

# Embedded Mission Decision-Making based on Dynamic Decision Networks in SoPC Platform

Hanan Chenini

Microwave Electronics Research Laboratory  
Faculty of Sciences of Tunis, University of Tunis El-Manar  
Tunis, Tunisia

**Abstract**—This paper tackles a Bayesian Decision Making approach for unmanned aerial vehicle (UAV) mission that allows UAV to quickly react to unexpected events under dynamic environments. From online observations and the mission statement, the proposed approach is designed by means of Dynamic Bayesian Networks (DBN) arising from the safety or performance failures analysis. After proposing a DBN model, a probabilistic approach based on Multiple-Criteria Decision-Making (MCDM) is then applied to find the best configuration reaching a balance between performance and energy consumption, thus decide which tasks will be implemented as SW and which as HW execution units, regarding the mission requirement. The proposal UAV mission decision-making is three-pronged, providing: (1) real time image pre-processing of sensor observations; (2) temporal and probabilistic approach based on Bayesian Networks to continuously update the mission plan during the flight; and (3) low-power hardware and software implementations for online and real time embedded Decision Making using Xilinx System on Programmable Chip (SoPC) platform. The proposed approach is then validated with a practical case UAV mission planning using the proposed dynamic decision-maker implemented on embedded system based on a hybrid device.

**Keywords**—*Bayesian Decision Making; Dynamic Bayesian Networks (DBN); Multiple-Criteria Decision-Making (MCDM); SoPC; practical case*

## I. INTRODUCTION

Autonomous system, such as drone known as UAV, is a robot system composed of several components (of hardware or software), fly autonomously according to a pre-programmed mission statement in unknown environment. Now-a-days, UAVs are being more widely used in different applications (military, aerospace reconnaissance, environmental and meteorological monitoring, aerial photography, and geophysical survey, etc.) achieving high performance and reliability combined with reduced size, weight, power consumption and cost. To execute successfully such missions, it was essential that the vehicle able to select appropriate scenario planning under consideration of the current state of the mission and uncertain environmental conditions. In each mission, the UAVs are generally set to perform a particular mission under several requirements and environmental conditions (unexpected obstacles, weather changes and sensor or hardware/software component failures, etc.). For UAVs, sensors are generally embedded on board, and consequently, the embedded systems are constrained to face the hard real-time constraints imposed by moving vehicle applications. In our work, we have focused on combining sensors and algorithms to understand the vehicle environment and to provide some autonomous processing

directly into the UAV and achieve the necessary processing power to run the algorithms near the sensors.

These issues increase the demand for providing an online, on-board scenario management system for UAV to produce a response to uncertain environmental events and emergent failures of sensors or software/ hardware components within a specified time. Indeed, it has to provide the system the capability to treat the real time data of uncertain or dynamic elements which might represent a threat for the UAV's mission execution.

Aiming at solving the above problem, the focus of our reasoning is to develop systems that behave differently in different contexts which means they continue to change over time, thus we need an approach capable to model such dynamic systems. For this reason, we are interested in how these systems that evolve over time using Bayesian Network (BN) and particularly its extension DBN [5], [6] to model the temporal evolution/influences of the variables of interest between multiple time series in the same model according to the mission statement. In fact, DBN used especially to relate variables to each other over adjacent time steps in the presence of uncertainty and anomalous observations.

Thanks to the proposed dynamic decision-making approach as depicted in Fig. 1, we may identify successfully the alternative decision that maximize the chance of achieving the mission goal at time  $t+1$  using several tasks as well as internal and external constraints of UAV that can affect the mission plan. During the flight, the UAV rely on several applications required to meet mission constraints (path planning, motion detection of aerial or ground targets, tracking, pose estimation to follow a precomputed path, obstacle avoidance, etc.), with computation demands that can vary during a mission. However, the UAV always require accurate measurements and information of the surrounding environment with respect to the current scene. Translating such complex applications for real-time implementation requires making specific choices so that the design meets the constraints. Some of the main processing requirements are speed of processing, accuracy of the results, cost and time involved in the implementation. To tackle with this problem, we are motivated to propose hardware reconfigurable architectures for implementing real time processing applications with stringent resource-consumption and runtime constraints. In this case, it can be beneficial to take advantage of the parallelism, low cost, and low power consumption offered by digital integrated circuits Field-programmable gate arrays (FPGAs) for implementing such reconfigurable systems dedicated to complex image processing algorithms.

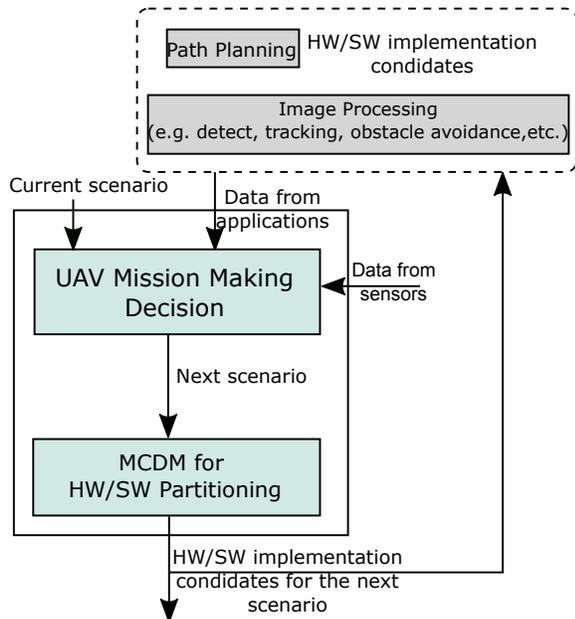


Fig. 1. UAV Mission decision making approach.

The design and implementation of the proposed Mission Decision Making approach can be divided into two main parts software and hardware within a single FPGA device. The software part is implemented on the dual-core ARM Cortex-A9 based Processing System (PS). The hardware partition is called the Programmable Logic (PL), which consists the FPGA programmable logic blocks. With regard to the latter, the different image processing applications is fully performed in FPGA fabric, whereas the control and the reconfiguration of the different functional blocks is managed by processes running on the ARM processor.

