Disturbance and Failure Classification in Walking Robots

Gustavo Schleyer, R. Andrew Russell
Intelligent Robotics Research Center,
Clayton Campus, Monash University, VIC 3800, AUSTRALIA
gustavo.schleyer@monash.edu, andy.russell@monash.edu

Abstract

This paper presents a technique which is being developed for the identification and subsequent classification of abnormal conditions undergone by robots. Initially, our experimental robot collects data under normal conditions. These sensor readings constitute the input of the Global Fuzzy C-means Clustering Algorithm which classifies the gathered data into different clusters. Then, the robot is subjected to a number of disturbances, while the degree of success in the accomplishment of the robot task is evaluated. The number of times that the robot is subject to the conditions represented by each cluster and is able to make progress or not is accumulated by a pair of counters. As a result, the creation of new clusters reveals the existence of abnormal situations and the counter values indicate the potential impact that the conditions represented by a cluster could have on the robot's mission. Further iterations of the algorithm allow the generation or absorption of clusters, changing the robot's notion of normal and abnormal. Hence the counter values are mapped to the new clustering configuration maintaining the information about the experience gained by the robot when facing similar conditions to those already experienced. The experimental results have shown the proposed method's capability to cluster an array of sensorial information into different types of situations experienced by a robot. This classification would facilitate the search for ways to overcome abnormal situations that could have a negative impact on a robot's performance.

1 Introduction

The identification and classification of abnormal conditions developed in this project is fundamental in the autonomous detection of a robot's failures and adaptability to unforeseen features of the robot's environment. By using this technique, it is expected to extend a robot's resilience to damage, increase the robots' lifespan and improve their autonomy in missions where human intervention is difficult or impossible, such as in extra-terrestrial exploration or other hostile environments.

A few related works can be found in the literature. For instance, the paper in [Murphy et al., 1996a] presents a method for classifying sensing failures in autonomous mobile robots. Here, the error classification is performed by a variation of the Generate and Test method [Lindsay et al., 1980]. This technique basically generates all of the possible causes of an error based on the symptoms, orders a list of associated tests and executes tests to confirm any of these causes, and terminates classification when all tests have been performed or an environmental change has been confirmed. Subsequent papers of the authors [Murphy et al., 1996b] and [Murphy et al., 1999] also describe works following this line. However, the presented methods seem to fail when they face different kinds of unexpected situations. In another paper [Jakimovski et al., 2008], the authors employ fuzzy logic with weight factors for detecting robot anomalies. The method described in this paper performs an error classification based on a fixed number of fuzzy rules not always suitable for the endless possible conditions that a robot could face during its operation. Therefore, a technique that considers the previous experience of the interaction of a robot with its environment is necessary.

This paper describes a method for identifying abnormal situations which can be generated by a robot failure or a robot interaction with previously unknown features of its environment. As a result, robots will still be able to make progress in their mission after some of their sensors or actuators experience a malfunction, or after changes in the features of their environment. For instance, the light conditions or the terrain's slope could change while a walking robot is trying to complete its task. This work is part of a larger project concerned about how robots can autonomously manage these kinds of conditions. A publication explaining a different aspect of this project can be found in [Schleyer et al., 2010].

This paper is organised as follows. A disturbance and failure classification method is presented in Section 2. Then, the experimental robot, where this technique has been implemented, is introduced in Section 3. Next, the obtained experimental results are shown in Section 4. Finally, conclusions and future related work are discussed in Section 5.

2 Failure Identification

2.1 Abnormal Situations: Malfunctions, Disturbances and Failures

From a robot's perspective an abnormal situation could be perceived as one or more sensor readings which are out of an expected range. It is also feasible that normal sensor values whose combined information is unexpected or generates contradictions could evidence an abnormal situation as well. Both cases are generated under two general scenarios. In the first one, there could be a malfunction in the robot's hardware. For instance, a robot could develop defective sensors, communication problems among its different control modules, faulty controllers or damaged parts of its body. On the other hand, unexpected sensor readings could be generated by certain features of a robot's environment that have not been previously experienced by the robot. Therefore, an abnormal situation can be produced intrinsically or extrinsically, depending on the previous robot experience and does not always evidence a malfunction. However, it is possible that both phenomena are detected by the same inconsistencies or/and unforeseen robot sensor readings, which complicates their classification.

In addition, there is no direct correlation between how a robot's performance is affected and the kind of abnormal situation that it is undergoing. Both, abnormal situations caused by a malfunction or the robot's environment have a large range of impacts on a robot's performance. As a result, during the classification process it is not possible to associate a negative effect in a robot's performance with the kind of situation that it is facing.

Finally, it is important to consider the duration of the negative effects that abnormal situations could have on a robot. Thus, long-term effects should be considered more seriously than short term impacts, which often disappear without taking any action. In this paper, we will refer to the negative short-term effects in the robot's performance as disturbances, whereas negative long term effects will be considered as failures.

