ANNALS

OF THE

UNIVERSITY OF CRAIOVA

Series: AUTOMATION, COMPUTERS, ELECTRONICS and MECHATRONICS

Vol. 11 (38), No. 1, 2014

ISSN 1841-0626

EDITURA UNIVERSITARIA

ANNALS OF THE UNIVERSITY OF CRAIOVA

Series: AUTOMATION, COMPUTERS, ELECTRONICS AND MECHATRONICS

Vol. 11 (38), No. 1, 2014 **ISSN 1841-0626**

***Note*:** The **"Automation, Computers, Electronics and Mechatronics Series"** emerged from **"Electrical Engineering Series" (ISSN 1223-530X)** in 2004.

Address for correspondence:

Vladimir RĂSVAN
University of Craiova
Faculty of Automation, Computers and Electronics
Al.I. Cuza Street, No. 13
RO-200585, Craiova, Romania
Phone: +40-251-438198, Fax: +40-251-438198
Email: vrasvan@automation.ucv.ro

This issue has been published under the responsibility of Emil PETRE.

We exchange publications with similar institutions from country and from abroad

CONTENTS

Computing Minimal Separating Polygons by Convexifying Non-Self-Intersecting Spanning Trees

Eliana-Dina Andreica*, Mugurel Ionut Andreica**

**Computer Science Department, Politehnica University of Bucharest*
Splaiul Independentei 313, 060042, sector 6, Bucharest, Romania
(email: eliana.andreica@cs.pub.ro)
***Computer Science Department, Politehnica University of Bucharest*
Splaiul Independentei 313, 060042, sector 6, Bucharest, Romania
(email: mugurel.andreica@cs.pub.ro)

Abstract: In this paper we present a novel approach for the problem of computing a small perimeter simple polygon which separates a set of *M* red points from a set of *N* blue points in the plane (i.e. all the points of the same color are inside or on the border of the polygon and all the points of the other color are outside or on the border of the polygon). Our approach is based on generating non-self-intersecting spanning trees of the points located inside the polygon and then convexifying these spanning trees in order to obtain a minimal separating polygon (i.e. a polygon whose perimeter cannot be decreased further by convexification operations and which completely contains the original spanning tree). By generating multiple spanning trees we are able to obtain separating polygons with various perimeter lengths. We consider both the case when the separating polygon may only contain the original points as vertices, as well as the case when new points can be used as vertices of the polygon. In the second case each spanning tree is convexified to (approximately) the same polygon irrespective of the order in which the convexification operations are performed.

Keywords: Adaptive algorithms, Polygons, Trees.

1. INTRODUCTION

In this paper we consider the well-studied problem of computing a minimum perimeter simple polygon which separates a set of *M* red points from a set of *N* blue points in the plane (i.e. all the points of one color are inside the polygon or on its border and all the points of the other color are outside the polygon or on its border). We consider that the *M+N* points are in general position (i.e. no 3 points are collinear) and the distance between two points is the usual Euclidean distance.

We propose a novel approach for addressing this problem which consists of generating non-self-intersecting spanning trees and then convexifying them in order to obtain various separating polygons. Then we can adapt almost any generic optimization algorithm for this problem in order to consider the spanning tree rather than the solution polygon. This provides several advantages: it is very easy to generate spanning trees which lead to very different separating polygons, while generating the separating polygons directly would be a more cumbersome task.

The rest of this paper is structured as follows. In Section 2 we discuss related work. In Section 3 we discuss the issue of generating non-self-intersecting spanning trees of all the points of a given color. In Section 4 we present our convexification algorithm, considering two cases: (1) when the vertices of the separating polygon must consist of only the given points; (2) when we can add new points as vertices to the separating polygon. In Section 5 we show how we can include our spanning tree generation and convexification algorithm into several generic optimization methods for finding the minimum perimeter separating polygon. In Section 6 we present experimental results. In Section 7 we conclude and discuss future work.

2. RELATED WORK

The problem of computing the minimum perimeter polygon which separates a set of *M* red points from a set of *N* blue points has been well studied in the scientific literature. The problem is known to be NP-hard (Eades and Rappaport, 1993). An approximation algorithm with an *O(log(M+N))* approximation ratio was presented in (Mata and Mitchell, 1995). A polynomial-time approximation scheme (PTAS) for this problem was presented in (Arora and Chang, 2003). Polygonizations of points of a given color which exclude the points of the other color were studied in (Fulek et al., 2010).

We are not aware of any previous publications which report attempts to compute minimum separating polygons by using generic optimization algorithms. In theory this should be possible by considering candidate separating

polygons and by defining operations which transform a separating polygon into another separating polygon which is "close" to the original one. However, defining such operations is not easy. Moreover, providing guarantees that the optimal separating polygon can be obtained by a sequence of these operations is difficult. On the other hand, it is easy to define such operations and provide such guarantees for our spanning tree-based approach.

Other related problems regarding constrained minimum perimeter enclosing polygons (Mitchell and Winters, 1991) and minimum perimeter polygon bipartitions (Provencal and Lachaud, 2009) were considered in the literature.

3. GENERATING NON-SELF-INTERSECTING SPANNING TREES

Let's consider the problem of generating non-self-intersecting spanning trees of all the points of the same color. Without loss of generality, we will consider that we need to generate non-self-intersecting spanning trees of the M red points. A spanning tree is defined to be non-self-intersecting if no two non-adjacent edges intersect. Two edges are adjacent if they share a common endpoint.

First of all, it is obvious that a minimum spanning tree of the M red points is non-self-intersecting (if two edges (u,v) and (w,t) of the minimum spanning tree intersect, then we could swap their endpoints and, thus, replace them with the edges (u,w) and (v,t), in order to obtain a spanning tree with a shorter total length, which would imply that the original spanning tree was not a minimum spanning tree). This leads us to a first approach for generating non-self-intersecting spanning trees: trying to generate spanning trees which are "close" to the minimum spanning tree, such that they do not contain self-intersections with a high probability. We considered a modified Prim's algorithm (Sedgewick and Wayne, 2011) for generating random spanning trees which are "close" to the minimum spanning tree. At each step of Prim's algorithm we have a set of points which are part of the spanning tree while the others are still outside of the spanning tree. For each point x outside of the spanning tree we have a potential parent y, which, in the standard version of Prim's algorithm, is the closest point to x which belongs to the spanning tree. We introduce the following two changes. At each step of the algorithm we first find the minimum distance $Dmin$ between a point outside the spanning tree and its potential parent. Then, all the points outside the tree whose distance to their potential parents is at most $C+Dmin$ will be considered candidates. C is a distance equal to a percentage CP of the average distance between all the pairs of red points. One of the candidates will be randomly chosen and added to the tree (by connecting it to its potential parent). Let the selected candidate be p. After this we consider all the points q which are still outside the spanning tree and compute their distance to p. If $distance(p,q)+C$ is smaller than the distance from q to its potential parent then we set p to be q's new potential parent. Otherwise, if $distance(p,q)$ is smaller than the distance from q to its potential parent we set p to be q's potential parent with a fixed probability $prob$.

A second possible approach is to generate any random spanning tree and then transform it into a spanning tree without self-intersections. The transformation can be performed by repeatedly applying *2-opt moves*, as was proposed in (Zhu et al., 1996) for transforming non-simple polygons into simple polygons on the same set of vertices.

In Fig. 1 we present a set of red and blue points and in Fig. 2, 3 and 4 we present three different non-self-intersecting spanning trees of the red points. We will use these three spanning trees in order to illustrate the convexification algorithm presented in the next section.

4. CONVEXIFYING NON-SELF-INTERSECTING SPANNING TREES

Let's consider that we have a non-self-intersecting spanning tree of the red points. It is easy to transform this tree into a non-simple separating polygon which contains all the red points on its border. Let's consider an arbitrary red point q and let's sort all of its tree neighbors r in ascending order according to the angle the segment r-q makes with the OX axis (from 0 to 2π). Let the order of the $K(q)$ tree neighbors of the red point q be $r(q,0), r(q,1), \ldots, r(q,K(q)-1)$. We will now perform an Euler tour (Tarjan and Vishkin, 1984) of the spanning tree starting from an arbitrary point u. The first edge to be traversed will be $(u, r(u,0))$. Then, let's assume that the most recent edge traversed was $(r(v,i), v)$. The next traversed edge will be $(v, r(v, (i+1) \text{ modulo } K(v)))$. The traversal will stop when the edge $(r(u, K(u)-1), u)$ is traversed. The initial polygon P will consist of all the traversed edges, in the order in which they are traversed. Note that P is not a simple polygon. Each point q will occur as a vertex of P a number of times equal to its degree $K(q)$ in the spanning tree.

After obtaining the initial polygon P corresponding to the spanning tree we will apply the following convexification algorithm:

1) Find three consecutive vertices u, v, w, on the border of P, such that v is a red point and the corner (u,v,w) can be convexified.

2) If no three consecutive vertices (u,v,w) such that the corner (u,v,w) can be convexified were found then stop. Otherwise convexify the corner (u,v,w) and then go back to step 1.

We will now discuss conditions for being able to convexify a corner (u,v,w) of P. First of all, the corner (u,v,w) must be a concave corner. The second condition depends on whether we can use new points as vertices of the separating polygon or only the original $M+N$ points can be used as vertices. We will obtain the set $S(u,v,w)$ of all the points located inside the triangle (u,v,w): points already located on the polygon border and blue points which are not among the polygon's vertices. We will compute the convex hull of the set

$$S'(u,v,w) = S(u,v,w) \cup \{u,w\} \quad (1)$$

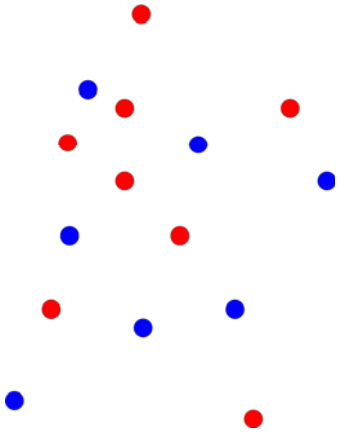

Fig. 1. A set of M=8 red and N=7 blue points.

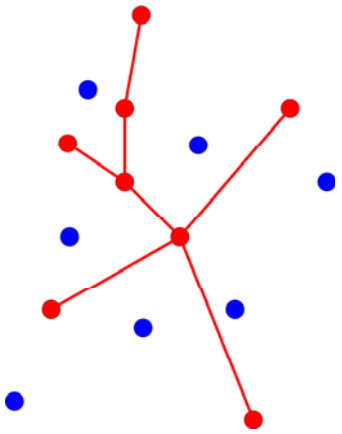

Fig. 2. Non-self-intersecting spanning tree 1 of the red points from Fig. 1.

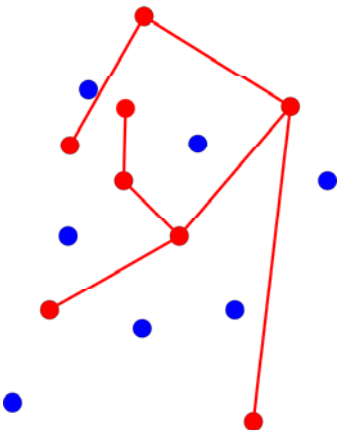

Fig. 3. Non-self-intersecting spanning tree 2 of the red points from Fig. 1.

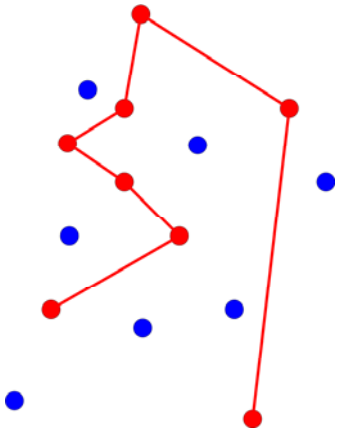

Fig. 4. Non-self-intersecting spanning tree 3 of the red points from Fig. 1.

The convex hull will consist of the segment *u*-*w* and a chain *C(u,v,w)* of points going from *u* to *w* (when *S(u,v,w)* is empty this chain is also the segment *u*-*w*). When new points cannot be used (case 1), then the second condition states that no other point from the polygon's border (except *u* and *w*) should be located on *C(u,v,w)*. When new points can be used (case 2) then there is no extra condition besides the corner *(u,v,w)* being a concave corner.

In case 1 we will replace the corner *(u,v,w)* by the chain *C(u,v,w)* on the border of the polygon *P*. In case 2 we will compute *C'(u,v,w)* by replacing each point *q* from *C(u,v,w)* which is already located on *P*'s border (except for *u* and *w*) by another point *q'* which is very close to *q*, but inside the polygon defined by *C(u,v,w)* and the segments *u*-*v* and *v*-*w*. We will consider these new points *q'* to be of a third color (e.g. violet). Then we will replace the corner *(u,v,w)* by the chain *C'(u,v,w)* on the border of *P*. Note that after this convexification operation the length of *P*'s perimeter decreases (because the length of *C(u,v,w)* or *C'(u,v,w)* is smaller than the sum of the lengths of the segments *u*-*v* and *v*-*w*).

An example of a successful corner convexification (case 1) is presented in Fig. 5 and an example of an impossible corner convexification (case 1) is presented in Fig. 6. Note how in Fig. 5 we can covexify the corner *(u,v,w)*, because *C(u,v,w)* consists of the sequence of points *u*-*x*-*y*-*w* which does not contain any other points from the polygon's border except *u* and *w* (despite the fact that *S(u,v,w)* contains the point *z* which belongs to the polygon's border). In Fig. 6 the corner *(u,v,w)* cannot be convexified because *C(u,v,w)* contains the point *x* which belongs to the polygon's border. When we switch from case 1 to case 2, the convexification of the corner *(u,v,w)* becomes possible by creating a new point *x'* very close to the point *x* (and inside the polygon *u*-*x*-*y*-*w*-*v*) and adding *x'* to the border of the polygon, as depicted in Fig. 7.

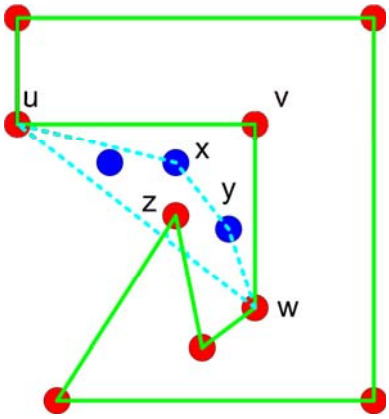

Fig. 5. Successful corner convexification (case 1).

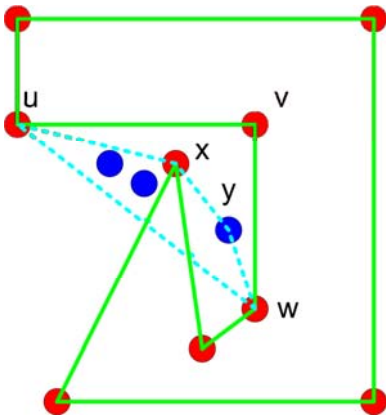

Fig. 6. Impossible corner convexification (case 1).

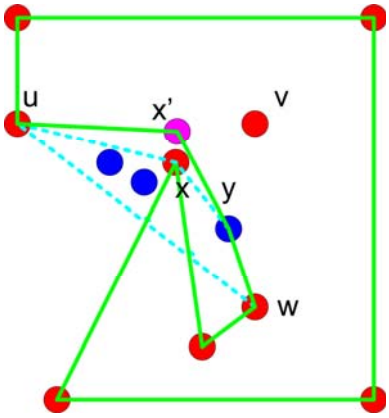

Fig. 7. Successful corner convexification (case 2).

In Fig. 8 we present the end result of the convexification algorithm (case 1) when applied to the spanning tree (and set of points) presented in Fig. 2. Solid red edges are spanning tree edges. Dashed greed edges are edges on the border of the separating polygon.

In Fig. 9 we present the result of the convexification algorithm (case 1) when applied on the spanning tree (and set of points) presented in Fig. 3. Solid red edges are spanning tree edges located inside the separating polygon, solid green edges are spanning tree edges located on the boundary of the separating polygon and dashed green edges are polygon edges not part of the spanning tree.