In this paper, we specifically propose an embedded Mission Decision Making by means of two contributions as illustrated in Fig. 1. The first one is the introduction based new DBN model is applied to schedule the information and to make on-line decision and provide the new scenario plan in constrained time when immediately required during the flight. Once the mission plan is updated, we then propose a new MCDM model to generate a suitable HW/SW configuration for the mission tasks regarding to performance constraints and resource availability that will allow the development of real time mission decision making module, on System-on-Chip (SoC) using FPGA for deployment on a UAV under extreme and uncertain environmental conditions.

The paper is organized as follows. Section 2 describes the proposed UAV mission decision making approach based on DBN. Section 3 explains how the MCDM deals with the problem of co-design to decide on the most appropriate HW/SW scheduling method and thus keep the embedded system's performance. Section 4 provides the hardware/software (HW/SW) implementations and few performance results concerning FPGA resource costs and execution times of the proposed Dynamic Decision Making module in real case study. Section 5 concludes the paper.

## II. PREVIOUS RELATED WORK

We confine the related work to Decision-Making approaches based artificial intelligence algorithms in embedded hardware and real-time architecture. Recently, many approaches of embedded Decision-Making are proposed in the literature [1] [2]. We can broadly classify these techniques into three categories: Multicriteria Decision Making [3], Artificial Intelligence and Mathematical Programming techniques [4]. In addition, current research tends to manage the uncertainty of the surrounding environment and the system such as Bayesian Networks introduced as stochastic/probabilistic models [7].

Thus, the development of embedded Decision Making system aims to continuously manage the UAV mission and propose the appropriate recovery action in the case of failure scenarios. This system is able to adapt to the variations in computation requirements during the UAV mission. Digital signal processors (DSP) or FPGAs are used for hardware implementation of such real-time systems. For this reason, a lot of researchers are interested in optimising hardware implementation for Decision-Making mechanism under the constraints of memory or computation.

To implement Decision Making based probabilistic (Bayesian) network onboard, some research has shown encouraging results with FPGA based reconfigurable hardware [19] [18] [15] [16]. In [9], the authors proposed an FPGA implementation based on a BN representation, that allows to continuously monitor the embedded system under time and resource constraints. For this purpose, they proposed off-line framework integrating a high level synthesis tool to generate the hardware version. In [15], the authors have suggested an efficient FPGA hardware design of a BN block written in Hardware Description Languages VHDL and used the development software ALTERA QUARTUS II12 to synthesize the design onto an Altera Cyclone IV EP4CE115 FPGA. Whereas, an embedded processor performed the monitoring of temporal sensor data using Linear Temporal Logic (LTL). In [17], the authors have developed a purely software implementation of their proposed DBN approach through the use of on-board software architecture called Anomaly resolution and prognostic health management for autonomy (ARPHA). This framework is developed to design and implement a specific failure scenario from a set of specifications (diagnosis, prognosis, and recovery) and provide as result an embedded software implementation of a DBN without any hardware alternative.

However, to the best of our knowledge, there is no prior work on hardware implementation of Decision making based on both DBN and MCDM with on-board vision processing on reconfigurable platforms. Our work provides a novel approach of combining DBN with utilizing MCDM techniques to choose between an HW (FPGA) or SW implementation (embedded processor) of the mission tasks.

## III. DYNAMIC BAYESIAN NETWORKS FOR UAV MISSION DECISION MAKING

### A. Dynamic Bayesian Network Model

The goal of this work consists of developing an autonomous on-board management system based on a set of

algorithms that will be embedded on-board. In our approach, we started by the model “glues together” proposed by [8] to control the behavior of different hardware/software components for a given task under uncertainty. By adding temporal specification in this model, the proposed mission decision making is a discrete time model of estimating the recovery tasks at the time  $t + 1$  based on the evidence obtained at the time  $t$ . To avoid critical situations while maintaining safety requirements, the proposed DBN is composed of two types of nodes (**Decision nodes**, **Sensor nodes**, **Data sources**) as depicted in the DBN hierarchy in Fig. 2 over two consecutive time slices.

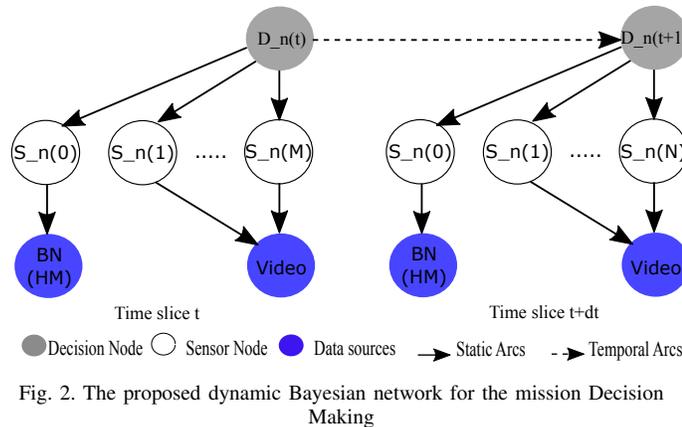


Fig. 2. The proposed dynamic Bayesian network for the mission Decision Making

- 1) **Decision nodes** ( $D_n$ ) includes the list of the available decision alternatives, in our case the list of possible scenarios to update the mission plan, modeled explicitly as possible states of the decision node.
- 2) **Sensor nodes** ( $S_n$ ) representing the data retrieved from the system environment by means of software applications. In practice, the outcome of each sensor node (Observed node) is known certainty with two possible states.
- 3) **Data sources** ( $D_s$ ) indicating process measurements by means of software or hardware sensors (in practice, we used imaging sensors and Health Management *HM* of the system providing information about the current state of the system environment).

Since the influence of decision nodes with sensor nodes is instantaneous, a regular Bayesian network arc is used as shown in Fig. 2. Considering two successive time slices in the modeling, each node at  $i^{th}$  time slice depend not only on its parents node at the same time slice but also on its parents and itself at previous time slices.

