



**HAL**  
open science

## An approach for Fault Tolerant and Performance Guarantee Autonomous Robotic Mission

Philippe Lambert, Lionel Lapierre, Didier Crestani

► **To cite this version:**

Philippe Lambert, Lionel Lapierre, Didier Crestani. An approach for Fault Tolerant and Performance Guarantee Autonomous Robotic Mission. AHS 2019 - NASA/ESA Conference on Adaptive Hardware and Systems, Jul 2019, Colchester, United Kingdom. pp.87-94, 10.1109/AHS.2019.00009 . hal-02160494

**HAL Id: hal-02160494**

**<https://hal.science/hal-02160494>**

Submitted on 19 Jun 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An approach for Fault Tolerant and Performance Guarantee Autonomous Robotic Mission

P. Lambert  
LIRMM, UMR 5506  
Université de Montpellier, CNRS  
Montpellier, France  
Philippe.Lambert@lirmm.fr

L. Lapierre  
LIRMM, UMR 5506  
Université de Montpellier, CNRS  
Montpellier, France  
Lionel.Lapierre@lirmm.fr

D. Crestani  
LIRMM, UMR 5506  
Université de Montpellier, CNRS  
Montpellier, France  
Didier.Crestani@lirmm.fr

**Abstract**—Long duration autonomous missions are still challenging objectives for robotics. This paper presents a new methodology using performance points of view to guide hardware and software resources management according to mission execution and fault occurrence. Experimental results on a patrolling mission are presented. It also detail how localization guarantee is managed and what impact it has on the overall methodology and its performances.

**Index Terms**—Autonomous robotics, performance guarantee, energy-aware management, safety, mission duration, localization, fault tolerance.

## I. INTRODUCTION

Long duration autonomous and complex missions are still challenging in robotics. Two main problems are the energy management and the fault tolerance [1]. This paper presents a novel methodology intending to manage the hardware and software resources allocation, before and during the mission, while satisfying performance objectives. Its aim is to address some Cyber-Physical Systems (CPS) issues and challenges [2] such as fault tolerance, energy management, resources management, (unanticipated) mission scenario.

After a state of the art, the experimental context is presented. Then the proposed methodology is summarized and mapped on an illustrative mission and experimental results are presented. Finally, before concluding, localization issue is addressed and simulation comparison is done.

## II. STATE OF THE ART

There is no common definition of autonomy. However, generally, decisional and behavioral autonomy can be distinguished [3]. Decisional autonomy concerns actions which are allowed to be performed and is related to the question “*what to do?*”. Behavioral autonomy concerns the ability to successfully perform actions despite the influence of environmental factors. It is related to the question “*How to perform what has been decided?*”. The work we present in this paper deals with behavioral autonomy.

Whatever the decision, behavioral autonomy raises 3 key factors: an objective to be satisfied, a finite set of alternatives to execute this objective, and a criterion to decide which

alternative must be chosen. As Sukhatme stated in [4], decision in robotics is related to multi-criteria points of view and performance is the relevant criteria to take a decision.

Performance is a widely used concept in robotics but it has not been yet clearly defined. Looking into business domain [5] performance is multi-dimensional, related to an objective, and is the result of an action, according to the involved resources. Measure (with regard to a reference), estimation (confidence range) and evaluation (interpretation) can be distinguished. Performance inductors influence the performance, performance indicators measure/estimate the performance.

In industrial robotics numerous well suited and accepted performance indicators are defined [6]. The most common ones are load capacity, workspace, speed, acceleration, repeatability, and accuracy. More specific ones can be used for manipulation or dexterity performance estimation. Some performance criteria have been standardized like (ISO/DIS 9283, 2015). However the industrial context is quite different from the autonomous mobile robotics. Most of the time the industrial environment can be considered as static, and the available energy can be considered as infinite.

It is clear that the performance concept is currently used in many works proposed to evaluate the efficiency of a new control law or a new localization technique. But in fact, few works address the mission level performance evaluation for autonomous mobile robots where many metrics have been proposed in the literature.

In [7] the authors argue that, for mobile robots, performance metrics are often limited to mission duration or path-length. Focusing on robot’s navigation, they propose other indicators that are security oriented (collisions per mission, obstacle clearance), quality oriented (path length, smoothness), mission success (successful mission percentage). A deeper analysis is performed in [8] where different robotic tasks like motion control (accuracy, speed, ...), SLAM (accuracy, ...), obstacle avoidance (collision counting, obstacle clearance, ...), visual-servoing (error positioning, stability, ...) are addressed.

Mission performance guarantee has been proposed in [9] [10] where a formal analysis of the robotic missions is done before the mission with regard to performance properties (safety, liveness, mission efficiency, ...). However no solution is proposed to guarantee the expected performances during the

The authors would like to thank Labex NUMEV, Site MUSE and FEDER funds for their support.

real mission.

As a conclusion, there is a lack of accepted performance criteria for mobile robotic mission. Surprisingly, the energy viewpoint is rarely mentioned. Moreover performance analysis concerns mainly post-experimental results, depending largely from experimental context. The work proposed here intends to decide before the mission if the mission performance objectives can be reached or not. It can also maintain the quality of service, as best as possible, during the mission, when some unforeseen events occur or when robot's hardware/software element fails.

