IA Mission Planning for Autonomous Vehicles: Probabilistic Models and Embedded Versions

Catherine Dezan, dezan@univ-brest.fr

Lab-STICC/University of Brest (Bretagne, France)

ICMC, USP, Sao Carlos, March 14th 2018

Outline

Outline

- Introduction and motivations
- Probabilistic models to diagnose and to decide
- Embedded versions
- Conclusion and perspectives

Mission planning

Some key words of the presentation

- **Autonomous vehicles:** What does it mean exactly?
- **Uncertainty:** What kind?
- **IA methods:** Why do we need IA methods?
- **Embedded Systems:** What is the pb with the system?

Autonomy of the vehicles

The number of onboard tasks increases:

- vision-based navigation, path-planning, obstacle avoidance
- application camera tasks, aerial sampling
- sense and avoid, diagnosis, emergency procedure
- health management, communication mangement, computational resource management, energy and storage management, decision making

Autonomy levels related to human independence, mission complexity and environmental complexity (ALFUS [Huang 2007])

Uncertainty during the mission

Anomalies occurring during missions (mission degradations and system failure):

- Hardware failures (battery failure, GPS outage, sensor failure,...)
- Application degradations (slow access to memory, no-real time computation, memory overflow, ...)
- External environmental factors (strong wind, high pressure, extreme temperature,...)
- System-environment interactions (unexpected obstacle, no GPS zone, not enough light for the camera, unknown zone ...)

Monitoring/Diagnoses necessary for the adaptivity of the mission

$$\text{uncertainty} \Longrightarrow \text{probability}$$

Making a decision with unexpected events: IA methods

SENSE, LEARN and DECIDE

SENSE: deal with uncertainty, usage of sensors, missing data.
Probabilistic model (Bayesian programming, Markov decision processes)
LEARN: explore data, extract knowledge (deep learning, reinforcement learning, data mining)
DECIDE: classify, compute (decision tree, neural networks)

Reasonning with the probabilities: inference to diagnose, to make decision.
$\longrightarrow$ focus on some probabilistic models like Bayesian model and Markov Decision Model

Embedded Systems to support online computing

HPC (High Performance Computing) for embedded system on autonomous vehicles: estimation of 100*GFLOPs* under power consumption constraints.

Emerging proposal:

- Nvidia Deep Learning solutions with Jetson TX2 for embedded machine learning based on Neural Network (GPU architecture based at 50*GFLOPS*) (https://www.nvidia.fr/autonomous-machines/embedded-systems/)
- UCLA Deep Learning (http://vast.cs.ucla.edu/projects/acceleration-deep-learning-cloud-and-edge-computing) on FPGA (up to 600*GFLOPS*)

Outline

Introduction and motivations

Probabilistic models to diagnose and to decide

Embedded versions

Conclusion and perspectives

Probabilistic models to diagnose and to make decision

Reasonning with the probabilities is the key point

Different types of models in this talk :

- Bayesian networks: to diagnose
- Makov Decision Process: to make decision

What is a BN?

- Probabilistic and graphical modelization of the relationship "cause to effect".
- Based on observed data, deduce the probability of unobserved ones (**inference**).
- In the model:
  - nodes represent random variables,
  - probability tables represent a priori or conditional probabilities,
  - arcs indicate the conditional dependencies.

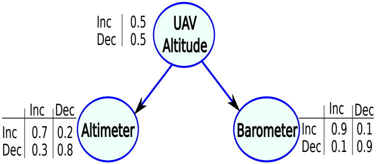


Figure: BN Example

What is a BN?

- Probabilistic and graphical modelization of the relationship "cause to effect".
- Based on observed data, deduce the probability of unobserved ones (**inference**).
- In the model:
  - nodes represent random variables,
  - probability tables represent a priori or conditional probabilities,
  - arcs indicate the conditional dependencies.

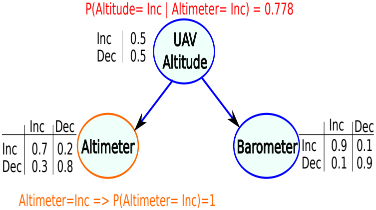


Figure: BN Example

What is a BN?