### B. The Proposed UAV Mission

As previously mentioned, UAVs are considered as critical systems operate in uncertain environments and have to face unexpected obstacles, weather changes and sensor, hardware or software failures. These constraints must be satisfied at all instants during the flight in interaction with the proposed decision maker, to impose the UAV to get to its final destination. At low altitude, we assume that the UAV always keeps the same height level from initial location to the destination location. The decision of re-planning a mission is taken by

evaluating the observations to obtain the state of the UAV on every time  $t + 1$ , from its previous state at time  $t$ . To move from a given initial position to a destination position, we attempted to introduce independent solutions for obstacle detection/avoidance, recognition and tracking of the desired object as well as the pose estimation in order to guarantee the accomplishment of detection and tracking mission goals during the flight.

Before we begin our application, the sensor data are aggregated from two cameras mounted in the front of the UAV for simultaneous capturing images in different angles. At any time, the camera can change its field of view  $\theta$  to detect the presence of ground and/or areal objects. Indeed, the angle  $\theta$ , which indicates the view angle between the UAV and the observed object, can take two possible values  $\theta = \{front, Bottom\}$ . Based on the output of these solutions, multiple decision alternatives ( $S_0, S_1, S_2, S_3, S_4, S_5, S_6$ ) are proposed to achieve new functionality/applications and to update the mission plan. For simplicity reasons, we suppose that we can receive information from only two processing tasks in each time  $t$  in addition to the HM data. After each time slice, we apply the decision maker and the alternatives decisions are sorted according to their priority (starting with urgent decision  $P_1$  and so on).

1) **Scenario S0:** As shown in Fig. 3, the initial position of the UAV is  $(x_0; y_0; z_0)$  and the target position is  $(x_d; y_d; z_d)$ . To move an UAV from the given initial position to the destination position, two independent image processing algorithms are performed simultaneously: (1) motion detection algorithm is applied in aerial images (captured by the front camera) to detect aerial targets (AT) and (2) pose estimation algorithm is applied in ground images (captured by the bottom camera) to find coordinates of the UAV’s location and then verify that it follows a precomputed path.

2) **Scenario S1:** For the second scenario  $S_1$ , if there is any obstacle detected in aerial image  $i$  before reaching its final destination, we aim to detect obstacles that are dangerous in the next aerial image  $i + 1$  where none, one or more aerial targets can be present. For this purpose, two independent image processing algorithms are performed simultaneously as shown in Fig. 4: (1) Region Labeling algorithm is applied in aerial images to compute region sizes of the detected aerial targets (AT) and then keep only the region with highest size (in our case, it represents obstacles) and (2) motion detection algorithm is applied to detect ground targets (GT).

3) **Scenario S2:** For the next scenario  $S_2$ , whenever there is any obstacle detected between images  $i$  and  $i + 1$ , it is highly required to know how far away the UAV is from the largest object observed (i.e. the object of the biggest label). For this purpose, distance measurement algorithm is applied using the two acquired images as shown in Fig. 5.

4) **Scenario S3:** For the scenario  $S_3$ , having found moving objects in ground image  $i$ , a method for recognizing a target object in image  $i + 1$  is proposed. For this purpose, distance measurement algorithm is applied using the two acquired images as shown in Fig. 6: (1) object recognition algorithm is applied to identify the nature of the detected object (GT) and then check if it can be considered the best match of the target object or not and (2) pose estimation algorithm is applied in

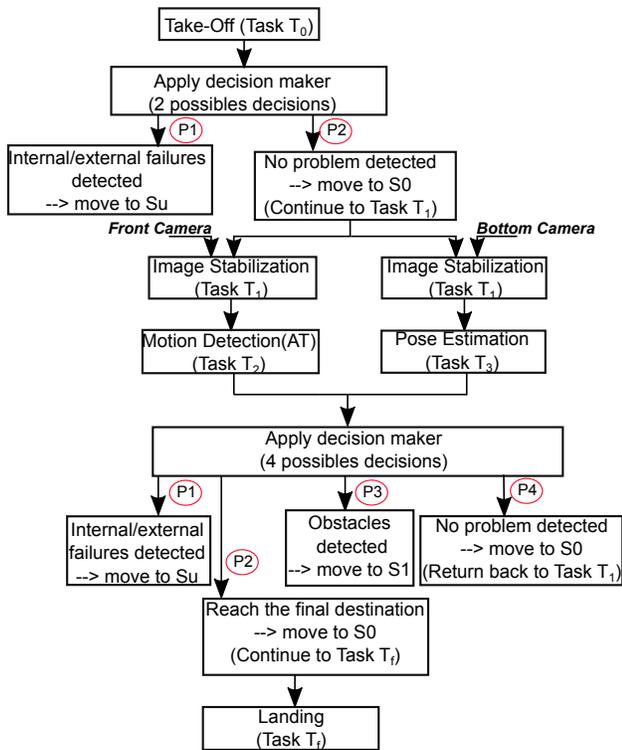


Fig. 3. The different steps of the scenario S0

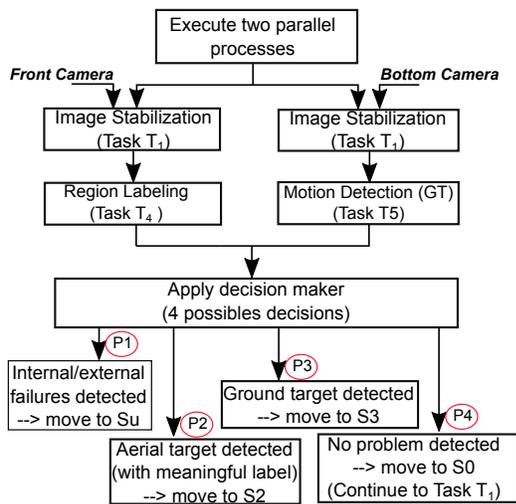


Fig. 4. The different steps of the scenario S1

ground images to find coordinates of the UAV's location and then verify that it follow a precomputed path.