Note how, in the end, the resulting polygon is not simple, because the concave corner *(u,v,w)* could not be convexified (because point *x* is part of *C(u,v,w)*) and, thus, points *v* and *w* occur twice on the boundary of the polygon. It is not easy to know before-hand (before starting the convexification procedure) if the convexification algorithm will end with a simple polygon *P* or not. However, this problem can be solved when considering case 2 (i.e. when we are allowed to add new points as vertices of the separating polygon), as depicted in Fig. 10. Note how, by adding a new point *x'* very close to point *x*, we are able to convexify the corner *(u,v,w)* (and then the corner *(x',w,y)*) and no point occurs multiple times on the boundary of the polygon. In fact, in case 2, we can start from the very beginning with a simple polygon. Let's consider the polygon *P* obtained from the Euler tour of the spanning tree. We will construct a new polygon *P'* as follows. For each corner *(r(q,i), q, r(q, (i+1) modulo K(q)))* the point *q* will be replaced by a copy *q(i)* of it which is located very close to *q*, in the wedge defined by its two neighbors *r(q,i)* and *r(q, (i+1) modulo K(q))*. For instance, a good candidate for placing *q(i)* is on the bisector of the angle formed by *r(q,i)* and *r(q, (i+1) modulo K(q))* with *q*, but very close to *q*. When *q* has only one spanning tree neighbor then *r(q,i)* = *r(q,(i+1) modulo K(q))*. In this case we consider the wedge to be the "exterior" wedge (whose angle is equal to 2π). The bisector of this angle goes in the opposite direction of the segment *(q,r(q,0))*. Fig. 11 shows the modified polygon *P'* for the spanning tree presented in Fig. 3, where the new points are the violet points. Then, the convexification algorithm will start from the modified polygon *P'* which is already a simple separating polygon.

The end result of the convexification algorithm applied to the spanning tree from Fig. 4 (considering case 1) is presented in Fig. 12. As in Fig. 9, solid red edges are spanning tree edges located inside the separating polygon, solid green edges are spanning tree edges located on the boundary of the separating polygon and dashed green edges are polygon edges not part of the spanning tree.

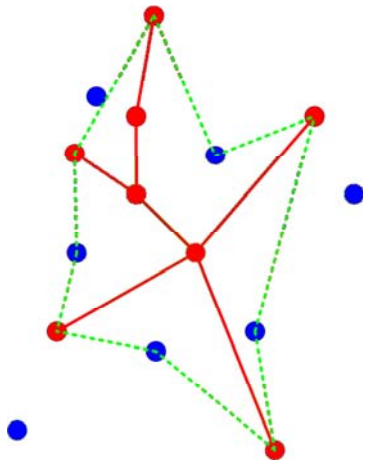

Fig. 8. End result of the convexification algorithm applied on the spanning tree and set of points from Fig. 2 (case 1).

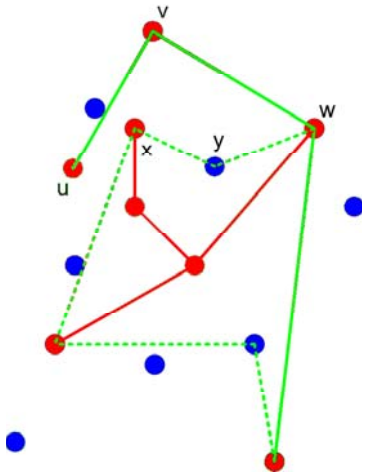

Fig. 9. End result of the convexification algorithm applied on the spanning tree and set of points from Fig. 3 (case 1).

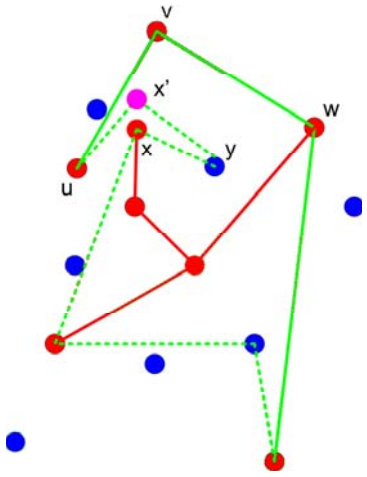

Fig. 10. End result of the convexification algorithm applied on the spanning tree and set of points from Fig. 3 (case 2).

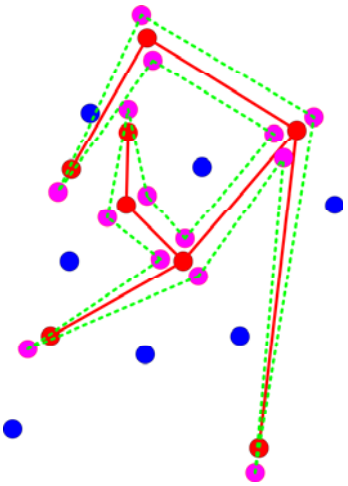

Fig. 11. Modified initial separating polygon (case 2).

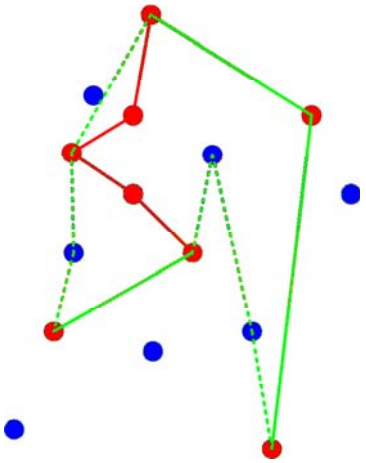

Fig. 12. End result of the convexification algorithm applied on the spanning tree and set of points from Fig. 4 (case 1).

We will now discuss about several implementation issues of the convexification algorithm presented in this section. First of all, in step 1, the algorithm does not specify which corner should be selected for convexification, in case there are multiple candidates. In general, the easiest approach would be to use a pointer u which tries to convexify the corner $(u, next(u), next(next(u)))$ (we denoted by $next(v)$ the next point on the polygon boundary after v). If the corner $(u, next(u), next(next(u)))$ could be convexified, then we do not update the pointer (because, perhaps, another convexification operation could be performed here); otherwise, we advance the pointer u to $next(u)$. After the pointer was moved completely along the polygon boundary without performing any more convexification operations (i.e. since the last convexification operation or since the beginning if no such operation has been performed) we can stop. Of course, other approaches could be used – for instance, at each step, all the corners could be evaluated and the corner to be convexified could be the one which reduces the length of the polygon's perimeter the most.

A question which arises is if the order in which we perform the convexification operations matters for the end result. In case 1 (when no extra points can be used as polygon vertices), this order indeed matters. In Fig. 13 we show an example in which three corners, (t,u,v), (u,v,w) and (v,w,x) can be convexified. If we convexify (t,u,v) or (v,w,x) first the final polygon will be the one depicted in Fig. 14. If we convexify (u,v,w) first the final polygon will be the one shown in Fig. 15.

In case 2 the order of the convexification operations does not matter and the end result will always be approximately the same (we say approximately because the locations of the new points will not necessarily be exactly the same, but they will be very close to some of the original points). It is easy to see that in case 2 the polygons from Fig. 14 and Fig. 15 can be convexified further to approximately the same final polygon (shown in Fig. 16).

For step 2 we need to find all the points located in a given triangle *(u,v,w)*. An easy *O(M+N)* approach is to independently test each point on the polygon boundary and each blue point outside of the polygon. However, more efficient approaches can be used. For case 1 we can preprocess the points into a static data structure which can efficiently answer triangle range reporting queries. Data structures with *O(M+N)* preprocessing and $O((M+N)^{0.5}+F)$ query time exist, as well as data structures with $O((M+N)^2)$ preprocessing and $O(\log^2(M+N)+F)$ query time or $O((M+N)^{2+\varepsilon})$ preprocessing and *O(log(M+N)+F)* query time (Matousek, 1994), (Chazelle et al., 1992), (Erickson, 2000), where *F* is the number of points located inside the query triangle. Computing the convex hull of the points from *S'(u,v,w)* can be easily performed in $O(|S'(u,v,w)|\cdot\log(|S'(u,v,w)|))$ time (Dave and Dave, 2008) or better if the points inside the triangle *(u,v,w)* are reported in sorted order.

In case 2, when new points can be created as vertices of the polygon, we may need to use dynamic data structures. A simple alternative is to use dynamic data structures for orthogonal range search (e.g. 2D range trees, quad-trees, kd-trees (de Berg et al., 2008), (Andreica and Tapus, 2010). Then, when performing a triangle query, we will first find all the points located in the minimum bounding box of the triangle (the sides of the bounding box are parallel to the OX and OY axes) and then we will keep only those which are also located inside the triangle.

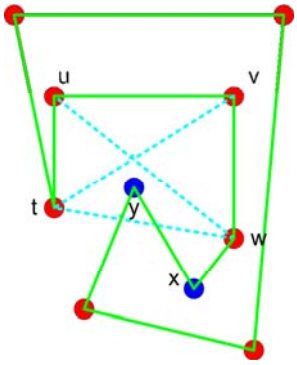

Fig. 13. Three corners can be convexified: *(t,u,v)*, *(u,v,w)* and *(v,w,x)*.

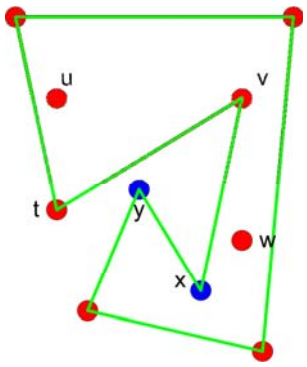

Fig. 14. Final polygon if we convexify the corners *(t,u,v)* or *(v,w,x)* first in Fig. 13 (considering case 1).

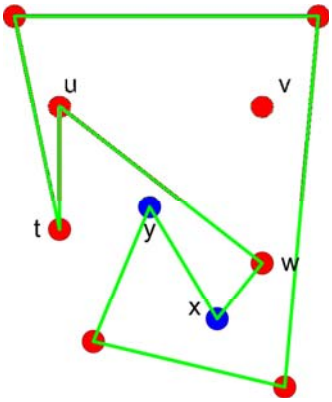

Fig. 15. Final polygon if we convexify the corner *(u,v,w)* first in Fig. 13 (considering case 1).

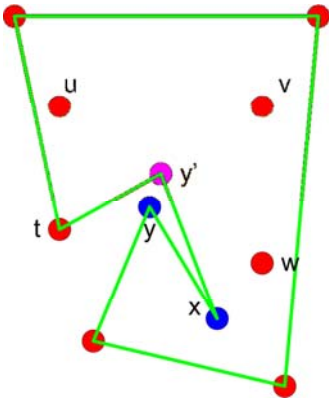

Fig. 16. Final polygon if we convexify the polygon depicted in Fig. 13 (considering case 2). The order of convexification operations does not matter.

5. FINDING A MINIMUM PERIMETER SEPARATING POLYGON

In this section we will assume that we want to compute a minimum perimeter separating polygon such that all the red points are inside or on the border of the polygon and all the blue points are outside or on the border of the polygon. Obviously, a minimum perimeter separating polygon may have either the red or the blue points inside. So, in order to try to find a minimum perimeter separating polygon we will need to run the algorithms presented below twice (once considering that the red points are kept inside the polygon and again considering that the blue points are kept inside the polygon). We will assume that there is a time limit for which the algorithms are allowed to run.

Algorithm 1 is an extremely simple algorithm. It will keep generating random non-intersecting spanning trees, convexify them and keep the best polygon found (*Popt*).

```
Algorithm 1.
(1)Popt = empty
(2)while the time limit was not exceeded do
```

```
   (2.1)T = a random non-self-intersecting spanning tree of the red points
   (2.2)P=Convexify(T)
   (2.3)if (P is simple) and ((Popt is empty) or (perimeter(P)<perimeter(Popt))) then
        (2.3.1)Popt=P
```

For Algorithms 2 and 3 we will introduce an operation which transforms a non-self-intersecting spanning tree T into another non-self-intersecting spanning tree T' by removing an edge and adding another one. This operation will first choose a random edge (u,v) of T and remove it. Then, the vertices of T will be split into two sets T_u and T_v (T_u and T_v are the two parts of T which were disconnected by the removal of the edge (u,v)). Then, a new edge (u',v') is selected randomly such that $u' \in T_u$ and $v' \in T_v$ and $(u',v') \neq (u,v)$. A new spanning tree T' is constructed by adding the edge (u',v') to T (from which the edge (u,v) was previously removed). If the edge (u',v') does not intersect any of the other edges of T' then T' is a non-self-intersecting spanning tree and the transformation operation is complete. Otherwise, a new edge (u',v') will be randomly selected. After a number of edges (u',v') were selected and none of them led to a non-self-intersecting spanning tree T' the edge (u,v) is added back to T and another edge (u,v) will be randomly selected for removal. The transformation operation is complete when a non-self-intersecting spanning tree T' is obtained or when the time limit is exceeded. With this operation defined, we can present Algorithms 2 and 3. Algorithm 2 corresponds to a random walk in the space of non-self-intersecting spanning trees which can be convexified to simple polygons. Algorithm 3 corresponds to an iterative improvement algorithm with hill-climbing, in which the "neighbor" of the current state (the tree T' into which T is transformed) is selected randomly (Kumar and Kumar, 2013). Algorithm 3 can also be viewed as an instance of a stochastic local search algorithm (Hoos and Stutzle, 2005).

Algorithm 2.

```
(1)Run Algorithm 1 for a shorter time limit and keep both Popt and the spanning tree T which generated Popt.
(2)while the time limit was not exceeded do
   (2.1)T'=Transform(T)
   (2.2)P=Convexify(T')
   (2.3)if (P is simple) then
        (2.3.1)T=T'
        (2.3.2)if (Popt is empty) or (perimeter(P)<perimeter(Popt)) then
             (2.3.2.1)Popt=P
```

Algorithm 3.

```
(1)Run Algorithm 1 for a shorter time limit and keep both Popt and the spanning tree T which generated Popt.
(2)while the time limit was not exceeded do
   (2.1)T'=Transform(T)
   (2.2)P=Convexify(T')
   (2.3)if (P is simple) and ((Popt is empty) or (perimeter(P)<perimeter(Popt))) then
        (2.3.1)T=T'
        (2.3.2)Popt=P
```

Our non-self-intersecting spanning tree approach can also be included within a simulated annealing algorithm (van Laarhoven and Aarts, 1992) or a genetic algorithm (Goldberg, 2013). Other types of optimization algorithms could also be adapted to use our spanning tree convexification approach, like, for instance, ant colony optimization algorithms (Dorigo and Stutzle, 2004), (Loukhaoukha, 2013) or physics-inspired algorithms (Biswas et al., 2013).

6. EXPERIMENTAL EVALUATION

We implemented Algorithms 1, 2 and 3 and evaluated them on *5* sets of $M=N=50$ random points in general position with real coordinates between *0* and *1000*. We chose a time limit of 10 seconds and ran the algorithms on an Intel Atom N570 1.66 GHz CPU. For Algorithms 2 and 3 the shorter time limit for running Algorithm 1 initially was chosen to be 2 seconds. We used the modified Prim's algorithm for generating random spanning trees in Algorithm 1 (with $CP=0.8$ and $prob=0.33$). For Algorithms 2 and 3 we used a slightly modified spanning tree transformation operation. The edge (u',v') to be added to the spanning tree must never be too long. In particular, edges (u',v') which were longer than IP times the average distance between all the pairs of red points were not considered (we used $IP=10$). We only considered case 1 for the convexification algorithm (i.e. the polygon's vertices are some of the given $M+N$ points). We computed and compared the perimeter of the solution found by each algorithm. The results are presented in Table 1.

Table 1. Perimeter length of the best separating polygon found by each algorithm on each of the 5 data sets.

	Set 1	Set 2	Set 3	Set 4	Set 5
Algorithm 1	5846.47	5878.32	5718.22	5527.29	5423.40
Algorithm 2	5856.18	5878.32	5749.51	5587.19	5433.03
Algorithm 3	5716.68	5248.01	5551.55	5370.97	5201.09

We can see that Algorithm 3 produces the best results, followed by Algorithm 1 and Algorithm 2. Algorithm 3 produces results which are around 4.6% better on average than the results produced by Algorithm 1. Algorithm 2 produces worse results than Algorithm 1 because it seems that random walks in the space of non-self-intersecting spanning trees are not efficient in finding good solutions. In fact, the random walk in Algorithm 2 never improved the initial solution produced by Algorithm 1 (which was run for the initial 2 seconds out of the 10 second time limit allowed for Algorithm 2).

