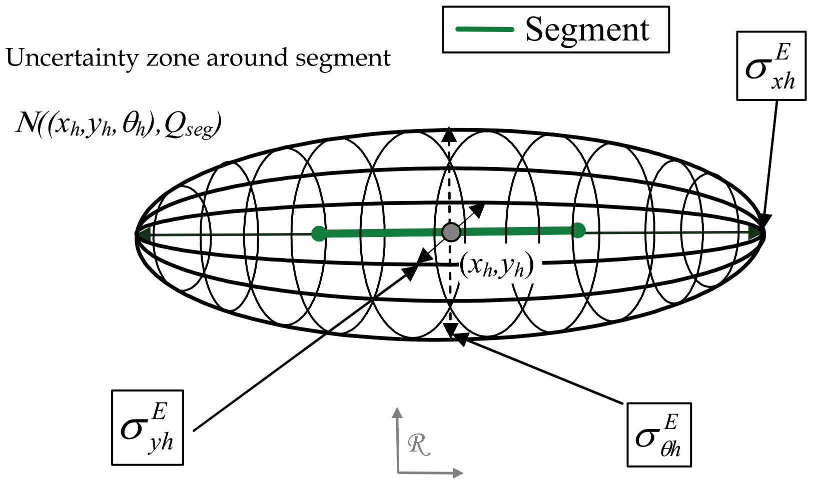


Fig.3. Ellipsoidal of probability construction representing zone around a segment for horizontal segment i.e. parallel with the east axes

The third axis of the ellipsoid represents the uncertainty of the estimation of the segment. This uncertainty is related to the relative error of the cartographical database. The covariance matrix of the cartographical observation error can be written:

$$Q_k^{carto} = \begin{pmatrix} \sigma_{x,h}^2 & Q_{xy,h} & 0 \\ Q_{xy,h} & \sigma_{y,h}^2 & 0 \\ 0 & 0 & \sigma_{\theta,h}^2 \end{pmatrix} \tag{3}$$

The GPS position measurement provides the GPS observation (x_{gps}, y_{gps}). The GPS measurement error can be provided also and in real time using the Standard NMEA sentence "GPGST" given by the Trimble AgGPS132 receiver which has been used in the experiments. The covariance matrix of the GPS error can be expressed as:

$$Q_k^{gps} = \begin{pmatrix} \sigma^2_{x,gps} & Q_{xy,gps} \\ Q_{xy,gps} & \sigma^2_{y,gps} \end{pmatrix}_k \tag{4}$$

The observation equation can be written:

$$Y_{GPS} = \begin{bmatrix} x_{gps} \\ y_{gps} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \beta_{gps} \tag{5}$$

3. Bayesian networks

A Bayesian network can be defined as a pair G= (S, Φ), where S is a directed acyclic graph and Φ is a parameterization of a set {P(X$_1$|Π$_1$),…, P(X$_n$|Π$_n$)} of conditional probability distributions, one for each variable, and Π$_i$ is the set of parents of node X$_i$ in S. Being directional and acyclic the structure of a Bayesian network provides a direct factorization of the joint probability distribution as (Castillo et al., 1997):

$$P(X_1,\ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid \Pi_i)) \tag{6}$$

The notation used in this work is adopted from (Cowell et al., 1999), (Murphy 2002) and (Murat 2001) where round nodes were used to denote continuous random variables and square nodes denote discrete random variables. Throughout this paper, X$_i$ denotes a continuous or discrete random variable. Values of the random variable will be indicated by lower case letters as in x$_i$. For a discrete variable that take r values, x_i^k denote a specific assignment for 1≤ k ≤ r. A set of variables is denoted in boldface letters **X**={X$_1$,…,X$_n$}.

3.1 Inference engine

An important issue in Bayesian networks is computation of posterior probabilities of some variables given observation. The inference problem in general Bayesian networks is still a hot research topic (Heckerman 1995). Several researchers have developed exact and approximate inference algorithms for different distributions (Murphy 2002). The most commonly used exact inference algorithm for discrete Bayesian networks is known as the JLO algorithm (Jensen et al., 1990). The JLO algorithm is a recursive message passing algorithm that works on the junction tree of the Bayesian network. The junction tree is constructed from the directed acyclic graph using some graph-theoretic tools. In the following we propose to detail an example of a discrete Bayesian network. A complete definition using continues and hybrid Bayesian networks are given in (Cowell et al., 1999); (Jensen, 2001) and (Murphy, 2002).

3.2 Constructing the junction tree

In order to define the junction tree we need to define the term clique. A clique is a complete set of nodes which is not a proper subset of another complete set (Castillo et al., 1997). A set of nodes is said to be complete if every pair of nodes in the set is linked. Fig.4 (a) contains the following cliques: C$_1$={A,B}, C$_2$={B,C}, C$_3$={C,D}, C$_4$={D,H}, C$_5$={D,E,G}, C$_6$={E,F,G} and

C_7={A,E}. However, if we add some extra links to the graph, some of the previous maximal complete sets are no longer maximal, and the graph contains different cliques. For example, the graph in Fig.4 (b) is obtained by adding three links to the graph in Fig.4 (a). The sets C_1, C_2, C_3, and C_7 are no longer complete. Thus, the graph in Fig.4 (b) contains five cliques: C_1={A,B,D,E}, C_2={B,C,D}, C_3={D,H}, C_4={D,E,G}, and C_5={E,F,G}.

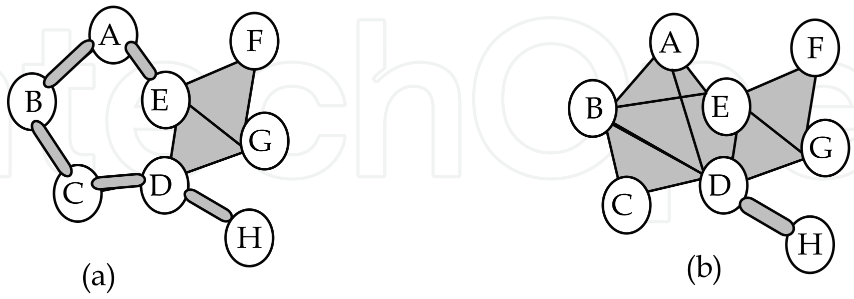


Fig.4. Examples of cliques associated with two different graphs

A junction tree is a tree of cliques that satisfies the running intersection property (RIP). RIP implies that if a node is contained in any two cliques, then it is contained in all the cliques in the unique path between them (see example in 3.2.3).

