Distributed to Embedded Bayesian Network for Diagnosis of a Networked Robot

Insaf Sassi, Alexia Gouin, Jean-Marc Thiriet

Distributed to Embedded Bayesian Network for Diagnosis of a Networked Robot

Insaf Sassi, Alexia Gouin, Jean-Marc Thiriet
Univ. Grenoble Alpes, CNRS, GIPSA-Lab, F-38000 Grenoble, France
Email: firstname.name@gipsa-lab.grenoble-inp.fr

***Abstract*—Mobile networked robots are distributed systems controlled by a distant station i.e, the controller is implemented on a control station. The main mission of the mobile robot is reaching a target position starting from an initial one while receiving the instructions from the control station via a wireless network. The wireless network is characterized by a stochastic behavior and is sensitive to perturbations. The unreliability of wireless networks does not guarantee data transmission between system components (the robot and the station) which can cause system performance degradation. A distributed Bayesian Network (BN) was proposed in previous work to monitor and diagnose the system performance and to model causal uncertainties between failures. The developed BN is a modular Bayesian Network (MBN) which is composed of three Bayesian modules shared between the robot and the control station. In the case of system performance degradation because of a bad network state, the robot switches to embedded controller implemented on-board (autonomous behavior). The distributed diagnosis architecture must be updated and the diagnosis tool becomes embedded on the robot. A procedure of BNs assembly is described in this work in order to implement one monolithic BN on-board. The obtained monolithic BN is the result of combining two Bayesian modules from the modular Bayesian Network: the control and the operative Bayesian modules.**

I. Introduction

A Wireless Networked Control System (WNCS) is composed of a number of distributed nodes (controllers, sensors and actuators), each of which collaborates in order to accomplish a mission. The communication between the WNCS components is insured by a wireless network. Such a network presents a number of advantages such as flexibility, lack of wiring, mobility and fast deployment [6]. However, the stochastic behavior of such networks does not guarantee a reliable communication between nodes, which can lead to problems that affect the Wireless Networked Control System (WNCS) performance as proved in [17]. Placing a wireless network between the WNCS nodes i.e, integrating it into the control loop, can generate performance and quality of control (QoC) degradation and instability because of the network quality of service (QoS) degradation [17], [8]. Networked mobile robots, as WNCS, have been increasingly deployed in a large number of applications. Such systems add more challenges like loss of connectivity because of the robots mobility, memory and calculation limitations and energy consumption constraints. A distributed diagnosis has become essential for a system having the constraints and problems explained above. A Bayesian Network (BN) for diagnosis of wireless networked mobile robot was consequently proposed in our work [14] to avoid performance degradation. The mobile robot is controlled via a wireless network by a control station in order to achieve a target solution. The developed BN is characterized by a multilayer structure that many works [1], [2], [13], [16] have proved its efficiency in a diagnosis procedure. Besides, the BN is distributed between the system nodes to have a Modular Bayesian Network (MBN). This distributed architecture is adopted to reduce the on-board processing overhead (i.e. load calculations, memory consumption) on the robot and because of the incomplete vision about the environment from both robot and station sides. The developed MBN is able to determine the cause of Quality of Control (QoC) degradation for a networked mobile robot, whether it comes from the robot internal state, the network or the controller. Based on the QoC state and the network Quality of Service (QoS) state, the remote station can order the robot to switch to its embedded controller to maintain the system in a good state. In this case, the robot is no more connected to the station. It becomes independent of the station until it finds a new one that guarantees a good QoS and potentially a better QoC. The network state does not impact the system performance after switching to embedded controller. The diagnosis architecture must be updated to be conform to the system structure and behavior. There are three Bayesian modules, in the proposed MBN, communicating via virtual linking between shared nodes [4]: the control module, the operative module and the network module as explained in [14]. When the robot becomes autonomous, the network module does not take part in the diagnosis procedure anymore.

In this paper, we propose a procedure to assembly two modules (control and operative modules) of the developed MBN to be implemented in a monolithic BN structure on the robot after switching to embedded controller. This diagnosis architecture update must also take into consideration QoS node elimination from the QoC parents set as we will explain later. This structure update is a solution to save the information about the robot and its mission until it finds a new control station.