5) Scenario S4: For the next scenario S4, if the distance between the UAV and the observed object (output of distance measurement application) is greater than a threshold distance before reaching its final destination, path planning algorithm is applied in ground image to compute another shortest path to the final destination as shown in Fig. 7

6) Scenario S5: For the next scenario S5, if the UAV has found the target object in image  $i$  before reaching its

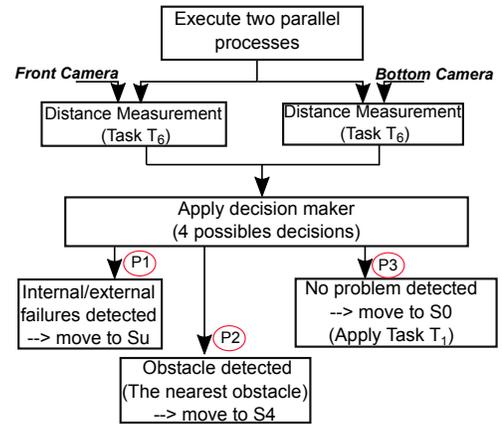


Fig. 5. The different steps of the scenario S2

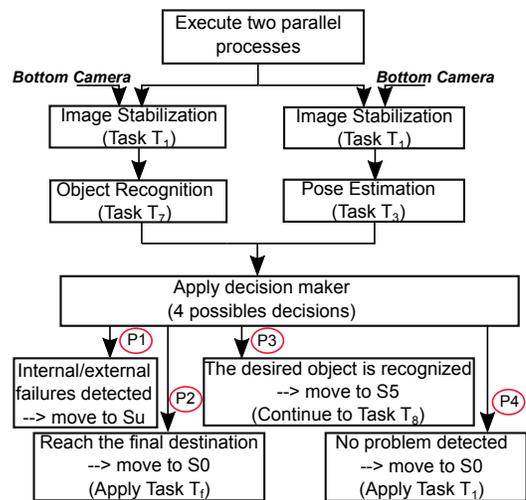


Fig. 6. The different steps of the scenario S3

final destination, two independent image processing algorithms are performed simultaneously as shown in Fig. 8: (1) motion detection algorithm is applied in aerial images to detect aerial targets (AT) and (2) video tracking application is then applied for locating the recognized object over time in ground image  $i + 1$  to ensure that the target object is not lost from the UAV's view.

7) Scenario S6: For the next scenario S6, if the detected object is tracked successfully in moving background before reaching its final destination, the UAV keep track of the target object. For this purpose, two independent image processing algorithms are performed simultaneously as shown in Fig. 9:(1) motion detection algorithm is applied in aerial images to detect aerial targets (AT) and (2) video tracking application is applied for locating the recognized object over time in ground image  $i + 1$ .

8) Scenario Su: For the scenario Su, if any anomalies or failures state are detected during the mission execution, the autonomous UAV need to perform the landing emergency task in the closest safe area.

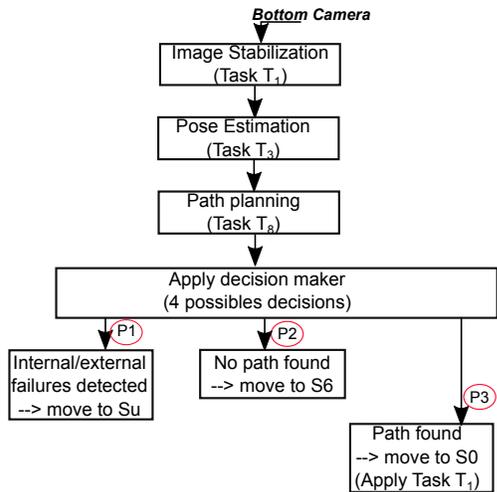


Fig. 7. The different steps of the scenario S4

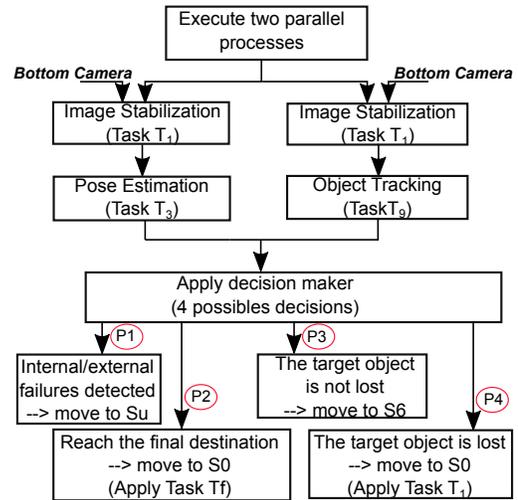


Fig. 9. The different steps of the scenario S6

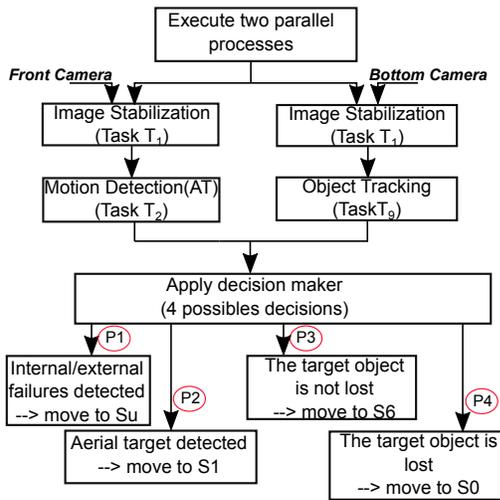


Fig. 8. The different steps of the scenario S5

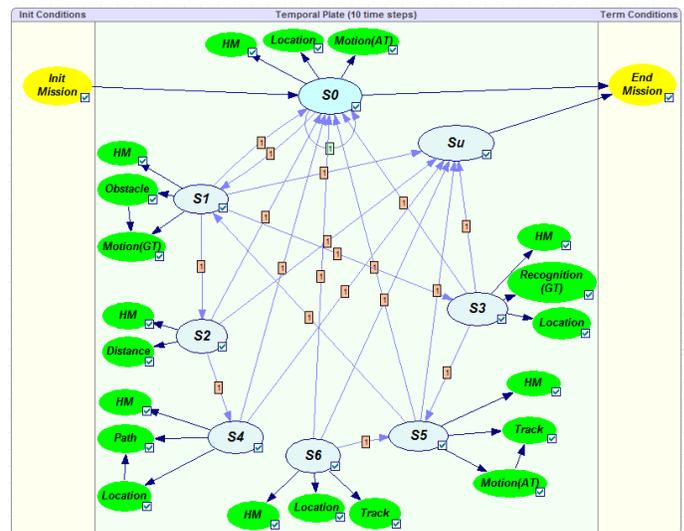


Fig. 10. The implementation of an example of the DBN model in software GeNie.