3.2.1 Moralization

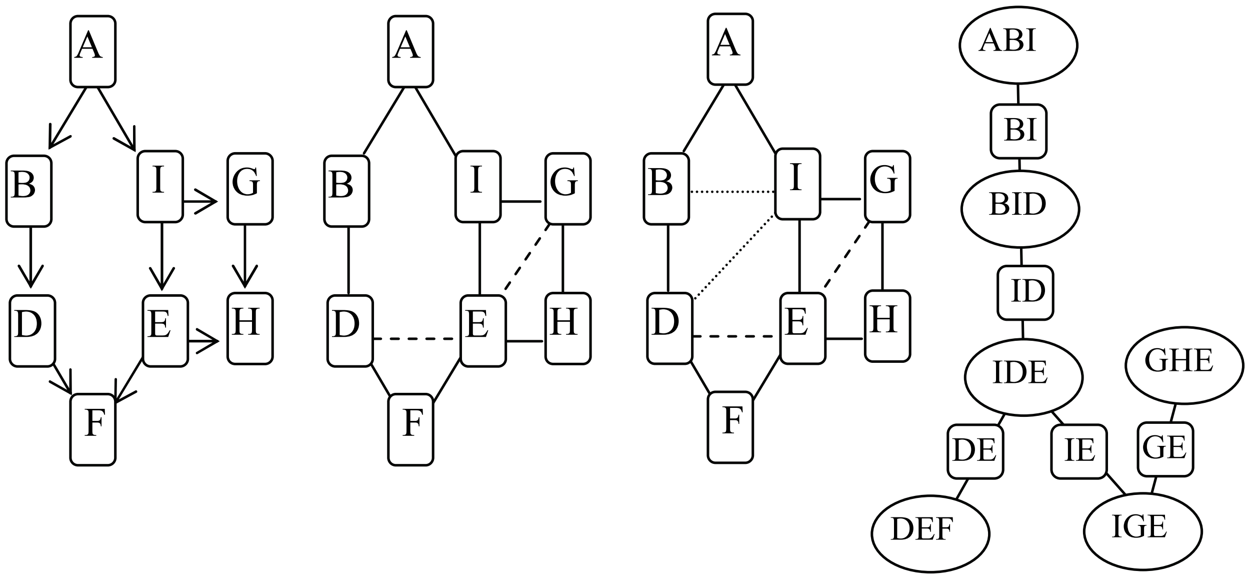
The moral graph of a directed acyclic graph is obtained by introducing additional undirected edges between any two nodes with a common child and subsequently replacing all directed edges with undirected ones (see Fig. 5 (b)). The moralization process guarantees that a family of nodes (a node with all of its parents) will occur together in a clique (Castillo et al., 1997).

3.2.2 Triangulation

The second step to construct a junction tree is to add sufficient edges to the moral graph to obtain a triangulated (chordal) graph. The aim of triangulation is to obtain a decomposable model such that the joint probability distribution can be factorized over the clique potential function (Cowell et al., 1999). An undirected graph is triangulated if every loop of length four or more $X_1 - X_2 - \ldots - X_{n-1} - X_1$ has at least one chord i.e. a link $X_i - X_j$ between two non-consecutive nodes X_i and X_j. If the required chords are not already in the set of edges, they are added, in order to get a triangulated graph. The triangulation process is not unique. In general it is desired to obtain a triangulation with a minimum number of additional edges. The additional edges have a major influence on the time complexity of the inference process (Yannakakis, 1981); (Kjaerulff, 1990). Fig.5 (c) is a trivial example for a triangulated graph. The moral graph (Fig.5 (b)) has a loop of length six: A—B—D— E— I—A. To get a triangulated graph, the links B—I and I—D can be added. As the triangulation process is not unique we can add the link B—E instead of I—D.

3.2.3 Junction tree

It is now possible to identify subgraphs, where all nodes are pairwise linked. Maximal subgraphs with this property are called cliques and are used as nodes afterwards in junction tree. The cliques of Fig.5 (C) are : {A,B,I}, {B,I,D}, {I,D,E},{D,E,F}, {I,G,E}, and {G,H,E}. To construct a junction tree the cliques are organized in a special way, so that all cliques C_i on a path $C_s - C_k - \ldots C_n - C_e$ between the start clique C_s and the end clique C_e contain the nodes of the intersection between C_s and C_e. Formally, $(C_s \cap C_e) \subseteq C_i, \forall C_i \in C_s - C_k - \ldots - C_n - C_e$. This property is known as the running intersection property (Cowell et al., 1999). According to (Jensen, 2001), there is always a way to organize the cliques of a triangulated graph into a junction tree. For inference purposes additional nodes containing the random variables in the intersection of two neighboured cliques are added (Cowell et al., 1999). These additional nodes are called separators (see Fig.5 (d)). The junction tree represented by Fig.5 (d) satisfies the running intersection property. For example we choose two cliques C_s = {A, B, I} and C_e= {I, G, E} with common nodes "I": {A, B, I} ∩ {I, G, E} = {I}. According to the definition of the running intersection property the variable "I" belong in all the cliques in the unique path between C_s and C_e. Effectively, all cliques and separators between the clique C_s= {A, B, I} and C_e= {I, G, E} contain the variable "I".



(a) Initial graph (b) Moralization (c) Triangulation (d) Junction tree

Fig. 5. Transformation steps from the initial Bayesian network to a junction tree

3.3 Initializing the junction tree

The junction tree should be initialized for use it in the inference of Bayesian network (computation of the posterior probabilities of some variables given observation). To enable the calculation of the distributions, tables are attached to each clique and separator of the junction tree, similar to the conditional probability tables of a Bayesian network. These tables are called potentials (Cowell et al., 1999), denoted by ψ_C and ψ_S, e.g. the potential of a clique C is denoted by ψ_C and potential of a separator S is denoted by ψ_S. Given the conditional probability distributions $P(X_i | \Pi_i)$ of the variables X_i (or $P(X_i)$ if there are no parents) the initialization of the junction tree can be performed as follows.

First, assign each variable X_i to just one clique that contains Π_i. The moralization process guarantees that a node X_i with all of its parent Π_i will occur together at least in a clique. For all instance of this variable affect 1. For example, at the clique {A, B, I} (see Fig.5 (d)) we assign the variables A, B and I. On the other hand, we assign just D at clique {B, I, D} because: the variable B was affected already to {A, B, I} and the other hand Π_B= {A} $\not\in$ {B, I, D}.

Second, for each clique C that is assigned with at least one variable define the potential function as the product of P(X_i | Π_i) over all X_i assigned to C. For all separators and remaining cliques define the potential function to be 1. Table 1 summarizes the process of initializing of junction tree given by Fig.5 (d).