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a novel approach for generating minimal polygons which separate M red points from N blue points in the plane based on convexifying non-self-intersecting spanning trees. We assumed that the points are in general position (i.e. no three points are collinear), but most of our results hold even without this condition. We described methods for generating non-self-intersecting spanning trees, for convexifying a spanning tree and for including the approach into generic optimization algorithms which seek to find the minimum perimeter separating polygon. We also provided experimental results for 3 optimization algorithms which use our approach in order to find a minimum perimeter separating polygon.

Our approach can also be used in order to easily generate minimal separating polygons, a problem which was otherwise not considered before (unlike the problem of generating random polygons with a given set of vertices (Zhu et al., 1996).

As future work we intend to implement and evaluate case 2 of the convexification algorithm as well as include our approach into other generic optimization algorithms. Moreover, we will consider the possibility of improving the time complexity of the algorithms presented in this paper. Our first non-self-intersecting spanning tree generation algorithm has a time complexity of $O(M^2)$, but it can definitely be improved by using spatial data structures for considering only "close" points when updating the potential parents and by using balanced tree-like data structures for obtaining all the candidates for addition to the spanning tree. The convexification algorithm has a time complexity of $O((M+N)^{1.5})$ in the best case. This time complexity can be achieved assuming that there are $O(M+N)$ corner convexification operations which may be performed and for each such operation we may find in $O((M+N)^{0.5}+F)$ time the F points located inside the query triangle. We make the reasonable assumption that the total number of points F' located in all the query triangles is not very large, so that the time complexity of computing their convex hulls (an upper bound is $O(F' \cdot log(F'))$) does not exceed $O((M+N)^{1.5})$.

REFERENCES

Andreica, M. I. and Tapus, N. (2010). Practical Range Aggregation, Selection and Set Maintenance Techniques, *Politehnica University of Bucharest Scientific Bulletin, Series C - Electrical Engineering and Computer Science*, vol. 72, issue 2, pp. 3-16.

Arora, S. and Chang, K. L. (2003). Approximation Schemes for Degree-Restricted MST and Red-Blue Separation Problem, *Proceedings of the 30th International Conference on Automata, Languages and Programming (ICALP)*, pp. 176-188.

Biswas, A., Mishra, K. K., Tiwari, S., and Misra, A. K. (2013). Physics-Inspired Optimization Algorithms: A Survey, *Journal of Optimization*, Article Id 438152.

Chazelle, B., Sharir, M., and Welzl, E. (1992). Quasioptimal Upper Bounds for Simplex Range Searching and New Zone Theorems, *Algorithmica*, Vol. 8, pp. 407-429.

Dave, P. H. and Dave, H. B. (2008). *Design and Analysis of Algorithms*. Pearson Education.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag.

Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimization*, Massachusetts Institute of Technology.

Eades, P. and Rappaport, D. (1993). The Complexity of Computing Minimum Separating Polygons, *Pattern Recognition Letters*, Vol. 14, pp. 715-718.

Erickson, J. (2000). Space-Time Tradeoffs for Emptiness Queries, *SIAM Journal of Computing*, Vol. 29 (6), pp. 1968-1996.

Fulek, R., Keszegh, B., Moric, F., and Uljarevic, I. (2010). On Polygons Excluding Point Sets, *Proceedings of the 22nd Canadian Conference on Computational Geometry*, pp. 273-276.

Goldberg, D. E. (2013). *Genetic Algorithms*. Pearson Education.

Hoos, H. and Stutzle, T. (2005). *Stochastic Local Search: Foundations and Applications*. Elsevier.

Kumar, T. V. V. and Kumar, S. (2013). Materialized View Selection Using Iterative Improvement. *Advances in Intelligent Systems and Computing*, Vol. 178, pp. 205-213.

Loukhaoukha, K. (2013). Image Watermarking Algorithm Based on Multiobjective Ant Colony Optimization and Singular Value Decomposition in Wavelet Domain, *Journal of Optimization*, Article Id 921270.

Mata, C. S. and Mitchell, J. (1995). Approximation Algorithms for Geometric Tour and Network Problems, *Proceedings of the 11th ACM Symposium on Computational Geometry*, pp. 360-369.

Matousek, J. (1994). Geometric Range Searching, *ACM Computing Survey*, Vol. 26 (4), pp. 421-461.

Mitchell, J. S. B. and Wynters, E. L. (1991). Finding Optimal Bipartitions of Points and Polygons, *Lecture Notes in Computer Science*, Vol. 519, pp. 202-213.

Provencal, X. and Lachaud, J.-O. (2009). Two Linear-Time Algorithms for Computing the Minimum Length Polygon of a Digital Contour, *Lecture Notes in Computer Science*, Vol. 5810, pp. 104-117.

Sedgewick, R. and Wayne, K. (2011). *Algorithms*. Pearson Education.

Tarjan, R. E. and Vishkin, U. (1984). Finding Biconnected Components and Computing Tree Functions in Logarithmic Parallel Time, *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 12-20.

van Laarhoven, P. J. M. and Aarts, E. H. L. (1992). *Simulated Annealing: Theory and Applications.* Kluwer Academic Publishers.

Zhu, C., Sundaram, G., Snoeyink, J., and Mitchell, J. S. B. (1996). Generating Random Polygons with Given Vertices. *Computational Geometry*, Vol. 6 (5), pp. 277-290.

A Neural Networks Approach for Churn Prediction Modeling in Mobile Telecommunications Industry

Ionut B. Brandusoiu, Gavril Toderean

Technical University of Cluj-Napoca, Cluj-Napoca, Romania
(e-mail: ionut.brandusoiu@gmail.com, toderean@pro3soft.ro)

Abstract: Nowadays, organizations are facing several challenges resulting from competition and market trends. Customer churn is a real issue for organizations in various industries, especially in the telecommunications sector with a churn rate of approximately 30%, placing this industry in the top of the list. Because higher expenses are involved when trying to attract a new customer than trying to retain an existing one, this is an important problem that needs an accurate resolution. This paper presents an advanced methodology for predicting customers churn in mobile telecommunication industry by applying data mining techniques on a data set consisting of call detail records (CRD). The data mining algorithms considered and compared in this paper are Multi-Layer Perceptron and Radial Basis Function neural networks.

Keywords: Data mining, churn prediction, neural networks, multi-layer perceptron networks, radial basis function networks.

1. INTRODUCTION

Customer retention in the telecom industry has become very important with the increasing competition and diversity of offerings on the market. In order to keep up in such a competitive marketplace, many telecommunication companies are using data mining techniques to overcome challenging issues such as: churn prediction, customer profiling, cross-selling and up-selling, lifetime value, and fraud detection (Shaw et al., 2001).

The word "*churn*" which has been obtained by combining the English words "*change*" and "*turn*", describes the phenomenon of loss of a customer. In mobile telecommunications industry (also referred to as customer attrition or subscriber churning) it refers to the movement of subscribers from one provider to another (Wei and Chiu, 2002). It is measured by the rate of churn and is an important indicator for organizations. The churn rate represents the percentage of lost customers over a given period divided by the total number of customers at the beginning of that period. The churn process is usually happening due to better rates or services, or due to different benefits that a competitor company offers when signing up.

The average monthly churn among mobile operators in Europe varies between 8% and 12% (Richeldi and Perrucci, 2002). The annual attrition rate varies from 20% to 40% in most mobile telecommunication operators (Ahn et al., 2006). In such a competitive market, a defensive marketing strategy is very important. Instead of trying to acquire new customers or to attract customers away from the competition, the defensive marketing is more interested in reducing the churning of its customers (Ahn et al., 2006), especially when it is 5 times more costly to acquire a new customer than to keep one (Verbeke et al., 2011). Being so expensive to acquire new subscribers, it is crucial to keep the existing ones. Building a churn prediction model will ease the customer retention process, and in this way the mobile telecommunication companies will success in this constantly increasing competitive marketplace.

2. LITERATURE REVIEW

The churn prediction modeling process is strongly dependent on the data mining process and techniques due to an increased performance generated by machine learning (ML) algorithms compared to the statistical techniques for nonparametric data (Garcia et al., 2012). Of several ML algorithms studied and used in other research papers for churn prediction, we can mention the following: decision trees (DT), logistic regression (LR), neural networks (NN), and Bayesian networks (BN).

This paper presents an advanced data mining methodology based on two well-known neural network models. Multi-Layer Perceptron neural network has been previously used to create a churn prediction model, but Radial Basis Function neural network is first presented and used in this research paper.

This paper is organized as follows: a short introduction to data mining is presented and theoretical aspects of Multi-Layer Perceptron and Radial Basis Functions neural networks and in the next section, the methodology with its corresponding phases. Finally, the conclusion and future work are presented.

2.1 Data Mining

Data mining is the practice of digging data to find trends and patterns, and can provide you with answers to questions that you should have asked (Whigham, 2011). Data mining methods lie at the intersection of artificial intelligence, machine learning, statistics, and database systems (Chakrabarti et al., 2006). Data mining techniques can help building prediction models in order to discover future trends and behaviors, allowing organizations to make smart decisions based on knowledge from data.

The methodology used is called modeling. Modeling is the process of creating a mining model, an implementation of specific data mining algorithms. These algorithms are mathematical functions that perform specific types of analysis on the associated data set (Fouche and Langit, 2011). Based on the business problems that need assistance, these data mining algorithms can be broadly classified into several categories, including classification, segmentation or clustering, association, regression or forecasting, sequence analysis and prediction, and deviation or anomaly analysis (Gorunescu, 2011). Data mining methods can be summarized in two main categories: predictive methods (use existing variables to predict values of other variables – classification, regression or forecasting, and anomaly analysis) and descriptive methods (reveal patterns and trends in data that can be easily interpreted by the user – segmentation or clustering, association, and sequence analysis) (Gorunescu, 2011).

Classification analysis is the process by which a model is built to categorize pre-classified instances (training examples) into classes. This classifier (classification model) is then used to classify future instances. Some of the most used classification algorithms are Decision Trees, Naïve Bayes, Neural Networks, Support Vector Machines, Genetic algorithm, and K-Nearest Neighbor.

Clustering analysis is a process whereby a set of heterogeneous data is segmented according to some similarities into more homogenous subgroups, called clusters. Unlike the classification process, the clustering process does not rely on predefined classes, and there is no training set or predefined examples.

Association rule discovery is the process that involves finding correlations between different groups of phenomena.

Regression or forecasting is the process whereby a continuous value(s) (predictor(s)) can be predicted based on multiple input variables (response variables) which usually contain time series data. Some of the most adopted techniques include Time Series Analysis, Linear Regression, and Logistic Regression.

Sequence and prediction analysis is the process that discovers patterns that are frequently repeated in a particular set (sequence) of data. Anomaly analysis is the process that deals with detection of abnormal behavior (outliers, anomalies, or deviations) in the data.

In the following section, the discussion is restricted to neural networks classification algorithm used later to create a predictive model in order to solve the subscribers churning problem. I decided to use the neural networks approach because they can build predictive models with minimal demands on model structure and assumptions, and have the ability to detect complex nonlinear relationships between the predictors and target variables.

2.2 Neural Networks

An artificial neural network (ANN), often just called neural network (NN), is a very popular machine learning algorithm based on modern mathematical concepts inspired by biological neural networks. The capacity of human beings to perceive and memorize new information through a learning process, motivated researchers to develop artificial systems which are able to perform certain functions based on a learning process (training) (Cremene and Zahan, 2009).

Artificial neural networks are obtained by connecting a number of elementary information processing units, called artificial neurons. When arranged and connected in parallel, these artificial neurons form a layer of a neural network. An ANN can have one or more layers of neurons (Ayoubi, 1996).

Artificial neural networks' ability to learn, consists in their property to change their parameters (weights and thresholds) based on the input data used during training (learning) process. ANN training could be supervised and unsupervised (Bishop, 2008). In our particular case we will use the supervised learning task because we will provide a training set which consists of input signals (predictors) and the correct output (target) which is explicitly given. In this case, the weights and thresholds values of the neurons are determined to lead to the minimization of a specified criterion function.

There are multiple types of neural networks, but in order to identify nonlinear processes, we will use the following two: Multi-Layer Perceptron (MLP) and Radial Basis Functions (RBF) neural networks.

2.3 Multi-Layer Perceptron (MLP) Networks

The multi-layer perceptron network consists of one or more layers connected in series, and each layer contains one or more processing units called perceptrons. The perceptrons are interconnected in a feedforward way, passing the network layer by layer until it reaches the output (Figure 1). The perceptron is the simplest form of a neural network, and has a single neuron that sums its weighted inputs, sum that is followed by an activation function.

Where *u* is the transfer value and is given by equation (1). Next, the nonlinear transformation that corresponds to the activation function is applied, thus obtaining the neuron output signal, *y*, given by (2) (Haykin, 2009):

$$u = \sum_{i=1}^{N} w_i x_i = w^T x \tag{1}$$

$$y = \sigma(u) \tag{2}$$

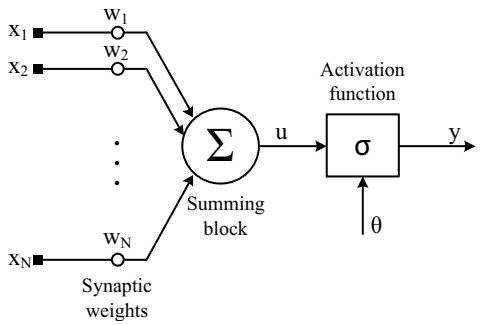

Fig. 1. Perceptron structure (Haykin, 2009).

The activation function used is the sigmoid function with its equation and its first derivative in (3), respectively (4):

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x)) \tag{4}$$

Arranging perceptrons in parallel in multiple layers connected in series forms a Multi-Layer Perceptron neural network. Typically, a MLP neural network has: an input layer, one or more hidden layers, and an output layer (Figure 2) (Haykin, 2009). The hidden layers confer the ability to learn to solve complex problems.

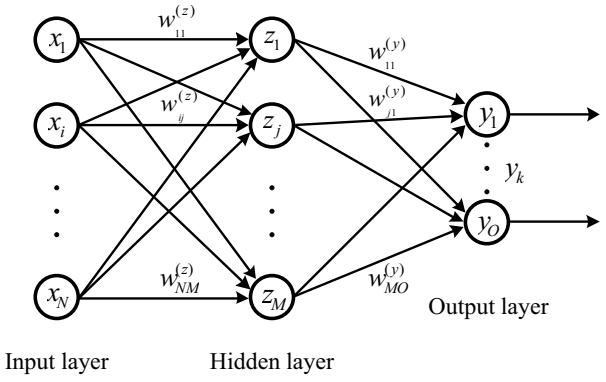

Fig. 2. Architecture of a MLP with one hidden layer.

MLP neural network training is done in a supervised manner, using the Back-Propagation (BP) learning algorithm. This learning algorithm is the most known training algorithm for neural networks. Approximately 95% of reported neural networks applications utilize multi-layer perceptron (MLP) neural network with the back-propagation learning algorithm (Wong et al., 1997).

This learning algorithm mechanism can be divided in two phases (Haykin, 2009):

- A forward pass – feedforward propagation of input signals. During this propagation the neurons' weights and thresholds have fixed values;
- A backward pass – propagates the error (5) backwards through the network, starting at output units (where the error is the difference between actual (*y*) and desired output (*t*) values) (Figure 3). During this back-propagation the weights and thresholds are adapted to obtain signals as closer as possible to the desired values.

$$e_k = t_k - y_k \tag{5}$$

The sum of squares error function is defined by the following equation:

$$E = \frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{O}(t_k^{(l)} - y_k^l)^2 \tag{6}$$

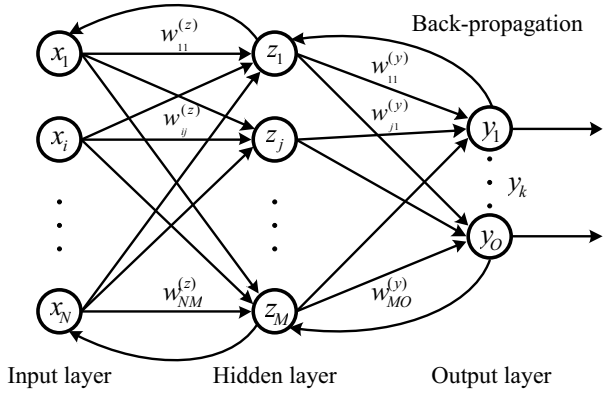

Fig. 3. Architecture of a MLP with back-propagation.

