

FROM TOPOLOGICAL KNOWLEDGE TO GEOMETRICAL MAP

François Tièche and Heinz Hügli

*Institute of Microtechnology, University of Neuchâtel
Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland
Phone: +41 32 718 34 55, Fax: +41 32 718 34 02
E-mail: heinz.hugli@imt.unine.ch*

Abstract: The behavioral approach to robot navigation, characterized by a representation of the environment that is topological and robot-environmental interactions that are reactive, is preferable to the pure geometrical navigation because it is far more robust to unpredictable changes of the environment. Nevertheless, there is still a need to obtain geometrical maps. This paper considers a geometrical map reconstruction that relies on the topological knowledge and uses redundant odometric measurements taken while the robot moves along the paths of the topological map. Five methods are presented and compared in experiments involving a Nomad200 mobile robot operating in a real environment.

Keywords: Autonomous mobile robots, Knowledge representation, Least-square method, Robot navigation, Robot vision, Robot control, Robotics

1. INTRODUCTION

A map of the environment is needed for a mobile robot to carry out navigation tasks. Various map representations and numerous map construction approaches have been considered. First there are geometrical maps, which integrate sensed data in a single frame of reference. In the *Certainty Grid* approach (Elfes, 1989), the certainty about the existence of obstacles, detected by sonar, is reported in a grid map. In an other approach, Crowley (1989) constructs geometric feature maps of line segments by means of an extended Kalman filter. Therefore these geometrical maps give an accurate description of the environment and can be used to compute optimal robot paths. However they provide a poor interface to symbolic planning units, use large amount of data, and require a complex process in order to maintain the map consistency in large environments.

Topological maps overcome some of these limitations. They represent the environment as neighborhood relationships of distinctive places. Places differ by their sensing signatures like sonar signatures (Kurz, 1993) or sonar and vision signatures (Kortenkamp and Weymouth, 1944). In another approach Thrun and Bücken (1996) use Voronoi squeletisation to extract identical topological regions from a grid map and then create a topological map.

All approaches described so far enter the class of "sense-map-plan-act" robot architectures, that is known to be inefficient to react quickly to unpredictable changes of a dynamic world. On the contrary the class of behavioral architectures allows the robot to move around safely, even in dynamic environments, by means of a set of individual behaviors that provide strong robot/environment interactions. Topological maps are suited to represent these interactions (Mataric, 1990). Such a topological map known as *cognitive map* has been proposed by Kuipers and Byun (1991). In this approach, distinctive places correspond to the activation of a particular class of behaviors, called *self-positioning* behaviors. These behaviors control the robot movements and lock it into a specific pose relative to particular environmental characteristics: the *self-positioning* site. Also neighborhood relations are expressed by behaviors which servo the robot between two *self-positioning* sites.

Although the behavioral navigation resists generally well to changes of the environment, the associated topological map is completely useless in the case of a loss of behavioral stimulation. These limitations can be avoided by extending the topological map with additional geometrical information. As a benefit, such a new map allows to determine paths that haven't been explored. It also provides an interface which is better understandable by a human operator.

This paper presents a way to extend the knowledge of a topological map of *self-positioning* sites by construction of a consistent associated geometrical map. This construction proceeds by integrating recorded odometric paths. Five methods are proposed to integrate these paths into a single frame of reference according to the topological map.

The paper is organized as follows: After a description of the mobile robot architecture in section 2, it describes in section 3 the behaviors that are used in connection with the topological map. Then, section 4 describes formally the topological map, and section 5 describes how the geometrical information is added to the map. Section 6 presents the five methods used to construct the consistent geometrical map. The experimental results are shown in section 7, and section 8 concludes this paper.

2. MOBILE ROBOT ARCHITECTURE

The robot architecture (Hügli *et al.*, 1994) follows the principles of the behavioral approach. It is composed of four hierarchical layers: sensorimotor, behavioral, sequencing, and planing. The lowest one, called sensorimotor layer, is based on control theory and on signal processing. It is responsible for the elementary movements of the robot and processes data acquired by the sensors. The second is the behavioral layer, composed of a set of behaviors that on one hand control the robot with respect to environmental characteristics, and on the other hand extract measures of the world in order to feed the robot internal world representation: the topological map. The sequencing layer implements tasks which are described as sequences of behaviors. Its kernel is formed of a state automaton which activates elementary behaviors, based on the interpretation of both the status of the various behaviors and the parameters transmitted by the planing layer. This latter activates and parametrizes sequencing tasks according to specifications given by a human operator, the information of the topological map and the feedback of sequencing tasks.