Cliques	Assigned variables	Potential cliques
{A, B, I}	A, B, I	$\psi_{\{A, B, I\}}$= P(A) . P(B | A) . P(I | A)
{B, I, D}	D	$\psi_{\{B, I, D\}}$= P(D | B)
{I, D, E}	E	$\psi_{\{I, D, E\}}$= P(E | I)
{D, E, F}	F	$\psi_{\{D, E, F\}}$= P(F | D, E)
{I, G, E}	G	$\psi_{\{I, G, E\}}$= P(G | I)
{G, H, E}	H	$\psi_{\{G, H, E\}}$= P(H | E, G)

Table 1. Cliques associated to initial directed acyclic graph (Fig.5 (a))

Associating a separator potential function to each separator set, we have the following factorization of the joint probability density (Cowell et al., 1999); (Murphy, 2002):

$$P(X_1,\dots,X_n)=\frac{\prod_{c\in C}\Psi_c(X_C)}{\prod_{s\in S}\Psi_s(X_S)} \tag{7}$$

3.3.1 Flow of information between adjacent cliques

The junction tree potential does indeed satisfy the equation (7), but it might not be consistent. Currently, comparing cliques they might tell you different stories about the distributions on certain variables. For instance, consider two adjacent cliques C_i and C_j with separator S, and both contains the variable "X". Marginalizing on both C_i and C_j to get "X", might not give the same result. The reason for this is that the individual cliques have not yet been affected by the information of other cliques in the tree. The information in the tree has to be distributed "equally" on. The global propagation algorithm performs a series of local manipulations that makes the junction tree locally consistent. To ensure consistency of the junction tree, messages are passed between the cliques of the junction tree. This results in a recalculation of the potentials. A clique C_j is said to absorb knowledge from a clique C_i, if the separator S between C_i and C_j gets as new potential Ψ_S^* (Cowell et al., 1999):

$$\Psi_S^*(X_S)=\sum_{C_i\setminus S}\Psi_{C_i}(X_{C_i}) \tag{8}$$

Namely, the separator potential $\Psi_S(X_S)$ is updated by marginalizing the clique potential over the variables that are in C_i but not in S. Then the update factor of the separator is used to update the potential of the destination clique C_j.

$$\lambda_S(X_S)=\frac{\Psi_S^*(X_S)}{\Psi_S(X_S)} \tag{9}$$

The new potential clique of C_j is given by:

$$\Psi^*_{C_j}(X_{C_j}) = \Psi_{C_j}(X_{C_j}).\lambda_S(X_S) \tag{10}$$

In order to distribute the information in each clique to the whole tree a two-phase propagation algorithm is used. Given a root clique in the tree, the collection-phase absorbs the flows (messages) starting from the leafs towards the root. Once all the flows are collected in the root, messages are send towards the leafs in the distribution-phase. After collection-phase and distribution-phase are finished, it is guaranteed that the junction tree is globally consistent. Namely, for all clique C_i and C_j which are connected with a separator S, and both contains the variable "X", marginalizing on both C_i and C_j to get "X", might give the same result.

3.3.2 Entering and propagation evidence

The algorithm of initializing the junction tree can be used to compute posterior distribution given an observation of a subset of variables. Once the clique tree is properly initialized with the observation, the two phase propagation algorithm diffuses the observation and the resultant potentials are the posterior marginal distributions in each clique and separator. The observations are used in the algorithm as follows. Let X^h be the subset of variables that are hidden or unobserved and let D_l= {X_i=x_i, X_j=x_j,...} be an observation of the set of variables D_l =X\X^h.. In order to compute P(X^h| D_l) we first define an evidence function such that:

$$g(x_i) = \begin{cases} 1 & if\ x_i = x_i^* \\ 0 & otherwise \end{cases} \tag{11}$$

After initializing the junction tree with the conditional probability distributions, we multiply each clique potential with the evidence function according to the variable assignment in the initialization step:

$$\Psi_C(X_C) = \Psi_C(X_C).\prod_{i:X_i \in X_C} g(x_i) \tag{12}$$

Once again, we call collection-phase and distribution-phase to propagate this evidence (observation) through the tree to yield the new (posterior) probabilities. The clique potentials are the joint probability of the local hidden variables and the observed evidence:

$$\Psi_C(X_C) = P(X^h, D_l) \tag{13}$$

If the potential function at a clique is normalized to sum to 1, one gets the conditional probability of the local hidden variables given the observation:

$$P(X^h \mid D_l) \tag{14}$$

If the clique potential is marginalized over the hidden variables the probability of the observed evidence is obtained:

$$P(D_l) = \sum_{X^h} P(X^h, D_l) \tag{15}$$

3.4 Bayesian network model for Mono-vehicle localization

Vehicle localization on respect of a road map involves applying a first filter which selects all the segments close to the estimated position of the vehicle. The goal is then to select the most likely segment(s) from this subset. Nowadays, since the geometry of roadmaps is more and more detailed, the number of segments representing roads is increasing. On the other hand, the map is not perfect and presents a known uncertainty. The road classification module is an important stage in the vehicle localization process because the robustness of the localization depends mainly on this stage. In order to take into account the error of several sensors or database used in this application, we introduce a concept which can manage multi-hypothesis in the formalism of Bayesian network.

For each selected segment $Carto_i$, we represent it by a Gaussian: $Carto_i \sim N(\mu_i, \Sigma_i)$. Where $\mu_i=(x_i,y_i,\theta_i)$ is the projection of the estimated position on this segment and θ_i is the heading of the segment. The proposed Bayesian network model for Mono-vehicle localization is illustrated in Fig. 6. In this model we used two hidden variables. The discrete variable Sk represents the segments of which the vehicle can be. The second is continuous variable; $X_k(x_k,y_k,\theta_k)$ represents the estimation of a vehicle for each candidate segment. The graph represented in Fig. 6. allows us to represent causal links between the variables. The continuous variable X_k is updated by observations $Carto_k$ and GPS_k if GPS measurement is available. This variable is multi-modal because it has been updated by the set of candidates segments ($Carto_k$). The discrete variable S_k is update by cartographical observation ($Carto_k$) and the estimation is given by the hidden variable X_k.