In the sequel, we concurrently consider the following viewpoints: **i) Energy**, the robot must have enough energy to be able to finish its mission; **ii) Safety**, the robot must be safe for itself and its environment; **iii) Duration**, the mission has dead-line duration; **iv) Localization**.

The objective of the localization is to estimate the position (x,y) of a mobile robot in a known map of its indoor environment (no GPS). Numerous localization techniques with their own strengths and weaknesses have been proposed in the literature. Dead-reckoning techniques build an estimation of the robot's movement from proprioceptive measurements. Repositioning techniques use exteroceptive information (vision or depth) to estimate the relative position of the robot with respect to environment features/landmarks (natural or artificial, active or passive). Position tracking can also be addressed by probabilistic techniques [14]. The position-tracking problem can be also addressed using techniques aiming to solve the global localization problem where the initial pose of the robot is unknown. Grid based localization techniques allow to deal with multi-modal and non-Gaussian densities using histograms representation of the robot's estimated position over a regular grid.

The localisation is supposed to be satisfied without guarantee in the first part of this work and is guaranteed at the end of this paper.

### III. EXPERIMENTAL CONTEXT

#### A. The Robot

The considered robot is a Pioneer 3DX (VRmax = 0.75 m/s) integrating 16 sonars (US) and 10 bumpers. Its weight is about 25 kg. Two top to tail URG-04 LX Hokuyo (laser1 laser2) are used for obstacle avoidance and robot localization. They allow 360° horizontal scanning of the surrounding environment. A Kinect camera is also used for localization using geo-referenced markers and for image capture. An embedded lead/acid battery generates theoretically up to 259 Wh of energy. The robot's energy consumption is monitored using an embedded wattmeter. The robot communicates with an embedded laptop supporting our different algorithms. The laptop has its own battery, which is also monitored. Switch boards allow for switching on/off some of robot's sensors (the kinect and each lasers independently). Depending on the used algorithms and sensors, the following Control Schemes (CS) are available: 7 Forward Motion control laws (FM), 3

TABLE I  
MISSION DECOMPOSITION

$x_i - x_{i-1}$ (m)	0 - 34	37	37	37	37 - 93.5	93.5	93.5	93.5	93.5 - 187								
Objective $O_i$	DS → V1	○	⊗	○	V1 → V2	○	⊗	○	V2 → DS								
$n_{alt_i}$	21	2	1	2	21	2	1	2	21								
Task $T_k$	FM / L	R	VD	R	FM / L	R	VD	R	FM / L								
$A_k^{c_j}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$n_{alt_k}$	21	9	3	2	1	2	9	21	21	7	2	1	2	21	21	9	21
$V_{max}$ (m/s)																	

Localization methods (L), a single on-place Rotation technique (R) and an image analysis (Valve Detection: VD).

#### B. Experimental environment

The experimental environment is presented in Fig. 1. It is a part of the LIRMM laboratory. On the right side there is an experimentation room (ER) of approximately 10 m long with a docking station (S) for the robotic system. This room crosses a long straight corridor (H1) of approximately 60 meters long and between 1.7 m and 2.2 m width. Within H1, a glazed area G is present. This corridor turns then to the right towards a new straight corridor (H2) of approximately 30 m long and more than 2.2 m width. Along this environment, there are two areas V1 and V2 with a valve which can be open or close. Moreover markers have been spread along these corridors to facilitate the robot localization.

#### C. The Patrolling Mission

The mission of the robot is to go from DS to inspect the state (open or closed) of V1 and V2 before returning back.

This global mission of about 187 meters long can be initially decomposed into a sequence of 9 objectives reported at Table I: to go from the docking station to the localization of V1; face V1; inspect the state of V1; turn back on track; go to the localization of V2; face V2; inspect the state of V2; turn back on track; then go back to the base.

The problem of performance guarantee is "how can we ensure the success of the mission", knowing that:

- The mission must be safe.
- The robot and the laptop have a limited (and known) amount of energy.
- The mission duration cannot exceed a fixed limit.

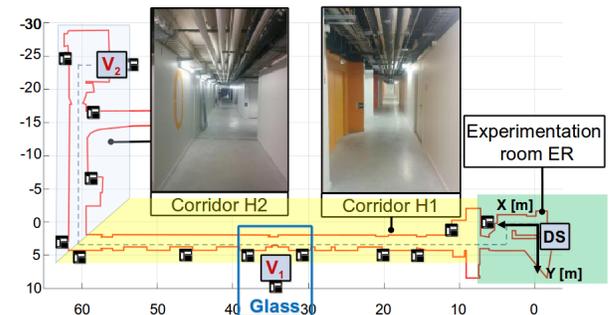


Fig. 1. Mission environment

- Some unforeseen events may appear during the mission execution like unexpected obstacle avoidance, or hardware and/or software faults.

The following assumptions are supposed to be verified:

- The control loop frequency (10 Hz) is sufficient to ensure the control stability of any of the considered algorithms.
- In the 1<sup>st</sup> part of this paper we consider that the used sensors and localization algorithms are sufficient to locate the robot with a desired accuracy.