C. DBN Model of the Case Study

For the case study, the proposed dynamic mission decision maker based on DBN, by analyzing influence relations between nodes is illustrated in Fig. 10. This model aims to describe the proposal decision maker that is dynamically evolving over time to update the mission planning and predict what should be the next activity of the UAV following unforeseen circumstances as time proceeds. According to the mission execution, based on online and real time readings given by sensors, the decision is taken whether or not to request intervention in case of anomaly alerts such as internal software and/or hardware failure, changes in the environment (weather, obstacles), or communication problems.

For simplicity reasons, in the example presented, only two sensors nodes can have influence on the same variable or on another variable in the same time slice  $t$  with the aim of detecting the anomalous or failed state of the mission plan. In fact, each node in the DBN model can have a different number of states.

Since the state of sensor nodes are observable (evidence indicators), this simplify the construction of the DBN model needed to implement the autonomous updating of the mission tasks during the fight. Similarly to inference in a regular Bayesian Network, the inference in a DBN starts by given the evidence indicators for the current state and then calculates the impact of observation of some of its nodes on the posterior probability distribution (pdf) over other nodes as time proceeds.

IV. MULTI-CRITERIA DECISION MAKING (MCDM) FOR HW/SW PARTITIONING

To make on-line decision, the autonomous embedded drones, in our case study, have on board management system to allow execution of complex tasks (motion detection, tracking, obstacle avoidance, pose estimation and so on), with computation demands that can vary during a mission. According

to some failure scenarios that may arise during the flight, as seen previously, the developed approach includes on-board decision making module to select the future mission tasks. The main objective of this section is to design a partitioning decision module that considers different types requirements to determine which task can be implemented either as a SW or HW components. For this purpose, a comparison between the hardware and software versions is established in the presence of multiple criteria. Indeed, hardware or software task allocation depends essentially on how quickly and how long tasks must be done.

In [10], an extended BN model with influence diagrams is proposed for optimal decision making depending in multiple criteria within the MCDM framework, including decisions alternatives. As illustrated in the BN hierarchy in Fig. 11, the MCDM decision making module of the case study has the following nodes:

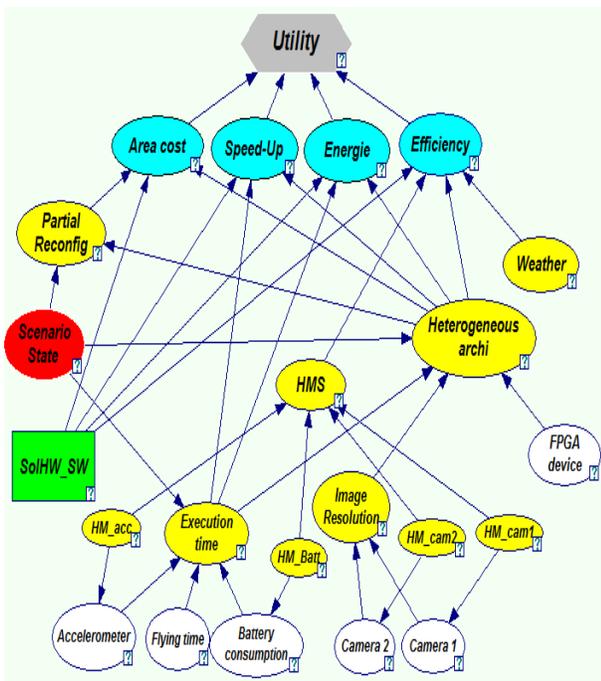


Fig. 11. The implementation of an example of the MCDM Model

- 1) Input node (red node, evidence node): current scenario state (Output of the decision making module).
- 2) Objective node (gray nodes, drawn as diamonds): to achieve a balance between different types of criteria cost area/energy consumption/speed-up/efficiency.
- 3) Decision node (Green node, drawn as rectangles): including the set of possible decision alternatives  $D_i$  (representing the possible combination of hardware and software : (1) pure HW/HW, (2) pure SW/SW, (3) mixed HW/SW, (4) mixed SW/HW).
- 4) Criteria nodes (blue nodes, chance nodes): including the set of decision criteria  $C_j$  such as cost area, energy consumption, speed up and the efficiency.
- 5) Sub-criteria nodes (yellow nodes, chance nodes): including a set of variables representing heterogeneous architecture, temporal constraints, internal states of

- 6) External factors nodes (white nodes, chance nodes): including a set of variables representing the sensors nodes (IMU, imaging sensors), FPGA platform, flying time (observed nodes).

In our case study, these nodes can have two states (state0,state1). Once the network structure is designed, the probabilities for each node are entered into the network in the form of Conditional Probability Table (CPT). For the criteria nodes, these probabilities are learned from the experimental results. Given observations of some external factors, the CPT entries are fixed on the basis of knowledge of interdependency between nodes and their interactions. The utility function  $Uf$  of each decision alternative  $Uf_{Di}$  ( $i$  from 0 to 3) is equal to the sum of the products of the performance of decision alternative  $Di$ , evaluated against the decision criterion  $C_j$  with the weights of relative importance of the decision criteria  $weight(j)$ :

$$Uf_{Di} = \sum_j (\sum_k \sum_t weight(j) * P(C_{jk}) * P(D_{it}))$$

with  $k=(state0,state1)$ ,  $t=(state0,state1)$

Once the overall utility scores are computed for all the alternatives, the best one is the alternative which has the highest score value and the decision alternative equal to the  $Max(Uf_{Di})$ .