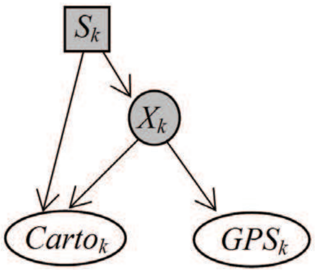


Fig. 6. Bayesian network model for Mono-vehicle localization

Bayesian inference gives us :

1. the probability of each candidate segment given by the posterior probabilité $P(S_k | Carto_k,GPS_k)$
2. for each segment, the estimation of the vehicle's position given by the probability density $P(X_k | Carto_k,GPS_k)$

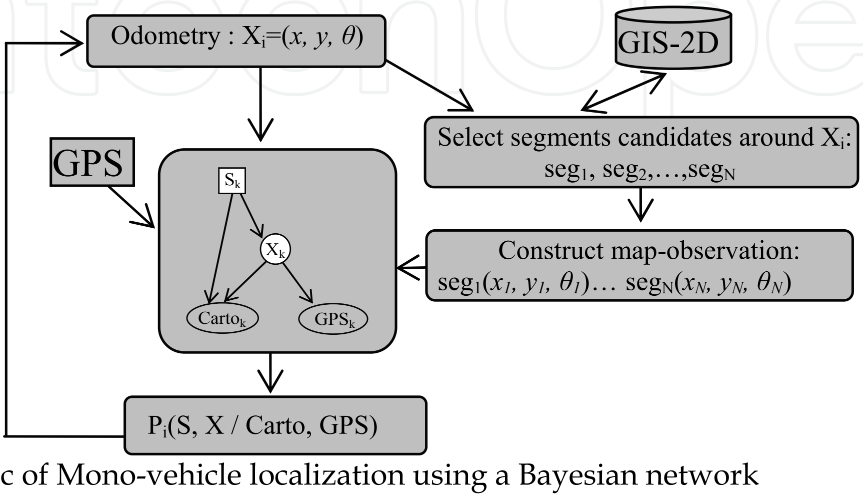

Odometry : $X_i=(x, y, \theta)$
GIS-2D
GPS
Select segments candidates around X_i: $seg_1, seg_2,\ldots,seg_N$
Construct map-observation: $seg_1(x_1, y_1, \theta_1)\ldots seg_N(x_N, y_N, \theta_N)$
S_k
X_k
$Carto_k$
GPS_k
$P_i(S, X / Carto, GPS)$

Fig. 7. Synoptic of Mono-vehicle localization using a Bayesian network

Let us use a specific case study to illustrate the method. In Fig. 8, the vehicle is travelling from the road 1 to road 8. At t=1, we have a predicted pose given by the odometer. We select all segments around the predicted pose. In this case we have one. This segment was used to generate one cartographical observation. Then, estimation is provided using Bayesian network. At t=2, the estimation errors and the digital map errors oblige to select segments 2 and 3 around the predicted pose. These segments were used to generate two observations. Then, using Bayesian network, two estimations were provided and so on, for the rest of the experiment. Table 2 summarizes the process of inference Bayesian.

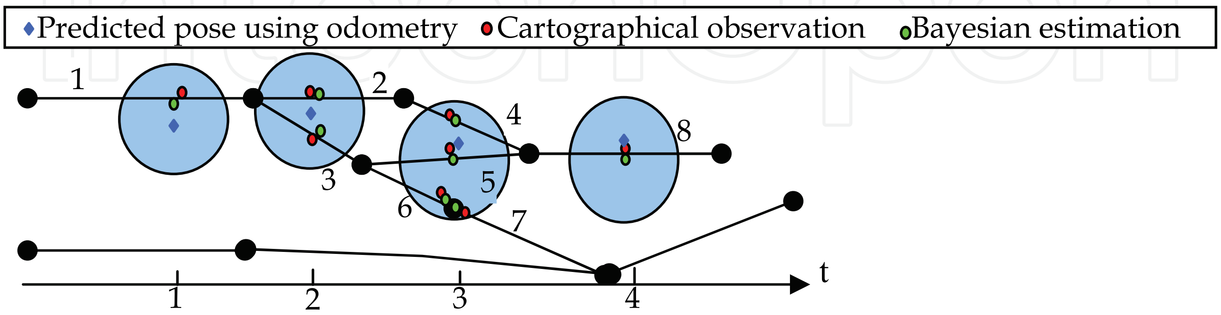


Fig. 8. Formalism of multi-hypothesis managing in Bayesian network

t	Carto	$P(S_t \mid \text{Carto})$	$P(X \mid \text{Carto})$
1	{1}	$P(S_t=\{1\} \mid \{1\})=1$	$X_{\{1\}}=(x_{\{1\}}, y_{\{1\}}, \theta_{\{1\}})$
2	{2,3}	$P(S_t=\{2\} \mid \{2,3\})=0.75$ $P(S_t=\{3\} \mid \{2,3\})=0.25$	$X_{\{2\}}=(x_{\{2\}}, y_{\{2\}}, \theta_{\{2\}})$ $X_{\{3\}}=(x_{\{3\}}, y_{\{3\}}, \theta_{\{3\}})$
3	{4,5,6,7}	$P(S_t=\{4\} \mid \{4,5,6,7\})=0.55$ $P(S_t=\{5\} \mid \{4,5,6,7\})=0.25$ $P(S_t=\{6\} \mid \{4,5,6,7\})=0.15$ $P(S_t=\{7\} \mid \{4,5,6,7\})=0.05$	$X_{\{4\}}=(x_{\{4\}}, y_{\{4\}}, \theta_{\{4\}})$ $X_{\{5\}}=(x_{\{5\}}, y_{\{5\}}, \theta_{\{5\}})$ $X_{\{6\}}=(x_{\{6\}}, y_{\{6\}}, \theta_{\{6\}})$ $X_{\{7\}}=(x_{\{7\}}, y_{\{7\}}, \theta_{\{7\}})$
4	{8}	$P(S_t=\{8\} \mid \{8\})=1$	$X_{\{8\}}=(x_{\{8\}}, y_{\{8\}}, \theta_{\{8\}})$

Table 2. Example of inference process given by Bayesian network

4. Bayesian network model for multi-vehicle localization

In order to move a set of vehicles in train configuration, each vehicle must to know its position. In our approach, we propose that the path of the leader vehicle is propagated in real time to follower's vehicle using wireless communication. Each vehicle in the platoon is equipped with: GPS sensors in order to determine its location and a Lidar to calculate the distance to vehicle in front. Only the leader vehicle is equipped with RTK-GPS and a special localization device. This device should provide geo-position with high accuracy.