In Table I, the first mission objective is : "Go from DS to VI". This task requires to realize simultaneously two tasks: FM (Forward Motion) and L (Localization). Each task can be implemented using and configuring different hardware (sensors) and software (algorithms) resources. That is to say that we have 7 choices for FM and 3 choices for L. That leads to 21 distinct choices ( $N_{alt}$ ) for the first objective. The same reasoning can be done for each mission objective. The underlying problem is the following: "How can we choose and parameterize the robot's resources to be used along the mission, to succeed and satisfy all the required performance objectives (Safety, Energy, and Duration)?" In the 1<sup>st</sup> part of this paper this problem is formulated as a Knapsack problem which is well known to be NP-complex. To reach this goal the following methodology has been developed, implemented and experimentally tested.

#### IV. THE PANORAMA METHODOLOGY

##### A. Main approach and concepts

The PANORAMA (Performance and AutoNomy using Resources Allocalization Management) methodology can be decomposed into 3 main phases (Fig. 2): preliminary (P1), off-line (P2) and on-line (P3).

The **preliminary phase** performs the identification of the performance inductors using cause-effects diagrams. A performance inductor corresponds to an element which can be controlled to influence a performance objective.

We project the performance constraints on the initial mission scenario to decompose it into a sequence of activities involving invariant constraints. Each activity  $A_k$  can be realized using  $n_{altk}$  Alternatives of Implementation (AI) (see Table I). This nominal scenario is called Nominal Mission Plan (NMP). From the NMP, the **off-line phase** objectives are: to estimate the nominal performance along the process; to identify resources allocalization alternatives and to specify the value of inductors in order to respect the performance constraints. These constraints can be respected if we are able to estimate the local performance of each AI. Each alternative must be characterized with respect to each performance axis. Hence, the activity duration, the corresponding energy consumption, the localization quality and the safety level, must be estimated. The second condition is the ability to estimate the global performance of the NMP mission, by composing the local performance estimation for each eligible AI.

The resources allocation resolution of the Knapsack problem is obtained using the algorithm proposed in [11]. It allows to

find a feasible solution in an efficient way, compatible with the real time constraint.

If a solution (determining which hardware and software resources can be used for each mission's activity, but also which Performance inductors values must be imposed) is found the final on-line phase can be engaged. Otherwise the mission cannot succeed with the current performance constraints.

Finally during the **on-line phase** the robot periodically monitors its actual energy consumption and mission duration and compares them with the expected ones to decide whether the performance objectives can be still guaranteed.

Two important concepts are defined:

- The **duration margin** corresponding to the difference between the estimated/real mission duration and the performance duration constraint  $D_{max}$ .
- The **energy margin** corresponding to the difference between the estimated/real mission energy consumption and the performance energy constraint  $E_{max}$ .

Which event can imply to reconsider the current Resource allocation Solution (RS)?

- Negative margin detection: that means that at the end of the mission, the energy and/or duration performance objectives cannot be satisfied: the mission cannot continue using the same resources allocation.
- Hardware or software fault detection: that means that the faulty element belongs to the current RS, so it is currently used or planned to be used. This resource can be eliminated from the possible alternatives of an activity.

If the current RS becomes unfeasible, a new one must be determined using the same approach that for the off-line

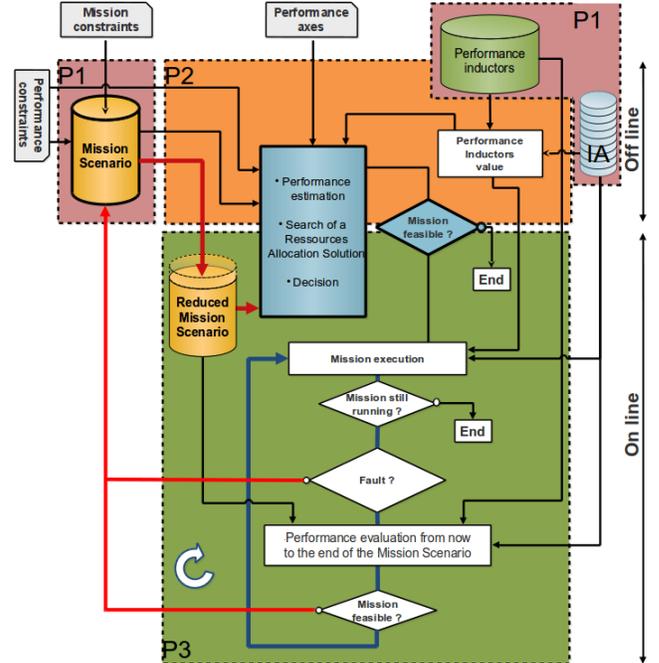


Fig. 2. PANORAMA Methodology principles

process. The only difference is that we just have to consider the rest of the mission from the current situation of the robot.

### B. Mapping the methodology on the considered mission

The following elements are considered to map the methodology on the different performance axes and to find a Resources allocation Solution.

- **Safety:** the safety must be ensured all along the mission. It is implemented considering obstacle avoidance capacity. To avoid an obstacle the robot must be able to detect it by selecting relevant sensors (Sonars must be used in the glazed area), and to avoid the obstacle using the Safe Manoeuvring Zone principle (SMZ). But also the robot must be harmless when an obstacle cannot be avoided (situation of a moving obstacle, potentially human). ISO-10128 imposes a velocity of 0.25 m/s. French law imposes that the impact energy must be less than 4J.
- **Duration:** The robot's speed is supposed to be constant during an activity. This choice depends on the security constraints and on the motion energy consumption.
- **Energy:** To estimate the energy consumption of an AI detailed and complete identifications of motion (depending on the chosen velocity), used sensors, and software energy consumption models have been realized [12]. It demonstrates that the amount of consumed motion energy to travel a given distance presents a unique minimum. It supposes straight line path and constant velocity. Energy margin helps to absorb the difference between these hypotheses and the reality.