- Probabilistic and graphical modelization of the relationship "cause to effect".
- Based on observed data, deduce the probability of unobserved ones (**inference**).
- In the model:
  - nodes represent random variables,
  - probability tables represent a priori or conditional probabilities,
  - arcs indicate the conditional dependencies.

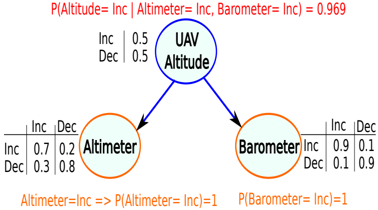
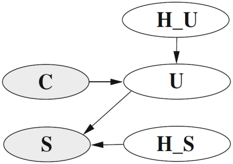


Figure: BN Example

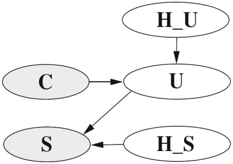
Health Management: NASA Proposal [Schumann 2011]



- Command node C (evidence): action or command from the exterior.
- Health node H (output): health of systems (H_U) or sensors (H_S).
- Sensor node S (evidence): software or hardware sensors (measure).
- Status node U: unobservable status of component.

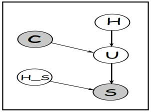
$\Rightarrow$ P(H|S,C): inference.

Health Management: NASA Proposal [Schumann 2011]



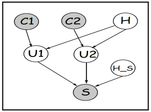
- Command node C (evidence): action or command from the exterior.
- Health node H (output): health of systems (H_U) or sensors (H_S).
- Sensor node S (evidence): software or hardware sensors (measure).
- Status node U: unobservable status of component.

$\Rightarrow$ P(H|S,C): inference.



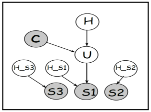
- External command of writing (node C).
- Logical sensor (node S) indicating the file space.

Application 1 Writing operation into a file.



- External command of pitching up or down (node C1 or C2).
- A hardware accelerometer sensor (S).

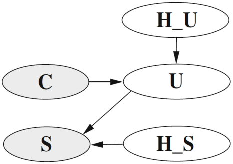
Application 2 Pitching up or down.



- External command of computing altitude(node C).
- Three hardware sensors (node S1, S2 and S3).

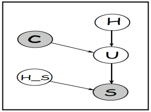
Application 3 Altitude computation.

Health Management: NASA Proposal [Schumann 2011]



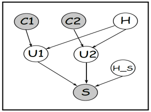
- Command node C (evidence): action or command from the exterior.
- Health node H (output): health of systems (H_U) or sensors (H_S).
- Sensor node S (evidence): software or hardware sensors (measure).
- Status node U: unobservable status of component.

$\Rightarrow$ P(H|S,C): inference.



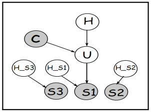
- External command of writing (node C).
- Logical sensor (node S) indicating the file space.

Application 1 Writing operation into a file.



- External command of pitching up or down (node C1 or C2).
- A hardware accelerometer sensor (S).

Application 2 Pitching up or down.



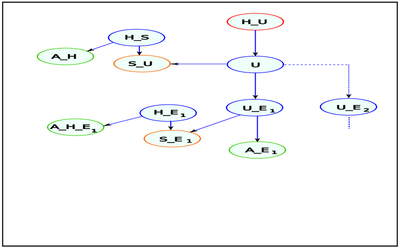
- External command of computing altitude(node C).
- Three hardware sensors (node S1, S2 and S3).

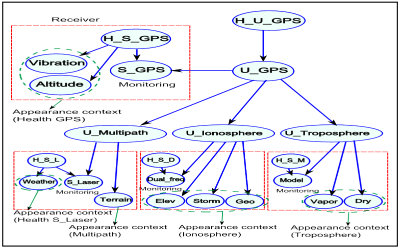
Application 3 Altitude computation.

