Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-31 (2022) 409–414

Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations

Martin Skaldebø * **Ingrid Schjølberg** **

* *Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: martin.b.skaldebo@ntnu.no).*

** *Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: ingrid.schjolberg@ntnu.no)*

Abstract: This paper presents a novel framework for modelling dynamic Bayesian belief networks (BBNs) for online risk assessment in underwater operations. Existing frameworks spans from commercial software with restricted code access to non-profit open source frameworks. Frameworks with restricted code access often provides general user interfaces and visualization tools, while open source frameworks provides access to code for developers. The model presented in this paper pursues a best of both worlds scenario, where the model implementation should be uncomplicated while also providing visualization and verification to provide the user with a clear perception of the implemented model. The presented method is an expansion of the Bayesian model of the pomegranate python library, and simplifies the procedure of building, verifying and utilizing BBN models. The method is applied to a conceptual design of an underwater scenario case study with a model for an underwater vehicle manipulator system.

Keywords: Bayesian belief network, online risk assessment, dynamic risk models, underwater intervention

1. INTRODUCTION

Humans conduct risk assessment in their decision making process, both intentionally and unconsciously. In underwater operations as in other maritime, aerial and land-based operations, risk assessment is a vital part of the human operator's decision making process. If such systems should move towards more autonomy, it is imperative that the risk assessment follows. However, it has proven to be difficult to map the human capability for risk assessment as a computational ability. This is due to the challenges in identifying the relevant knowledge in an ocean of available information, and difficulties in computerizing the background knowledge of the process. "We are drowning in information and starving for knowledge" - *Rutherford D. Roger*. Increased autonomy in various types of operations is not a goal in itself, rather a results of reducing costs and risk and increasing safety. Technological advances in autonomous systems can arguably improve the safety, efficiency and performance by aiding in decision-making [1]. In this paper, an approach of using dynamic Bayesian belief networks (BBNs) is proposed, presenting *Another Bayesian Belief Analyser* (ABBA) which is a novel BBN framework designed for dynamic and online risk assessment purposes. A clear advantage of using BBNs when assessing risk is enabling a network to assess risk by analysing data while also exploiting the human knowledge of the process. Using historic data to train the models can allow for robust models specialized for a specific task or scenario. BBNs can also use expert judgement to decide causal dependencies and probabilities in the networks. Consulting experts when designing the internal behaviour of BBNs can however lead to flawed models and conclusions, due to limited and subjective knowledge. In both approaches, deciding arcs and nodes in the networks allows for human knowledge to be transferred to the risk assessment process. When doing so, identifying relevant nodes becomes imperative. If important causal dependencies or nodes are missing in the BBN model, it would significantly affect the results. ABBA attempts to circumvent the use of expert judgement by analysing historic data of the process, however leaving it to the user to decide the causal dependencies to exploit the users overall knowledge of the process.

The main contributions of this paper is summarized with:

- Discussing the importance of risk assessment in underwater operations and presenting how the increase in autonomy brings higher requirements for risk assessment.
- An overview of existing BBN frameworks will be provided along with discussion about the novelties of ABBA compared to existing solutions.
- The development of ABBA, a dynamic approach of designing BBNs with graphical visualization and adaptable source code for implementation with other systems.
- A conceptual design of a case study presenting how ABBA is utilized to reduce uncertainties in an underwater vehicle manipulator system (UVMS).

The paper is structured as follows. Section 2 presents information about risk assessment and the use of risk in

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2022.10.462

Table 1. Processes with dynamic and online risk assessment

	Process	Dynamic	Online
1	Risk assessment performed by the captain on a fishing vessel regarding the risk for going out at sea for a mission	YES	NO
2	Advanced Guidance and navigation control with risk assessment for collision for Autonomous Surface Vessels (ASV) and Autonomous underwater vehicles (AUV)	YES	YES
3	Updating evidences in a Bayesian belief network (BBN) to update risk values	NO	YES
4	Updating evidences in a BBN to update risk values while also using gathered information to update and retrain the model	YES	YES

underwater operations. Section 3 discuss Bayesian belief networks and existing solutions for developing BBNs. Section 4 presents the novel ABBA framework with information about configuration, data formats and verification method, and a model of ABBA for a UVMS is presented in Section 5. Section 6 provides results which are discussed in Section 7 and conclusions and suggestions for further work are provided in Section 8.