Back-propagation algorithm seeks to minimize the error function defined in the space generated by the weights using the descent gradient optimization algorithm. The set of weights that minimize the error function is considered the solution to the training (learning) (Haykin, 2009).

For each output node the local gradient is computed by using the formula:

$$\delta_k = y_k(1-y_k)(t_k - y_k) \tag{7}$$

For each hidden node the local gradient is computed by using the formula:

$$\delta_j = y_j(1-y_j)\sum_{k=1}^{O}\delta_k w_{jk} \tag{8}$$

The weights are updated bringing the weight in the opposite direction of the gradient by subtracting the learning rate (η) from the weight. The speed and quality of learning depend on this ratio.

$$w_{new} = w_{old} + \Delta w \quad (9)$$

$$\Delta w = -\eta \delta_l y_{l-1} \quad (10)$$

These two phases are repeated until the performance of the network is satisfactory.

2.4 Radial Basis Function (RBF) Networks

A radial basis function (RBF) network is a feedforward, supervised learning network with an input layer, only one hidden layer called the radial basis function layer, and an output layer (Bishop, 2008). Like MLP, RBF network can be used for prediction and classification problems. Compared to MLP networks, RBF networks take a different approach, in terms that the nodes from the hidden layer implement a set of radial basis functions. In our case we will use Gaussian function which has the equation (11). This function is localized in the sense that for large values it tends to zero.

$$\phi(x) = \exp(-\frac{x^2}{2\sigma^2}) \quad (11)$$

Instead of every input data point (x_n) having its own influence on the Gaussian radial basis function, we are going to elect a number ($K<<N$) of important centers for the data, and have them influence the neighborhood around them. These centers (μ_k) will be the centers of the Gaussian functions, and are in the same space as the input data points x_n.

The output equation for the j[th] hidden node is:

$$z_k = \exp(-\frac{1}{2\sigma_k^2}\left\|x - \mu_k\right\|^2) \quad (12)$$

where $\|x-\mu_k\|$ is the Euclidean distance between the input data vector x and the corresponding center μ_k.

Regarding the output nodes, they implement linear summation functions as in multi-layer perceptron networks (Bishop, 2008):

$$y = \sum_{k=1}^{K} w_k \exp(-\frac{1}{2\sigma_k^2}\left\|x_n - \mu_k\right\|^2) \quad (13)$$

The architecture of a RBF network is depicted in Figure 4:

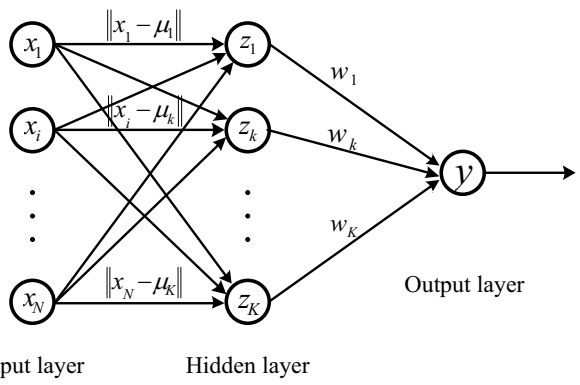

Fig. 4. Architecture of a RBF network (Bishop, 2008).

The process of finding the w_k and μ_k parameters from equation (13) is called training. The RBF network is trained in two stages (Zhang et al., 1996):

- Determine the centers μ_k and their widths by using the two-step clustering algorithm.
- Estimate the synaptic weights w_k given the radial basis functions.

Because of the simplicity of these two stages the RBF network is trained much quicker than MLP.

The Two Step cluster algorithm (Zhang et al., 1996) can handle very large data sets, and both continuous and categorical variables. It consists of two steps:

- Pre-cluster the input data into small sub-clusters – scans the input data and if the conditions are met the current input record is merged with the previously formed clusters or a new one is created based on the distance criterion (ISL, 2007).
- Cluster the sub-clusters into the desired number of clusters by using an agglomerative hierarchical clustering method (Zhang et al., 1996).

For each cluster the mean and standard deviation for each variable is computed (Zhang et al., 1996). The center μ_k of the k[th] radial basis function is equal to:

- The k[th] cluster mean of the i[th] input variable, if it is a continuous variable.

$$\mu_k = \bar{x}_{ki} \quad (14)$$

- The proportion of the category of a categorical variable that the i[th] variable corresponds to.

$$\mu_k = \pi_{ki} \quad (15)$$

The width σ_k of the k[th] radial basis function is equal to:

- The k[th] cluster standard deviation of the i[th] input variable, if it is a continuous variable; and $h>0$ is the RBF overlapping factor.

$$\sigma_k = h^{1/2} s_{ki} \quad (16)$$

- Equation (17), if the input variable is categorical.

$$\sigma_k = h^{1/2} \sqrt{n_{ki}(1-n_{ki})} \quad (17)$$

The error function used in RBF network's case is the sum of squares (6) and the activation function is the identity function (18). The sum of squares error is minimized by using the ordinary least squares regression method (Gonzalez-Camacho et al., 2012).

$$f(x) = x \quad (18)$$

3. METHODOLOGY

The data mining approach proposed in this paper is based on the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. This format is the most complete available and is applied in the present paper to extract information, interpret it and to propose solutions. The CRISP-DM major consisting phases are related in Figure 5 as it is applied to churn prediction modeling.

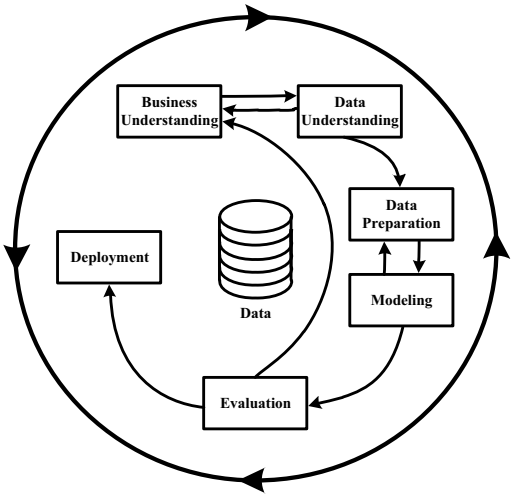

Fig. 5. Architecture of a RBF network (Chapman et al., 2000).

3.1 Business Understanding

Instead of the existing reactive retention program, where company representatives attempt to convince the customers to stay only when they are calling to leave the company, the mobile telecommunication company wants to implement a proactive retention program which should increase subscribers' loyalty before they decide to leave the company and target all their efforts towards clients who are at risk of churning.

The Churn Model analyzes the Call Detail Records (CRD) of subscribers that have left the company and of those that have remained and determines customers that are at the risk of churning.

3.2 Data Understanding

The data set used in this article is from University of California, Department of Information and Computer Science, Irvine, CA (Blake and Merz, 1998). This data set contains historical records of customer churn, how they turned out in hindsight, their previous behavior – if it turned out that they are churners or not.

The Call Detail Records data set used for this purpose has a total number of 3333 subscribers with 21 variables each. For each of these subscribers we can find information about their corresponding inbound/outbound calls count, inbound/outbound SMS count, and voice mail. The "Churn" variable will be used as the target variable, and all other 20 attributes will be used as input variables.

3.3 Data Understanding

For the data preparation part and all the following we decided to use IBM SPSS (Statistical Product and Service Solutions), a statistical and data mining software package used to build predictive models (IBM SPSS, 2012).

Before proceeding to create the churn model, our data need to be cleaned, transformed and prepared in a proper format, suitable for analytical modeling. The first step taken was to audit the data in order to see if there are outliers, extreme or missing values for a particular variable, and to have a first visual contact about advanced statistics pertaining to the variables used. It showed that this is a complete data set, meaning that for each subscriber there is no attribute missing. This saves us additional work, because otherwise, we should have imputed them by choosing the proper method.

During data preparation phase we discovered that between some variables there is a perfect correlation with the R-squared statistic precisely 1. The four *charge* variables are linear functions of the *minutes* variables, so we decided to arbitrarily eliminate all four of them to avoid incoherent results. The *area code* variable contains only three different values (from CA only) for all the records, while the *state* variable contains all 51 states, so we decided not include these two variables as well, as it can be bad data. The *phone* variable has been eliminated because it does not contain relevant data that can be used for prediction; it is useful only for identification purposes. We have therefore reduced the number of predictors from 20 to 13. Table 1 lists all attributes together with their corresponding type.

Table 1. Data set variables

Variable Name	Type
Account Length	Continuous
International Plan	Categorical
Voice Mail Plan	Categorical

# Voice Mail Messages	Continuous
# Day Minutes	Continuous
# Day Calls	Continuous
# Evening Minutes	Continuous
# Evening Calls	Continuous
# Night Minutes	Continuous
# Night Calls	Continuous
# International Minutes	Continuous
# International Calls	Continuous
Customer Service Calls	Continuous
Churn?	Categorical
Omitted	
State	Categorical
Area Code	Categorical
Phone	Categorical
# Day Charge	Continuous
# Evening Charge	Continuous
# Night Charge	Continuous
# International Charge	Continuous

3.4 Modeling

The modeling phase involves choosing the modeling algorithms and the modeling architecture, and finally building the model. Because we intend to build a classification model based on neural networks we must partition the data set in two: a training set and a testing set (Sumathi and Sivanandam, 2006). The training set has been randomly partitioned to be 80% of the original data set, consisting of 2666 subscribers, whereas the testing set has been randomly partitioned to be 20% of the original data set, consisting of 667 subscribers, as shown in Figure 6. This figure also shows the churn distribution within both sets, and it is as follows: in the testing set we have 107 churners (15%) and 559 non-churners (85%), and in the training set we have 376 churners (14%) and 2291 non-churners (86%).

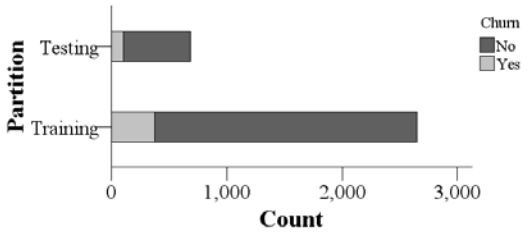

Fig. 6. Samples and churn distributions.

For implementing the churn predictive model a neural network approach was taken. The Multi-Layer Perceptron and Radial Basis Function neural networks algorithms have been chosen to build the predictive models. After multiple configurations of the MLP and RBF neural networks the architectures that are listed in Table 2 and Table 3 are those that drove to better performance in the predictive models.

Table 2. MLP network architecture

MLP network architecture			
Input Layer	# of units	Factors	2
		Covariates	11
Hidden Layer	# of hidden layers		1
	# of units		6
	Activation function		Sigmoid
Output Layer	Dependent variables		1 (churn?)
	# of units		2 (yes/no)
	Activation function		Sigmoid
	Error function		Sum of squares
Optimization algorithm			Gradient descent

Table 3. RBF network architecture

RBF network architecture			
Input Layer	# of units	Factors	2
		Covariates	11
Hidden Layer	# of hidden layers		1
	# of units		11
	Activation function		Gaussian Softmax
Output Layer	Dependent variables		1 (churn?)
	# of units		2 (yes/no)
	Activation function		Identity
	Error function		Sum of squares

The selected stopping rules that determine when to stop training the MLP neural network algorithm have been set on automatically to choose the data used for computing the prediction error as well to compute the maximum training epochs (complete data passes). When the RBF neural network algorithm is used these settings are ignored.

3.5 Evaluation

In this part both models are evaluated in order to decide if the prediction model is a success or not. The performance of both algorithms used can be visualized by using the confusion matrix. The cells on the diagonal of the cross-classification of cases are correct predictions, whilst those off the diagonal are incorrect predictions. Table 4 and Table 5 are the confusion matrices for MLP and RBF neural networks algorithms used in creating the model.

When MLP network was used to create the model, 254 of the 376 subscribers who previously churned are classified correctly. 2253 of the 2291 non-churners are classified correctly. Overall, 94% of the customers from training data set are classified correctly, and 6% are classified incorrectly as shown in Table 4. In the testing data set, 73 of the 107 subscribers who previously churned are classified correctly, and 551 of the 559 non-churners are classified correctly. Overall, 93.7% of the customers from training data set are classified correctly, and 6.7% are classified incorrectly as shown in Table 4.

Table 4. Confusion matrix for MLP algorithm

Confusion matrix MLP network				
		Predicted		
Sample	Observed	No	Yes	% correct
Training	No	2253	38	98.3%
	Yes	122	254	67.6%
	Overall %	89.1%	10.9%	**94. 0%**
Testing	No	551	8	98.6%
	Yes	34	73	68.2%
	Overall %	87.8%	12.2%	**93.7%**

Table 5. Confusion matrix for RBF algorithm

Confusion matrix RBF network				
		Predicted		
Sample	Observed	No	Yes	% correct
Training	No	2244	47	97.9%
	Yes	181	195	51.9%
	Overall %	90.9%	9.1%	**91.5%**
Testing	No	544	15	97.3%
	Yes	49	58	54.2%
	Overall %	89.0%	11.0%	**90.4%**

When RBF network was used to create the model, 195 of the 376 subscribers who previously churned are classified correctly. 2244 of the 2291 non-churners are classified correctly. Overall, 91.5% of the customers from training data set are classified correctly, and 8.5% are classified incorrectly as shown in Table 5 (has a lower performance than MLP model). In the testing data set, 58 of the 107 subscribers who previously churned are classified correctly, and 544 of the 559 non-churners are classified correctly. Overall, 90.4% of the customers from training data set are classified correctly, and 9.6% are classified incorrectly (much lower performance than MLP model) as shown in Table 5.

This suggest that, overall, our models will correctly classify 9 out of 10 subscribers. But, since our target is to identify customers that are at risk of churning, our MLP model has a percent of correct classification of 68.2% meaning that 2 out of 3 customers will be correctly classified as churners; while our RBF model a percent of 54.2%, meaning that 1 out of 2 customers will be correctly classified as churners. In our case, MLP model has a slightly better performance.

The model summaries also tell that because we have almost equal percentages of incorrect predictions in both, training and testing samples, our models haven't overtrained.

Another way of interpreting the results are the lift charts. The lift chart sorts the predicted pseudo-probabilities in descending order and display the corresponding curve. There are two types of lift charts: cumulative and incremental. The cumulative lift chart shows how better the prediction rate produced by our models is, compared to the random expectation (red line).

In Figure 7 we can see the curves corresponding to MLP and RBF models and by reading the graph on the horizontal axis, we can see that for the 20th percentile, both models have approximately a 4 lift index on the vertical axis.

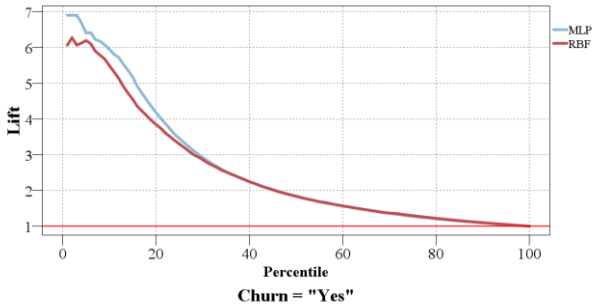

Fig. 7. Cumulative lift chart for MLP and RBF

The incremental lift chart displayed in Figure 8, shows the lift in each percentile without any accumulation. The lift lines corresponding to the MLP and RBF models (blue and respectively burgundy line) descend below the random line at about 20th percentile, meaning that compared to random expectation our models achieve their best performance in the first 20% of the records.