HM extensions: HM proposals from Lab-STICC

Sara Zermani [MECO 2016]: An automatic translation from FMEA to BN

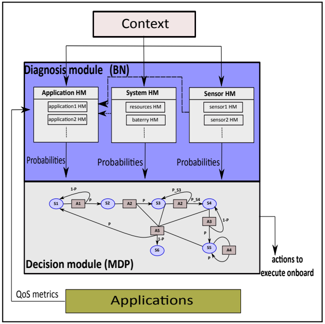
Error type (E)	Monitor (M)	context E (error type)	Context M (Monitor)
U_E_i	S_E_i	$A\_E_{i\_j}$	$A\_H\_E_{i\_u}$

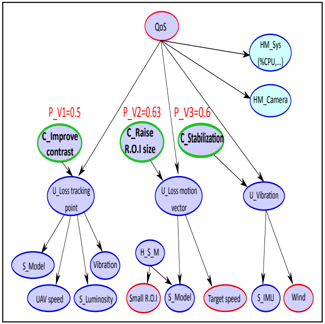




HM extensions: HM proposals from Lab-STICC

BN to mitigate errors
Chabha Hireche [ICRA 2018]: case of application adaptation with BN





Probabilistic Models to make a decision

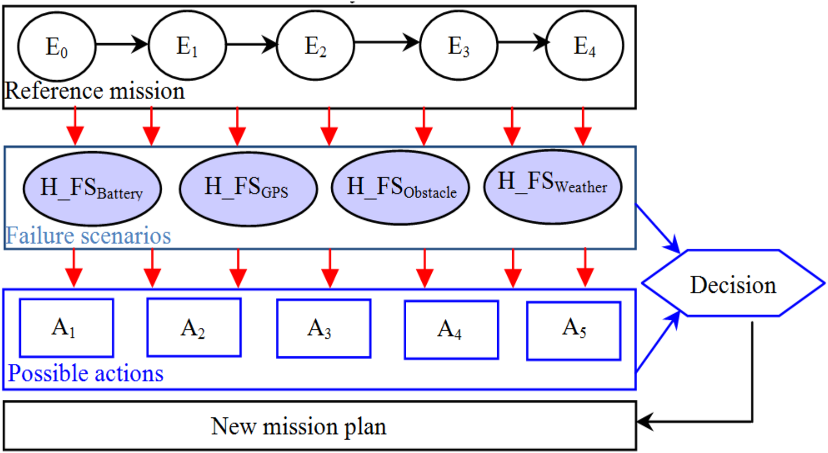
An action is chosen but hazards can alter the result of the choice. The alternatives are known.
Type of hazards to face during a mission:

- Internal hazards: defective sensors (GPS, Camera, altimeter), defective on-board computer (memory overflow, Memory leakage, bandwith, CPU), defective battery, ...
- External hazards: unexpected weather (strong wind, high luminosity, low temperature, ...), unexpected obstacle (birds, drones, unexplored area,...)

Ex: GPS is likely not to work anymore, commute to the SLAM system to get the localization
Ex: There is some strong wind making vibration at the UAV level, we probably need to activate extra-stabilization for image processing during the tracking
Decision making models can incorporate the prob computed by BN

Decision Making under failure scenarios

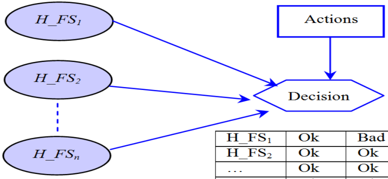
Sara Zermani [MECO 2017]



Influence Diagram for the Decision Making module

▷ Influence Diagram that determines the most appropriate recovery action for the mission based on:

- evaluation of the Health status given by the Failure Scenario Management (H_FS) module,
- the possible recovery actions.

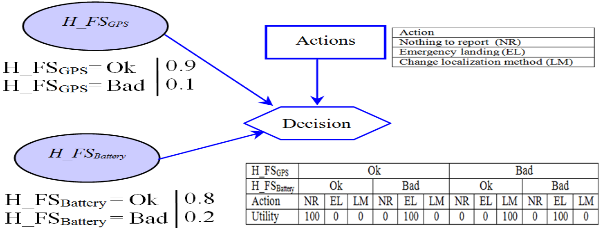


Action
A_1
A_2
...
A_m

H_FS_1	Ok	Bad	...	Bad
H_FS_2	Ok	Ok	...	Bad
...	Ok	Ok	...	Bad
H_FS_n	Ok	Ok	...	Bad
Action	A_1	A_2	...	A_m
Utility	U_1	U_2	...	U_{m*2^n}

Solving the Decision Making is done by maximizing a utility function, calculated by the utility table and the probabilities of the failure scenarios Health status.