The architecture is implemented in the form of a development environment MANO, that encompasses a mobile robot Nomad200 (Nomadics, 1992) moving in a room environment, a set of different sensors, dedicated vision hardware, a collection of sensory-based behaviors as well as a versatile control unit. The successful implementation of several tasks in real environment speaks for the validity of this architecture (Tièche *et al.*, 1995)

3. BEHAVIORS

The behavioral layer comprises various behaviors. Some of them are directly related to the self-

positioning sites and others to the displacements between sites.

Two kind of behaviors are related to the sites: the *self-positioning* behaviors which move the robot into sites and the *localization* behavior which identifies the sites. Among the *self-positioning* behaviors, the *homing on corner* behavior, (Facchinetti and Hügli, 1994) controls the robot to a fixed pose defined with respect to particular configurations of the environment: salient corners and reflex corners. The pose, where the robot is moved, is defined on the bisectrice of the corner, at a fixed distance from it. This behavior receives range profiles from the structured light vision system Sensus500 and moves the robot such as to minimize the errors between a reference corner and the observed corner. Another vision-based *self-positioning* behavior is the *homing on target* behavior which positions the robot with respect to a pair of visual landmarks.

The behavior which distinguishes the different *homing* sites is called *localization* behavior (Tièche *et al.*, 1996). It uses a gray scale video camera pointing to the ceiling and identifies a site by comparing snapshots taken when the robot stands in a *self-positioning* site with a set of reference images stored in a database. It returns the identification of the unknown place. The combination of both a *homing* behavior and the *localization* behavior allows to define the distinctive places very accurately and in a non-ambiguous way.

The behaviors which are related to the robot displacements between sites are called the *move to* behaviors. One of these behaviors controls the robot to follow a wall detected by means of the structured light vision system Sensus500. Another one is activated when a reflective landmark is seen. It moves the robot towards the landmark and stops it at a fixed distance to the site.

4. TOPOLOGICAL MAP

Topological maps represent the environment by neighborhood relationships between distinctive places. Formally, the topological map consists of a graph $G = (V, E)$, where $V = \{v_1, \dots, v_N\}$ is the set of N nodes, and $E = \{e_{ij}\} = \{(v_i, v_j)\}$ the set of M edges. It may be considered in two ways. From the topological point of view, it is centered on a symbolic representation of the environment. From the robot resources point of view, the map is based on the interactions of robot sensors and actuators performed by the behaviors. In the frame of this work, each node corresponds to a self-positioning site and an edge to the displacement of the robot between two such sites. The behavior associated to the nodes are the *homing on corner* and the *localization* behaviors, while a *move to* behavior goes with edges.

The choice of corners as environmental characteristics for *self-positioning* behaviors is justified by the fact that the corners are easily detected, represented in a large number in man-made environments and that they appear in stable parts of the environment such as tables, walls, doors, etc. This gives the map a high accuracy and a good stability.

5. ADDITION OF GEOMETRICAL INFORMATION

This section considers the extension of the topological map by adding geometrical information. The idea is to record the odometer path while the robot moves, between sites, along the edges of the topological graph. The result is a serie of odometric paths which must be integrated to form a consistent global map.

More precisely, the topological map is built by moving the robot, manually or with adequate behaviors from corner to corners. This building process provides a sequence of visited nodes that is stored in a list: $\Sigma = \{v_i\}_{1 \leq i \leq M+1}$. The robot pose $p = (x, y, \varphi)^t$ is a 3-dimensional value that defines the position and the turret orientation of the robot, in a single frame of reference.

The odometric paths provide geometrical relations between poses the robot takes in *self-positioning* sites. A path w_{AB} between two sites $A(x_A, y_A, \varphi_A)^t$ and $B(x_B, y_B, \varphi_B)^t$, is represented by a 3-dimensional vector $w_{AB} = (d_{AB}, \alpha_{AB}, \beta_{AB})^t$.