## V. EXPERIMENTAL RESULTS

The experimental section is divided mainly into two parts. First, we present the hardware (FPGA) and software (CPU) implementations of the dynamic decision making module, including the MCDM module for HW/SW Partitioning an on-board heterogeneous architecture. Second, the proposed approach has implemented, and tested using a real case study.

### A. The Heterogeneous Architecture

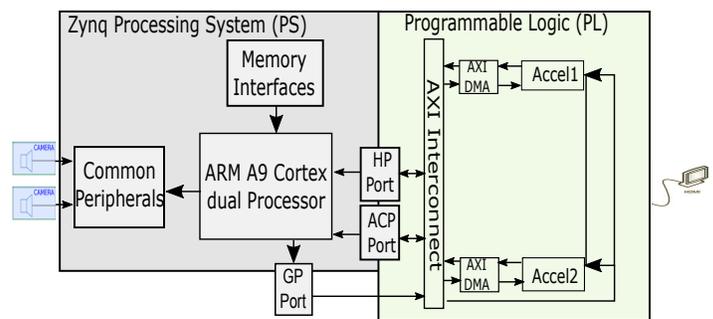


Fig. 12. Overview of the proposed hardware architecture

Our works aim at embedding the decision making module in a mixed hardware and software architecture (CPU and hardware cells) as the best way for low power consumption and high computational performances. In order to do so, the proposed design targets Xilinx Zynq XC7Z020 FPGA as the main processing chip. As shown in Fig. 12, the architecture is built around the ARM CortexA9 processor (Zynq processing system PS). In order to speed up image processing applications, the use of computational HW accelerators components

dedicated to some functions make the ARM CortexA9 processor available to adapt its functionality to perform other mission tasks according to mission objectives before the take off or even during the flight. As shown in Fig. 13, the ARM A9 Cortex dual processor can: (1) obtain the sensor data from two USB webcams, (2) communicate with dedicated hardware (HW) accelerators using the programmable logic through an Advanced eXtensible Interface (AXI) bus and (3) display the on a TV screen with the HDMI output.



Fig. 13. Setup used to validate the proposed UAV mission decision approach.

### B. HW/SW Implementations of The Proposed Mission Decision Making

In this work, we propose an extended framework [9] that can generate an FPGA bitstream from the BN specification. In fact, the hardware implementation of BN inference on synthesizable hardware as FPGAs can be divided into two main phases:

(1) Off-line phase: Once a BN is built, the designer has first to determine, some input BN specifications including evidence indicators (sensor measurement) and network parameters. Based on these specification choices, the developed tool generates the internal representation (Arithmetic circuit (AC) form) and then generates the corresponding C/C++ code (software version). Taken as input, the application written C or C++ languages, Vivado HLS (High Level Synthesis) generates low-level HDL description (Hardware version). The generated HDL code can be instantiated on SoPC Xilinx platform using high level design tools (Vivado) from Xilinx. Afterwards, the HDL code can be verified. Here, if the design does not meet the performance requirements, a first loop enables to re-configure and re-generate the HDL code, regarding resources utilization, speed up and energy consumption.

(2) Execution phase: during the execution of the program, data given by the sensors (evidence indicators) are transferred to the circuit and the developed BN module can be executed and the failure scenario is selected by computing the probability distribution function (pdf) of each node.

In the first experiment, we decide to put the network parameters and the evidence parameters into off-chip memories, with respect to real time processing, and then transmitted to hardware accelerators via the GP AXI bus. After synthesis, the resource requirement for the complete design (using the hardware version) is reported in Table I. Those results demonstrate

TABLE I. ZYNQ XC7Z20-1CLG484 - LOGIC SLICES UTILIZATION

Slice Logic Utilization	Used	Available	Utilization
Slice LUTs	32172	153200	21%
FF	7446	106400	19%
DSP48E	73	220	34%
BRAM (18k)	36	280	13%

that the mapping of our application is efficient in terms of resource usage.

In the second experiment, we use our design tool to evaluate the performance of the decision making based Bayesian Network module mapped onto the FPGA. In order to test the performance of the FPGA implementation, we measured the experimental times taken by the pure hardware and the pure software implementations to process a single decision making block. In the current implementation, the execution of the decision making module, using its hardware version, took  $42ms$ . Whereas, the execution of the decision making module (using its software version) running as a program on the CPU took  $155ms$ . Summarizing the measurements presented above, we obtain a speedup of 3.7 obtained when comparing the software version with the hardware version at clock frequency  $100MHz$ . However, the processing time obtained during the experiments has shown a considerable shortening of time and we can even improve these results by using high-level transformations.

### C. The Proposed Algorithms Applied in our Approach

Before we begin our application, the input for updating the proposed dynamic decision maker comes from the various sensors cameras placed in critical parts of the UAV. The principle is to acquire a sequence of images while the cameras (which will be embedded on UAV) is in motion. In our case, the UAV acquires data from working environment (immediate surroundings), in particular the data given by two cameras putted in the front of the UAV. Thus, the estimate of the decision can be updated based on the dataset collected up to that point in time. To show the validity of the proposed approach, we have conducted a set of experiments on real image sequences with resolution  $720 \times 480$  pixels captured with two Logitech webcams. To achieve the mission goals, the algorithms used in the work to verify the viability of our approach include in particular image processing techniques such as: motion detection based on mixture of Gaussian/Background subtraction (Fig. 14), image stabilization [11]), pose estimation [12] (Fig. 15), object recognition [14], object tracking [13]), region labeling and path planning. These applications based image processing are already embedded on-board FPGA in our autonomous UAV. Table II summarizes the different algorithms used in our approach.