Each follower vehicle can transmit its position and velocity. To reproduce the path of the leader vehicle, each vehicle constructs the trajectory of the leader vehicle by linking up the position transmitted by the leader. Then the follower can calculate the lateral and longitudinal controls to be close to the leader vehicle path and to keep a constant distance within vehicles. Follower's vehicles positioning is improved by using the Lidar and propagated measure of leader vehicle. According to Fig.9, the Cartesian coordinates of first follower (f_1) are given by:

$$X_{Vobs}^{f_1} = \begin{cases} \mathrm{X}_{\mathrm{Leader}} \text{ - D. cos } (\beta) \; + \; \mathrm{N}\,(\mu,\sigma) \\ \mathrm{Y}_{\mathrm{Leader}} \text{ - D. sin } (\beta) \; + \; \mathrm{N}\,(\mu,\sigma) \end{cases} \tag{16}$$

D is the distance (given by rangefinder) between the follower and the leader. The same process is carrying out between follower (f_{i+1}) and follower (f_i):

$$X_{Vobs}^{f_i} = \begin{cases} \mathrm{X}_{\mathrm{f_{i-1}}} \text{ - D. cos } (\beta) \; + \; \mathrm{N}\,(\mu,\sigma) \\ \mathrm{Y}_{\mathrm{f_{i-1}}} \text{ - D. sin } (\beta) \; + \; \mathrm{N}\,(\mu,\sigma) \end{cases} \tag{17}$$

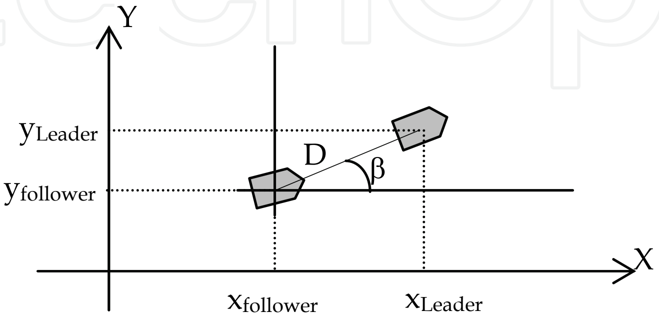


Fig.9. Approving follower's position according to leader's position and rangefinder data

The proposed Bayesian network model for multi-vehicle localization is illustrated in Fig.10. In this model we used the continuous hidden variable $X_{S_i}(k+1) = (x_{S_i}^{k+1}, y_{S_i}^{k+1}, \theta_{S_i}^{k+1})$ for each follower vehicle to estimate its position. This variable is updated by the observations $X_{gps\_Si}(k+1)$ and $X_{Vobs\_Si}(k+1)$ and depends on the precedent state $X_{Si}(k)$ and law command $U_{Si}(k)$. Finally, the law command (lateral and longitudinal) depends on the trajectory of leader the vehicle $X_L(k+1) = (x_L^{k+1}, y_L^{k+1}, \theta_L^{k+1})$.

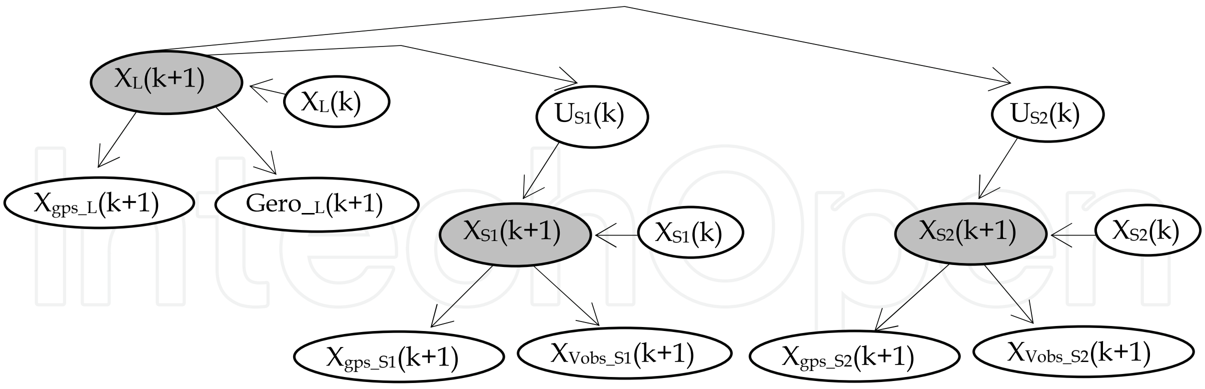


Fig.10. Bayesian network model for Multi-vehicle localization

Each vehicle has to know its position to derive an adequate control law which allows it to:

- preserve a constant distance (DistS) to the vehicle in front
- reduce lateral distance (DistL) between the trajectory of the follower and the leader one (Fig.11).

Each vehicle's position is represented by the (x_k,y_k) Cartesian coordinates of M. The heading angle is denoted θ_k. The motion model can be expressed as:

$$X_{k+1} = \begin{cases} x_{k+1} = x_k + d_s.\cos(\theta_k + \frac{\omega_\theta}{2}) \\ y_{k+1} = y_k + d_s.\sin(\theta_k + \frac{\omega_\theta}{2}) \\ \theta_{k+1} = \theta_k + \omega_\theta \end{cases} \tag{18}$$

Where d_s is the length of the circular arc followed by M and ω_θ is the elementary rotation of the mobile frame.

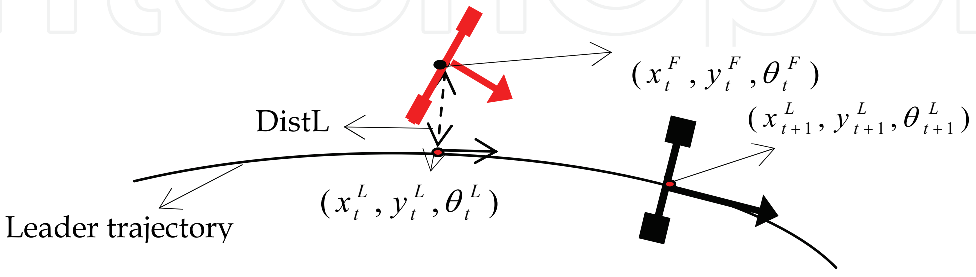

$(x_t^F, y_t^F, \theta_t^F)$
DistL
$(x_{t+1}^L, y_{t+1}^L, \theta_{t+1}^L)$
$(x_t^L, y_t^L, \theta_t^L)$
Leader trajectory

Fig.11. Model for leader vehicle with one follower

5. Experiment results

A test trajectory for a mono-vehicle localization has been carried out at Compiègne in France with an experimental vehicle. The used GPS is a differential Trimble AgGPS132 receiver. For odometry, we have used the ABS sensors of the rear wheels of the experimental vehicle.
For multi-vehicle localization, a test trajectory was carried out at the place Stanislas at Nancy-France for the leader vehicle. For followers' vehicle GPS and Lidar are simulated by adding Gaussian noise for followers' vehicles.