2.2 Abnormal Situation Detection

Data Normalisation

The method proposed in this paper utilises all the sensorial information collected by a robot. The sensor readings are normalised between 0 and 100 and are grouped in the vector R, as it is described by expression (1).

$$R = [r_1, r_2, \dots, r_d] \tag{1}$$

Where

$r_i = 100\frac{|e_i - a_i|}{s_i}$, if expected sensor reading i is available.

$r_i = 100\frac{|a_i|}{s_i}$, otherwise.

e_i: $expected\ sensor\ reading\ i.$
a_i: $actual\ sensor\ reading\ i.$
si: $span\ of\ sensor\ reading\ i.$
$i = 1,2, \dots, d\ (number\ of\ sensor\ readings).$

The vector R, called a status report, is created each time that the robot reports its status. This vector comprises all of the sensor readings available in the target robot, at a given moment. After κ status reports, the status report matrix (S_R) of κ rows and d columns is created. This matrix will be the input of the clustering algorithm presented next. Therefore, the selected κ value determines the number of rows that S_R will have. This directly affects the processing time of the clustering algorithm. In other words, the greater the value of κ, the longer the processing time of the clustering algorithm. On the other hand, the κ value establishes how often this algorithm is executed. As a result, the selection of smaller κ values is traduced firstly as a faster execution of the clustering algorithm and finally as an increment in the number of times that this is executed during the period that the target robot is performing its task. In this paper $\kappa = 5$ has been selected. This value has provided a good balance between processing speed and the refreshment rate of the cluster centroids. However, different κ values could be adopted depending on the type of application, the dimension of the clustered data and the required and available processing speed.

Each time that κ status reports have been submitted by the target robot, the resulting S_R matrix is utilised as the input of the Fast Global Fuzzy C-Means Clustering Algorithm (FGFCM) proposed by [Wang et al., 2006]. This paper is briefly summarised below. For a full description please read the reference.

Fast Global Fuzzy C-Means Clustering Algorithm

The main disadvantage of the Fuzzy C-Means Clustering Algorithm (FCM) is that it always converges to a local minimum, and whether this corresponds or not to the general solution of a problem depends on the initial selection of all the cluster centroids. However, the information necessary for the correct selection of these values usually is not available. FGFCM attempts to solve this problem by finding the optimal initial centroids using an incremental approach. It starts with one cluster at a time, and incorporates a new cluster centre at each stage. Once all of the initial cluster centroids have been calculated, these are used as initial conditions on FCM, improving the obtained results. The FGFCM algorithm can be described as follows:

Step 1:

Execute the FCM to find the optimal clustering centroid v_1 of the fuzzy partition ($C = 1$) and let J_m be the corresponding value of the objective function calculated by means of expression (2). This expression represents a sum of squared error and is one of the most popular and well studied data clustering criteria.

$$J_m = \sum_{i=1}^{N}\sum_{c=1}^{C} u_{ic}^m \|x_i - v_c\|^2 \tag{2}$$

Where

$X = [x_1, x_2, \dots, x_N] = [R_1, R_2, \dots, R_N]$ is a numerical array containing N d-dimensional data points (each element of X is a status report R).

$V = [v_1, v_2, \dots, v_C]$ is a numerical array containing C d-dimensional cluster centroids.

m = weighting exponent $1 \le m \le \infty$ ($m = 2$ in this paper).

u_{ic} is the degree of membership of the data point x_i in the cluster c. This value is restricted using the following criterion and can be calculated by means of expression (3):

$$\forall i, \sum_{c=1}^{C} u_{ic} = 1;\ \forall i, c, u_{ic} \in [0,1]; \forall c, \sum_{i=1}^{N} u_{ic} > 0$$

$$u_{ic} = \left(\sum_{k=1}^{C}\left(\frac{\|x_i - v_c\|^2}{\|x_i - v_k\|^2}\right)^{\frac{2}{m-1}}\right)^{-1} \tag{3}$$

Where $1 \le i \le N, 1 \le c \le C$.

The weighting exponent, m in expression (3), controls the weight placed on each squared error. When $m = 1$, the partitions that minimise J_m are hard. On the other hand, as $m \to \infty$, each data point has the same probability of belonging to each cluster. There is no way to determine an optimal value of m. However, $m = 2$ is a value widely used because it eases the calculation of J_m while delivers a fuzzy classification. A more detailed description of FCM can be found on [Bezdek et al., 1984].

Step 2:
Calculate the value of the objection function for each data point used as initial state, by means of expression (4). This data clustering criterion is a reformulation of expression (2). The derivation of this is beyond the scope of this paper, but can be found in [Hathaway et al., 1995].

$$J_m = \sum_{i=1}^{N}\left(\sum_{c=1}^{C}\|x_i - v_c\|^{2(1-m)}\right)^{(1-m)} \qquad (4)$$

Where $\|x_i - v_c\| \neq 0$.