Furthermore, we apply a simple method to measure distance between the UAV and the aerial object. For this purpose, we took two cameras slightly separated where the first one must be fixed in the front of the UAV and the second one must be at distance forward as shown in Fig. 16 to capture

TABLE II. THE PROPOSED ALGORITHMS

Application	Description
Region Labeling	Thresholding+Aggregation+Assigning labels
Path Planning	Breadth-First Search
Motion Detection	Application based on mixture of Gaussian Background subtraction
Pose estimation	Harris+Matching+3D movement estimation (RANSAC)
Image Stabilization	Application inspired from [11] Harris+Matching+2D Movement Estimation+ Motion Compensation
Object Recognition	Bottom-up visual attention approach
Object tracking	Dynamic Neural Fields (DNFs)



Fig. 15. FPGA pose estimation implementation

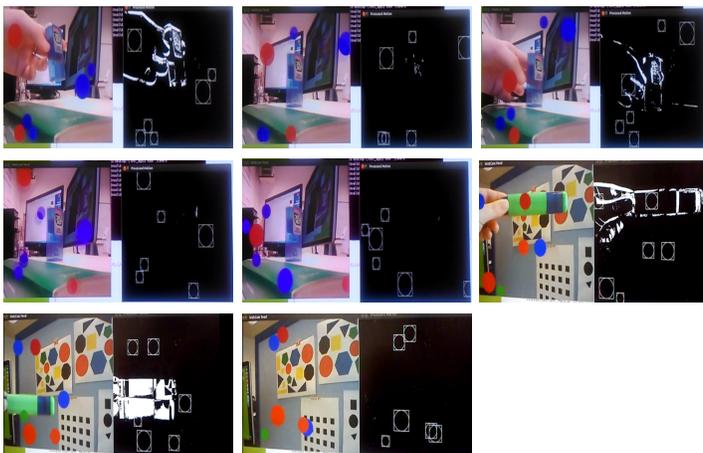


Fig. 14. FPGA motion detection implementation

the same image from two different camera angles at the same time. Thus, the distance between the camera and the aerial object  $d$  is given by:

$$d = \frac{m}{1 - \frac{x_1}{x_2}} \quad (1)$$

#### D. Case Study of the Proposed Approach

In this section, the goal is to validate our proposal on the Zynq platform through a real UAV mission with no anomalous or failed state detected. For example, we consider the following experiment that address only two scenarios ( $S_0$ ,  $S_1$ ):

- 1) At time  $t_0 = 0$ , the UAV starts moving in our indoor environment (Fig. 17) and continues moving in the same direction. The two cameras keep the same field of view as the time proceed (front and bottom). During this phase, the task is to detect any motion in the acquired image.
- 2) At time  $t_1 = 4s$ , the front camera detects movement (aerial target). After motion detection, region labeling application is applied to determine characteristics (shape, size, position, etc.) of each connected group and therefore to determine whether the detected motion is an object or just a noise.

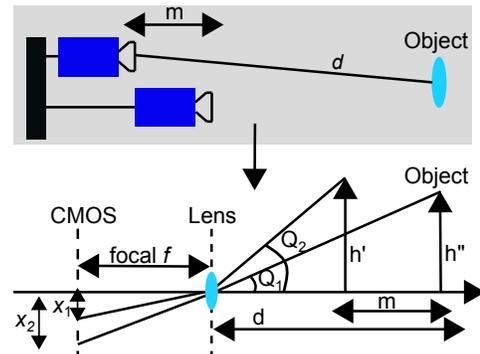


Fig. 16. Distance measurement method

- 3) At next time  $t_2 = 5s$ , no ground target is detected, and returns no threat detected in the front of the UAV.

After generating the bitstream of the embedded applications, we start to boot Linux on the Zedboard from the SD card divided into two partitions: (1) FAT32 partition including boot.bin (boot image for the Zynq), system.bit (Bitstream file of the custom IPs in the PL), application.elf, zynq\_fsb.elf (First Stage Boot Loader (FSBL)) (2) EXT4 partition including the Linaro file system. For the software implementation on ARM Cortex A9 dual core processor, the pose estimation algorithm takes more than  $252ms$  and the execution of the motion detection algorithm based on background subtraction took  $78ms$  and finally the region labeling task takes around  $42ms$ . Table III presents the synthesis results (synthesis under time constraints) in term of FPGA resources utilization and the processing time of each image processing application using hardware accelerators generated by the HLS Vivado tool.

On the basis of the dynamic decision making model, the moving vehicle pursues its goals by means of the developed

TABLE III. FPGA SYNTHESIS RESULTS

Application	Pose estimation	Motion detection
Slice LUTs	9%	13%
FF	2%	7%
BRAM (18k)	31%	28%
Execution times (ms)	127	35



Fig. 17. The indoor environment used to test the different applications

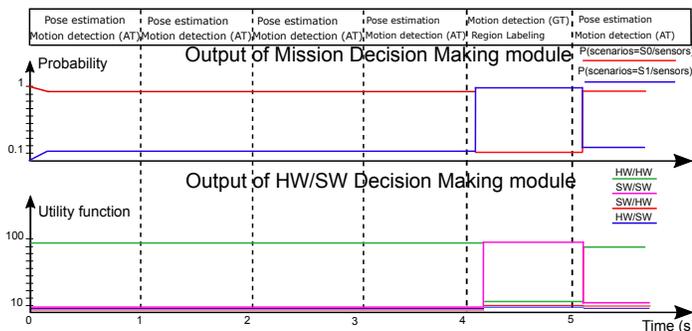


Fig. 18. Example of application of our approach

scenarios according to its available data about the state of the surrounding environment. Our mission decision making will be reactive, that means it will look for new scenarios when something happens according to its current knowledge. As first step, the embedded mission decision making module (the hardware accelerator) takes as evidences the output signals provided by the software and hardware components of the embedded applications. These evidences parameters are stored inside the FPGA using on-chip memories and transmitted by the ARM processor via general purpose (GP) AXI ports. Then, as second step, the choice of a functionality implementation (software version or hardware FPGA version) during the mission is driven by the the developed mission decision making module.