The case study is a simulated networked mobile robot moving toward a target position. When the networked robot switches to the embedded controller (autonomous behavior), the QoS node is removed from the control Bayesian subnet. The control Bayesian module and the operative Bayesian module are assembled and combined in one Bayesian structure implemented on the robot. The assembly procedure is described in this work.

The paper is organized as follows. Section 2 presents some

basics review of BNs and MBN. Section 3 describes the procedure of BNs assembly. Section 4 is dedicated to diagnosis architecture reconfiguration procedure (from distributed to embedded structure). Finally, conclusions and perspectives are given in section 6.

II. Background

A. Bayesian Networks

Definition: BNs are acyclic directed graphs representing probabilistic causal dependencies between random variables [12]. These dependencies are a set of edges linking the nodes corresponding to a set of variables (discrete/continuous). The causal relationship is represented by the conditional probability of the variable knowing the state of its parents nodes. Let $G = (V, E)$ be an acyclic directed graph corresponding to a BN where V is the set of random variables associated to the graph nodes and E is the set of directed edges. The joint probability distribution in a BN is expressed as following [10]:

$$P(V_1, V_2, .., V_n) = \prod_{i=1}^{n} p(V_i | pa(V_i)) \quad (1)$$

where $pa(V_i)$ is the set of V_i parents.

D-separation property: the D-separation property describes the conditional independence between the BN variables. Let X, Y and Z a set of variables in G. X and Y are d-separated by Z ($X \perp Y|Z$) means that Z blocks the information path between X and Y. It implies that knowing the state of X (apply a hard evidence on X) does not change our knowledge about Y state. For all paths between X and Y in serial or diverging connection, if Z undergoes a hard observation, the information circuit between X and Y is blocked. Meanwhile, in case of converging connection, if neither Z nor any of its descendants has received evidence, the information path is not open between X and Y.
The d-separation property allows to reduce the computation time in the inference algorithm in BNs which is useful for inference optimization in MBNs.

B. Modular Bayesian Network

MBN is the efficient solution for diagnosis of physically distributed systems and is adequate to model and map between observations coming from different domains [11]. Local causal dependencies related to a specific sub-domain can be represented by local BN which is defined as a BN module. The set of BN modules, collaborating with each other for the same objective, is called MBN. The contribution in [11] proposed a modular bayesian structure whose BN modules communicate with posterior marginal probabilities via shared nodes. This solution reduces constraints and dependencies between BN modules in the inference procedure unlike the Multiply Sectioned Bayesian Network (MSBN) in [19]. However, the inference policy enforces a unique direction of beliefs sharing making BN modules in an hierarchical structure [11]: each BN module is inferred sequentially from the low level to the high level [5]. Another work [4] proposes a MBN as a solution for complex and large scale systems in order to reduce computational complexity. It is designed for mobile environment and devices with resource limitations. MBN, as defined in [4] and then in [5] and [7], guarantees more extensibility and efficiency. Besides, it is possible to reuse BN modules in different applications.

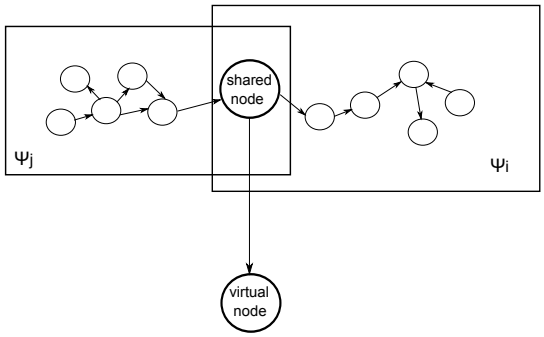

Fig. 1. Modular Bayesian Network

MBN is a distributed Bayesian structure obtained by dividing one large BN into several Bayesian modules using the modularization procedure detailed in [4]. The modularization consists in defining the modules by selecting the corresponding variables to each domain. An acyclic sub-graph is then built for each module based on the global acyclic graph of the monolithic BN. Each obtained BN module is thus a sub-graph of the monolithic BN. A Bayesian Module has the same corresponding conditional probability tables (CPTs) definition as the monolithic BN. Every Bayesian module (Bayesian sub-net) presents a sub-domain of the global system and shares variables (set of nodes) with other Bayesian subnets. Finally, MBN Ω is defined as following [5]:

- 2-tuple (Ψ, R) where Ψ is the set of BN modules in Ω and R represents the causality between the BN modules.
- Two modules $\Psi_i = ((V_i, E_i), P_i)$ and $\Psi_j = ((V_j, E_j), P_j)$, where P_i and P_j correspond to the local probability distributions of Ψ_i and Ψ_j respectively, have influence on each other if there is a link $R = \{(\Psi_i, \Psi_j) | i \neq j, V_i \cap V_j \neq \emptyset\}$ representing a causal-effect dependency between two modules as shown in figure 1.

The d-separation property between BN modules is similar to the d-separation concept between nodes in a BN for the three types of connection. Ψ_i and Ψ_j are d-separated by the shared node S means that any changes in Ψ_i or Ψ_j do not affect the other connected module. BN inference algorithm is used for inference in MBN structure. The inference in MBN is a two-step inference. The first step consists in applying hard evidence in every Bayesian Module. The result of the first step inference is then communicated between the connected modules. Virtual nodes are added as a child to every shared node to pass information between the BN modules as explained in [5]. The virtual node has two possible states; yes and no and its CPT is defined according to the received soft evidence of the shared variable. A hard evidence is then applied on the virtual

node and the BN module performs a second inference. If two connected modules Ψ_i and Ψ_j are d-separated, there is no need to re-inference the other module e.g Ψ_j because the changes in Ψ_i do not affect Ψ_j.

III. Bayesian Networks Assembly

In order to assembly BN modules into one monolithic BN, we have to verify that the assembled BN can satisfy globally correct inference. Our assembly procedure is inspired from the work in [11] that consists in checking the global correct inference to construct its MBN. We use the method of [11] not to build the MBN but to assembly the BN modules of our previous work [14] developed using the algorithm construction of MBN in [5]. The MBN construction procedure as described in [5] must guarantee that the corresponding monolithic BN is an acyclic graph and the algorithm in [11] ensures a global correct inference by constructing a correct inference graph from the BN modules. The assembly procedure adds the dependencies constraints that [5] simplifies in the distributed MBN. Two Bayesian modules can be assembled into one monolithic BN if they construct an inference graph that satisfies coherent message passing (inference algorithm). Let $C = \{C_1, C_2, ..., C_n\}$ be the set of clusters each corresponding to a BN module in the MBN and $S = \{S_{i,j} | 1 \leq i \leq n, 1 \leq j \leq n, i \neq j\}$ devote the set of separators between the clusters where $S_{i,j} = C_i \cap C_j \neq \emptyset$ as defined in [11]. The Bayesian Modules communicate via the shared nodes $(V_i \cap V_j)$ as explained above by passing marginal posterior probabilities. The inference graph $G_{inference} = (C, S)$ is by definition [11] a graph that contains all the set of separators corresponding to shared nodes passing the posterior marginal probabilities between BN modules. Let Ψ_i and Ψ_j be two BN modules to be assembled, corresponding to DAGs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$ such that $G_i \cap G_j \neq \emptyset$. The inference graph is composed of the two clusters corresponding to each Bayesian module and their separators. An important step of the assembly procedure is to find the cycles set Σ in the obtained BN and eliminate them in order to build an acyclic cluster graph that can support a global inference [11]:

- Elimination of separator: if a set of clusters create a cycle i.e, $\Sigma \neq \emptyset$ because of a redundant separator or variable, remove the smallest separator between clusters belonging to Σ knowing that this separator is contained in all the set of separators between clusters in cycle. The smallest separator is composed of one variable shared between clusters.
- Elimination of the variable: if the cycle is not created because of a redundant separator or variable, the cycle is eliminated by removing a variable from all the cycle separators and clusters.