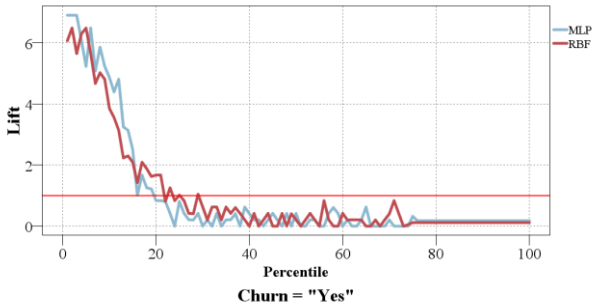

Fig. 8. Incremental lift chart for MLP and RBF

If, for instance, a mobile telecommunication company wants to send offers containing incentives to stay with them, they can easily select the top 20% subscribers in this sorted scored list, and expect to contact more than four times (lift index 4) the number of subscribers that are categorized as churners than normal.

3.6 Deployment

The deployment part of the CRISP-DM process involves organizing knowledge or information gained through data mining models in a way that stakeholders can use it in order to make decision. The deployment phase could have different outputs, from a simple report to a more complex repeatable data mining process built in a business intelligence environment. The output obtained in this paper is summarized in a report that can be handled to decision making employees in a spread sheet or document format. Complex reports integrated in dashboards can be generated if a business intelligence approach is taken into

consideration, where data used in the data mining process is extracted from data warehouse and the results are then observed by stakeholders using reporting tools.

4. CONCLUSIONS

In this paper we built two predictive models for subscribers churn in mobile telecommunication companies, using MLP and RBF neural networks. By evaluating the results from the technical point of view, we observed that for predicting non-churners, MLP model has slightly the same performance as RBF, but for predicting churners it has a better performance, 68.2 % compared to 54.2% in RBF's case. There is no need to neglect RBF neural networks, because the results are characteristic only for this data set. RBF networks can handle outliers better than an ordinary neural network, and for real-life extremely large data sets the computational performance is better and worth considering when using the neural networks approach.

From the practical point of view, these two models have a very good performance in predicting churners in a mobile telecom company. Decision making employees can build different marketing approaches to retain churners based on the predictors that have higher importance in scoring the model performance. These churn prediction models can be used in other customer response models, such as cross-selling, up-selling, or customer acquisition.

As future research we intend to study and create predictive models using other neural networks, such as Probabilistic neural networks (PNN) and General Regression neural networks (GRNN), as well as other machine learning algorithms, including Support Vector Machine, Random Forest, and Bayesian network.

REFERENCES

Ahn, J.H., Han, S.P., and Lee, Y.S. (2006). Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry. *Telecommunications Policy*, Vol. 30, Issues 10–11, pp. 552–568. Elsevier.

Ayoubi, M. (1996). *Nonlinear System Identification Based on Neural Networks with Locally Distributed Dynamics and Applications to Technical Process*. VDI Verlag. Düsseldorf.

Bishop, C.M. (2008). *Neural Networks for Pattern Recognition*. Oxford University Press. New York.

Blake C.L. and Merz, C.J. (1998). Churn data set. *UCI Repository of Machine Learning Databases*. http://www.ics.uci.edu/~mlearn/MLRepository.html. University of California, Department of Information and Computer Science, Irvine, CA.

Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., and Wang, W. (2006). Data mining curriculum: A proposal. Version 1.0.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. et al. (2000). *CRISP-DM 1.0*. Chicago, IL: SPSS.

Cremene, M. and Zahan, S. (2009). *Inteligenta Artificiala, Retele neuronale – teorie si aplicatii in telecomunicatii*. U.T. Press. Cluj-Napoca.

Fouché, G. and Langit, L. (2011). *Foundations of SQL Server 2008 R2 Business Intelligence 2nd Edition*. Apress.

García, V., Marqués, A.I., and Sánchez, J.S. (2012). Non-parametric statistical analysis of machine learning methods for credit scoring. *Advances in Intelligent Systems and Computing*, Volume 171, pp. 263-272. Springer Berlin Heidelberg.

González-Camacho, J.M., De los Campos, G., Pérez, P., Gianola, D., Cairns, J.E., Mahuku, G., Babu, R., Crossa, J. (2012). Genome-enabled prediction of genetic values using radial basis function neural networks. *Theoretical and Applied Genetics*, 125(4), pp. 759–771. Springer.

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques*. Springer-Verlag Berlin Heidelberg.

Haykin, S. O. (2009). *Neural Networks and Learning Machines, 3rd Edition*. Prentice Hall. New York.

IBM SPSS Modeler, IBM Corporation, http://www-01.ibm.com/software/analytics/spss/

Integral Solutions Limited (2007). *Clementine® 12.0, Algorithms Guide*. Chapter 10.

Richeldi, M. and Perrucci, A. (2002). Churn analysis case study. *Telecom Italia Lab.*. MiningMart, Torino.

Shaw, M., Subramaniam, C., Tan, G.W., and Welge, M.E. (2001). Knowledge management and data mining for marketing. *Decision Support Systems*, Vol. 31, no. 1, pp. 127–137. Elsevier.

Sumathi, S. and Sivanandam, S.N. (2006). *Introduction to Data Mining and its Applications*. Studies in Computational Intelligence, Vol. 29. Springer.

Verbeke, W., Martens, D., Mues, C., and Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, Vol. 38, No. 3, pp. 2354–2364. Elsevier.

Wei, C.P. and Chiu, I.T. (2002). Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems with Applications*, Vol. 23, pp. 103–112. Pergamon.

Whigham, S. (2011). *What is data mining in SQL Server 2008/R2 Analysis Services?*. www.youtube.com, min. 2:00.

Wong, B.K., Bodnovich, T.A., and Selvi, Y. (1997). Neural network applications in business: A review and analysis of the literature (1988–1995). *Decision Support Systems*, Vol. 19, pp. 301–320. Elsevier.

Zhang, T., Ramakrishnon, R., and Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 103–114. Montreal, Canada.

Particular Aspects Regarding Asynchronous Motors Controlled by Principle of Orientation by Rotor Flux

Monica-Adela Enache, Aurel Campeanu,
Sorin Enache, Ion Vlad

University of Craiova, Craiova, 200440
Romania (e-mail: menache@em.ucv.ro)

Abstract: This paper presents the results of a study regarding dynamic regimes of asynchronous motors supplied by static converters which operate by the principle of orientation by field. Two concrete cases have been considered: control by the rotor flux with the implementation of an inverter voltage source and the control by the rotor flux with the implementation of a voltage inverter controlled in current. For this, there are detailed the equations of asynchronous machine controlled by the principle of the orientation by the rotor flux and the Simulink models of the motor and of the two types of converters used. There are also presented the simulations obtained with the help of these models for the cases when the values of the rotor resistance and of the rotor inertia moment are modified. The paper ends with conclusions obtained by finalizing the research and with references.

Keywords: asynchronous motor, vector control, modelling, simulation, testing.

1. INTRODUCTION

The problem of the vectorial control of asynchronous motors is a very present one. Important international conferences (Baby et al. (2014), Ly (2013), Saman et al. (2011), Smiththisomboon et al. (2014), Yinhai et al. (2012) etc.) and the most important technical publications (Buyukdegirmenci et al (2014), Jiefeng et al. (2013), Niguchi et al (2013), Terras et al (2013), Xiaodong et al (2014) etc.) provide wide spaces to this variant of control.

Because of the fact that in case of asynchronous machine we do not have a direct and controlled access to the electromagnetic torque, it does not have an adequate behaviour in dynamic regime (economic fast speed variation between large limits or, on the contrary, keeping it constant irrespective of the torque conditions, fast damping of the afferent transient electromechanic processes).

In order to eliminate this drawback the method of the orientation by field has been introduced; in essence, it consists in establishing a convenient electric angle between the representative phasors of the stator current, considered as command quantity, respectively of the rotating fluxes linking the stator winding, the rotor winding or in the air-gap. The modification of this angle is obtained by means of the stator current components defined on the flux direction, respectively orthogonal to it.

2. EQUATIONS OF ASYNCHRONOUS MACHINE

In the following there will be used the equations of squirrel cage asynchronous machine, written in a reference frame rotating with the angular speed ω_B (Campeanu et al. (2012)).

$$\begin{aligned}
\underline{u}_s &= R_s \underline{i}_s + \frac{d}{dt}\underline{\Psi}_s + j\omega_B \underline{\Psi}_s \\
0 &= R_r' \underline{i}_r' + \frac{d}{dt}\underline{\Psi}_r' + j(\omega_B - \omega)\underline{\Psi}_r' \\
\underline{\Psi}_s &= L_s \underline{i}_s + L_{sh} \underline{i}_r' \\
\underline{\Psi}_r' &= L_r' \underline{i}_r' + L_{sh} \underline{i}_s \\
\underline{\Psi}_{sh} &= \underline{\Psi}_{rh}' = L_{sh}(\underline{i}_s + \underline{i}_r') = L_{sh}\underline{i}_m \\
m &= \frac{3}{2} p \underline{\Psi}_s \times \underline{i}_s = \frac{3}{2} p \frac{L_{sh}}{L_r'} \underline{\Psi}_r' \times \underline{i}_s \\
m - m_r &= \frac{J}{p}\frac{d\omega}{dt}
\end{aligned} \tag{1}$$

For a convenient analysis of the processes, the d-axis of the coordinates system, rotating with the speed ω_B, is overlapped to the axis of the flux phasor $\underline{\Psi}_i$ considered (i = s, r, sh), which rotates with the angular speed $\omega\Psi_i$ (defined in literature as coordinates system orientated by field).

Among the possibilities of orientation by field, the orientation by the rotor flux has proved as being the most convenient one and that is what we approach in the following.

In this situation, $\underline{\Psi}_i = \underline{\Psi}_r'$, respectively $\omega_B = \omega_{\Psi_r'}$ (see figure 1).

Replacing now $\underline{i}_s$, obtained from the third equation of the previous system, in the computation relation of $\underline{u}_s$, it is obtained:

$$\underline{u}_s = \frac{d}{dt}\underline{\Psi}_s + (\frac{R_s}{L_s} + j\omega_B)\underline{\Psi}_s - \frac{R_s}{L_s}L_{sh}\underline{i}_r^{/} \qquad (2)$$

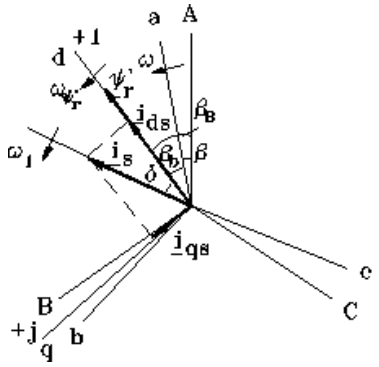

Fig. 1. Representation of the phasors $\underline{i}_s$ and $\underline{\Psi}_r^{/}$ in the complex plane.

Proceeding in a similar manner with equations 2 and 4 of the same system, it results:

$$0 = \frac{d}{dt}\underline{\Psi}_r^{/} + [\frac{R_r^{/}}{L_r^{/}} + j(\omega_{\Psi_r^{/}} - \omega)\underline{\Psi}_r^{/} - \frac{R_r^{/}}{L_r^{/}}L_{sh}\underline{i}_s \qquad (3)$$

Because $\underline{\Psi}_r^{/}$ is overlapped to the d-axis, it may be written:

$\underline{\Psi}_r^{/} = \Psi_r^{/} = \Psi_{dr}^{/}$ and respectively, $\Psi_{qr}^{/} = 0$.

Moreover, the time constant of the rotor will be noted with $T_r = L_r^{/} / R_r^{/}$.

From the previous figure it results:

$$\underline{i}_s = i_{ds} + i_{qs} = i_s e^{j\delta} \qquad (4)$$

With this, the projections of the relation (3) on the two axes become:

$$\frac{d}{dt}\Psi_r^{/} + \frac{1}{T_r}\Psi_r^{/} = \frac{L_{sh}}{T_r}i_{ds} \qquad (5)$$

$$(\omega_{\Psi_r^{/}} - \omega)\Psi_r^{/} = \frac{L_{sh}}{T_r}i_{qs} \qquad (6)$$

From the second equation of system (1), by projecting on the two axes it results:

$$i_{dr}^{/} = -\frac{1}{R_r^{/}}\frac{d}{dt}\Psi_r^{/} \qquad (7)$$

$$i_{qr}^{/} = -\frac{1}{R_r^{/}}(\omega_{\Psi_r^{/}} - \omega)\Psi_r^{/}$$

Replacing $\underline{\Psi}_r^{/}$, given by (6), in (7), it is obtained:

$$i_{qr}^{/} = -\frac{1}{1+\sigma_2}i_{qs} \qquad (8)$$

With the help of the relation 6 of the system (1) it is obtained successively:

$$m = \frac{3}{2}P\frac{L_{sh}}{L_r^{/}}\Psi_r^{/}i_s\sin\delta = \frac{3}{2}P\frac{1}{1+\sigma_2}\Psi_r^{/}i_{qs} \qquad (9)$$

$$m = \frac{3}{2}P\Psi_r^{/2}(\omega_{\Psi_r^{/}} - \omega)\frac{1}{R_r^{/}} \qquad (10)$$

It its turn, the rotor flux may be written as:

$$\Psi_r^{/} = L_{sh}[\underline{i}_s + (1+\sigma_2)\underline{i}_r^{/}] = L_{sh}(\underline{i}_m + \sigma_2\underline{i}_r^{/}) = L_{sh}\underline{i}_{mr} \qquad (11)$$

where $\underline{i}_{mr}$ is the representative phasor of the modified magnetization current (of the rotor).

In the case when $L_{sh} = ct.$, taking (11) into account, the equations (5), (6) and (9) become:

$$\begin{aligned} &\frac{di_{mr}}{dt} + \frac{1}{T_r}i_{mr} = \frac{1}{T_r}i_{ds} \\ &(\omega_{\Psi_r^{/}} - \omega)i_{mr} = \frac{1}{T_r}i_{qs} \\ &m = \frac{3}{2}pL_{sh}\frac{1}{1+\sigma_2}i_{mr}i_{qs} \end{aligned} \qquad (12)$$

From the previous relations, a few interesting conclusions result. Thus, the flux $\Psi_r^{/}$ is dependent on the component i_{ds} of the stator current.

When $\Psi_r^{/} = ct.$, it is obtained that $i_{dr}^{/} = 0$, which leads to the conclusion that the two phasors are orthogonal.

The electromagnetic torque can be modified by means of $\Psi_r^{/}$ and i_{qs}.

If $\Psi_r^{/}$ is variable, because the dynamic process is conditioned by the time constant T_r, the electromagnetic torque is slowly modified. But a high value of T_r has the advantage that it blots the effect of possible harmonics overlapped to the continuous component i_{ds} on $\Psi_r^{/}$ (and implicitly of m).

If $\Psi_r^{/}$ is kept constant and i_{qs} is modified, the electromagnetic torque varies practically instantaneously, without override. In conclusion, by means of i_{ds} the rotor flux is set, therefore the level of stress and with the help of i_{qs} the torque is modified. Proceeding this way a decoupling of the circuits is made, similar to that of direct

current machine. The currents i_{ds} and i_{qs} are the currents corresponding to some hypothetical rotating orthogonal stator windings, identical with the phase windings of the machine, which have a position fixed by the electrical angle β_B in any moment.

3. MOTOR MODELLING

The motor model has been carried out with the help of the equations obtained in the case when the saturation was neglected. For implementing it the Matlab-Simulink program has been used (Matlab users guide (2010)).

In these conditions the block scheme of this model is that presented in figure 2.

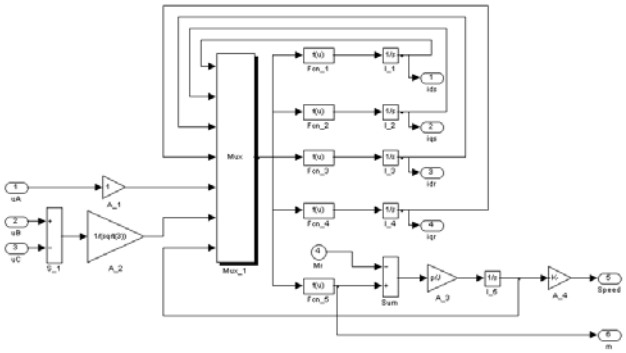

Fig. 2.Simulink model of the asynchronous motor.