Decision Making for the GPS and Battery failure scenarios



H_FSGPS	Ok						Bad					
H_FSBattery	Ok			Bad			Ok			Bad		
Action	NR	EL	LM	NR	EL	LM	NR	EL	LM	NR	EL	LM
Utility	100	0	0	0	100	0	0	0	100	0	100	0

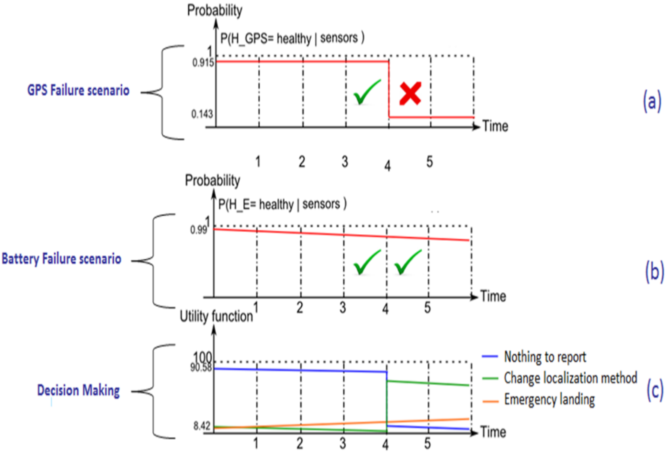
Recovery action= Max (U_F_{NR} , U_F_{EL} ,U_F_{LM}),
$U\_F_k(k = \{NR, EL, LM\})$ is equal to:

$$U\_F_k = \sum_i \sum_j U(A = k, H\_G = i, H\_B = j) * P(H\_G = i) * P(H\_B = j)$$

(i= Ok, Bad and j= Ok, Bad, G for FS_{GPS}, B for $FS_{Battery}$, U for Utility and A for Action)

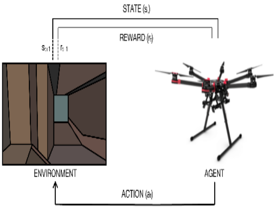
$U\_F_{NR} = 72$, $U\_F_{EL} = 20$ and $U\_F_{LM} = 8 \Rightarrow$ nothing to report.

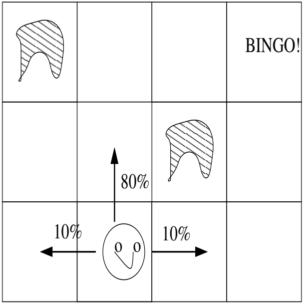
Decision Making for the GPS and Battery failure scenarios

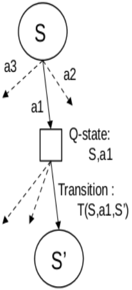


Other probabilistic model for decision making

- Limitation of Influence Diagram: maximize the current return
- Markov Decision Model (MDP): maximize the expected return including the future steps (cumulative, discounted reward).

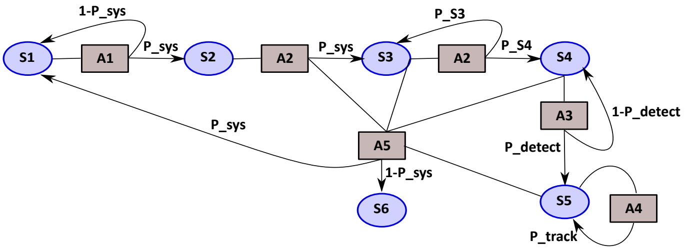






MDP example: UAV mission

Case of tracking mission. Chabha Hireche [ICRA 2018]



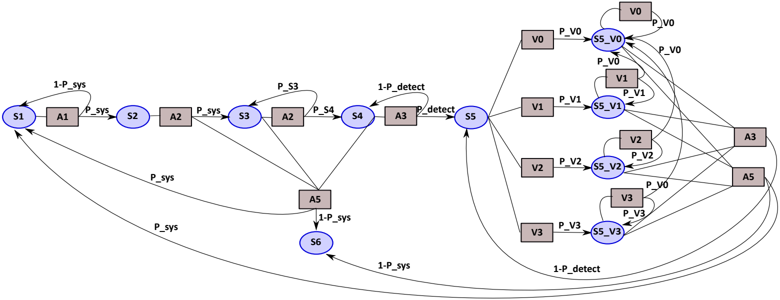
Actions: Take-off (A1), Follow trajectory (A2), detect (A3), track (A4), return to base (A5)
Probabilities on the transition functions: *P_sys*, *P_S3*,*P_S4*, *P_detect*, *P_track* (BN computation)