2. RISK ASSESSMENT

2.1 Dynamic and online risk

There are contradicting beliefs and opinions concerning some terminology in the risk community, especially dynamic risk and online risk are two terms that are interchangeably used. Time dependency, varying models, parameter updates and connectivity are aspects that are emphasized when attempting to define these terms. In this paper, the elemental definitions of the adjectives are used to formalize definitions of the risk terms. Thus, in this paper, dynamic risk is defined as a process exposed to constant change, activity or progress, while online risk is defined as a process that is controlled by, or connected to, a computer. The definitions are taken from the online dictionary `Lexico` provided by Dictionary.com and Oxford University Press [2]. Table 1 illustrates some examples of processes and how they would be characterized with the definitions used in this paper. Examples 3 and 4 from Table 1 are the most relevant for the work presented in this paper.

The reason example 3 is considered not to be dynamic while the other processes are considered dynamic is the fixed model used for the assessment. Advanced risk models applied in e.g. ASVs and AUVs (ex. 2) can include changing systems considering different variables for different situations, and more intelligent solutions may even include models able to evolve during operations with the use of internal learning. Example 3 represents a risk process that involves performing risk assessment with a fixed model. Moreover, the input to the model will be updated with sensor data and the calculated risk will be included in the decision-making process of the system, hence it is online, yet not dynamic. Example 4 however can be considered to be dynamic. Utilizing ABBA in such a process allows for the user to alternate the model how ever desired and train a new model that considers new sensors and new data. The framework is also capable of being incorporated as part of an intelligent system where it can decide to train a new model with new data if the system finds this beneficial.

2.2 Risk assessment in increasingly autonomous operations

Risk is often defined as the product of likelihood of an event and consequence if the event occurs,

$$\text{Risk} = \text{Likelihood} \cdot \text{Consequence}. \qquad (1)$$

Considering this definition and further analysing the risk often attempts to answer three main questions [3].

(1) What can go wrong?
(2) What is the likelihood of something going wrong?
(3) What are the consequences if something goes wrong?

These three question demonstrates why risk assessment are challenging for increasingly complex systems. Identifying everything that can go wrong have proven to be problematic, especially since a person's information and knowledge about a complex system can be inadequate. Lack of historic data for new systems can decrease the knowledge about the likelihood of something going wrong as well as the severity of the consequences. It is therefore essential that risk analysis becomes a vital part of the design, building and operation process, especially for increasingly autonomous and intelligent systems [1].

Introducing more autonomy in operations will increase the necessity of risk assessment performed by the system. Take the underwater segment, that is an area highly focused on increased autonomy. The underwater scene provides a harsh environment in constant change, which introduce the importance of dynamic and online solutions. With increased autonomy follows increased significance of decisions, which again increase the importance of making correct decisions. Autonomous underwater vehicles (AUVs) may possess collision avoidance systems, path planning systems and other systems analysing when the risk of continuing operation is too high and counteractions should be taken. In such operations the humans usually do not have the ability nor the accessibility to the system in order to oversee the system, and the system must perform the decision-making independently. Loss of AUV during operations is not an uncommon phenomena which proves that existing risk assessment solutions have room for improvements.