After eliminating the cycles to obtain the adequate inference graph, the remaining shared nodes are called d-sepnode in the monolithic BN because all the parents of the shared node are defined in one BN module in the MBN structure [11]. Consequently, the conditional probability distributions defined in the monolithic BN and in the MBN are the same.

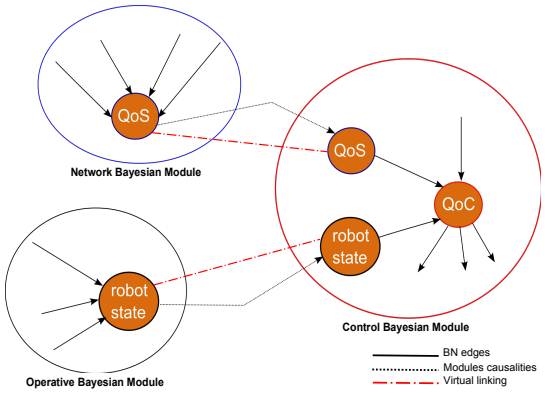

Fig. 2. Modular Bayesian Network for diagnosis of a networked mobile robot

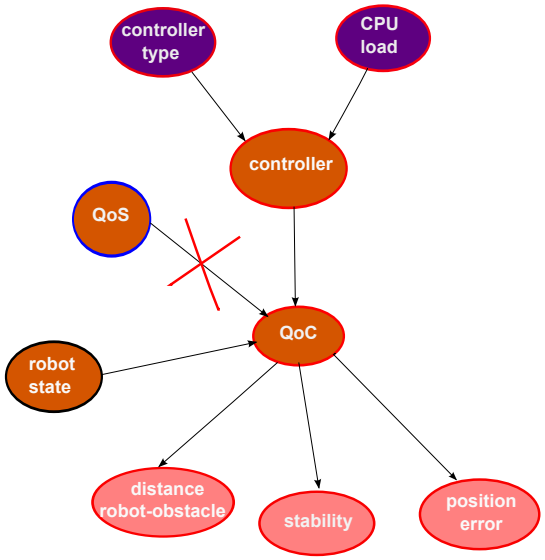

Fig. 3. Control Bayesian Module Update

IV. Diagnosis Architecture Update

Our previous work [14] proposed a diagnosis solution for a distributed system: a networked mobile robot controlled by a control station. A collaborative and distributed BN was developed to maintain the robot in an operational state and lead it to its target position. The MBN developed in [14] is composed of three Bayesian modules as shown in figure 2: the control module, the operative module and the network module. Once the QoC becomes bad because of a bad QoS (after inferring the diagnosis result), the station orders the robot to switch to embedded controller. The robot becomes autonomous and independent of the control station. The diagnosis architecture must thus be updated to be adequate to the new system structure and behavior. The network module does not take part of the diagnosis procedure anymore and the network state does not affect the system performance. The causal dependency between QoS and QoC is thus removed from the control module as shown in figure 3 which imply eliminating the QoS variable from the QoC parents set. A monolithic BN, implemented on-board, is then used combining