Finally, for each mission activity, a velocity can be defined like shown in color on the last line of Table I. Green boxes correspond to the maximal possible velocity (blue 0.32 m/s, green 0.4 m/s, orange 0.52 m/s and grey 0 m/s). Two main strategies are applied at the same time:

- To choose the velocity during an activity we choose to move as fast as possible while remaining safe. This strategy maximizes the duration margin and enhances the robustness to unforeseen obstacles for example.
- For the resources allocation algorithm we choose the most energetic solutions **while** satisfying the global mission energy constraints.

It can be noticed in Table I that the initial 9 mission objectives have been decomposed into 17 activities to get homogeneous performance constraint for each activity, each having  $n_{altk}$  AI respecting the constraints of the activity. Hence, globally the Knapsack allocation algorithm must find a solution within a state space equal to  $\prod n_{altk} > 10^{13}$  for the patrolling mission.

## V. THE EXPERIMENT

The experiment [13] takes place in the corridor described in section III-B with the robot presented in section III-A. The following performances objectives are defined for the mission:

- Duration axis: D max = 600 s
- Energy axis: Two performance indicators

- max robot energy: ER max = 2.4 Wh
- max laptop energy: EL max = 2.8 Wh
- Safety axis: Two performance indicators
  - Obstacle avoidance: Verified
  - Harmlessness: SH max = 4 J

The experimental mission succeeded satisfying all the performance constraints.

Fig. 3 shows the observed velocities along the mission.

Fig. 4 shows the variation of the 3 measured margins allong the mission:  $E_{LM}$  for the energy margin of the laptop battery,  $E_{rM}$  for the energy margin of the robot battery and  $D_M$  the Duration margin.

In Fig. 4 RSi represents an event requiring a new Resources allocation Solution:

- RS0: 1<sup>st</sup> RS solution generated at the end of the off-line phase of PANORAMA.
- RS1: Negative  $E_{rM}$  energy margin detection due to odometry drift, obstacle avoidance and consumption model error.
- RS2: Simulated sensor fault on laser2. This sensor is used in current and planned activities.
- RS3: Negative  $E_{rM}$  energy margin detection due to odometry drift, obstacle avoidance and consumption model error.

Table II shows the different Resources allocation Solutions (RS), and, for each RS the chosen IA (Implementation Alternative) planned for the mission numbered from 1 to 7. The green boxes correspond to the real executed control schemes.

The following modifications occur on RSi events:

- RS1: On activity 17, AI (4) switches to AI (5). The laser 1 is planned to be disconnected for activity 17 to decrease (in the future) the energy consumption.
- RS2: Due to laser2 fault the selected IA (6 or 7) uses only laser1, US or Kinect sensors.

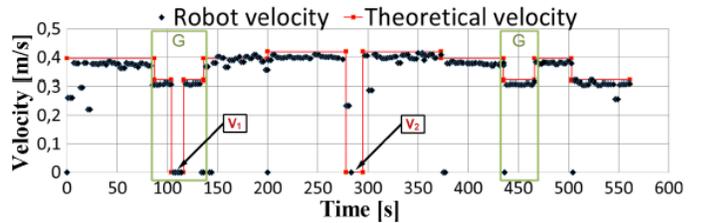


Fig. 3. Experimental versus theoretical velocities

TABLE II  
GENERATED RESOURCES ALLOCATION SOLUTIONS

$A_k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
RS <sub>0</sub>	1	1	1	2	3	2	1	1	1	1	2	3	2	1	1	1	4
RS <sub>1</sub>								1	1	1	2	3	2	1	1	1	5
RS <sub>2</sub>														6	6	7	6
RS <sub>3</sub>																	7

AI = {(1): SMZ-2LAS-US/KIN, (2): OPR/KIN, (3): VALVE ANALYSIS, (4): SMZ-1LAS/KIN, (5): PF/KIN, (6): SMZ-1LAS/KIN, (7): SMZ-US/KIN, etc.}

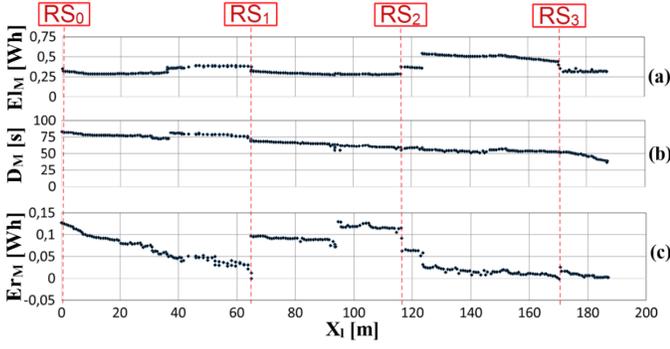


Fig. 4. Performance margins

- RS3: The last IA (7) disconnects laser1 to use US sensors and decrease the energy consumption.

#### A. Strengths and limitations

This approach has many interesting strengths:

- It addresses the performance issue at the mission level.
- It addresses the concept of performance guarantee which is important in real mission.
- It manages the energy point of view that is rarely considered and is central for real autonomous mission.
- It enhances the mission robustness finding adapted recovery solutions in case of hardware or software resources failures, or performance drift.