In the underwater scene, increasing autonomy in remotely operated vehicles (ROVs) is also a growing field of interest. Since humans are still an important participant in ROV operations, it may not require the system to act and decide independently in critical situations, but the systems should however possess strong risk awareness applications. In a human supervisory system, humans should not intervene unless necessary, thus the system should be able to perform decision-making independently up to a certain significance of potential consequences. Such systems should also be able to provide the human supervisor with a simple situational awareness. If the human supervisor should be able to assess the overall risk of operation, it could be problematic to inspect all available data and sensory feedback. The semi-autonomous systems should therefore be able to collect available data, analyse it and provide a simpler understanding of the potential risk of operation for the human supervisor to understand. Consequently, risk assessment is important in all levels of autonomy, whereas requirements for robustness and assurance of correct and sufficient decisions increases with the increased levels of autonomy.

3. BAYESIAN BELIEF NETWORK SOFTWARE

There are several existing software and frameworks for modeling BBNs, both licensed products and free to use open source codes. Table 2 lists the most used available software and frameworks. The graphical user interface (GUI) based software provide visualization tools that the command line based software does not. However, such frameworks often have free access to the source code, which makes it highly adaptable for different applications because the user can modify the source code to fit different purposes.

3.1 Commercial software

Some of the most known commercial software used when computing BBNs are GeNie, HUGIN and Netica. The software all have a graphical user interface (GUI) and restricted source code. GeNie is a GUI to the SMILE Engine which is a reasoning and causal discovery engine for graphical models i.e. BBNs and others. Smile and GeNie are provided by BayesFusion which specializes in using Bayesian networks in artificial intelligence modelling and machine learning software. SMILE is the software library which can be embedded into existing user software and GeNie is BayesFusion's GUI for the library where interactive modelling is supported. HUGIN EXPERT A/S provides decision support software and uses graphical models based on Bayesian network technology. Their software mainly provide analytic solutions for detecting fraud, credit card fraud and assisting corporations in analyzing their customers in suspicion of money laundering. Netica is a program for BBNs and influence diagrams developed by Norsys Software Corp., which is used for drawing networks and calculation of causal relationships either by individual probabilities from expert judgement or learned from data. BayesFusion provides free academic versions of their software GeNie, while HUGIN and Netica are only provided without cost through lite test-versions. [6, 7, 8].

3.2 Free to use software

Existing non-commercial software provides free use of software, but are often less developed than the commercial software. However, with access to source code, the software can be highly adaptable and easily incorporated with other systems and code. This often requires a good comprehension of the code and can provide problems for users not that familiar with either the programming language, architecture of the code, or the utilized models. Some available software, e.g. the pomegranate package, are powerful tools that provide a range of models and applications. For advanced users these tools are exceptional, however the large libraries can be hard to comprehend for less advanced users. See [9, 10, 11, 12, 13, 14, 15] for more information on non-commercial software listed in Table 2.

4. ANOTHER BAYESIAN BELIEF ANALYZER (ABBA)

ABBA is an open source python library based on the Bayesian module of the pomegranate library. ABBA extends pomegranate in two ways by (1) enabling easy calculations and implementations of conditional probability tables (CPTs), and (2) visualizing the Bayesian network similar to other graphical user interface (GUI) based software. Thereby incorporating the best of both worlds. The library takes a .txt file as input where all the relevant nodes are described. The .txt file has the format `<<name1;name2;scale;limits;parent nodes>>` where the attributes are described in further details in Table 3. Note that the input file differentiate between attributes with semicolon.

4.1 Historic Data Format

ABBA uses historic data to calculate the conditional probability tables (CPTs). This circumvents the need for expert judgements. This data should be in a specific format for the library to be able to correctly read it. The data should be collected in a .csv file, where each node has its own column of historic data and where each row represents a measurement. The name of the node in the .csv file should match the attribute `name2` from the input .txt file. If the historic data is collected from a larger database it can contain insignificant or irrelevant datapoints. This is of no concern because the relevant datapoints that should be included in the BBN are specified in the input .txt file. Thereby, extracting only information about the relevant nodes from the .csv file.