Step 3:
Find the minimum value of the objection function J_m and the corresponding initial state v_c from Step 2. Let $V = [v_1, \dots, v_c]$ be the c initial clustering centres of the new fuzzy partition considering an extra cluster.

Step 4:
Perform the FCM algorithm with c clusters from the initial state *V* and obtain the final clustering centres.

Step 5:
If $c = C$, stop; otherwise set $c = c + 1$ and go to Step 2.

Clustering Sensor Readings

In order to be able to apply the FGFCM algorithm it is necessary to determine C, the total number of clusters. Ideally, each cluster should represent a different situation experienced by a robot. As the robot gains more experience, the number of clusters is incremented. This value can be found while FGFCM is executed by considering two values. The first one is d_{c_min}, the minimum distance between all of the possible pairs of cluster's centroids. If the clustering has C clusters, then it is necessary to calculate $\binom{C}{2}$ distances between centroids. Then, d_{c_min} is the minimum of these distances and it is a measure of the separation of the clusters. Hence, the larger the value of d_{c_min} the more different the data contained in each cluster are. Another important value to consider is ϵ_{max}, which is the maximum ratio between the mean distance between all of the points of a cluster and its centroid; and the number of elements of the cluster. Then, it is necessary to calculate an ϵ value for each cluster. This variable is a measure of the dispersion of the data in a cluster. Therefore, ϵ_{max} determines the cluster with the larger dispersion in the classification process. By using d_{c_min} and ϵ_{max}, it is possible to determine E_c, the error of the classification. This is expressed in equation (5).

$$E_c = 100\left(\frac{\epsilon_{max}}{\epsilon_1} - \frac{d_{c\_min}}{d_1}\right) \qquad (5)$$

Where
ϵ_1 is the ϵ value calculated after grouping all of the data points into only one cluster $(C = 1)$.
d_1 is the maximum distance between the centroid and all of the data points calculated after grouping all of the data into only one cluster $(C = 1)$.
Expression (5) uses values ϵ_1 and d_1 which describe all of the data points when they are clustered into one group. Therefore, classifications using more clusters are compared with the original state of the data. The method for establishing C, the total number of clusters, consists of calculating E_c for an increasing number of clusters, starting with only one cluster. The E_c value decreases until the optimal number of clusters (the one that minimises E_c) is reached. However, because there is no pair of centroids when the number of clusters is one, it is not possible to calculate d_c. Therefore, only ϵ_{max} is employed for comparing the clustering process using one cluster and two clusters. The criterion used here is represented by expression (6).

$$\begin{aligned} &C = 1, \quad if\ |\epsilon_{max1} - \epsilon_{max2}| < \delta \\ &C = 2, \qquad\quad otherwise \end{aligned} \qquad (6)$$

Where ϵ_{max1} and ϵ_{max2} are the ϵ_{max} values associated with the clustering process using 1 and 2 clusters, respectively. δ is a parameter of the method which controls how different the data must be in order to be classified by more than one cluster. $\delta = 0.1$ was used in this paper.

The result of applying the clustering process to a data set of 677 two-dimensional points is shown in Fig. 1. Each data point corresponds to the x-y coordinates of an accelerometer mounted on the top of our experimental robot. When the robot is parallel to the ground, alpha (roll angle) and beta (pitch angle) in Fig. 1. are 90°. Then, alpha increases if the robot rotates clockwise and decreases if the robot rotates counter-clockwise about the axis running from the front to the back of the robot. On the other hand, the same relation is valid for beta. In this case, the rotation is about the axis running from the left side to the right side of the robot. Fig. 1 shows the accelerometer readings collected while the robot's tilt was changed. Firstly, the robot was parallel to the ground. Then, the robot's roll angle was quickly incremented. Next, the roll angle was suddenly decremented. Finally, the same procedure was applied to the pitch angle.

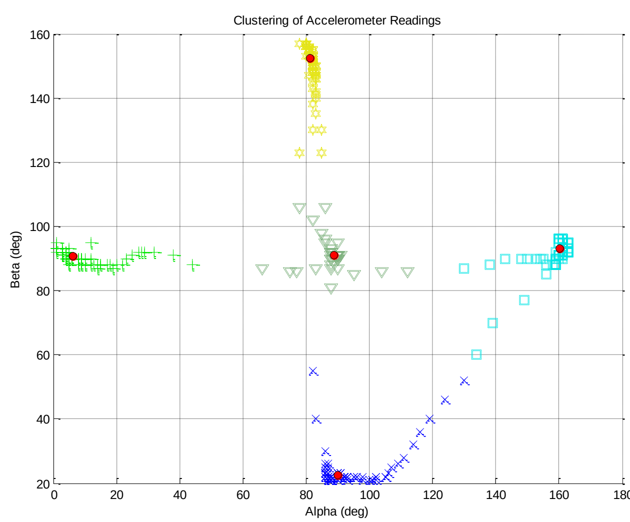

Fig. 1. Clustering Results.

The number of clusters calculated by the method previously discussed for this data set is five. The final classification performed by FGFMC is represented by using different shapes and colours for each cluster. The centroid of each cluster are shown by using red circles.