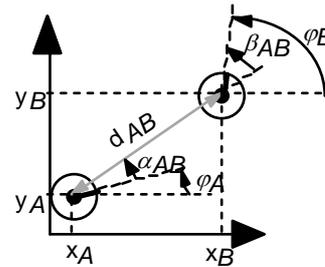


Fig 1: The path w_{AB} between two robot poses A and B is defined by the 3-dimensional vector $(d_{AB}, \alpha_{AB}, \beta_{AB})^t$

Assuming the robot is in pose A , α_{AB} is the rotation angle which brings the turret to point towards the pose B , d_{AB} is the distance between the two poses A and B , and β_{AB} is the rotation angle that aligns the turret to the pose B . A compounding operation is defined to express a pose p_B , in term of a pose p_A and a path w_{AB} linking p_A and p_B . This compounding operation is denoted as: $p_B = p_A \oplus r_{AB}$.

$$\begin{aligned} \begin{pmatrix} x_B \\ y_B \\ \varphi_B \end{pmatrix} &= \begin{pmatrix} x_A \\ y_A \\ \varphi_A \end{pmatrix} \oplus \begin{pmatrix} d_{AB} \\ \alpha_{AB} \\ \beta_{AB} \end{pmatrix} \\ &= \begin{pmatrix} x_A + d_{AB} \cos(\varphi_A + \alpha_{AB}) \\ y_A + d_{AB} \sin(\varphi_A + \alpha_{AB}) \\ \varphi_A + \alpha_{AB} + \beta_{AB} \end{pmatrix} \end{aligned} \quad (1)$$

The compound operation is associative on the right $p = ((p_o \oplus w_1) \oplus w_2) \dots \oplus w_k$, and a sequence of compounding operations is denoted as:

$$p = p_o \bigoplus_{i=1}^k w_i \quad (2)$$

In the same way, the inverse compounding operation $p_B^* p_A = w_{AB}$ expresses the path between two sites in terms of the poses of both sites.

$$\begin{aligned} \begin{pmatrix} d_{AB} \\ \alpha_{AB} \\ \beta_{AB} \end{pmatrix} &= \begin{pmatrix} x_B \\ y_B \\ \varphi_B \end{pmatrix} * \begin{pmatrix} x_A \\ y_A \\ \varphi_A \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \\ \arctan\left(\frac{y_B - y_A}{x_B - x_A}\right) - \varphi_A \\ \varphi_B - \arctan\left(\frac{y_B - y_A}{x_B - x_A}\right) \end{pmatrix} \end{aligned} \quad (3)$$

The inverse path can also be defined: $* w_{AB} = w_{BA}$. It implies that if a path is known, the inverse path can be computed. These compounding operators are close to those used by Lu and Milios (1997), but differ because the paths are not defined the same way.

6. CONSISTENT GEOMETRICAL MAP CONSTRUCTION

Given the topological map G and the associated information: the M measured geometrical paths w_{ij}^m , and the sequence of explored nodes Σ . The geometrical map building problem is to determine $N-1$ robot poses $\hat{p}_i = (\hat{x}_i, \hat{y}_i, \hat{\varphi}_i)^t$ in a single coordinate system. One pose is given *a priori* and defines the origin of the system. Arbitrarily, the pose of the first explored node is chosen: $p_{\Sigma(1)} = (0, 0, 0)^t$.

Five methods to solve this problem are proposed hereafter.

6.1 M1: Path integration along the exploration sequence.

This method takes the nodes from the exploration list one by one and finds their poses by simple integration of successive paths.

Formally, the pose $\hat{p}_{\Sigma(l)}$ of the node $\Sigma(l)$ can be expressed by compounding the origin pose with the sequence of paths joining $\Sigma(l)$:

$$\hat{p}_{\Sigma(l)} = p_{\Sigma(1)} \bigoplus_{k=2}^l w_{\Sigma(k-1)\Sigma(k)}^m \quad (4)$$

As soon as circuits appear in the graph, nodes are visited more than one time, hence their poses are computed several times. In order to assign a single pose to each node, this method keeps only the first computed pose and discards the remaining.

The complexity of M1 is $O(M)$, where M is the number of edges.