4. CONVERTERS MODELLING

In the frame of this analysis two situations of converters have been considered:
- case of control by the rotor flux with voltage inverter controlled in voltage;
- case of control by the rotor flux with voltage inverter controlled in current.

Simulink schemes of the simulation programs are detailed in figures 3 and 4 (Enache et al. (2009)).

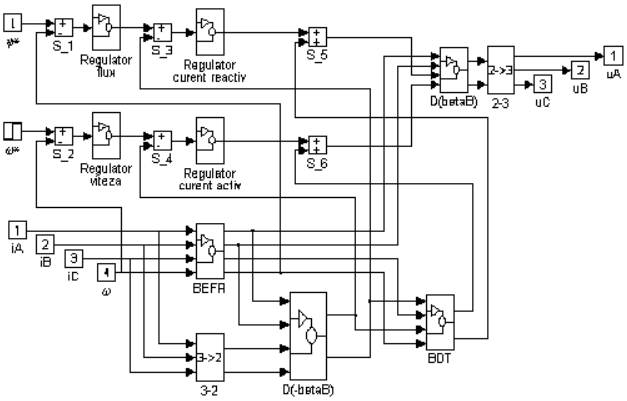

Fig. 3. Simulink model of the converter in case of control by the rotor flux with voltage inverter controlled in voltage.

For the case of the model from figure 4 a PWM inverter with sinusoidal currents has been modelled; its structure is depicted in figure 5.

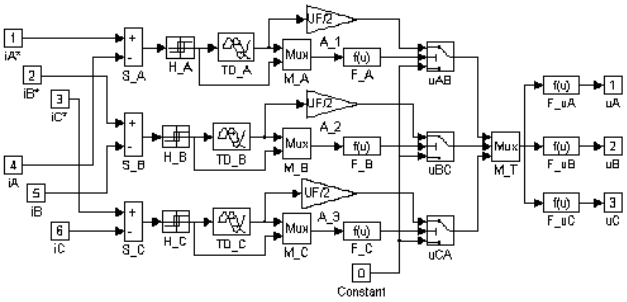

Fig. 4. Simulink model of the converter in case of control by the rotor flux with voltage inverter controlled in current.

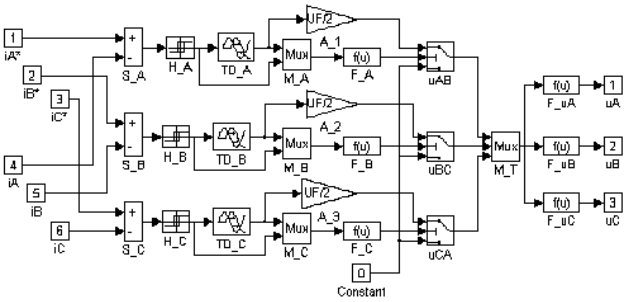

Fig. 5. Structure of the PWM inverter with sinusoidal currents.

5. SIMULATIONS

With the help of the program conceived, for the case of the control by the rotor flux with voltage inverter controlled in voltage, there have been simulated a few particular situations:
- the dynamic regime of the phase current when the value of the rotor resistance is modified;
- the same dynamic regime for the case when the value of the inertia moment is modified.

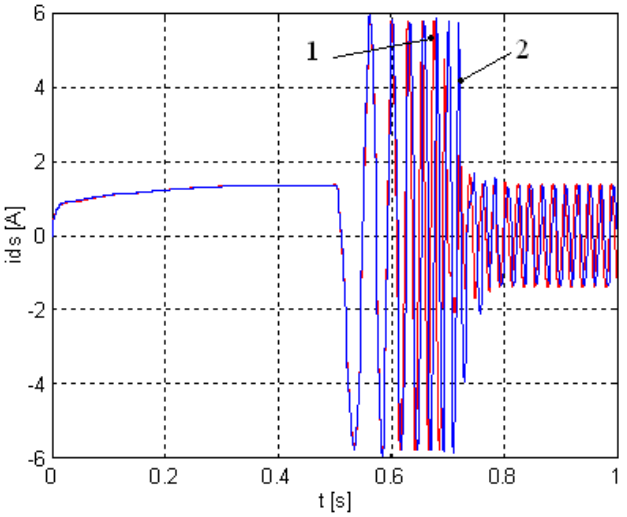

Fig. 6. Variations of current obtained for R'_r=5,5 Ω (curve 1) and R'_r=5 Ω (curve 2).

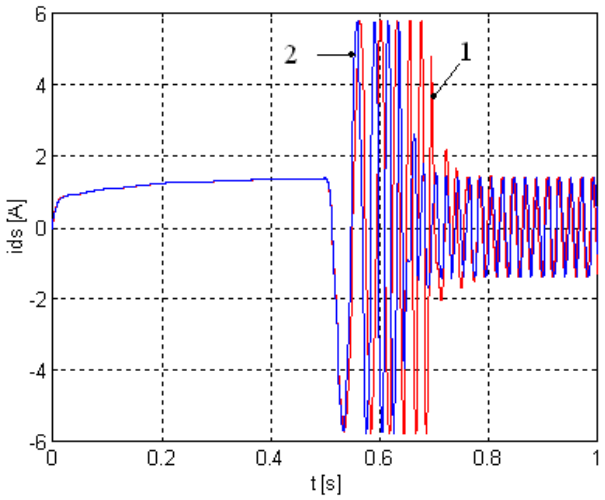

Fig. 7. Variations of current obtained for J=0,014 kg m^2 (curve 1) and J=0,01 kg m^2 (curve 2).

The analysis of these graphics leads to the following conclusions:
- the decrease of the rotor resistance causes the increase of the time of dynamic regime stabilization;
- the increase of the inertia moment leads to the increase of the stabilization time.

For the case of the control by the rotor flux with voltage inverter controlled in current, two particular situations have been simulated:
- the dynamic regime of the phase current when the value of the rotor resistance is modified (Fig. 8);
- the same dynamic regime for the case when the value of the inertia moment is modified (Fig. 9).

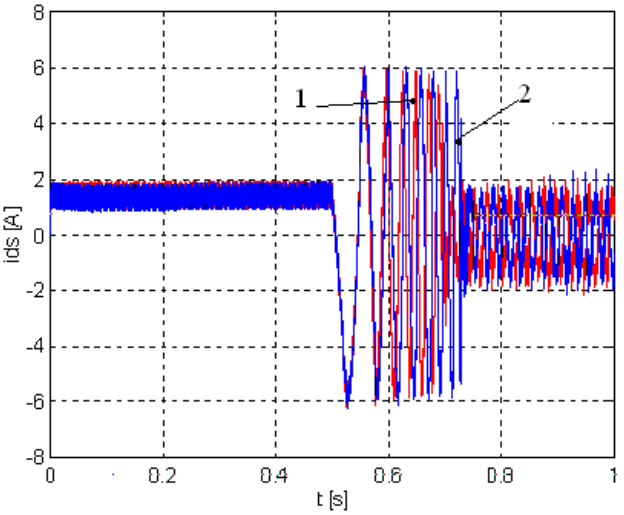

Fig. 8. Variations of current obtained for R'_r=5,5 Ω (curve 1) and R'_r=5 Ω (curve 2).

The graphics from figures 8 and 9 confirm the validity of the conclusions obtained in case of control by the rotor flux with voltage inverter controlled in voltage.

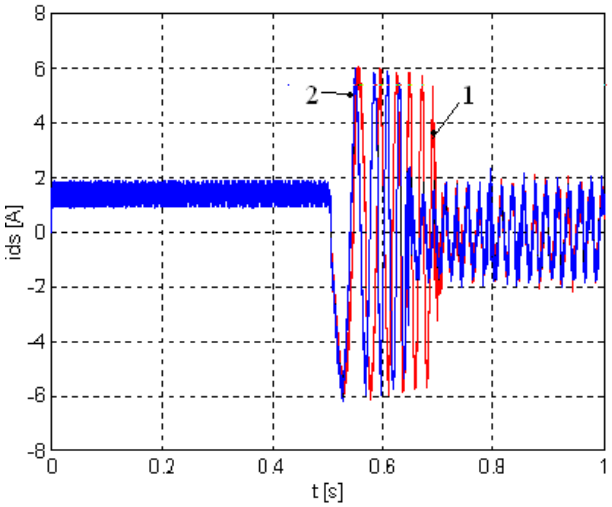

Fig. 9. Variations of current obtained for J=0,014 kg m^2 (curve 1) and J=0,01 kg m^2 (curve 2).

Observation
A similar analysis may also be carried out for the situations when other parameters of the motor are modified (the stator resistance, the stator inductivity, the rotor inductivity etc.). But it found that these parameters do not influence significantly the duration of the transient analysis analyzed.

6. EXPERIMENTAL VERIFICATIONS

In order to verify experimentally the previous characteristics the circuit detailed in figure 10 has been carried out.
The central element of the circuit is the data acquisition board KPCI 3102 which ensures a sampling frequency of 100 kS/sec (Enache et al. (2007)).

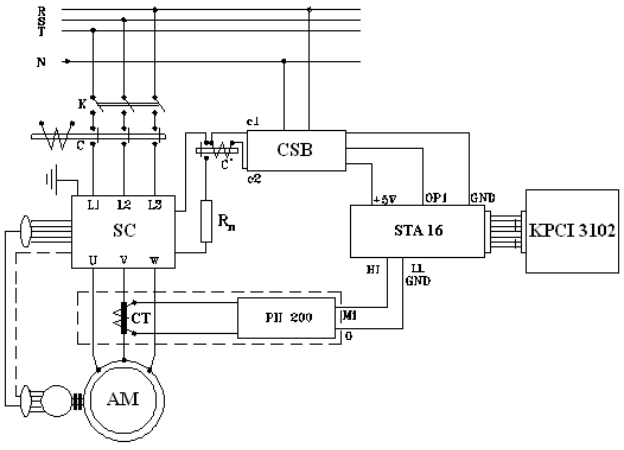

Fig. 10. Experimental circuit for monitoring the dynamic regime of current.

The notations used in figure 10 have the following significance:
- AM asynchronous motor;
- SC static converter;
- KPCI 3102 acquisition board;

- CT current transformer;
- CSB command and synchronization block;
- STA 16 connection block;
- PII 200 adaption and protection block.

The rated data of the asynchronous motor are:
- rated power P_N=1,1 kW;
- rated voltage U_{1N} =220 / 380 V;
- rated current I_{1N}= 5,02 / 2,9 A;
- rated frequency f_{1N}=50 Hz;
- rated speed n_N=1405 r.p.m;
- rated power factor cos φ_N=0,78;
- squirrel cage rotor;
-stator winding connection: star;
- protection degree IP 44.

The previous circuit has been used for acquiring the phase current of the motor during the variation by saltus of the reference quantity of the converter (Fig. 11). In this case a converter with voltage inverter controlled in current.

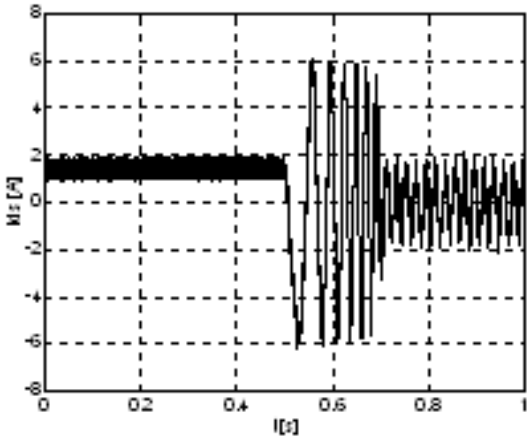

Fig. 11. Experimental variation of the A phase current.

Comparing this variation with the characteristic 1 from figure 8 it is noticed that the experiment validates the simulations quantitatively and qualitatively.

7. CONCLUSIONS

This paper aimed at analyzing the influences of certain parameters of motor on the operation in dynamic regime for the cases when supplying by converters with orientation by the rotor flux.

The conclusions resulted from this analysis are:

a) for the case of implementation of a voltage inverter controlled in voltage:

- the decrease of the rotor resistance causes the increase of the stabilization time of dynamic regime;
- the increase of the inertia moment leads to the increase of the stabilization time.

b) for the case of implementation of a voltage inverter controlled in current:
- the conclusions are generally the same as in the previous case;
- the rotor resistance influences in a less degree than in the previous case the duration of the transient process analyzed.

ACKNOWLEDGMENT

This work was supported by the strategic grant POSDRU/159/1.5/S/133255, Project ID 133255 (2014), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007 – 2013.

REFERENCES

Baby, B., Shajilal, A.S. (2014). An improved indirect vector controlled current source inverter fed induction motor drive with rotor resistance adaptation. 2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), DOI: 10.1109/ AICERA. 2014. 6908258, pp. 1 – 6.

Buyukdegirmenci, V.T., Bazzi, A.M., Krein, P.T. (2014). Evaluation of Induction and Permanent-Magnet Synchronous Machines Using Drive-Cycle Energy and Loss Minimization in Traction Applications. IEEE Transactions on Industry Applications, Vol. 50, Issue 1, DOI: 10.1109/TIA.2013.2266352, pp. 395 – 403.

Campeanu, A., Cautil, I., Vlad, I., Enache, S. (2012). Modelling and simulation for alternating current electrical machines. Romanian Academy Publishing House, Bucharest, ISBN 978-973-27-2233-6.

Enache, S., Campeanu, A., Vlad, I., Enache, M.A.(2009). Considerations Regarding Asynchronous Motors Control by the Stator Flux. Proceedings of 3rd International Symposium on Electrical Engineering and Energy Converters, Suceava, 24-25 September 2009, ISSN 2066-853X, pp. 185-188.

Enache, S., Vlad, I. (2007). Test Bench for Reluctance Synchronous Motors. Analele Universităţii din Craiova, Seria: Inginerie electrică, Anul 31, nr. 31, Vol.II, 2007, ISSN 1842-4805, p. 122-125.

Jiefeng Hu, Jianguo Zhu, Yongchang Zhang, Platt, G., Qishuang Ma, Dorrell, D.G. (2013). Predictive Direct Virtual Torque and Power Control of Doubly Fed Induction Generators for Fast and Smooth Grid Synchronization and Flexible Power Regulation. IEEE Transactions on Power Electronics, Vol. 28, Issue 7, DOI: 10.1109/TPEL.2012.2219321, pp. 3182 – 3194.

Lv Ming (2013). Mathematical Modeling and Simulation Analysis on Speed Control System of Asynchronous Motor Based on Vector Frequency Conversion Technology. 2013 Fourth International Conference on Digital Manufacturing and Automation (ICDMA), DOI: 10.1109/ICDMA.2013.183, pp. 770 – 772.

Niguchi, N., Hirata, K. (2013). Torque-Speed Characteristics Analysis of a Magnetic-Geared Motor Using Finite Element Method Coupled With Vector Control. IEEE Transactions on Magnetics, Vol. 49, Issue 5, DOI: 10.1109/TMAG.2013.2239271, pp. 2401 – 2404.

Saman, M., Karaköse, M., Akin, E. (2011). Grey system based on flux estimation for vector control of asynchronous motors, 2011 IEEE 19th Conference on Signal Processing and Communications Applications (SIU), pp. 666 – 669.

Smiththisomboon, S., Suwankawin, S. (2014). A stator-equation-based reduced-order observer for position-sensorless vector control system of doubly-fed induction machines. 2014 International Power Electronics Conference (IPEC-Hiroshima 2014 - ECCE-ASIA), pp. 3885 – 3892.

Terras, T., Hadjeri, S., Mezouar, A., Chikouche, T.M. (2013). Robust speed control with rotor resistance estimation. Canadian Journal of Electrical and Computer Engineering, Vol. 36, Issue 2, pp. 43 – 51.

Xiaodong Shi, Krishnamurthy, M. (2014). Survivable Operation of Induction Machine Drives With Smooth Transition Strategy for EV Applications. IEEE Journal of Emerging and Selected Topics in Power Electronics, Vol. 2, Issue 3, DOI: 10.1109/ JESTPE.2014.2303488, pp. 609 – 617.

Yinhai, Zhang, Zhikui, Chen (2012). An Asynchronous Motor Vector Control System Considering the Variation of Rotor Resistance. 2012 Third International Conference on Networking and Distributed Computing (ICNDC), DOI: 10.1109/ ICNDC.2012.41, pp. 140 – 143.