5.1 Performance of mono vehicle localization method

The test trajectory is presented in Fig.12. In this experience, the GPS measurements were available in the beginning of the test trajectory. Then, the GPS was not used for 1.5Km. One can remark that in spite of the long GPS mask, the vehicle location is matched correctly. As a matter of fact, the final estimated positions stay close to the GPS points. In Fig.12, we only presented the most probable Bayesian network estimation of the pose.
In second experiment (see Fig. 13), we want to show how the Bayesian network handles and treats ambiguous situations (junction road and parallel road). To simulate satellite masks occurring in urban environments we remove GPS data and put them back again. In the first situation, GPS was not available before the junction. One can see that the method manages two hypotheses (two segments) for three steps then wrong hypothesis was eliminated by presence the GPS. One can remark that among the different hypotheses the good segment was always given the highest probability by the Bayesian network inference. On the same figure, a second ambiguous situation appears: close and parallel road. The method detects the ambiguity of this situation and selects all probable segments in this parallel roads. The Bayesian network manages all hypotheses until the elimination of the ambiguity. One can remark that in spite the ambiguity the road on which the vehicle is running presents the highest probability.

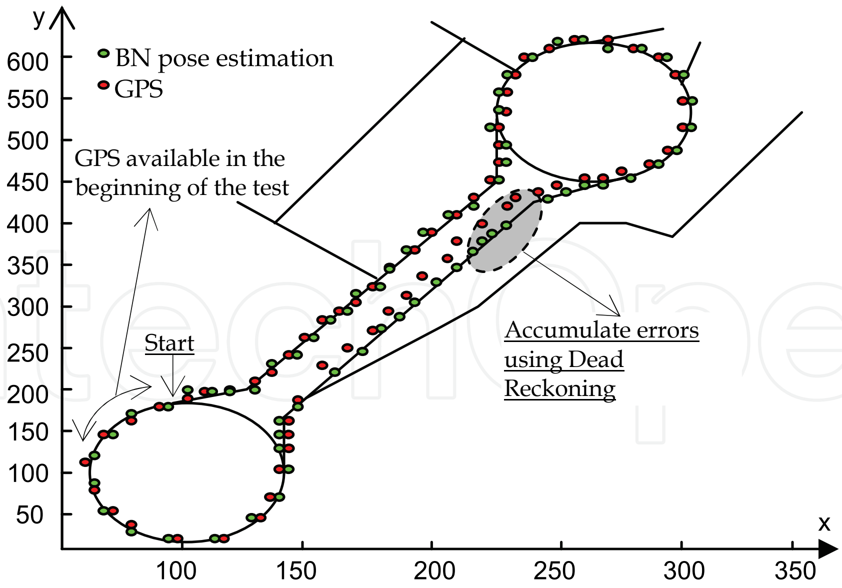


Fig.12. Pose estimation with Bayesian network using odometry and cartographical observation (GPS was masked)

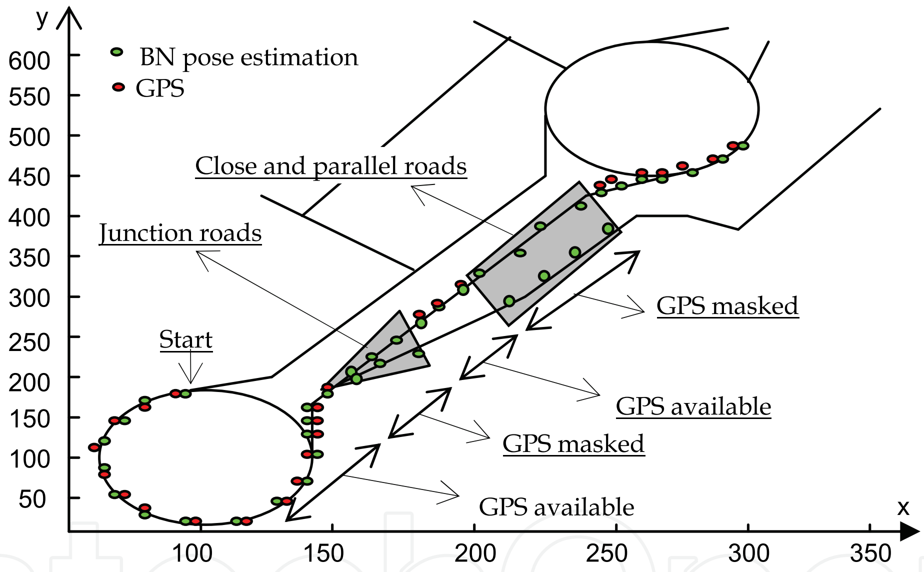


Fig.13. Multi-hypothesis managed with Bayesian network to treat junction road and parallel road situation

5.2 Performance of multi-vehicle localization

The real trajectory of the vehicle leader is presented by the green line on Fig.14. This trajectory represents the centimetric precision GPS position. These geo-positions are provided by the THALES Sagitta02 which is a centimetric GPS LRK. In this experiment, we assume that all of two follows vehicles (blue and black respectively) are initially parked on the reference path and not necessarily have the same leader's heading. Followers dispose by WiFi and in real time the path of the leader vehicle.

The leader's path, given by centimetric GPS-LRK, is plotted by green line in Fig.14. The figure 15 shows the time evolution of the vehicles positions provided by the proposed approach. Thus followers' vehicles are supposed to be equipped by low cost GPS sensors (3 meters accuracy) for localization task and a rangefinder to determine distance between vehicles. The obtained followers' vehicles paths are plotted in Fig.15 by blue path for first follower and red path for second one.