4.2 Verification

In order to use the model for predicting new scenarios, we have to be confident in the model's legitimacy. In order to determine if the model is able to predict unseen scenarios it should be properly verified. The verification of the model will provide insight in how well it predicts unseen situations and a good verification contribute with a sense of belief in the model. The verification is used in order to validate if the software performs as intended.

ABBA uses root mean square error (RMSE) with K-fold cross validation to validate the model. The training data is divided into K partitions and the training is performed K times. For each iteration the model trains on $K-1$ partitions and is validated on the last partition, which is repeated until all partitions have been used for validation. The RMSE represents the difference in the observed or true values and the estimated/predicted values. Considering all K samples the overall RMSE used for verification in ABBA is computed with

$$RMSE = \sqrt{\frac{\sum_{i=1}^{K} \frac{\sum_{j=1}^{n_i}(y_i-\hat{y}_i)^2}{n_i}}{K}}, \tag{2}$$

Table 2. Review of Bayesian network modeling software. Updated and extended from an earlier review from [4] and [5]

Name	Programming language	Type of interface	Licensing type	Access to source code
GeNie	C++, Java, R, Python, .NET	GUI	Commercial	Restricted
HUGIN	Undisclosed	GUI	Commercial	Restricted
Netica	Undisclosed	GUI	Commercial	Restricted
VIBES	Java	GUI	Free	Restricted
SamIam	Java	GUI	Free	Restricted
UnBBayes	Java	GUI	Free	Restricted
BNT	Matlab, C	GUI, Command line	Free	Open
Pomegranate	Python	Command line	Free	Open
bnlearn	R	Command line	Free	Open
gRain	R	Command line	Free	Open

Table 3. Description of the attributes in the input file to ABBA

Attribute	Description
name1	Name of the node (should not contain spaces)
name2	Name of the attribute in the .cvs file
scale	Scale defining different states in the CPTs
limits	Limits for classification within the relevant scales
parent nodes	name1 of all parent nodes. If none, leave blank

where y_i and $\hat{y}_i$ are model variables and predicted values respectively, n_i is the number of samples in iteration i, and K is the number of partitions in the cross validation.

5. CASE STUDY

The case study is an attempt to further improve the methods and results presented in [16] and [17]. [16] presented a relative dynamic positioning system of a low cost underwater vehicle relative to a object of interest using monocular camera for detection, and [17] continued this work by introducing a manipulator to simultaneously perform intervention of the relevant object.

In the previous work, the solution encounter problems when the manipulator occludes the object of interest. The dynamic positioning system is dependent on the distance calculated by a scaling function, and occlusion of the object results in a faulty detection and estimated distance. The goal of this case study is thus to investigate if ABBA can be used to formulate a value of belief in the estimated distance. In this way, a low belief in the estimated distance should result in the system not trusting this estimation and hence find other measurements to use for the estimation, and a high belief should results in the system being confident in the estimation.

5.1 BBN modeling

The methodology used for modeling the architecture of the BBN in this case study is adopted from [18]. The authors of the paper states three requirements for the nodes that have to be met in order to develop a BBN.

(1) The nodes can be defined
(2) The state of the nodes can be represented by measurable variables
(3) The target node and any other node in the network have known traceable direct/indirect relationships

The final structure of the BBN is depicted in Fig. 1.

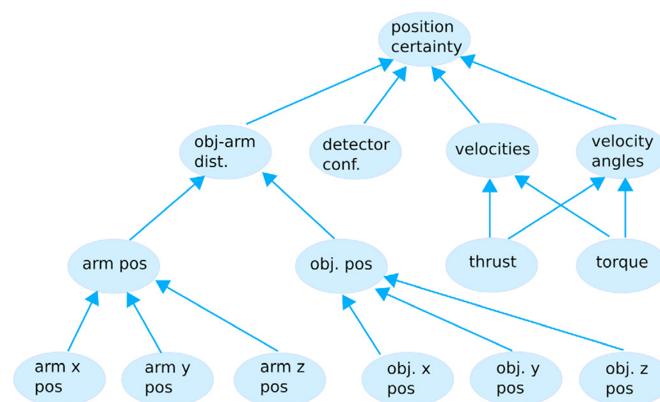

Fig. 1. BBN structure with causal dependencies between nodes.