After the FGFCM algorithm is applied, the optimal initial conditions for the FCM algorithm are known. Thus, FCM is executed and, as a result, the sensorial information supplied by the robot during the first κ status reports is classified into C clusters.

As was discussed in subsection 2.2, the execution of the FGFCM after each status report could demand an excessive amount of processing time. Hence, the algorithm is only employed every time that κ status reports have been submitted by the target robot. However, each time that a new status report is delivered, a membership error, m_error, is calculated by means of expression (7).

$$m\_error_c = \|R - v_c\|,\ c = 1,2,\dots,C \tag{7}$$

This data point is associated with the cluster c where m_error_c is minimum, provided that this error is less than the classification error. Otherwise, a new cluster is created with its centroid and elements being the data point x itself. This allows the identification of an abnormal situation each time that a status report is received. If a new cluster is created in this way, the classification error is updated so its value is equal to the minimum membership error of the x data point previously calculated utilising the relation (7).

2.3 Disturbance and Failure Detection

So far, abnormal situations are detected by the proposed method, but failures are not yet identified. In order to do so, a robot must be able to assess its own performance. Therefore, a quantitative performance analysis should indicate if a robot is making progress in its task or not. Let G represent a robot status once its goal is accomplished and S the current robot status. So, once a robot has accomplished its mission $S = G$. If this robot has not yet completed its task, then the difference d_p between S and G could be determined by utilising expression (8).

$$d_p = \|G - S\| \tag{8}$$

A robot where the proposed method is implemented, must supply the d_p difference with each status report that follows the first κ reports. By keeping the d_p value of the previous status report (d_{p-1}) it is possible to evaluate if the robot is making progress or not. This can be represented by the P variable described in expression (9).

$$P = d_{p-1} - d_p \tag{9}$$

Where $P > 0$ indicates than the target robot is making progress in the completion of its task. While $P \le 0$ shows the opposite. Therefore, a disturbance could be identified simply by finding when P does not have a positive value. However, in order to determine if this disturbance is in fact a failure, a long-term analysis is required. Hence, the proposed method utilises two counters associated with each cluster. The first one, g_c, counts the number of times that a status report is classified into the cluster $\boldsymbol{c}$ and the robot makes progress in its task $(P > 0)$. On the other hand, a second counter, b_c, counts how many times a robot does not advance in the completion of its mission after its status report is classified into the cluster c $(P \le 0)$. As a result, as more status reports are submitted, there is more information about how potentially detrimental or beneficial is the current status of the robot for successful completion of its mission. Here, a probabilistic approach seems to be more suitable in order to decide if a cluster represents a failure or not. Thus, by considering the expression (10) it is possible to calculate the probability, p_c^f, that the cluster c represents a failure.

$$p_c^f = \frac{b_c}{b_c + g_c} \tag{10}$$

2.4 Experience Storage

After another κ status reports have been submitted by a robot, it is time to perform a new FGFCM cluster classification. Because at this point the robot has already gained some experience, this should be included in the clustering process. However, if all of the status reports were the new FGFCM input, the required processing time and memory would be excessive. The technique presented in this paper tackles this problem by considering only the cluster's centroids as input data. Here, the cluster's centroids created during the previous and between two consecutive FGFCM executions are included.

Once the FGFCM algorithm is executed again, almost all of the proposed method's variables are recalculated as was explained during this section. The only exceptions are g_c and b_c, which preserve the experience acquired by the robot so far. However, after each new clustering process, these variables must be mapped to the new cluster distribution. One feature of the proposed method is that from the second execution of FGFCM, each time that this algorithm is employed, the number of clusters decreases or remains constant. This simplifies the mapping of g_c and b_c to one or both of the following operations.

a) If the cluster c_x is labelled as cluster c_y in the new cluster distribution, then $g_y = g_x$ and $b_y = b_x$.
b) If the cluster c_x is merged with the cluster c_y and labelled as cluster c_z in the new cluster distribution, then $g_z = g_x + g_y$ and $b_z = b_x + b_y$.

2.5 Failure Classification

After a disturbance or a failure has been detected it is possible to determine which sensor readings are evidencing it. In practice, under normal circumstances, most of the sensor readings show some degree of difference with respect to their expected values. However, when a robot is undergoing a disturbance or a failure one or more sensor readings show an increment of this difference. This prevents a data point, x^f, that represents this situation, from being associated with a cluster m, where $g_m > b_m$. Let assume that the cluster m, with centroid v^m, is the closest cluster to x^f that does not represent a failure. Then, the percentage of contribution of the ith x^f's sensor reading to the total difference between x^f and v^m, E_i, can be calculated by means of expression (11).

(11)

$$E_i = 100 \frac{|x_i^f - v_i^m|}{\|x^f - v^m\|}$$

Where $i = 1,2, \dots, number\ of\ elements\ of\ x^f$.