*** (2010). Matlab users Guide. Mathworks.

Fault Detection in Superheater Using dSpace and Electronic Simulator

Camelia Maican

Automation and Electronics Department, University of Craiova, Romania
(e-mail: camelia@automation.ucv.ro)

Abstract: In this paper is study the actuator's faults detection at the steam superheaters using an electronic simulator and data acquisition board dSpace. The electronic simulator reproduces the real function of the superheater in normal condition and in fault condition. The possibility of the fault exists - blocking of the actuator on the flow injection of the second injector by a switch. In case of fault, the value of the second flow injection is set in a potentiometer. This command can be used to study the function in the manual regime of the simulator.

Keywords: fault detection, electronic simulator, steam temperature, steam superheater, data acquisition.

1. INTRODUCTION

The steam temperature must be constant before steam hits the tips of turbine's blades. In this conditions, for steam temperature control, the superheater is divided in three parts, in every points of connection are mounted the devices that allow injection of condensate for cooling steam (Iancu & Vinatoru 2003). The control of the output temperature is difficult because there is a transfer time delay between the points where the water is sprayed and the points where steam temperature is measured.

The paper proposes to use an electronic simulator to reproduce the real function of the superheater in normal conditions and in fault conditions. It is necessary to establish a correspondence between the real temperature inside the superheater and the voltage in equivalent point of the superheater's electronic simulation. With this electronic simulator we can study the possibility to fault detection of the actuators.

For these experiments there were used the data acquisition board dSpace, the data acquisition board Cassy lab, an electronic simulator for the three superheaters and Matlab/ Simulink.

Real-Time Interface (RTI) is the link between dSPACE hardware and the development software MATLAB/ Simulink/ Stateflow from the MathWorks. It extends Simulink® Coder™ for the seamless, automatic implementation of ours Simulink and Stateflow models on the real-time hardware. The implementation time is greatly reduced. The hardware configuration for the real-time application is guided by automatic consistency checks to prevent parameterization errors.

To connect ours model to a dSPACE I/O board, just drag & drop the I/O module from the RTI block library and then connect it to the Simulink blocks (RTI and RTI-MP 2007, Automation Guide For Release 5.3, ControlDesk, 2007).

The entire build process for a real-time application is handled by ConfigurationDesk. ConfigurationDesk is an intuitive, graphical configuration and implementation tool. With Configuration Desk, it is easy to implement the behavior model code (from MATLAB/Simulink/Simulink Coder) and the I/O function code (from ConfigurationDesk) on the dSPACE SCALEXIO hardware (ControlDesk Experiment Guide 2007).

2. DESIGN OF THE SIMULATOR'S ELECTRONIC CIRCUIT

The mathematical model of the steam superheater is obtained using equations for mass and heat transfer balance for each heat exchanger and injectors.

SA1: $$T_{ta1}\frac{dT_{a1}}{dt} = (a_{a1i}T_{a1i} - a_{a1}T_{a1})\cdot F_{a1} + b_{a1}(T_{g3} - T_{a1}) \quad (1)$$

SA2: $$T_{ta2}\frac{dT_{a2}}{dt} = (a_{a2i}T_{a2i} - a_{a2}T_{a2})\cdot F_{a2} + b_{a2}(T_{g1} - T_{a2}) \quad (2)$$

SA3: $$T_{ta3}\frac{dT_{a3}}{dt} = (a_{a3i}T_{a3i} - a_{a3}T_{a3})\cdot F_{a3} + b_{a3}(T_{g2} - T_{a3}) \quad (3)$$

SG1: $$T_{tg1}\frac{dT_{g1}}{dt} = (a_{g1i}T_{g1i} - a_{g1}T_{g1})\cdot F_{g1} + b_{g1}(T_{g1} - T_{a2}) \quad (4)$$

SG2: $$T_{tg2}\frac{dT_{g2}}{dt} = (a_{g2i}T_{g2i} - a_{g2}T_{g2})\cdot F_{g2} + b_{g2}(T_{g2} - T_{a3}) \quad (5)$$

SG3: $$T_{tg3}\frac{dT_{g3}}{dt} = (a_{g3i}T_{g3i} - a_{g3}T_{g3})\cdot F_{g3} + b_{g3}(T_{g3} - T_{a1}) \quad (6)$$

$$T_{Fa1}\frac{dF_{a1}}{dt} = -F_{a1} + F_{a2} - W_{inj1} \quad (7)$$

$$T_{Fa2}\frac{dF_{a2}}{dt} = -F_{a2} + F_{a3} - W_{inj2} \quad (8)$$

$$T_{Fa3}\frac{dF_{a3}}{dt} = -F_{a3} + F_T \quad (9)$$

$$T_{a2i}=T_{a1}-1.97034*K_f W_{inj1}$$

$$T_{a3i}=T_{a2}-1.97034*K_2 W_{inj2} \quad (10)$$

$$T_{a1i}=T_{ai}=618,15K; \qquad T_{gi}=1183,15K$$

Where: $T_{tax} = \frac{\rho_{ax} c_{ax} V_{ax}}{F_T c_T}$; $T_{tgx} = \frac{\rho_{gx} c_{gx} V_{gx}}{F_g \rho_{gi} c_{gi}}$; $x = 1, 2, 3;$

And a_{ax}, a_{gx} and b_{gx} are constants determined from the boiler's data (Maican et al. 2010).

Each module of the electronic simulator corresponds to the equation 1÷6 for the steam and gas zones of the superheaters and 7÷9 for the injection blocks.

Table 1 presents the real temperature in the points of the superheater for steam (T_a) and gas (T_g) and the correspondence with equivalent voltage in the correspondence point of the electronic simulation.

We consider the temperature variation in area 0^0C-1000^0C and the equivalent voltage area 0-10V of the electronic simulation modules with a conversion factor of 1/100 [V/ ^{0}C] (Maican et al. 2010).

Thus out of the drum, the steam has a temperature of 345°C out of the first superheater, 396°C out of the second superheater 453°C and out of the third superheater and at input has into turbine has 550°C. The same thing can be said about the gas, whose temperature varies along the three sections, as seen in Table 1.

To enter the delays that occur during the overheating of the steam and gas corresponding to the (1-9) an RC circuit assembly was used.

The operational amplifiers have the advantage of performing the mathematical operations such as assembly, integrals or differentials computing, by simply changing the various components on the feedback part (resistances and condensers).

Each differential equation of the mathematical model was simulated with an electronic circuit consisting of a RC group and an operational amplifier in summation assembly, with direct or reverse action according to the sign terms of the equations.

The simulator reproduces the same dynamics of the real process.

In order to be able to use the electronic simulator from the heat of the steam superheateres the configuration presented in Fig. 1 was accomplished.

The scheme has provided a mounting number of clamps that allow the connection of the measuring instruments or the ports of entry and exit of the data acquisition to an electronic simulator.

On the board of the simulator, on the right of each terminal of clamp string, process variables are passed: measurable variables; the temperature gas out of each segment analyzed namely T_{g1}, T_{g2}, T_{g3} and the temperatures steam T_{a1}, T_{a2}, T_{a3}; the command variables corresponding to order flow injection W_{inj1} and W_{inj2}; the principle disturbances of the process T_{gi} – the temperature gases in the area of the furnace and T_{ai} – the steam temperature in the area vaporizations [3]. These perturbations may be locally amended by of the potentiometers P1 and P2 or from computer via signals applied to the terminals T_{ai} and T_{gi} of the string clamps (Fig. 1). Switch K1-K2 on the board assembly allows the selection of the two situations: C - computers, M – manually (Maican et al. 2010).

Table 1. The temperature of the steam and exhaust gas

	Section			
Temp.	input SA1	output SA1	output SA2	output SA3
T_a [°C]	345	396	453	550
T_g [°C]	910	847	769	547
Voltage				
U_a [V]	3,45	3,96	4,53	5,5
U_g [V]	9,1	8,47	7,69	5,47

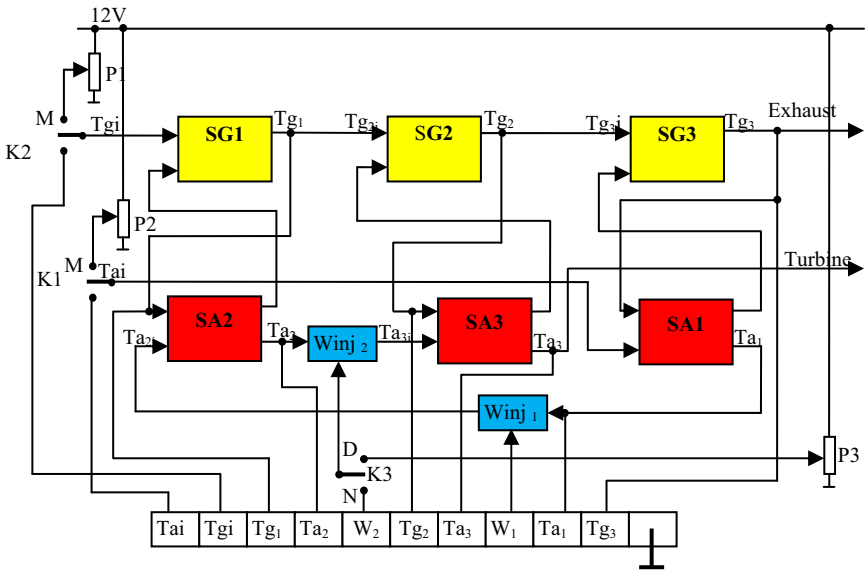

Fig. 1. Block scheme of the electronic simulator

The possibility of the fault exists - blocking of the actuator on the flow injection of injector W_{inj2} by the switch K3. In case of fault, the value of the flow injection W_{inj2} is set in potentiometer P3. This command can be used to study the function in the manual regime of the simulator.

On this simulator, a final test of the stationing values voltages has been conducted in various points of the electronic circuit corresponding to scale 1/100 of the temperatures in various points of the whole real steam superheateres from Isalnita Power Station boilers.

These final tests on the simulator have been achieved in parallel with tests on the simulation models used in the design phase and making of the electronic circuit.

The simulations were made in stationary regime.

The tests used to the layout of a stabilizing source of continue voltage with the following values: V1=10V - input steam and gases; V2=12V - supplying operational amplifier; V3=5V - input injectors W_{inj1} and W_{inj2}.

3. THE EXPERIMENTAL SIMULATION

To realize the simulation it use the electrical diagram of the electronic simulator presented in Fig. 2 and the data acquisition board dSpace. The steam temperatures T_{a2} and T_{a3} are connected to the input of the data acquisition board dSpace *ADC5* and *ADC6*. The command variables corresponding to injection flow W_{inj1} and W_{inj2} are connected to the output of the data acquisition board dSpace *DAC1* respectively *DAC2*.

For applications with dSPACE 1104 system, was used Matlab Simulink to implement the block diagram of Fig. 3, which used a PI control law. The scheme contains: 2 signal generators with frequency and amplitude adjustable, 2 input constants (T^*_{a2}=4.53 and T^*_{a3}=5.5), 2 input channels of the dSPACE board (*ADC5, ADC6*) and 2 output channels of the dSPACE board (*DAC1, DAC2*). Once you have made the appropriate settings for dSPACE 1104 board, was opened ControlDesk, were added plotters to viewing and were obtained the experimental results as we can see in Fig. 4..

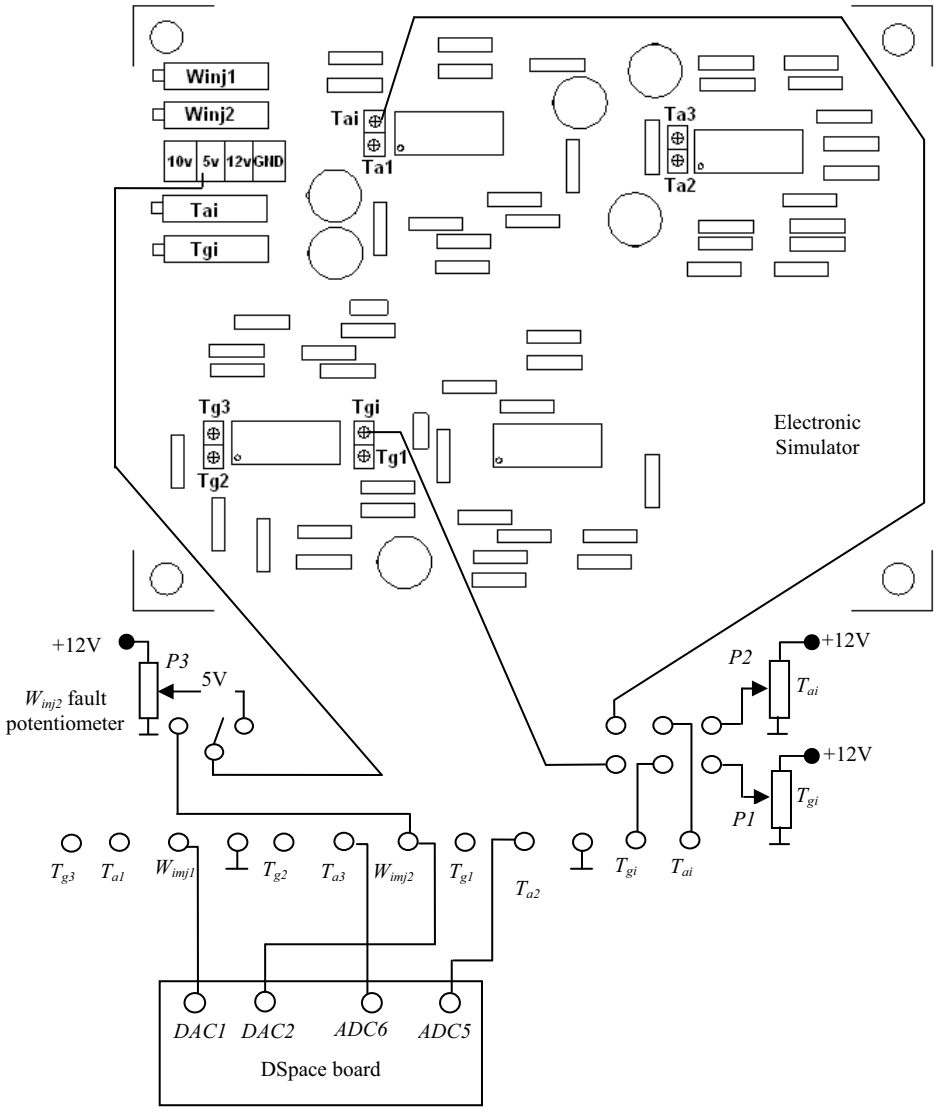

Fig. 2. Electrical diagram of the electronic simulator and the data acquisition board dSpace