6.2 M2: Path integration without circuits along the exploration sequence

This method also takes the nodes from the exploration list one by one. The pose is found by integration of successive paths, but when a circuit is closed on the explored sequence, the integration is interrupted and restarted from the first node belonging to the circuit. In this case, one pose is assigned to each node.

The complexity of M2 is $O(M)$

6.3 M3: Path integration along the minimum distance tree

This method determines the pose of a node by compounding the original pose with a sequence of paths. In the graph, several sequences possibly link the origin to the current node. The chosen sequence is the one which has the minimum "distance" cost, defined as the sum of the distance d of each path along the sequence. This method finds the minimum spanning tree for a given root.

The complexity of M3 is $O(MN)$.

6.4 M4: Path integration along the minimum orientation tree

This method determines the pose of a node by compounding the original pose with a sequence of paths. In the graph, several sequences possibly link the origin to the current node. The chosen sequence has the minimum "angular" cost, defined as the sum of the angular variation $|\alpha| + |\beta|$ of each path along the sequence. This method finds the minimum spanning tree for a given root.

The complexity of M4 is $O(MN)$.

6.5 M5: Least square minimization

The least square method minimizes the error between the measured paths w_{ij}^m and the estimated paths \hat{w}_{ij} . The function to minimize is:

$f(\hat{w}) = (w^m - \hat{w})^t P (w^m - \hat{w})$ where P is a matrix of weights.

The estimated relations can be expressed as a non linear function of the estimated poses: $\hat{w}_{ij} = \hat{p}_j @ \hat{p}_i$. Hence the function to minimize depends on the estimated robot poses $f(\hat{p})$. It is minimum or maximum if its gradient is equal to zero.

$$\nabla f(\hat{p}) = \frac{\partial f(\hat{p})}{\partial \hat{p}} = 0 \quad (5)$$

This provides a system of $3N$ non linear equations with $3N$ unknown variables. It is solved by means of the Newton-Raphson iterative method.

The complexity of M5 is $O(N^3)$.

7. EXPERIMENTAL RESULTS

This section presents the geometrical map reconstruction for a real environment explored by a Nomad200 mobile robot. The results of the five methods are compared.

7.1 Exact map

The real environment is composed of 28 homing sites (11 reflex corners, 17 salient corners) distributed over a 10x12m surface (figure 2a). In order to compare the reconstructed maps of robot poses $\hat{p}_i = (\hat{x}_i, \hat{y}_i, \hat{\varphi}_i)^t$, an exact map of the robot poses $p_i^e = (x_i^e, y_i^e, \varphi_i^e)^t$ is measured. It is constructed in two steps. First, the corners are mapped by means of a precise measure. Then, the robot pose with respect to a corner is established, by averaging several measurements. Finally, these values are added to the precise map of corners, in order to obtain the exact map of robot poses.

7.2 Comparison of exact and estimated map

After a rigid alignment transformation, the exact and the estimated maps are compared. The difference between both maps is expressed as the root mean square of the distance Δd and the difference of orientation $\Delta \varphi$, between corresponding site poses.

$$\Delta d = \sqrt{\frac{1}{N} \sum (x_i^e - \hat{x}_i)^2 + (y_i^e - \hat{y}_i)^2} \quad (6)$$

$$\Delta \varphi = \sqrt{\frac{1}{N} \sum (\varphi_i^e - \hat{\varphi}_i)^2} \quad (7)$$

7.3 List of explored paths

The paths have been measured by odometers while the robot was exploring its environment. Seventy-two paths between the twenty-eight *self-positioning* sites were measured. The figure 2b shows the compounding of the starting pose with the 72 paths, along the sequence of exploration. Note that if a node is visited more than one time, it is represented by several site poses.