BN model to migate errors

Possible adaptations related to the context

Errors	Possible monitoring	Appearance context	Solution (algorithms)
Vibration	IMU Vibration sensor	Wind Vibration	Activate the stabilization (V1)
Tracking point lost	Model based on: number of features detected (Harris)	Drone speed variations of luminosity	Improve the contrast (V2)
Motion vector lost	Model based on: motion vector between 2 images	Target speed Small R.O.I	Raise the R.O.I size (V3)

Resource/context Aware MDP



Probabilities to find the most suitable version of the tracking application

BFM model: Bayesian Networks built from FMEA tables for MDP

BFM benefit

Expected benefit:

Scenario	Tracking time (nbr steps)		# Cycles (10^6)
	Reference	BFM	Tracking version
Nominal	51	51	103 (320x240 frame)
Vibrations	40	65	292 (nominal + stabilization)
High Speed	40	72	264 (640x480 frame)
Wind	35	56	264

Extension of BFM model for UAV mission

"just-enough" quality management with

- Concurrent MDPs to handle different applications during the mission: MDP navigation (obstacle avoidance), MDP landing (emergency area), MDP tracking
- Resolution of behavior conflicts (Return base is antagonist with Landing), Resource conflits (Resource sharing)
- Interaction with the application scheduler (reconfiguration controller)
- Experiment on FPGA-SoC from Altera for HPeC project

Outline

Introduction and motivations

Probabilistic models to diagnose and to decide

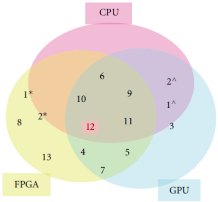
Embedded versions

Conclusion and perspectives

Embedded versions for autonomous vehicles

Embedded versions driven by performance, resource and energy constraints
What are the most suitable execution support (MCPU, GPU, FPGA)?

	Dwarf	Examples/Applications
1	Dense Matrix	Linear algebra (dense matrices)
2	Sparse Matrix	Linear algebra (sparse matrices)
3	Spectral	FFT-based methods
4	N-Body	Particle-particle interactions
5	Structured Grid	Fluid dynamics, meteorology
6	Unstructured Grid	Adaptive mesh FEM
7	MapReduce	Monte Carlo integration
8	Combinational Logic	Logic gates (e.g., Toffoli gates)
9	Graph traversal	Searching, selection
10	Dynamic Programming	Tower of Hanoi problem
11	Backtrack/ Branch-and-Bound	Global optimization
12	Graphical Models	Probabilistic networks
13	Finite State Machine	TTL counter



Source: R. Palmer(paralleldwarfs.codeplex.com)

Parallel pattern for HPC

Source: Ra Inta, Chimera plateform, 2012

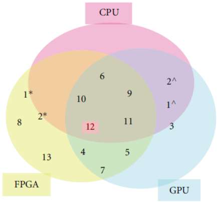
The most appropriate hardware for speed-up

It is only a question of speed-up?

Embedded versions for autonomous vehicles

Embedded versions driven by performance, resource and energy constraints
What are the most suitable execution support (MCPU, GPU, FPGA)?

	Dwarf	Examples/Applications
1	Dense Matrix	Linear algebra (dense matrices)
2	Sparse Matrix	Linear algebra (sparse matrices)
3	Spectral	FFT-based methods
4	N-Body	Particle-particle interactions
5	Structured Grid	Fluid dynamics, meteorology
6	Unstructured Grid	Adaptive mesh FEM
7	MapReduce	Monte Carlo integration
8	Combinational Logic	Logic gates (e.g., Toffoli gates)
9	Graph traversal	Searching, selection
10	Dynamic Programming	Tower of Hanoi problem
11	Backtrack/ Branch-and-Bound	Global optimization
12	Graphical Models	Probabilistic networks
13	Finite State Machine	TTL counter