However many limitations can be also identified:

- The mapping of the performance constraints on mission scenario is a complex process.
- The sensors, actuators, electronic cards and software components consumption identification is a long process.
- The chosen energy strategy maximizing locally the energy consumption is efficient since the energy constraint is globally satisfied. However, as it will be exposed in the sequel, a minimum energy optimization can be performed when considering the localization performance axis.

The underlying assumption for this strategy is that the more efficient localization technique is, the more energy it needs. If this assumption was not done, the odometry based localization technique will be always selected. It is well known that this kind of localization technique drifts significantly with the traveled path. Hence, it cannot be always selected to ensure a localization with prescribed quality along the mission. The following section addresses in detail the localization issue and allows for considering a minimization criterion for the performances points of view.

## VI. LOCALIZATION METHODS AND ACCURACY

It is very important for the PANORAMA approach to be able to monitor during the mission the performance indicators. Unfortunately, GPS can't be used in indoor environment. Only external landmarks with known localization can be used. This is why the localization guarantee objective is particularly difficult to address and verify.

#### A. Localization uncertainty: multiple errors origins

To reach its goals, a robot needs to know its current location. However, it exists a set of uncertainties that affects the true localization of the robot before the moment where this information is used to compute the control of the robot. The objective here is to identify the cause of uncertainty in the localization estimation.

- The environment itself is the true reference, but the environment model where the robot must locate itself has necessarily approximations. Some elements (landmarks) cannot be identified at their true position. We call this class of error Landmark Position Error LPE.
- The sensor measurements present uncertainty. We call this class of error Sensor Measure Error SME.
- The used localization methods induces, due to its own principles, a localization error. We called this class of error Localization Method Error LME.
- The position of the sensor device with respect to the robot's reference frame must also be considered. We call this class of error Mechanical Design Error MDE.
- The delay between the measurement and the use of localization estimation implies that the robot will have moved during this time interval. We call this software error Architecture Implementation Error AIE.

When the pose uncertainty estimation is made with a local method (1) is used. When the estimation is done with a global method (2) is used.

$$\widehat{X}_i = \widehat{\Delta}_{dt} + \widehat{T}_K * (\widehat{T}_{cr} * \widehat{M}_{i,loc})^{-1} \quad (1)$$

$$\widehat{X}_i = \widehat{\Delta}_{dt} + (\widehat{T}_{cr}^{-1} * \widehat{M}_{i,glb}) \quad (2)$$

where:

- $\widehat{X}_i$  is the final pose estimation.
- $\widehat{\Delta}_{dt}$  is the estimation of the robot's displacement since the last sensor measurement: affected by AIE.
- $\widehat{T}_K$  is the transformation to replace the robot in the world frame: affected by LPE.
- $\widehat{T}_{cr}$  is the transformation from the sensor frame to the robot frame: affected by MDE.
- $\widehat{M}_{i,-}$  is the raw estimation: affected by SME, LME, LPE.
- The symbol  $\hat{x}$  is the estimation of  $x$ .

#### B. Experimental localization methods and error models

To be able to project constraints on our mission scenario and predict our performances, we need to model the behavior of our localization methods and their errors. 3 different localization methods are considered.

1) *Odometry*: the odometry is a Dead-reckoning technique widely used in mobile robotics. It is possible to get a displacement estimation from the robot odometer measurements. We use the Borenstein square methodology [17] to estimate the static error parameters of the robot. We then use (3) to describe the odometry error. Experimental results allow to get the covariance matrix parameters  $k_r$  and  $k_l$  of (4) where

$\Delta s_r$  and  $\Delta s_l$  are the linear length of right and left wheel displacements.

$$\sum_{p'} = \nabla_p f \sum_p (\nabla_p f)^T + \nabla_{\Delta ag} f \sum_{\Delta} (\nabla_{\Delta ag} f)^T \quad (3)$$

$$\sum_{\Delta} = cov(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r * \Delta s_r & 0 \\ 0 & k_l * \Delta s_l \end{bmatrix} \quad (4)$$

2) *KIN*: the robot is equipped with a Kinect camera and geo-referenced markers are use as a repositioning technique. The camera estimates the position of the markers and the transformation given in (1) provides the estimation of the global robot position. This method is prone to errors due its sensitivity of cumulative errors of orientation estimation (present in LPE, SME, LME and MDE). We use interval analysis to determine the range of position error depending on the distance where the marker has been seen. Markers are regularly spread in the environment. Therefore the Kinect oriented localization is coupled with odometry to have a permanent localization method using odometry between each marker detection.

3) *Grid Based Localization (GBL)*: GBL is a grid based localization technique using 360° distance reading to locate the robot in a 20cm grid. In a searching window, in-line lidar measurements are compared with several precomputed primitives, extracted from simulation, and indexed to each cell of the grid. A correlation is realized with each cell-candidate to select the best pose. Experimentally this method, in this regular environment of laboratory corridors is able to correctly locate the robot at least every 5s. Between two estimations, the robot localization is based on odometry.

Due to the composition of these structural localization errors the pose uncertainty remains important, whatever the localization method.