5.2 ABBA in the control loop

ABBA should be incorporated in the control loop in order to influence the estimation from the detector. Fig. 2 presents how ABBA influences the motion control system (MCS), where the Position estimator block and ABBA block is new relative to the control loop from [17] and the dotted lines represents dependencies introduced in the new control loop. In this model, ABBA represents a trained BBN with fixed structure as presented in Fig. 1. The nodes *arm x pos*, *arm y pos*, *arm z pos*, *obj. x pos*, *obj. y pos*, *obj. z pos*, *thrust* and *torque* are updated online in the control loop and influence the other nodes as well as the target node, *position certainty*. In order to preserve as much of the original MCS as possible, the guidance system is kept original. The relative distance fed to the MCS however is influenced by the indication from ABBA. Previously the MCS received relative distances from the detector which was used for calculating reference velocities in the guidance system. In the new control loop the MCS receives relative distances from the position estimator, which are already adjusted by the influence of the BBN. The position estimator will use the position certainty from ABBA to adjust the distances from the detector. If the system no longer trust the measures from the distance estimator, it will put higher value to other sensor measures in the system and less on the object detection.

5.3 Collection and augmentation of data

Sufficient amount of data have to be collected in order to train the BBN. The data consist of measurements with a constant time step. A satisfactory dataset should include

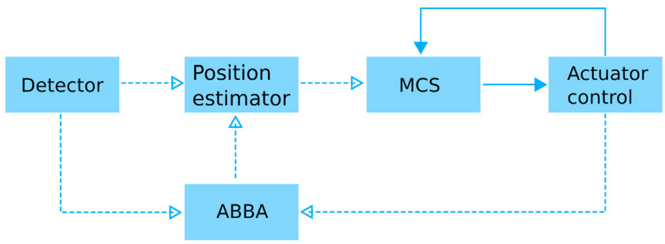

Fig. 2. Control loop.

Table 4. Data collection for each node in the BBN.

Node	Measure procedure
Arm x position	Through forward kinematics in kinematic control
Arm y position	Through forward kinematics in kinematic control
Arm z position	Through forward kinematics in kinematic control
Object x position	Extracted from the detector
Object y position	Extracted from the detector
Object z position	Extracted from the detector
Arm position	Calculated based on parent nodes
Object position	Calculated based on parent nodes
Thrust	Equal the thrust given to the propellers
Torque	Equal to the torque given to the propellers
Distance between object and arm	Calculated based on parent nodes
Detector confidence	Extracted from the detector. The confidence is a measure of how confident the detector is in the detection of the object and is a result of the training of the detector.
Velocities	Estimated with a Kalman filter based on position data
Velocity angles	Estimated with a Kalman filter based on position data
Position certainty	Calculated by considering difference in believed position and true position from Qualisys

values for every node involved in the BBN. Some of the nodes in the BBN architecture were included after performing parent divorcing and does not represent a direct measurement or a direct causal relationship with the parent nodes. Take the node *arm pos*, which has definitional relationships with the parent nodes, where the position can be defined from the nodes *arm x pos*, *arm y pos* and *arm z pos*. To ensure a sufficient dataset, the calculated values will be evaluated equivalent to direct measurements and recorded equally in a .csv file. The position certainty as the target node also needs to be measured for collected data. If this value could be measured at all times, the need for the BBN to estimate it would be unnecessary. Moreover, for the collection of the dataset, this value will be measured using the motion capture system Qualisys. Qualisys is installed and calibrated in the marine cybernetics laboratory (MC-Lab) at NTNU and uses optical tracking technology to measure position of objects using cameras and known markers attached to the objects. An overview of how the specific nodes in the BBN are measured is listed in Table 4.