Source: R. Palmer(paralleldwarfs.codeplex.com)

Parallel pattern for HPC

Source: Ra Inta, Chimera plateform, 2012

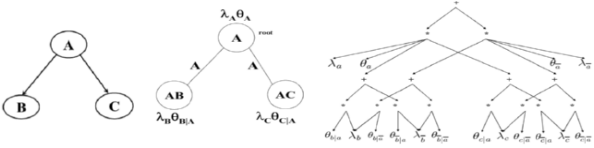
The most appropriate hardware for speed-up

It is only a question of speed-up?
FPGA can be interesting for different aspects: energy efficency, real-time, non-intrusive...

Embedded versions for diagnosis

Different algorithms for BN inference but complexity problem

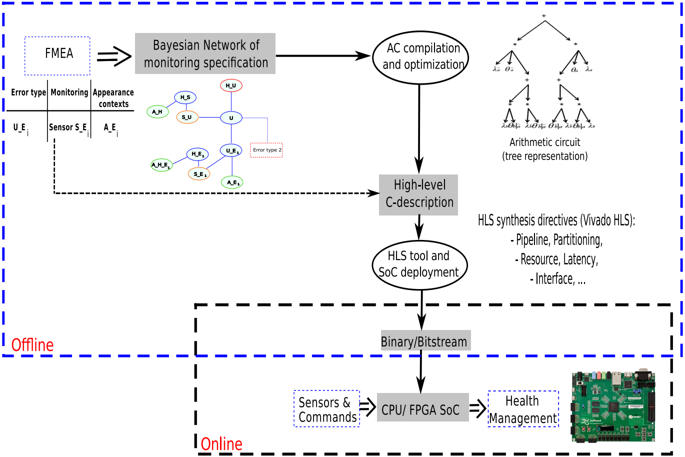
- conditionning [Pearl 88], variable elimination [Zhang 96]
- junction tree [Jensen 90]
- Differential methods with Arithmetic circuit [Darwiche 2003]
- Approximated inference: stochastic simulation, Gibbs sampling, ...



Bayesian network   Jointree   arithmetic circuit

Tools: Hugin, Genie, BNT ToolBox (Matlab), Samlam,...
Few embedded version: with AC on FPGA [Schumann 2015]

Design tool for embedded FPGA/SoC version



Sara Zermani [AHS 2015, MicroProc 2017]

SoC/FPGA experiment

ZedBoard from Xilinx incoporates in a the Zynq SOC (XC7Z020)

- Dual ARM Cortex-A9 processor (Zynq processing system PS) 866MHz + L1,L2 cache
- Programmable logic (FPGA part for HW accelerators) with 53K LUT, 100K FF, 560 BRAM, 220 DSP slices
- communicating with HW accelerators through AXI bus 32/64 bits
- storage on-chip or off-chip

CPU/FPGA experiment with BN

Examples of speed-up

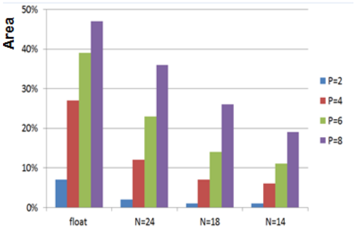
- JT/AC compilation on CPU

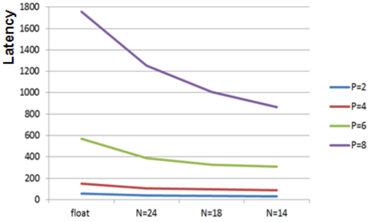
Virual AC	JT inference time (ms)	AC inference time (ms)	Speed-Up SW
P=2 (7 nodes, 14 λ)	13.0	12.1	1.07
P=4 (31 nodes, 62 λ)	42.4	18.7	2.26
P=6 (127 nodes, 254 λ)	162.5	24.4	6.65
P=8 (511 nodes,1022 λ)	627.3	31.6	19.85