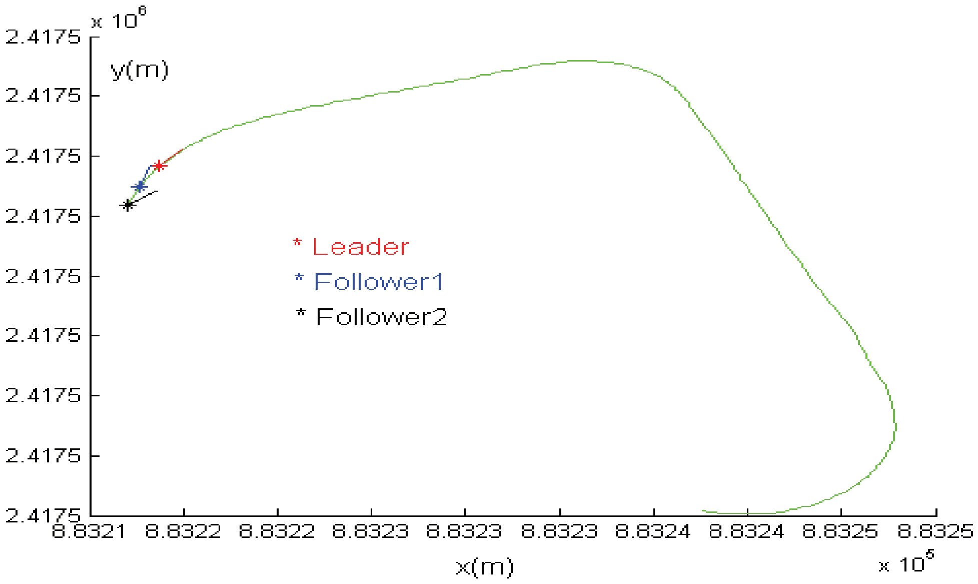


Fig.14. Leader's trajectory and initial position of vehicles

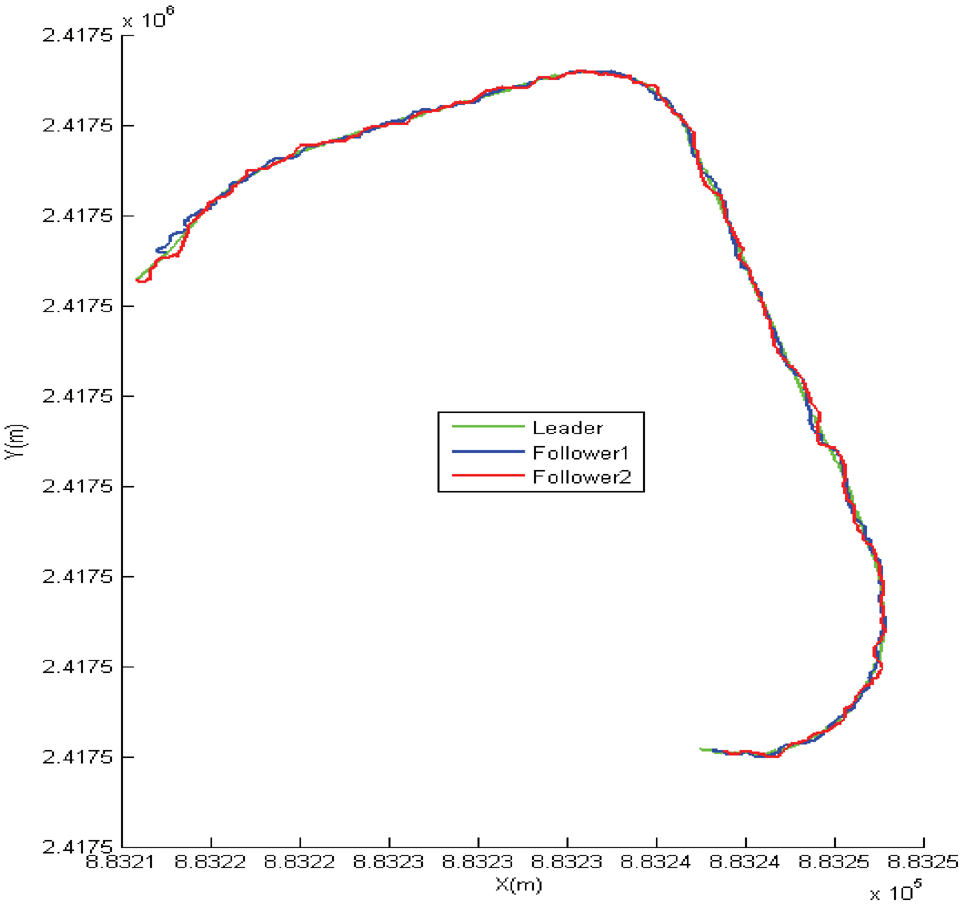


Fig.15. Followers' trajectory given by Bayesian network compared to leader's trajectory

One can remark that the followers' paths shown in figure 15 present oscillations all around leader path. We believe that oscillations come from the used control law. In this work, a simple proportional law control is used.

Figure 16 show the distance between vehicles. Initially, inter-distance between vehicles is chosen to be for about 2 meters. In this simulation, the desired security distance between vehicles is chosen to be 1 meter. Namely, the objective is to control vehicles in order to respect this distance. It can be noticed that the chosen control law in each vehicles commands the vehicle velocity in order to respect the desired security distance. One can remark that the security distance is generally respected.

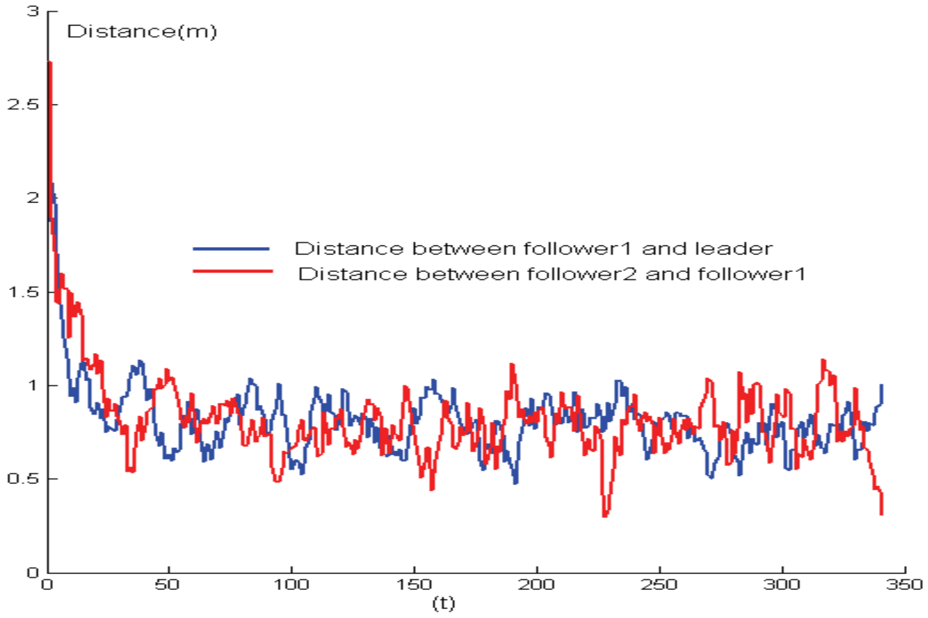


Fig.16. Distance between vehicles

7. Conclusion

This article has presented a multi-sensor fusion method for vehicle localization. The main contributions of this work are the formalization of a multi-sensor fusion method in the Bayesian Network context and an experimental validation with real data.