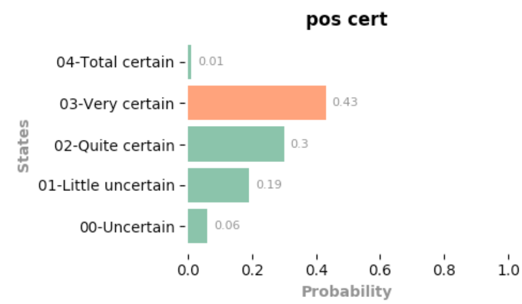

Fig. 3. CPT for the target node.

6. RESULTS

ABBA has been utilized to build a BBN model with architecture and nodes as presented in Fig. 1. Each node in the BBN includes CPTs as presented in Fig. 3 for the target node. Note that no experiments have yet been conducted and thus the data presented in the figure are only from a conceptual design.

ABBA is able to build the BBN with corresponding CPTs from the input .csv file and the descriptive .txt file. A built in generic classifier categorize measurements from the .csv file based on the limits and scales described in the .txt file and builds causal dependencies as described in the same file.

A conceptual design of the BBN in the control loop is presented, where the BBN will affect the MCS by invoking evidence on child nodes based on real time sensor measurements. As explained in Section 5.2, the nodes *arm x pos*, *arm y pos*, *arm z pos*, *obj. x pos*, *obj. y pos*, *obj. z pos*, *thrust* and *torque* will be updated real time. These measurements can be extracted at real time with the same procedure used when logging the data. These procedures are discussed in Table 4.

7. DISCUSSION

A BBN for the presented case was built using ABBA. Once the data is collected in a .cpt file and the parameters are correctly described in a .txt file ABBA does the rest. The tool is simple to use and provides good visualisation of the network. A built-in generic classifier enables the tool to be directly incorporated in dynamic systems where evidence derives from measurements. The classifier characterizes measured values within the scales and limits as described in the input .txt file and updates the network based on these measurements. The BBN model is also capable of being updated real time if measurements, time dependencies or other factors in the overall system requires it. Such scenarios could appear if the system operates in a dynamic environment where updates of the models are required in order to detain a compatible system.

As discussed in Section 1, an advantage in the BBN approach is exploiting human knowledge of a process while still using historic data to train a model. The human knowledge is exploited in the modelling of the network. When modelling the BBN presented in the conceptual design of the case study, a generic 8-step approach was used. Challenges that emerged when modeling the BBN are partially discussed in Section 5.1, however other challenges occurred as well. Parent divorcing and introduction of intermediary nodes solved challenges related to complex CPT calculations and computational load. Another investigated solution here was the altercation of scales and limits corresponding to the relevant nodes. The number of feasible states for a node determines the accuracy of the measurements and a should not be too low nor too high. The scales and limits determines in which state the measurements are categorized and should be carefully considered. An analyse of the available data was also performed here, although not explicitly explained in the modeling description. The available data should resemble a normal distribution which can be arranged by alternating the scales and limits to fit the data. If the scales for a node categorize all available data to one single state, there is no existing information for the other nodes. Information and knowledge about what happens if another state is achieved are then missing from the model. During the modeling of the BBN, the data should therefore be thoroughly analysed.

An analyse tool for analysing the data before training the CPTs could easily be implemented in the ABBA tool. Such a tool could analyse the available data before determine, or at least suggest, scales and limits for nodes in the model. The analyse tool could then provide scales and limits that would ensure normal distributed distributions. A limitation with such a tool is the decrease of human knowledge in the process. Automatically chosen scales and limits could also provide worse overall achievement of the model if the data is not good. If the original data is skewed or flawed in any other way, the analyse tool would adjust the scales and limits according to bad information. However, it could still be beneficial to create this analysing tool to at least suggest scales and limits. Manually analysing the data and numerous iterations in the modeling steps for determining the scales and limits were time consuming, and a suggestive tool could help streamline this process.