The sensor readings that have greater E values are the ones which are evidencing a failure. These values constitute the symptoms that a robot is experiencing and provide valuable information for understanding the cause of the problem. Although these data can only reduce the spectrum of search, they facilitate the search process for the exact identification of a failure's cause. Thus, a more precise diagnosis would require some processing techniques that are beyond the scope of this paper. For instance, a set of actions taken by the robot could be planned for discarding wrong assumptions of a failure's cause. After an action that targets a particular supposition is executed, the feedback provided by the robot's sensors could be analysed in order to determine if this assumption is indeed correct or not.

3 Experimental Robot

The method discussed in the previous section has been implemented in the 18-degrees-of-freedom hexapod robot shown in Fig. 2. Each one of the robot's legs has an independent controller. These are utilised for the interpretation of the information obtained from the legs' sensors and the communication with the on-board central control. This main controller is responsible for a number of tasks. First of all, it manages the communication and collects the information provided by the leg controllers. In addition, it obtains information from sensors connected to the robot's body. Finally, it wirelessly sends all the collected information, in the form of a status report, to a PC. Here, the clustering algorithm is executed and the next coordinates of the robot's legs are calculated.

Fig. 2. Experimental Hexapod Robot.

The readings supplied by the experimental robot come from sensors located on the robot's legs and body. Each robot's leg possesses force and position sensors for each one of its three servo-motors. In addition, the robot's body has been equipped with an accelerometer and five light sensors. The information provided by all of these sensors is grouped in a status report, which structure is shown in Table 1.

Acc_x	Acc_y	Ps_{ij}	Fs_{ij}	Ls_k

Table.1. Data Frame of Experimental Robot's Status Report.

Where

Acc_x and Acc_y represent the x-axis and y-axis accelerometer readings, respectively.

Ps_{ij} represents the position sensor reading of the servo motor *j* located in the robot's leg i.

Fs_{ij} represents the force sensor reading of the servo motor *j* located in the robot's leg i.

$i = 1,2, \dots, 6$ represents each robot's leg. These have been enumerated in counterclockwise order starting from the right middle leg.

$j = 1,2,3$ represent each leg's servo motor. Servos 1, 2 and 3 work in the robot's leg as coxa, tibia and femur articulations, respectively.

Ls_k represents the light sensor reading *k*, with $k = 1,2, \dots, 5$.

Some of the information contained in the status report has been compared with its respective expected value. For instance, the coordinates of the legs sent to the robot are the expected values of the position of the legs. Therefore, these coordinates are compared with the information provided by the leg position sensors. In addition, the coordinates of the legs have been utilised for calculating the direction angles of the 3D plane formed by the tip of the legs. These angles have been compared with Acc_x and Acc_y. In the other hand, there are not expected values for the force or light sensors.

The task assigned to our experimental robot is simply following the light. The progress that the robot makes in its mission is quantitatively easy to measure by means of the light sensors. These sensors are located on the robot's body. They are separated by 90º, so the robot is able to sense the light in every direction. There are two sensors together at the front of the robot pointing straight ahead. This array allows more precision in the calculation of the light source direction. The remaining three sensors are located at each side and at the back of the robot, respectively. These are used for rougher preliminary calculations, detecting large changes in the light direction, or turning the robot towards the light direction.

4 Experimental Results

The experimental Robot has been subject to five tests in order to analyse the results obtained by the proposed method. Basically, these are expressed by means of the number of clusters, the respective clusters' centroid and the associated values of the *g* and *b* counters. First of all, each cluster's centroid represents a different robot status. These clusters are formed by grouping the data points resulting from processing the information provided by the robot as status reports. After this information is processed, it is converted into three types of data. The first type of data, symbolised by P_{LxSy} in Tables 2-5, is the percentage of difference between the expected angle of the servo y, belonging to the leg x, and the value measured by the respective position sensor. The second kind of data, represented by F_{Lx} in Tables 2-5, is the maximum percentage of force that the servos of the leg x are applying. This value is measured by the force sensors of the respective leg. Finally, Acc_x and Acc_y in Tables 2-5 symbolise the percentage of difference between the x-axis and y-axis tilt angles, measured by an accelerometer, and the respective x-axis and y-axis tilt angles of the plane formed by the tip of the robot's legs. On the other hand, as was discussed in Section 2, the g and b counters show the number of times that the robot has made progress or not, respectively, in the completion of its task. Therefore, clusters or states where $b>g$ should be avoided by the robot. In addition, the $b+g$ amount indicates the number of times that the robot has been in a particular status. In this paper, we have discarded clusters where $b+g$ is much lower than the number of submitted status reports. This is because the information provided by these clusters is not significant and also for space economy reasons.

4.1 Normal Conditions

In the first experiment, the robot starts to walk towards a light source over a horizontal surface. The robot's hardware is working normally and these conditions remain constant during the whole experiment.