## VII. LOCALIZATION PERFORMANCE MANAGEMENT

### A. Description of the localization performance axis

Localization guarantee means that during the mission one must be able to estimate the position of the robot (with a prescribed uncertainty) in x and y. As for any performance axis, the Localization integration process requires to map the Localization constraints on the mission scenario:

- Firstly, the expected performance guarantee of Localization must be defined by the user. That means that the desired localization uncertainty can vary along the mission.
- Secondly, we suppose that, knowing the nature of the environment, the domain of application of a localization method M and its involved sensors, we are able to identify on the planned scenario, the different zones where a given method can be used.
- We suppose that for each method a model of the uncertainty is known.
- Some experimental constraints can also be added like a maximal robot velocity when using some sensor (e.g. Kinect camera to be able to obtain stable camera frame).

### B. Initialisation management

A problem set by the localization axis is to be able to guarantee the localization constraint (uncertainty) from the instant we switch between activities. Localization methods often requires conditions to be able to produce result. It has to be launched before its use, creating a performance correlation between 2 activities. This settling period does not exist if the same localization method is used in two consecutive activities. Therefore these initialisations requirements must be considered in addition to the Localization performance capabilities.

The previous energy performance principle was to maximize the energy consumption while satisfying globally the consumption limits. If we are able to identify, for each activity, the relevant localization techniques capable of satisfying the localization accuracy prescription, we will be able to select the less consuming one. Then, globally we will be able to address, for the mission at least, the question of energy consumption minimization.

### C. Ressources allocation with performance minimization

In a first step, to be able to guarantee the end-mission performances (Energy/Duration) we still use the previous filtering algorithm [11]. However each AI is initialized considering the worst possible local conditions to verify globally the end-mission performances guarantees. In a second step instead of selecting the most consuming solution in energy, we will use the Viterbi algorithm to find a minimal cost solution. The Viterbi algorithm [15] has been initially developed to find the most likely sequence of hidden states according to a sequence of observations in a Hidden Markov Model. This problem is isomorphic in finding the minimum cost path through this class of graph. Considering a Trellis graph T having N states (T width) and a sequence of length M (T Length), there is  $N^M$  possible state sequences. The dynamic programming approach proposed by Viterbi leads to a polynomial time complexity in  $O(N^2M)$ . Some complexity improvements can be found in [16].

Using the notations of the previous section, the following identifications can be done to map the Viterbi algorithm with the PANORAMA methodology:

- The trellis length M, corresponds to the number of mission's activities.
- N is the total number of independent available AI.
- In an activity, a state (node) is characterized by the performances of the considered AI depending on the estimated activity duration and the required sensors.
- Between two nodes (states) of the trellis, the edge is valued by the incremental performance cost corresponding to the transition from the node i to the node j.  $Cs(i,j)$  denotes the settling delay needed for an usable Localization technique of section s+1 to be operational from the section s.
- The Viterbi algorithm considers a trellis structure. But the allocation problem formulation leads to a less complex graph since, for each activity, only one subset can be

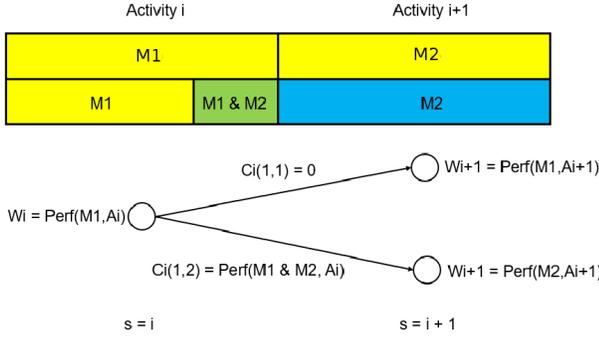


Fig. 5. Simplified Viterbi graph between two activities

considered. Therefore, between two activities the edges only connect each possible state of activity  $s$  to each possible state of activity  $s+1$ . Obviously, if an edge connects two identical states, its corresponding valuation is equal to 0 since no settling delay is needed (Fig. 5).

Therefore, using the Viterbi algorithm, it is possible to determine a path through the mission scenario, leading from the starting state to a final state, minimizing an utility performance function.

The proposed utility function (5) is composed by the maximum duration and energy limits ( $D_{MaxMission}$  and both of  $E_{MaxLap}$  and  $E_{MaxRob}$ ).

$$F(E_{Rob}, E_{Lap}, D) = \frac{1}{N} (\alpha * \frac{E_{Rob}}{E_{MaxRob}} + \beta * \frac{E_{Lap}}{E_{MaxLap}} + \gamma * \frac{D}{D_{MaxMission}}) \quad (5)$$

Where:

- $\alpha$ ,  $\beta$  and  $\gamma$  denotes the weight of each considered point of view. They are defined by the user before the off-line phase.
- $N$  denotes the normalisation coefficient:  $\alpha + \beta + \gamma = N$ .
- $E_{MaxRob}$ ,  $E_{MaxLap}$  and  $D_{MaxMission}$  are the maximal usable performance resources for the robot and laptop energy consumption and the mission duration fixed by the user.
- $E_{Rob}$ ,  $E_{Lap}$  and  $D$  are the estimation of the level of energy consumption and duration for a considered state or transition cost.

This utility function provides the possibility to modulate dynamically the weights if, during the mission, some performance points of view are difficult to satisfy.