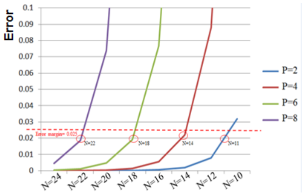
- FPGA implementation on Zedboard

Virtual AC	Area	AXI GP Mem On HW Speed-Up	AXI ACP Mem Off HW Speed-up
P=2	(20% DSP, 02% FF, 07% LUT)	1.52	0.699
P=4	(60% DSP, 12% FF, 38% LUT)	3.73	1.83
P=6	(60% DSP, 28% FF, 68% LUT)	5.14	3.27
P=8	(60% DSP, 38% FF, 92% LUT)	6.15	4.81

Bitwidth exploration with FPGA for BN





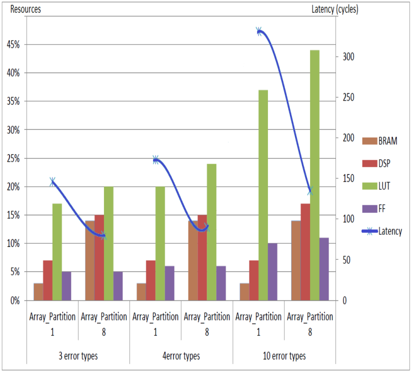


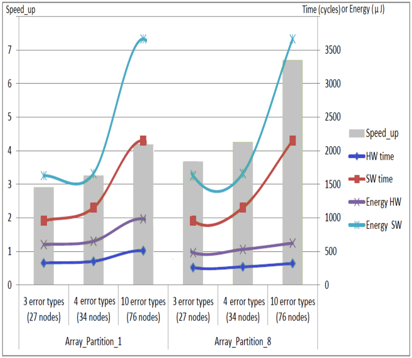
Trade-off between latency, resources and error margin

e.g:if an accepted error margin is $0,025$, then
$N = 14$ is enough for $P = 4$
$N = 18$ is enough for $P = 6$
...

High-level directives for parallel versions of HM

High-level HLS directives: ALLOCATION (resource constraint), ARRAY_PARTITION (Memory mapping), LATENCY (latency constraint), INLINE, PIPELINE, UNROLL, ...





Embedded versions for decision making

Example with influence diagram: utility function (HW/SW) + BN (HW)

			Resource BRAM	DSP	LUT	FF	Timing (cycles)	SpeedUp	Energy. (μ J)
2 BN (54 nodes)	2 Actions	HW	14%	14%	23%	12%	310	5.96	6.33
		HW/SW	14%	6%	20%	11%	528	3.50	10.50
	10 Actions	HW	14%	24%	32%	16%	355	7.33	7.56
		HW/SW	14%	6%	20%	11%	1328	1.96	26.42
10 BN (270 nodes)	2 Actions	HW	61%	41%	78%	34%	6001	11.12	170.42
		HW/SW	60%	6%	44%	24%	60370	1.10	1521.32
	10 Actions	HW	80%	60%	80%	40%	11025	21.3	330.84
		HW/SW	60%	6%	44%	23%	232475	1.02	5835.12

Strategy for a SoC/FPGA implementation:

- HW/SW version only for a small number of BN
- linear progression of HW timing related to number of actions
- exponential progression of HW timing related to number of BN

Outline

Conclusion

Autonomy + uncertainty => embedded probabilistic methods

- Bayesian Networks for diagnosis, for mitigation
- Influence Diagrams and Markov Decision Process (MDP) for decision making

- Embedded versions: tradeoff between performance/resource/energy => hybrid device (CPU/GPU/FPGA)
- Need of design tools/frameworks for designers and automatic mapping on heterogeneous systems : SDSOC, HLS, openCL, openCV, ...

Example of UAV mission planning

Perspectives

- HW/SW embedded MDP, BFM for advanced UAV decision making
- Dynamic reconfiguration to embed more applications on SWaP constraints (Size Weight and Power) => HPeC project
- Security perpectives => IoMoT project with Kalinka Branco (USP/ICMC)
- Embedded IA perspectives: Deep Learning + probabilistic model (Deep Reinforcement learning)

Thanks Girls!

PhD students involved: Sara Zermani (2013-2017), Chabha Hireche (2015-2018) and Arwa Khannoussi (2016-2019)
French CNRS PICS project (2013-1015) and French HPeC project (2015-2019)

Partnership: Jean-Philippe Diguet (CNRS/Lab-STICC), Luis Mejias (ARCAA/QUT), Kalinka Branco (USP/ICMC), Rosana Braga (USP/ICMC)