The obtained results show that after 95 status reports have been submitted only one cluster has been created. This indicates that the conditions faced by the experimental robot have remained constant. In addition, the cluster's centroid shown in Table 2, indicates that there are not large errors between measured and expected values. This confirms the fact that the robot is facing normal conditions.

P_{L1S1}	P_{L1S2}	P_{L1S3}	P_{L2S1}	P_{L2S2}	P_{L2S3}	P_{L3S1}	P_{L3S2}	P_{L3S3}
0.88	2.56	2.21	3.54	1.18	1.31	1.43	2.91	2.58
P_{L4S1}	P_{L4S2}	P_{L4S3}	P_{L5S1}	P_{L5S2}	P_{L5S3}	P_{L6S1}	P_{L6S2}	P_{L6S3}
0.99	2.77	2.14	1.23	3.01	0.75	1.36	2.01	2.92
F_{L1}	F_{L2}	F_{L3}	F_{L4}	F_{L5}	F_{L6}	Acc_x	Acc_y	
0.00	0.02	0.02	0.69	0.79	0.02	2.29	0.93	

Table 2. Obtained Centroid (Normal Conditions).

Finally, the counters associated with this cluster are *g = 126* and *b = 26*. As expected, these values show that the robot is making a very good progress in the completion of its task while it is facing the current conditions.

4.2 Uphill

The conditions are slightly changed in this second experiment. Initially, the robot walks over a horizontal surface as before. But then the inclination is increased to around 30 degrees. Therefore, as is shown in Fig. 3, the experimental robot must walk uphill.

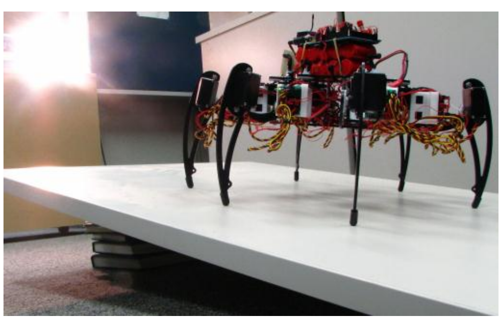

Fig. 3. Experimental Robot Walking Uphill.

The obtained results show that after 40 status reports have been submitted two clusters have been created. This indicates that two different conditions have been faced by the experimental robot. As can be inferred from the clusters' centroids shown in Table 3, these differences have been mainly detected thanks to Acc_x and Acc_y. In the first situation (the robot walking over a horizontal surface) Acc_x and Acc_y have similar values.

However, when the robot is walking uphill (cluster 2 in Table 3), Acc_y is incremented and Acc_x is decremented. This shows that the main factor responsible of the error between the expected and the real tilt is Acc_y.

#	P_{L1S1}	P_{L1S2}	P_{L1S3}	P_{L2S1}	P_{L2S2}	P_{L2S3}	P_{L3S1}	P_{L3S2}	P_{L3S3}
1	0.80	3.35	2.64	3.71	1.95	1.91	2.73	3.41	3.89
2	0.59	3.31	2.67	4.43	2.30	1.71	1.47	3.82	3.76
	P_{L4S1}	P_{L4S2}	P_{L4S3}	P_{L5S1}	P_{L5S2}	P_{L5S3}	P_{L6S1}	P_{L6S2}	P_{L6S3}
1	1.43	2.96	3.65	1.74	3.55	2.55	3.32	3.08	3.78
2	1.14	2.73	3.36	1.58	3.16	1.99	1.86	3.15	3.71
	F_{L1}	F_{L2}	F_{L3}	F_{L4}	F_{L5}	F_{L6}	Acc_x	Acc_y	
1	0.01	1.23	0.76	0.06	0.63	0.13	90.74	99.2	
2	0.00	0.43	0.79	0.04	0.30	0.07	90.85	100.4	

Table 3. Obtained Centroids (Uphill).

Finally, the counters associated with these clusters are $g_1 = 20$, $g_2 = 4$, $b_1 = 2$ and $b_2 = 3$. These values show that the robot is making a good progress in the completion of its task while it is facing either of the two analysed conditions.

4.3 Tied Leg

In this experiment, leg 4 of the robot has been tied as is shown in Fig. 4.

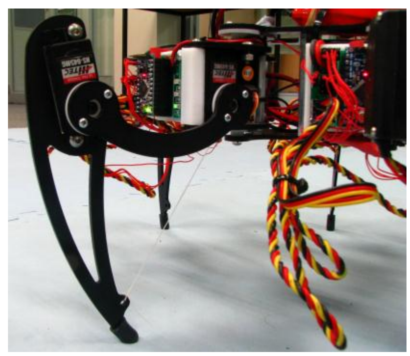

Fig. 4. Experimental Robot with Tied Leg.

The obtained results show that after 25 status reports have been submitted only one cluster has been created. This indicates that the conditions faced by the experimental robot have remained constant. In addition, the cluster's centroid shown in Table 4, indicates an increment in the force applied by legs 3 and 4. This shows the propagation effect of a disturbance.