An interesting characteristic of this approach is that it is flexible and modular in the sense that it can easily integrate other sensors. This feature is interesting because adding other sensors is a way to increase the robustness of the localization.

In this approach, the use of the digital map as an observation of the state space representation has been introduced. This observation is used in the Bayesian Network framework in the same way that the GPS measurement. It turned out in the experiments that the GPS measurements are not necessary all the time, since the merging of odometry and roadmap data can provide a good estimation of the position over a substantial period. The strategy presented in this paper doesn't keep only the most likely segment. When approaching an intersection, several roads can be good candidates for this reason we manage several hypotheses until the situation becomes unambiguous.

Multi-vehicle localization method presented in this work can be seen like an extension of the Mono-vehicle method, in the sense that we have duplicate the (BN) used to fuse measurements sensors to localize one vehicle for several ones. Then, we have added vehicles inter-connection to represent finally the train of vehicles in the context of Bayesian network. The multi-sensor fusion of leader's vehicle measurements with the Lidar measurement in Bayesian network formalism can provide continuous and accurate geo-position of followers' vehicles. Thus this data fusion method allows computing an accurate follower's position without using an expensive sensor on each follower's vehicle.

The proposed method for multi-sensors fusion for multi-vehicle localization in train configuration permits to implement control law based on near-to-near approach, which can only be seen as a first step in platoon control design. For both proposed approach, real data was used in this work to test and quantify the quality of results.

8. Future work

In this work, we have not treated the control law of the follower vehicles. The perspective of this work is to use the tricycle model validated by numerous laboratories (Daviet & Parent); (Thuilot et al., 2004). The tricycle model allows us to manipulate the curvilinear distance instead of rectilinear distance. The main advantage of curvilinear distance is that it agrees with the distance travelled and is perfectly consistent when following reference paths with high curvature (which is not the case with rectilinear distance) (Bom et al., 2005). Using curvilinear distance allows us to reduce the difference between follower's trajectory and leader's one. Moreover, platoon control design relies on nonlinear techniques, instead of control approaches based on linear approximations. Since no approximation is achieved, performances provided by the nonlinear control law are more satisfactory and more robust than those offered by linear control. This control approach allows to fully decouple longitudinal and lateral controls (Bom et al., 2005).

9. References

Abbott, E. & Powell, D. (1999). Land-Vehicle Navigation using GPS. *Proceedings of IEEE, vol.87, N.1*

Bom, J.; Thuilot, B.; Bom, J.; Marmoiton, F. & Martinet, P. (2005). Nonlinear Control for Urban Vehicles Platooning, Relying upon a Unique Kinematic GPS. *Proceedings of the 2005 IEEE International Conference*

Castillo, E.; José M. G. & Ali, S. H. (1997). *Expert Systems and Probabilistic Network Models,* Publisher, ISBN, New York Berlin Heidelberg

Cowell, R. G.; Dawid, A. P.; Steffen, L.; & David, J. S. (1999). *Probabilistic Network and Expert Systems,* Publisher, ISBN, New York Berlin Heidelberg

Daviet, P. & Parent, M. (1995). Platooning for small public urban vehicles, *In 4t hIntern. Symposium Experimental Robotics, pp 345-35417*, Stanford, CA (USA), July

Daviet, P. & Parent, M. (1996). Longitudinal and Lateral Servoing of Vehicles in a Platoon. *In Intelligent Vehicles Symposium*. Tokyo Japan

Dissanayake, M.W.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; & Csorba, M. (2001). A Solution to the Simultaneous Locakization and Map Building (SLAM) Problem. IEEE Transaction on Robotics and Automation, Vol. 17, N0.3

EL Najjar, M. E. & Bonnifait, P. (2003). A Road Matching Method for Precise Vehicle Localization using Belief Theory and Kalman Filtering. *IEEE/EJS/ISR 11th Int. Conference on Advanced Robotics, pp.1677-1682*

EL Najjar, M. E. & Bonnifait, Ph. (2005). To wards an Estimate of Confidence In a Road-Matched Location. IEEE International Conference on Robotics and Automation

Heckerman, D. (1995). A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division

Judea, P. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Publisher, ISBN, San Francisco

Jensen, F. V.; Lauritzen, S. L.; Steffen, & Olsen, K. G. (1990). *Bayesian updating in recursive graphical models by local computations,* Computational Statistics and Data Analysis Publisher, ISBN

Jensen, F. V. (2001). *Bayesian networks and Decision Graphs,* Statistics for Engineering and Information Science. Springer, Berlin, Heidelberg

Kjaerulff, U. (1990). Triangulation of graphs-algorithms giving small total space. Department of Mathematics and Computer Science. Institute of Electronic Systems. Aalborg University

Murat Deviren (2001). "Structural Learning of Dynamic Bayesian Networks in Speech Recognition". Technical Report.

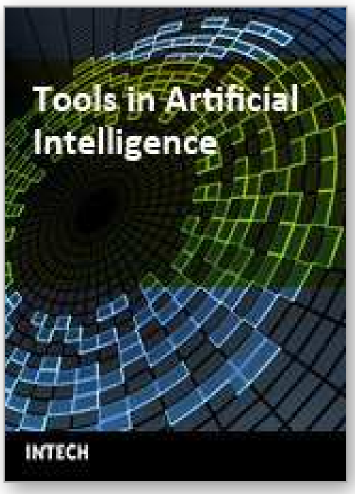
Murphy, K. P. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, UC Berkley, Computer Science Division

Thrun, S.; Fox, D.; Burgard, W. & Dellaert, F. (2001). Robust Monte-Carlo localization for mobile robots. *Journal of Artificial Intelligence (AI). Vol. 128, No 1-2. pp 99-141*

Thuilot, B.; Bom, J.; Marmoiton, F. & Martinet, P. (2004). Accurate automatic guidance of an urban electric vehicle relying on kenematic GPS sensor. *in 5th IFAC Symposium on Intelligent Autonomous Vehicles (IVAC'04),* July

Yannakakis, M. (1981). Computing the minimal fill-in in np-complete. *SIAM Journal on Matrix Analysis and Applications,* Vol. 2, Issue 1, page 77-79

Zhao, Y. (1997). *Vehicle Location Navigation Systems,* Artech House Publishers



Tools in Artificial Intelligence

Edited by Paula Fritzsche

ISBN 978-953-7619-03-9
Hard cover, 488 pages
Publisher InTech
Published online 01, August, 2008
Published in print edition August, 2008

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821