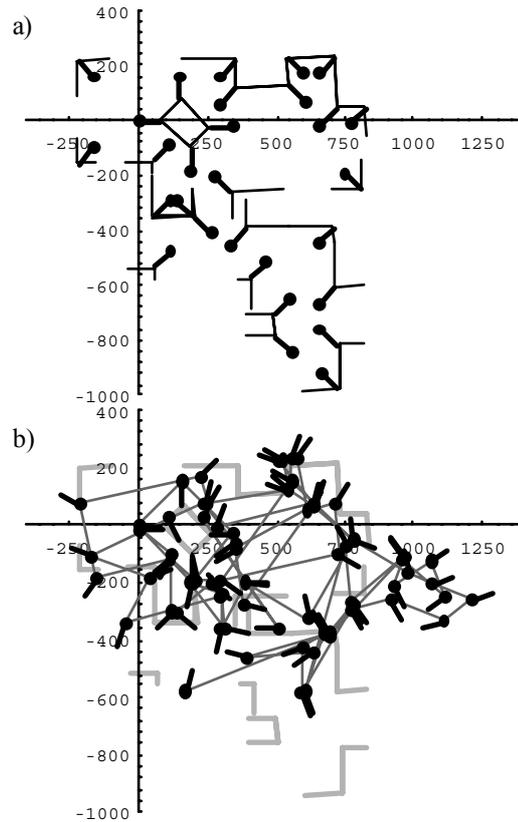


Fig. 2. a) Exact robot poses map b) Integration of all measured paths

7.4 Estimated maps

Figure 3 compares the reconstructed geometrical maps with the exact maps for the five methods M1...M5. The edges correspond to the paths needed to build the estimated maps. Note that with methods M2, M3, M4 many edges are not taken into account.

Obviously, the map provided by M1 is bad, those provided by M2, M3 and M4 show acceptable results, while M5 is excellent. The visual results are confirmed by comparing the reconstruction errors reported in table 1, expressed by the root mean square value of distance and angular differences between site poses of exact and reconstructed map.

Table 1 Difference between exact and estimated map

	M1	M2	M3	M4	M5
Δd [cm]	153.0	47.9	40.4	28.3	10.7
$\Delta \varphi$ [$^\circ$]	28.4	11.1	5.3	5.7	2.0

For every method except M5, the errors are accumulated along the paths. Hence the poses become less accurate as soon as they are far from the origin. Furthermore, a variation in the measures can modify the map significantly. Since M5 is stable and very accurate, it will be preferred even if the processing time is longer.

8. CONCLUSIONS

This paper shows how to extend the knowledge of a topological map of *self-positioning* sites by construction of a consistent associated geometrical map. Five methods have been proposed to determine the robot poses in a single frame of reference, using the topological map knowledge and odometric measurements along the paths linking robot poses. Four methods integrate paths according to different strategies and one uses a global minimization. These geometrical map building methods were implemented in a development environment involving a mobile robot Nomad200, and were tested on a map reconstruction problem with 28 *self-positioning* sites. The five methods were evaluated numerically by comparison of reconstruction errors and graphically by comparing the maps they deliver

with an exact geometrical map. The least square method shows best accuracy and gives excellent results even for a large environment.

REFERENCES

- Crowley J. (1989). World Modeling and Position Estimation for a Mobile Robot using Ultrasonic Ranging. *Proc. Int. Conf. on Robotics and Automation*, Scottsdale, AZ, May, 674-680.
- Elfes A. (1989). Using Occupancy Grids for Mobile Robot Perception and Navigation. *IEEE Computer Magazine, Special Issue On Autonomous Intelligent Machines*, June, 46-57.
- Facchinetti C. and H. Hügli (1994). Using and Learning Vision-Based Self-Positioning for Autonomous Robot Navigation. *Proc. on the Third Int. Conf. on Automation, Robotics and Computer Vision*, Singapore.
- Hügli H., J.-P. Müller, Y. Gat, M. Rodriguez, C. Facchinetti and F. Tièche (1994). Architecture of an autonomous system: application to mobile robot navigation. In: *Proc. of the Second Symposium on Artificial Intelligence and Robotics*, EPFL-Ecublens, Switzerland, September, 97-110.
- Kortenkamp D. and T. Weymouth (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. *Proc. of the Twelfth National Conf. on AI (AAAI-94)*.