A conceptual design of the model incorporated in the control loop have been presented. Furthermore, this will provide more robust estimates of the relative position of the object. However, so far this is only a conceptual design and no experiments have been conducted to verify the suggested solution. Experiments of the preliminary system without the use of ABBA was conducted in MC-Lab at the department of marine technology at NTNU. Preferably, new experiments should be conducted at the same facility using the same equipment. In this way the experiments are comparable with previous experiments. This is essential in order to draw any conclusions regarding the improvements of the system. A proper verification of the model has nor been conducted in the lack of available data. The quality of the model depends both on the quality of the data as well as the architecture of the BBN model. In order to ensure that the presented model is valid it should be verified using the presented verification method with collected data from MC-Lab.

8. CONCLUSIONS

A framework for modeling dynamic BBNs have been presented in this paper. The presented framework, ABBA, enables simple calculations of CPTs and provides strong verification and visualization tools. A conceptual design is presented, where ABBA should help increase estimations in a UVMS performing DP and autonomous intervention. In this case, a static BBN model is built using ABBA where relevant nodes in the network can be updated in real time in order to perform online real time risk assessment. The natural next steps are to expand the presented work with collection of historic data and experimental testing of the system.

REFERENCES

[1] *Risk Management of Autonomous Marine Systems and Operations*, volume Volume 3B: Structures, Safety and Reliability of *International Conference on Offshore Mechanics and Arctic Engineering*, 06 2017. V03BT02A020.

[2] *Lexico, Powered by Oxford.* Provided by Dictionary.com and Oxford University Press (OUP), November 2019.

[3] Stanley Kaplan and B. John Garrick. On the quantitative definition of risk. *Risk Analysis*, 1(1):11–27, 1981.

[4] Mohamed Ali Mahjoub and Karim Kalti. Software comparison dealing with bayesian networks. In Derong Liu, Huaguang Zhang, Marios Polycarpou, Cesare Alippi, and Haibo He, editors, *Advances in Neural Networks – ISNN 2011*, pages 168–177, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[5] Kevin Murphy. Software for graphical models: A review. *Int Soc Bayesian Anal Bull*, 14, 01 2007.

[6] *GeNie Modeler: Complete Modeling Freedom.* BayesFusion LCC, 2019.

[7] *Explore Bayesian Network Technology.* Hugin Expert A/S, 2019.

[8] *Netica Application.* Norsys Software Corp., 2019.

[9] *Variational Inference for Bayesian Networks.* Sourceforge, 2019.

[10] *Sensitivity Analysis, Modeling, Interfercens And More.* Automated Reasoning Group, UCLA, 2019.

[11] *UnBBayes Overview.* University of Brasilia, 2019.

[12] *Bayes Net Toolbox for Matlab.* Kevin Murphy - Github, 2019.

[13] Jacob Schreiber. Pomegranate: fast and flexible probabilistic modeling in python. *CoRR*, abs/1711.00137, 2017.

[14] *bnlearn - an R package for Bayesian network learning and inference.* Marco Scutari, 2019.

[15] *gRain: Graphical Independence Networks.* Søren Højsgaard, 2019.

[16] M. Skaldebø, B. O. Arnesen, and I. Schjølberg. Dynamic positioning of an underwater vehicle using monocular vision-based object detection with machine learning. *OCEANS 2019 Seattle Conference Exposition. Seattle, USA*, 2019.

[17] B. O. Arnesen, M. Skaldebø, and I. Schjølberg. Monocular vision-based gripping of objects. *Robotics and Autonomous Systems*, 2019.

[18] Jeevith Hegde, Ingrid Bouwer Utne, Ingrid Schjølberg, and Brede Thorkildsen. A bayesian approach to risk modeling of autonomous subsea intervention operations. *Reliability Engineering System Safety*, 175:142 – 159, 2018.