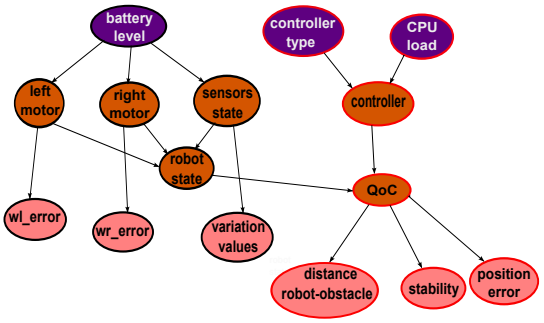

Fig. 4. Monolithic Bayesian Network

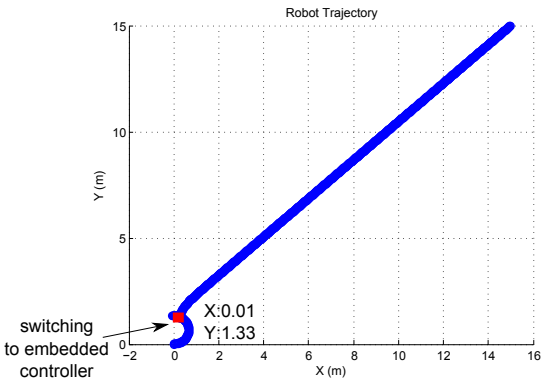

Fig. 5. Robot Trajectory

the control module and the operative module as shown in figure 4. The QoC conditional probability distribution must be reconfigured because the QoC state depends only on the robot and the controller states. The assembly procedure defined in the previous section is used to assembly the operative module and the control module such that the robot state node (the shared node) becomes the d-sepnode in the monolithic BN. The simulation scenario in figure 5 represents the robot mission dynamic. The robot mission is to achieve a target position $(15, 15)$ starting from the initial position $(0, 0)$. Truetime toolbox (2.0 version) [3] is used to simulate the networked system while BNT toolbox [9] is used for BN implementation. The robot receives control data from the first station whose position is (1,1). The network conditions in the network are bad because of a high packet loss rate (packet loss rate=90%). This degradation in the network conditions affects the system behavior as shown in figure 5 i.e, the robot does not navigate toward the target position $(15, 15)$ and it deviates from the goal. This deterioration in system behavior is observed in position $(X, Y) = (0.01, 1.33)$ (see figure 5). The control module in the MBN receives hard evidences on observation nodes such that $P(stability = no) = 1$ and $P(positionerror = nodecrease) = 1$. This degradation corresponds to posterior probabilities $P(QoC = bad) = 1$, $P(QoS = bad) = 0.87$ after setting hard evidences on the corresponding observations nodes. Using the MBN diagnosis output and the decision making algorithm defined in [14], the robot switches to embedded controller as shown in figure 5 because the system performance degradation is caused by a bad QoS. At this point, we have $[P(robot = good) = 0.74, P(robot = degraded) = 0.11, P(robot = bad) = 0.15$. Before assembling the control and the operative modules, the QoS node must be removed from the QoC conditional probabilities table. In order to determine the new QoC conditional probability distribution from the previous one, we use the following equation for each QoC state knowing the states of the robot and the controller:

$$P(QoC|R, C) = \sum_{QoS} P(QoC|R, C, QoS) \qquad (2)$$

where R and C refer to the robot state and the controller state respectively.

The parents of the shared node Robot state are all defined in one module corresponding to a cluster. The control module is the second cluster in the clusters graph and the robot state is the d-sepnode that guarantee a global inference in the monolithic BN. Assembling the two clusters corresponding to the control and the operative modules, we obtain two clusters $C_1 = \{$battery level, left motor, right motor, sensor states, wl_{error}, wr_{error}, variation values, robot state$\}$ and $C_2 = \{$controller type, CPU load, controller, robot state, QoC, distance robot-obstacle, stability, position error$\}$ with a separator $S = \{$robot state$\}$. No cycle appears in the cluster graph corresponding to the monolithic BN of figure 4 and there is not need to use separator or variable elimination.

V. Conclusion

A collaborative and distributed diagnosis is the good solution for distributed systems to optimize the computations and communications resources. Our system dynamic is susceptible to change while moving toward its goal. The communication between the robot and the control station could be interrupted and the robot switches to its embedded controller (the on-board controller). Switching from distributed to embedded controller implies that a diagnosis configuration must be considered in order to adapt the new system architecture. The proposed MBN (the distributed BN) is adapted to the new system architecture and so the diagnosis method is updated so that we have a monolithic BN. The obtained monolithic BN is the result of the assembly procedure of two bayesian modules: the control Bayesian module and the operative one. The assembly procedure consists in restoring the dependencies between the bayesian modules by verifying the cluster graph structure. We check the cluster graph structure to avoid having cycles in the inference graph of the monolithic BN and to ensure a global inference which is mandatory. This step is important to maintain the continuity of the system mission and so its monitoring. In addition, removing the QoS node from the QoC parents set is considered in order to have an adequate diagnosis tool which is conform to the system behavior and architecture. The switching from embedded to distributed BN has been studied in another work [15]. Besides, the distributed diagnosis method will be extended to the case of a system composed of multiple networked robots controlled by a control station.