P_{L1S1}	P_{L1S2}	P_{L1S3}	P_{L2S1}	P_{L2S2}	P_{L2S3}	P_{L3S1}	P_{L3S2}	P_{L3S3}
3.24	9.36	3.83	4.30	0.21	2.63	4.36	2.28	3.88
P_{L4S1}	P_{L4S2}	P_{L4S3}	P_{L5S1}	P_{L5S2}	P_{L5S3}	P_{L6S1}	P_{L6S2}	P_{L6S3}
0.31	3.81	4.15	1.30	3.55	2.66	1.86	4.32	3.62
F_{L1}	F_{L2}	F_{L3}	F_{L4}	F_{L5}	F_{L6}	Acc_x	Acc_y	
0.06	11.75	0.02	0.06	0.07	0.06	90.98	90.69	

Table 4. Obtained Centroid (Tied Leg).

Finally, the counters associated with this cluster are *g = 21* and *b = 5*. These values show that the robot is having occasional difficulties but it is still making progress in the completion of its task.

4.4 Deactivated Position Sensor

During this experiment all the position sensors of leg 2 have been disconnected. This intends to emulate some malfunctions in these sensors. The obtained results show that after 110 status reports have been submitted, a new cluster has been created. As expected, the cluster's centroid shown in Table 5, indicates that there is an increment in the error related to the position of the servo motors belonging to leg 2.

P_{L1S1}	P_{L1S2}	P_{L1S3}	P_{L2S1}	P_{L2S2}	P_{L2S3}	P_{L3S1}	P_{L3S2}	P_{L3S3}
1.68	4.28	3.34	17.15	17.29	2.54	4.10	4.68	1.91
P_{L4S1}	P_{L4S2}	P_{L4S3}	P_{L5S1}	P_{L5S2}	P_{L5S3}	P_{L6S1}	P_{L6S2}	P_{L6S3}
3.39	4.75	1.26	3.57	2.99	4.59	1.57	4.29	0.00
F_{L1}	F_{L2}	F_{L3}	F_{L4}	F_{L5}	F_{L6}	Acc_x	Acc_y	
0.02	0.02	0.03	0.02	0.03	0.02	91.87	90.60	

Table 5. Obtained Centroid (Deactivated Position Sensor).

Finally, the counters associated with this cluster are *g = 32* and *b = 10*. This indicates that the robot is making very good progress in the completion of its task despite the fact that some of its sensors have been disconnected.

4.5 Changeable Light Conditions

In this last experiment, the light conditions were changed as is shown in Fig. 4a and Fig. 4b. Basically, the light source was turn on and off a number of times during the experiment. This intends to produce a type of robot's environmental change that directly affects the performance evaluation of the robot.

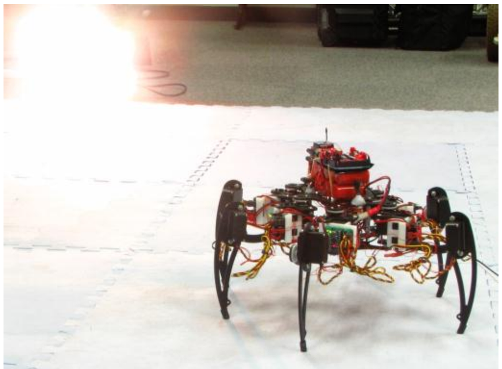

Fig. 4a. Experimental Robot in a Well Lit Environment.

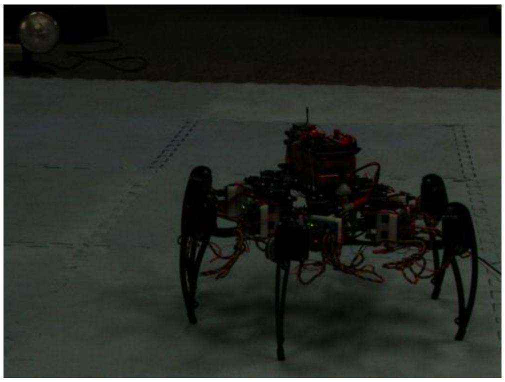

Fig. 4b. Experimental Robot in a Dark Environment.

In this case, the obtained results show that after 55 status reports have been submitted only one cluster has been created. This indicates that the changes in the light conditions faced by the experimental robot are insignificant from a performance point of view. Therefore, the cluster's centroid shown in Table 5, indicates that everything is normal.

P_{L1S1}	P_{L1S2}	P_{L1S3}	P_{L2S1}	P_{L2S2}	P_{L2S3}	P_{L3S1}	P_{L3S2}	P_{L3S3}
0.78	4.36	3.70	5.24	2.71	2.20	1.58	4.73	5.27
P_{L4S1}	P_{L4S2}	P_{L4S3}	P_{L5S1}	P_{L5S2}	P_{L5S3}	P_{L6S1}	P_{L6S2}	P_{L6S3}
1.19	3.9	4.52	1.90	3.74	3.03	3.03	2.80	4.19
F_{L1}	F_{L2}	F_{L3}	F_{L4}	F_{L5}	F_{L6}	Acc_x	Acc_y	
0.00	0.18	0.02	0.02	0.02	0.36	90.6	90.1	

Table 6. Obtained Centroid (Changeable Light Conditions).