Fig. 18 presents the results obtained for the mission decision making over time with three applications of the case study. From time  $t_0 = 0s$  to  $t_1 = 4s$ , no problem in the mission is detected. Scenario  $S_0$  is the best possible scenario in this case since the probability  $P(S_0/evidence\ on\ sensors)$  is equal to 0.9 and the maximum of the utility function is attained for the decision alternative “SW/HW”, i.e., the motion detection is implemented as a hardware accelerator on the FPGA and the pose estimation software implementation on the ARM processor. At time  $t_1 = 4s$ , a moving object is detected. Scenario  $S_1$  is the best possible scenario in this situation since the probability  $P(S_1/evidence\ on\ sensors)$  increases and the maximum of the utility function is attained for the decision alternative “HW/SW”, i.e., the motion detection is implemented as a hardware accelerator on the FPGA and the region labeling software implementation on the ARM processor. These obtained results showed the ability of the decision making approach to choose the most suitable alternative scenario.

## VI. CONCLUSION

In this paper, we have presented an approach to Decision Making among alternative scenarios for UAV by using mixed DBN for dynamic decision making and MCDM for HW/SW partitioning according to the flying environment conditions as well as the imposed constrains. Thus, the current scenario of UAV can be modified to adapt its functionality to perform different tasks depending on the objectives of the UAV mission. Moreover, real time implementations (hardware and software) of the embedded Decision Making module on FPGA board are given in term of FPGA resource utilization and timing performance. To evaluate the proposed approach, we have demonstrated the implementation of real case study in heterogeneous architecture based on an hybrid device. In the current work, we attempt to validate the complete mission through the use of dynamic and partial reconfiguration at runtime to update the Decision Making and the complete embedded system under certainty. In future work, we may wish to focus on evaluating the performance of our proposal in outdoor environments.

## REFERENCES

- [1] K. Noori and K. Jenab, *Fuzzy reliability-based traction control model for intelligent transportation systems*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(1):229–234, 2013.
- [2] Hsin-Han Chiang, Yen-Lin Chen, Bing-Fei Wu, and Tsu-Tian Lee, *Embedded driver-assistance system using multiple sensors for safe overtaking maneuver*. IEEE Systems Journal, 8(3):681–698, 2014.
- [3] W. Ho, X. Xu, & P. K. Dey, *Multi-criteria decision making approaches for supplier evaluation and selection: A literature review*. European Journal of Operational Research, 202(1), 16–24.
- [4] J. Chai, J. N. K. Liu, & E. W. T. Ngai(2013), *Application of decision-making techniques in supplier selection: A systematic review of literature*. Expert Systems with Applications, 40(10), 3872–3885.
- [5] P. Dagum, A. Galper, E. Horvitz, A. Seiver, *Uncertain Reasoning and Forecasting*. International Journal of Forecasting, 11 (1): 73–87, 1995.
- [6] FV. Jensen, TD. Nielsen, *Bayesian networks and decision graphs*. 2nd ed. New York, NY:Springer;2007.
- [7] J. Pearl *Probabilistic reasoning in intelligent systems*. San Francisco, CA: Morgan Kaufmann;1988.
- [8] J. Schumann, T. Mbaya, O. J. Mengshoel, K. Pipatsrisawat, A. N. Srivastava, A. Choi, and A. Darwiche, *Software health management with Bayesian networks*, ISSE, vol. 9, no. 4, pp. 271–292, 2013.
- [9] S. Zermani, C. Dezan, H. Chenini, J.P. Diguët, R. Euler, *FPGA implementation of bayesian network inference for an embedded diagnosis-Prognostics and Health Management (PHM)*, 2015.
- [10] Y. W. Peng Waththayu, *A Bayesian network based framework for multi-criteria decision making*. In Proceedings of the 17th international conference on multiple criteria decision analysis. Whistler, British Columbia CA, 2004.
- [11] L. Domez, L. Sieler, A. Landrault, J.P. Derutin: *Embedding of a real time image stabilization algorithm on a parameterizable SoPC architecture a chip multi-processor approach*. J. Real-Time Image Processing 6(1): 47-58 (2011).
- [12] H. Chenini, D. Heller, C. Dezan, J.P. Diguët, D. Campbell, *Embedded real-time localization of UAV based on an hybrid device*. ICASSP 2015: 1543-1547.
- [13] Chenini. Hanen, *Pipelined architecture for real time detection and tracking of moving objects on an hybrid device*. ATISIP 2017: 1-6.
- [14] Chenini. Hanen, *A Bottom-up Approach for Visual Object Recognition on FPGA based Embedded Multiprocessor Architecture*. International Journal of Advanced Computer Science and Applications (IJACSA), 8(5) (2017): 474-482.
- [15] Schumann. Johann, Rozier. Kristin Y, Reinbacher. Thomas, Mengshoel. Ole J, Mbaya. Timmy, Ippolito. Corey: *Towards Real-time, On-board, Hardware-supported Sensor and Software Health Management for*

*Unmanned Aerial Systems*. Int. J. of Prognostics and Health Management, 6(1) (2015).

- [16] I. Sugiarto, C. Axenie, & J. Conradt, *FPGA-Based Hardware Accelerator for an Embedded Factor Graph with Configurable Optimization*. Journal of Circuits, Systems and Computers, 1950031, 2018.
- [17] D. Codetta-Raiteri, & L. Portinale, *Dynamic Bayesian Networks for Fault Detection, Identification, and Recovery in Autonomous Spacecraft*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(1), 13–24, 2015.
- [18] M. Lin, I. Lebedev, & J. Wawrzynek (2010), *High-throughput bayesian computing machine with reconfigurable hardware*. Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays - FPGA '10.
- [19] Z. Kulesza, & W. Tylman, *Implementation Of Bayesian Network In FPGA Circuit*. Proceedings of the International Conference Mixed Design of Integrated Circuits and System, 2006. MIXDES 2006.