REFERENCES

[1] Bottone, S., Lee, D., O'Sullivan, M., and Spivack, M. (2008). Failure prediction and diagnosis for satellite monitoring systems using Bayesian networks. In Military Communications Conference, 2008. MILCOM 2008. IEEE (pp. 1-7). IEEE.

[2] Cheng, Y., Xu, T., and Yang, L. (2013). Bayesian network based fault diagnosis and maintenance for high-speed train control systems. In Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2013 International Conference on (pp. 1753-1757). IEEE.

[3] Cervin, A., Henriksson, D., and Ohlin, M. (2009). TrueTime 2.0reference manual. Technical Report.

[4] Hwang, K. S., and Cho, S. B. (2009). Landmark detection from mobile life log using a modular Bayesian network model. Expert Systems with Applications, 36(10), 12065-12076.

[5] Lee, S. H., Yang, K. M., and Cho, S. B. (2015). Integrated modular Bayesian networks with selective inference for context-aware decision making. Neurocomputing, 163, 38-46.

[6] Li, X., Li, D., Wan, J., Vasilakos, A. V., Lai, C. F., and Wang, S. (2015). A review of industrial wireless networks in the context of industry 4.0. Wireless Networks, 1-19.

[7] Lim, S., Lee, S. H., and Cho, S. B. (2016). A modular approach to landmark detection based on a Bayesian network and categorized context logs. Information Sciences, 330, 145-156.

[8] Mechraoui, A. (2010). Co-conception d'un systme command en rseau sans fil l'aide de rseaux baysiens distribus (Doctoral dissertation, Institut National Polytechnique de Grenoble-INPG).

[9] Murphy, K. (2001). The bayes net toolbox for matlab. Computing science and statistics, 33(2), 1024-1034.

[10] Naim, P., Wuillemin, P. H., Leray, P., Pourret, O., and Becker, A. (2011). Réseaux bayésiens. Editions Eyrolles.

[11] Oude, P. (2010). Modular Bayesian networks: reasoning, verification and model inaccuracies.

[12] Pearl, J. (1988). Probabilistic reasoning in intelligent systems. Palo Alto. Morgan Kaufmann. PEAT, J., VAN DEN BERG, R., and GREEN, W.. Changing prevalence of asthma in australian children. British Medical Journal, 308, 1591-1596.

[13] Przytula, K. W., and Choi, A. (2007). Reasoning framework for diagnosis and prognosis. In Aerospace Conference, 2007 IEEE (pp. 1-10). IEEE.

[14] Sassi, I., Gouin, A., and Thiriet, J. M. (2016). A Bayesian network for diagnosis of networked mobile robots. Risk, Reliability and Safety: Innovating Theory and Practice: Proceedings of ESREL 2016 (Glasgow, Scotland, 25-29 September 2016), 340.

[15] Sassi, I., Gouin, A., and Thiriet, J. M. (2017). Diagnosis architecture reconfiguration for a networked mobile robot. Submitted to ESREL 2017 (Portoroz, Slovenia, 18-22 June 2017).

[16] Schumann, J., Mbaya, T., and Mengshoel, O. J. (2012). Software and system health management for autonomous robotics missions. Proc. of i-SAIRAS 2012.

[17] Simon, D., Song, Y. Q., and Aubrun, C. (2013). Co-design approaches to dependable networked control systems. John Wiley and Sons.

[18] Zhang, L., Gao, H., and Kaynak, O. (2013). Network-induced constraints in networked control systemsa survey. IEEE transactions on industrial informatics, 9(1), 403-416.

[19] Xiang, Y., and Lesser, V. (2000). Justifying multiply sectioned Bayesian networks. In MultiAgent Systems, 2000. Proceedings. Fourth International Conference on (pp. 349-356). IEEE.