### VIII. COMPARISON OF PERFORMANCE MISSION GUARANTEE WITH OR WITHOUT LOCALIZATION MANAGEMENT

#### A. Localization in the mission description

The basic mission used in this simulation is the same as in Sec.III-C and other objectives (Security, Energy, Duration) have the same values as in Sec.V. In the following description the localization accuracy performance is  $L[Xs, Ys]$  where  $Xs$  denotes the uncertainty constraint (in meter) along the axis followed by the robot and  $Ys$  the lateral axis.

TABLE III  
RESOURCES ALLOCATION SOLUTIONS AND ENERGY MARGINS FOR SED AND SLED APPROACHES

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Erob Margin	Elap Margin	Duration Margin			
SED RS <sub>0</sub>	1	1	1	2	3	2	1	1	1	2	3	2	1	1	1	1	4	0.13Wh	0.14Wh	79.0s			
SED RS <sub>1</sub>															1	1	4	12	0.021Wh	0.032Wh	51.4s		
SED RS <sub>2</sub>																	8	0.023Wh	0.0059Wh	51.4s			
SLED RS <sub>0</sub>	1-1	1-2	2	3	4	5	6	7	8	9-1	9-2	10	11	12	13	14	15	16-1	16-2	17	Erob Margin	Elap Margin	Duration Margin
SLED RS <sub>1</sub>	8	8	9	7	10	3	2	9	8	8	8	7	10	3	2	8	8	9	9	7	0.44Wh	0.27Wh	109.05s
SLED RS <sub>2</sub>																					0.31Wh	0.16Wh	49.27s

AI={(1):SMZ-2LAS-US/KIN, (2):OPR/KIN, (3):VALVE\_ANALYSIS, (4):SMZ-LAS1-US/KIN, (6):SMZ-LAS1/KIN, (7):SMZ-US/KIN, (8):SMZ-LAS1/GBL, (9):SMZ-US/GBL, (10):OPR/ODOM, (11):SMZ-2LAS/GBL, (12):CENTERING-2LAS/ODOM, ect }

The following localization constraints have been introduced for the robot in the mission scenario:

- $L[0.65 \ 1]$  in H1.
- $L[0.75 \ 2]$  in H2.
- $L[0.45 \ 1]$  approaching V1 and V2.
- The Viterbi cost function has the following static parameters:  $\alpha = 1$ ,  $\beta = 0$  and  $\gamma = 0$ . We are only trying to minimize the robot energy consumption. But every performances criteria will still be verified.

#### B. Description

We consider two simulations, according to the chosen performance guarantees: **SED** for Security, Energy and Duration and **SLED** adding Localization.

#### C. Comparison of SED and SLED performances

We compare here, the efficiency of the SED (Maximal energy) and SLED (Minimal energy) approaches with the following scenario events:

- SED RS<sub>0</sub>: 1<sup>st</sup> RS solution generated at the end of the off-line phase.
- SED RS<sub>1</sub>: Negative  $E_{lap}$  energy margin detection due to obstacle avoidance at S=120m.
- SED RS<sub>2</sub>: Simulated sensor fault on US at S=160m. This sensor is used in current and planned activities.
- SLED RS<sub>0</sub>: 1<sup>st</sup> RS solution generated at the end of the off-line phase.
- SLED RS<sub>1</sub>: Simulated sensor fault on US at S=160m. This sensor is used in current and planned activities.

Table III shows the different Resources allocation Solutions (RS). For each RS the chosen IA planned from 1 to 13 are presented and the expected end-mission margins are specified. The green boxes correspond to the actual executed control schemes in each simulation.

As shown in Table III SLED approach leads to a huge increase of energy margins (1348% for the robot and 2712% for the laptop) and a little decrease of duration margin (4.1%). Moreover the SLED approach prevents undesired reconfigurations (like SED RS<sub>1</sub>). Table IV shows the full description of the final mission scenario according to SLED RS<sub>0</sub>. We can notice that there are more activities (20) in SLED simulation with regard to SED (17). The three new activities come from the necessary KIN settling delays. A settling method is active but is not used for localization.

TABLE IV  
SLED  $RS_0$  TASK ASSIGNMENT

Ak	1-1	1-2	2	3	4	5	6	7	8	9-1
D_Start (m)	0.0	26.5	31.0	34.0	37.0	37.0	37.0	37.0	41.0	63.5
D_End (m)	26.5	31.0	34.0	37.0	37.0	37.0	37.0	37.0	63.5	86.5
Task	Moving	Moving	Moving	Precise Moving	Turning	Image Analysis	Turning	Moving	Moving	Moving
Task Method	SMZ LAS1	SMZ LAS1	SMZ US	SMZ US	TURNING	Image Analysis	TURNING	SMZ US	SMZ LAS1	SMZ LAS1
Localization Method	BGL	BGL	BGL	KIN	ODOM	ODOM	KIN	BGL	BGL	BGL
Settling Method		KIN	KIN		KIN	KIN				

Ak	9-2	10	11	12	13	14	15	16-1	16-2	17
D_Start (m)	86.5	91.5	93.5	93.5	93.5	93.5	123.5	146.0	146.5	156.0
D_End (m)	91.5	93.5	93.5	93.5	93.5	123.5	146.0	146.5	156.0	187.0
Task	Moving	Precise Moving	Turning	Image Analysis	Turning	Moving	Moving	Moving	Moving	Moving
Task Method	SMZ LAS1	SMZ LAS1	TURNING	Image Analysis	TURNING	SMZ LAS1	SMZ LAS1	SMZ US	SMZ US	SMZ US
Localization Method	BGL	KIN	ODOM	ODOM	KIN	BGL	BGL	BGL	BGL	KIN
Settling Method	KIN		KIN	KIN					KIN	