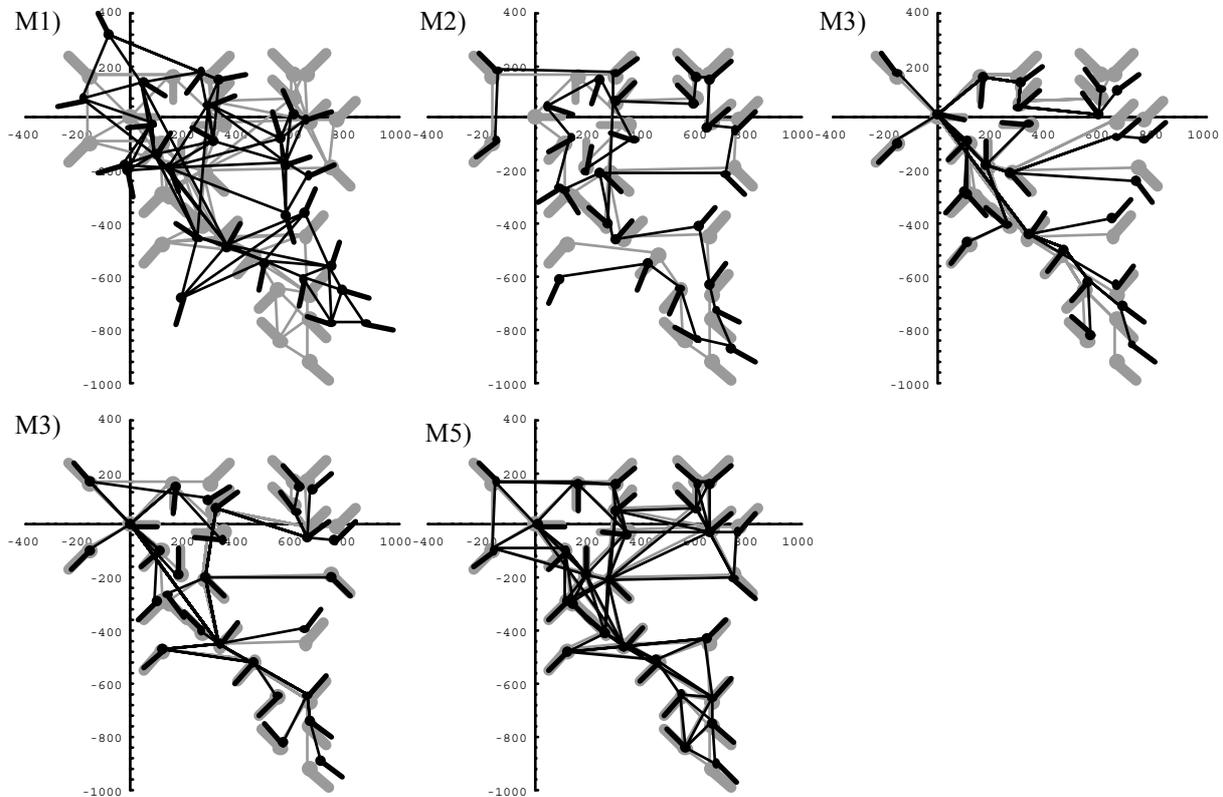


Fig. 3. Comparison of reconstructed (black) and exact (gray) maps for each method. Shown edges are the ones used for reconstruction.

- Kuipers B.J. and Y.-T. Byun (1991). A robot exploration and mapping strategy based on semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, **8**, 47-63.
- Kurz A. (1993). Building maps on learned classification of ultrasonic range data. *Proc of the 1st IFAC Int. Workshop on Intelligent Autonomous Systems*, 193-198.
- Lu F. and E. Miliotis (1997). Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robot 4*, 333-349.
- Mataric M.J. (1990). Environment learning using a distributed representation. *Proc. Int. Conf. on Robotics and Automation*, Cincinnati.
- Nomadics (1992). *Nomad 200 Users's Guide*. Nomadic Technologies, Mountain View, CA.
- Thurn S. and A. Bücken (1996). Integrating grid-based and topological maps for mobile robot navigation. *Proc. of the Thirteenth Conf. on Artificial Intelligence*, AAAI, MIT Press, Menlo Park.
- Tièche F., C. Facchinetti and H. Hügli (1995). A Behavior-Based Architecture to Control an Autonomous Mobile Robot. *Proc. of the Fourth Int. Conf. on Intelligent Autonomous Systems (IAS-4)*, Karlsruhe, March.
- Tièche F., N. S. Ghai and H. Hügli (1996). Self-positioning and localization of a mobile robot using vision-based behaviors. *Proc. of 3rd France-Japan Congress & 1st Europe-Asian Congress on Mechatronics*, **2**, Besançon, France, 872-877