However, the counters associated with this cluster are *g = 44* and *b = 11*. These values show that the robot is having difficulties but it is still making progress in the completion of its task. In fact, the *b* value is indicating the number of times that the light source has been disconnected during this experiment.

5 Conclusions and Future Work

This paper has presented a technique which allows the identification and subsequent classification of abnormal conditions undergone by a robot. The experimental results have shown the proposed method's capability to cluster an array of sensorial information into different types of situations experienced by a robot. Consequently, each created cluster corresponds to a particular situation. Those clusters that are not representative are easily discarded by considering the number of data points, or number of times that the robot has experienced a particular situation, belonging to this kind of clusters. In addition, the incorporation of the *g* and *b* counters has allowed further cluster classification according to the progress that a robot makes in its task. By using this technique, the experience gained by a robot, when facing the situation described by a cluster, is accumulated. As a result, it is possible to associate clusters with the percentage of times that they have represented a failure or a success. Therefore, it is feasible for a robot to know to what degree a particular situation must be avoided. Moreover, the proposed method only utilises the clusters' centroid as input data. This provides a faster classification speed and less memory requirements. Nevertheless, this technique is still able to preserve the experience gained by the robot and to use this information in the disturbance and failure classification process.

As future work, we intend to more deeply analyse the data provided by the proposed method. The idea is to provide robots with a technique for the automatic generation of assumptions related to the cause of a failure. Then, a robot should be able to test every assumption until a way for overcoming the failure is autonomously found. The development of this technique will improve the autonomy of robots in missions where human intervention is difficult or impossible, such as in extra-terrestrial exploration or other hostile environments.

References

[Bezdek et al., 1984] Bezdec, J.; Ehrlich, R. and Full, W., "FCM: The Fuzzy c-Means Clustering Algorithm", *Computers & Geosciences*, Vol. 10, N° 2-3, 1984, pp. 191-203.

[Garcia et al., 2008] Garcia, E.; Gonzalez de Santos, P. and Matia, F., "Dealing with Internal and External Perturbation on Walking Robots", *Autonomous Robots*, Vol. 24, 2008, pp. 213-227.

[Hathaway et al., 1995] Hathaway, R. and Bezdek, J., "Optimization of clustering criteria by reformulation", *IEEE Transactions on Fuzzy Systems*, Vol. 3, 1995, pp. 241–245.

[Jakimovski et al., 2008] Jakimovski, B. and Maehle, E., "Artificial Immune System Based Robot Anomaly Detection Engine for Fault Tolerant Robots", *5th International Conference on Autonomic and Trusted Computing (ATC-08)*, Oslo, Norway, pp. 177-190, 2008.

[Johnson et al., 2010] Johnson, A.; Haynes, G. and Koditschek, D., "Disturbance Detection, Identification, and Recovery by Gait Transition in Legged Robots", *International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5347-5353.

[Lindsay et al., 1980] Lindsay, R.; Buchanan. E.; Feigenbaum, E. and Lederberg, J., "Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project", New York, McGraw-Hill, 1980.

[Murphy et al., 1996a] Murphy, R. and Hershberger, D., "Classifying and Recovering from Sensing Failures in Autonomous Mobile Robots", *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI'96, Vol. 2, August 1996.

[Murphy et al., 1996b] Murphy, R., "Biological and Cognitive Foundations of Intelligent Sensor Fusion", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 1996*, Vol. 26, pp. 42-51.

[Murphy et al., 1999] Murphy, R. and Hershberger, D., "Handling Sensing Failures in Autonomous Mobile Robots", *The International Journal of Robotics Research*, 1999, Vol. 18, pp. 382-400.

[Roumeliotis et al., 1998] Roumeliotis, S.; Sukhatime, G. and Bekey, G., "Fault Detection and Identification in a Mobile Robot using Multiple-Model Estimation", International Conference on Robotics and Automation (ICRA), 1998, pp. 2223-2228.

[Schleyer et al., 2010] Schleyer, G. and Russell, A., "Adaptable Gait Generation for Autotomised Legged Robots", *Proceedings of the 2010 Australasian Conference on Robotics & Automation (ACRA)*.

[Wang et al., 2006] Wang, W.; Zhang, Y.; Li, Y. and Zhang, X., "The Global Fuzzy C-Means Clustering Algorithm," *Sixth World Congress on Intelligent Control and Automation, WCICA 2006*, Vol. 1, pp.3604-3607.

[Wang et al., 2006] Wang, W.; Zhang, Y.; Li, Y. and Zhang, X., "The Global Fuzzy C-Means Clustering Algorithm," *Sixth World Congress on Intelligent Control and Automation, WCICA 2006*, Vol. 1, pp.3604-3607.