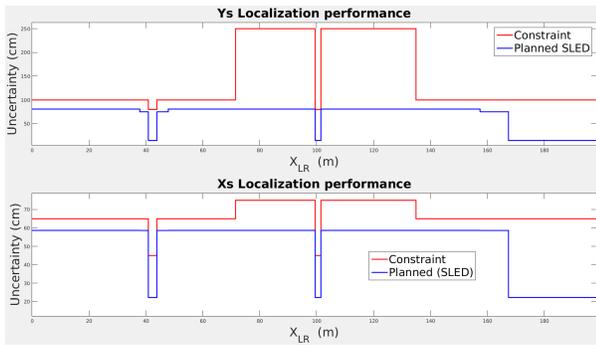


Fig. 6. Localization constraint and uncertainty for SLED  $SR_0$

Fig. 6 shows the shape of the localization performances along SLED  $RS_0$ . The localization margin is the difference between the localization constraint and the planned localization uncertainty. The combination 8 SMZ-LAS1/GBL seems to be the most efficient since it is used when the constraints are the less demanding: Not glazed, Average uncertainty. The Kinect is used sparsely when more precise localization constraint is needed near V1 and V2. The GBL planned uncertainty is close to the localization constraint. If this method cannot produce a satisfying localization every 5s, the odometry drift may induce a localization fault.

## IX. CONCLUSION

This paper presents a practical methodology designed to manage dynamically hardware and software resources allocation at mission level while satisfying performance constraints objectives. The energy, duration and localization margins concepts and efficient allocation algorithms allow for developing and implementing in real-time a strongly robust approach which is able to deal with many type of faults (environment, hardware or software, etc.). The proposed work permits to address many needed CPS properties for self protecting and healing and for the self optimizing and configuring points of view.

Further researches are planned. The experimental implementation of the SLED PANORAMA approach integrating the localization performance dimension is in progress. In the

future we intend to integrate this efficient behavioral autonomy principle with decisional autonomy approaches to address globally the autonomy issue.

## REFERENCES

- [1] J. Carlson and R. Murphy, "How UGVs physically fail in the field", In: IEEE Trans. Robot. 21.3 (2005), pp. 423-437. Issn:15523098. doi:10.1109/TRO. 2004.838027.
- [2] J. M. Bradley and E. M. Atkins, "Optimization and control of cyber-physical vehicle systems", In: Sensors 15.9 (2015), pp. 23020 – 23049.
- [3] J. M. Bradshaw, P. J. Feltovich, H. Jung, S. Kulkarni, W. Taysom and A. Uszok, "Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction", Agent and Computational Autonomy, Springer-Verlag, pp. 235-268, 2004.
- [4] G. S. Sukhatme, "On the Evaluation of Autonomous Mobile Robots", PhD., University of Southern California, 1997.
- [5] A. Neely, Business Performance Measurement: Theory and Practice, Cambridge University Press, 2004.
- [6] V. Scheinman and J. M. McCarthy, Mechanisms and actuation, in Springer Handbook of Robotics. Springer, pp. 67-86, 2008.
- [7] N. Ceballos, J. Valencia and N. Ospina, "Quantitative performance metrics for mobile robots navigation", Intech Open Access Publisher, 2010.
- [8] F. Bonsignorio, J. Hallan and A. P. Del Pobil, "Good experimental methodology guidelines". In: Special Interest Group on Good Experimental Methodology in Robotics European Robotics Research Network (EURON), Tech. Rep (2008).
- [9] D. M. Lyons, R. C. Arkin, S. D. Fox, P. Nirmal and J. Shu, "Designing autonomous robot missions with performance guarantees", IEEE/RSSJ Int. Conference on Intelligent Robots and Systems (IROS) 2012, pp. 2583-2590, Vila Moura, Algarve Portugal, Oct. 7-12th 2012.
- [10] M. O'Brien, R. C. Arkin, D. Harrington, D. M. Lyons and S. Jiang, Automatic verification of autonomous robot missions", Simulation, Modelling and Programming for Autonomous Robots (Springer Lecture Notes in AI: 8810), pp. 462-473, Bergamo Italy, Oct. 2014.
- [11] M. Bennour, D. Crestani, O. Crespo and F. Prunet, "Computer aided decision for human task allocation with mono and multi performance evaluation", International Journal of Production Research, Vol. 43 (No. 21), pp. 4559-4588, 2005.
- [12] L. Jaiem, S. Druon, L. Lapiere and D. Crestani, "A step towards autonomous robotic systems", In Towards Autonomous Robotic Systems, p 189-195, 17th Annual Conference, TAROS 2016, Sheffield, UK.
- [13] L. Jaiem and D. Crestani, (2016), <https://youtu.be/jdTiLjeKC78>
- [14] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, The MIT Press, 2005.
- [15] G. D. J. Forney, "The Viterbi algorithm", Proceedings of the IEEE, 61(3), pp. 268-278, 1973.
- [16] A. Backurs and C. Tzamos, "Improving Viterbi is hard: better runtimes imply faster clique algorithms", Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 311-321, 2017.
- [17] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots", IEEE Transactions on Robotics and Automation-Volume12, pp. 869 – 880, 1996.