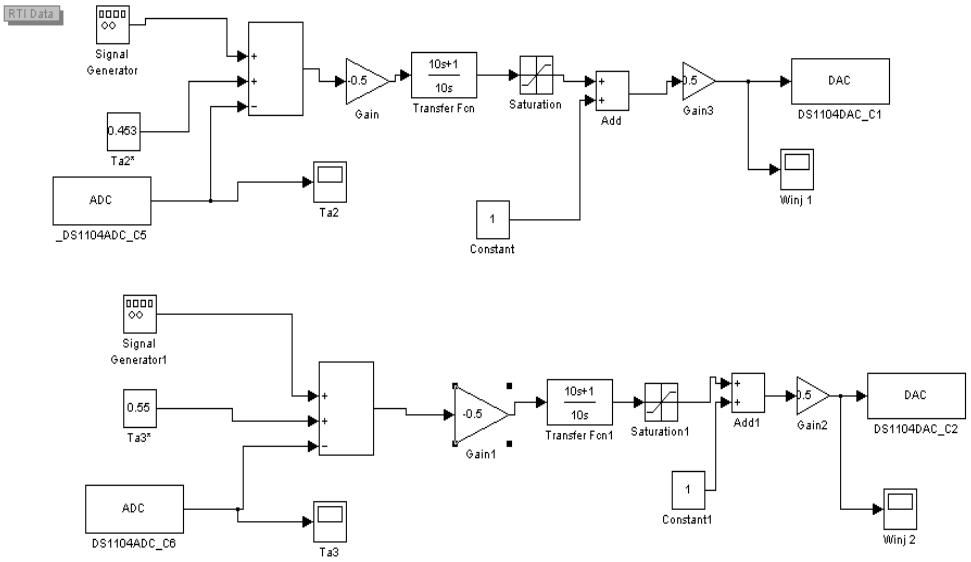

Fig. 3. Simulink model

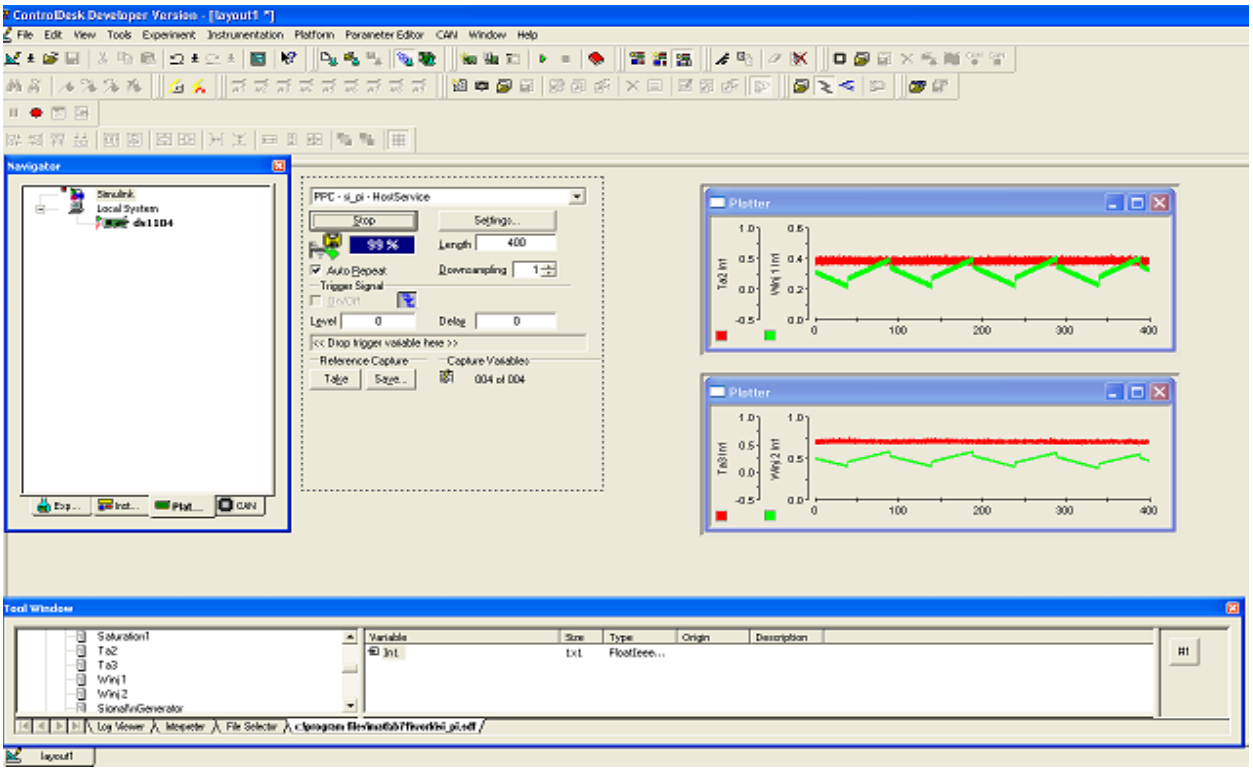

Fig. 4. Graphical configuration in ControlDesck

The time constant of the electronic models are very high and the display system does not allow full view of the response due to the memory buffer allocated by the program to display in real time. Increasing the memory buffer (runtime of the chart) leads to blocking program. Would need a PC with a processor more powerful to graphics processing.

It can make a trick by increasing the sampling period for data acquisition, but this increases the conversion noise of CAN, as noted on the graph as a acquisition noise (the display shows a band of values instead of a continuous curve). To eliminate this noise can introduce low pass filters (LPF) in the control structure of the simulator, made in Simulink, but these would require additional processing time, loading more the processor and thus no longer fulfil the functioning condition in real time. In this conditions, to eliminate the acquisition noise we use the block oriented simulation hardware BORIS and the hardware interface (CASSY I/O). For this, the data acquisition board CASSY is connected to the terminals corresponding to the steam temperatureT_{a2} and T_{a3} of the electronic simulator. Using the block scheme presented in Fig. 5, we obtain the filtered signal for the steam temperature T_{a2} and T_{a3}.

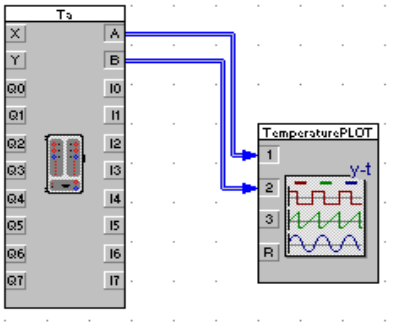

Fig. 5. Block scheme using BORIS

The next figures present some experimental results related to electronic simulator response at the control commands W_{inj1} and W_{inj2} in normal conditions and in fault conditions.

In Fig 6 it is shown the response of the steam temperature T_{a2} and T_{a3} at the variation of the control commands W_{inj1} and W_{inj2}. The controller parameters are K_p=-0.5 and T_I=10s. The filtered signal for the steam temperature T_{a2} and T_{a3} using BORIS software is presented in Fig. 7, where it can be seen that the fault occurred at the actuator corresponding to injection flow W_{inj2}, was detected.

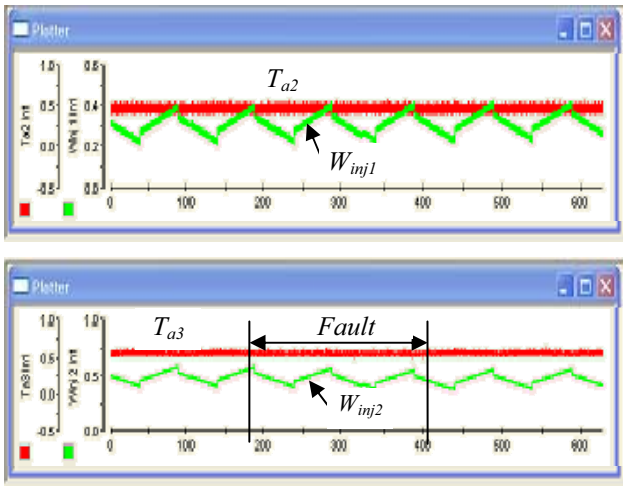

Fig. 6. dSpace graphics for the steam temperature and the injection flow (K_p=-0.5; T_I=10s)

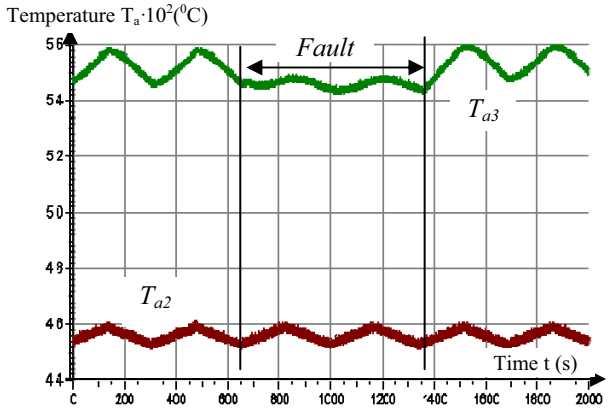

Fig. 7. BORIS graphics for the steam temperature

In both case it can be observed that the injection flow influence the steam temperature at the output of each superheater, meaning that when W_{inj1} and W_{inj2} increase, then T_{a2} respectively T_{a3} decrease and when W_{inj1} and W_{inj2} decrease, then increase T_{a2} respectively T_{a3}.

As a conclusion we can say that for specific case, the technological system represented by the electronic simulator responds to the command given by regulators, represented by W_{inj1} and W_{inj2}.

In Fig. 8 we consider the same parameters for controllers, but the fault appears in another time interval. The filtered signal for the steam temperature T_{a2} and T_{a3} using BORIS software is presented in Fig. 9 and in this case it can observed that the fault occurred at the actuator corresponding to injection flow W_{inj2}, was detected.

For the controller parameters K_p=-0.5 and T_I=5s, we have the steam temperature T_{a2} , T_{a3} and the control commands W_{inj1}, W_{inj2} represented in Fig. 10. In this case, the fault occurs from the initial moment until t=500s and the filtered signal is shown in Fig. 11.

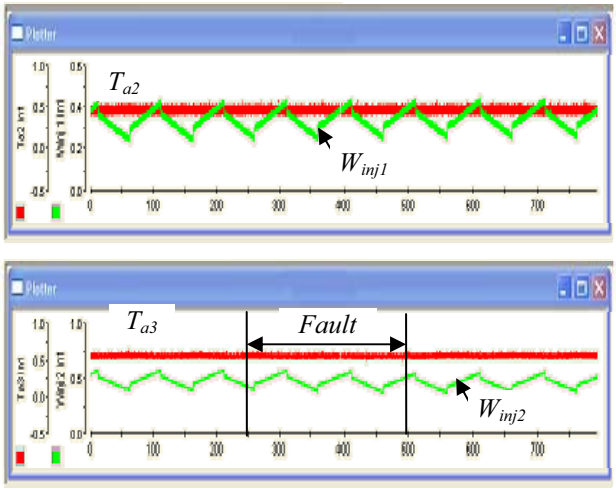

Fig. 8. dSpace graphics for the steam temperature and the the injection flow

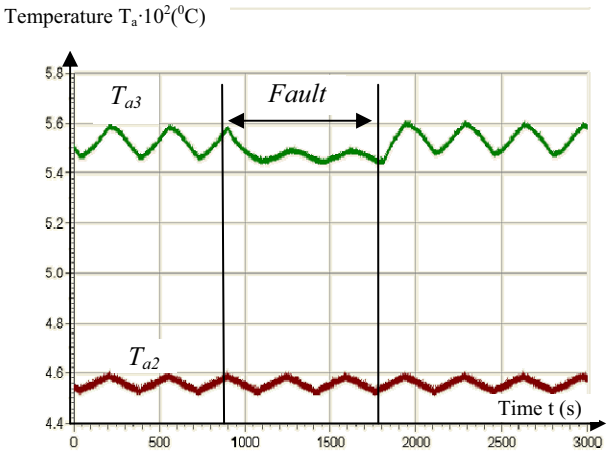

Fig. 9. BORIS graphics for the steam temperature (the fault occurs from t=900s÷1800s)

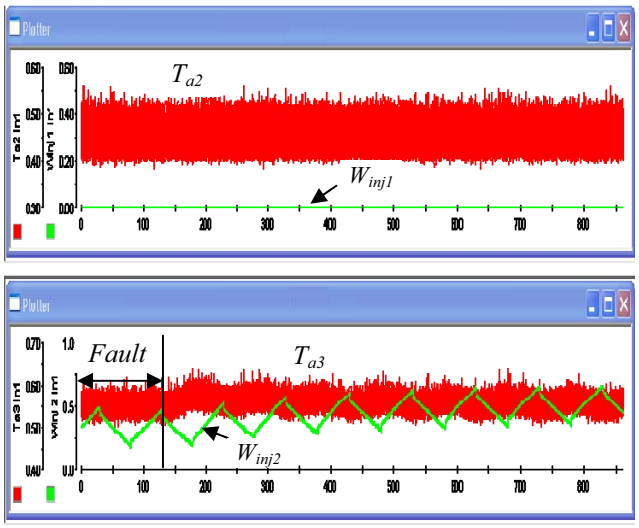

Fig. 10. dSpace graphics for the steam temperature and the injection flow (K_p=-0.5; T_I=5s)

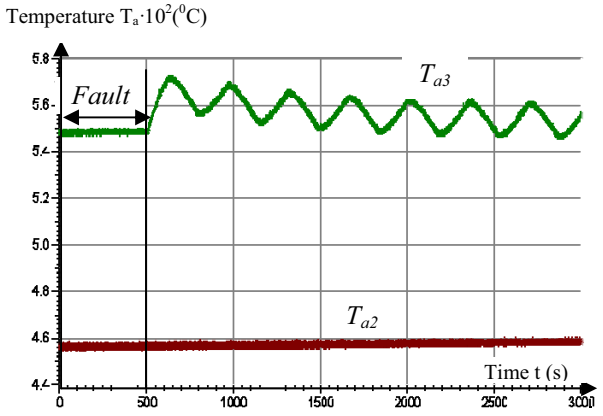

Fig. 11. BORIS graphics for the steam temperature (K_p=-0.5; T_I=5s)

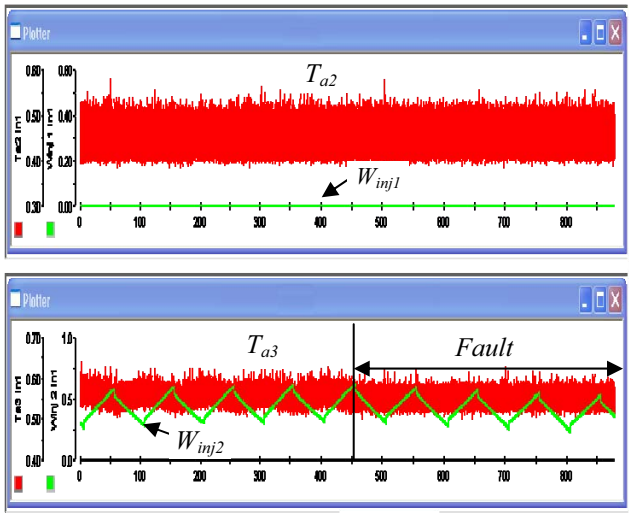

Fig. 12. dSpace graphics for the steam temperature and the injection flow

In Fig. 12 we consider the same parameters for controllers (K_p=-0.5 and T_I=5s), but the fault occurs at the time t=1600s. The filtered signal is shown in Fig. 13.

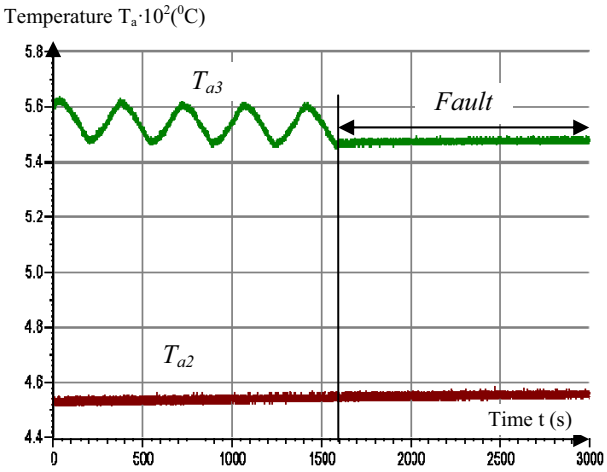

Fig. 13. BORIS graphics for the steam temperature (the fault occurs at t=1600s)

6. CONCLUSIONS

Using a software model of the superheater, the electronic simulator together with the computer system PC and the analogical input-output data acquisition system take into consideration the delays and the noises that appear on the channels of transmission of the information between process and digital control equipment.

On the electronic simulator one can create different fault events that cannot be artificially created in the real process for security reasons.

In this paper were considered a number of possible faults that may appear in the superheater, and using data acquisition system these faults have been detected.

ACKNOWLEDGMENT

This work was supported by the strategic grant POSDRU/159/1.5/S/133255, Project ID 133255 (2014), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007 - 2013.

REFERENCES

Automation Guide For Release 5.3, ControlDesk, (2007)

ControlDesk Experiment Guide for Release 5.4, ControlDesk, (2007)

Iancu E., and Vinatoru M., (2003). *Analytical method for fault detection and isolation in dynamic systems study case*, Ed. Universitaria Craiova.

Implementation Guide for Release 5.4, Real-Time Interface (RTI and RTI-MP), (2007)

Maican C., Vinatoru M., and Canureci G., (2010), Fault detection and control in superheater using electronic simulator, *IEEE International Conference on Automation, Quality and Testing, Robotics AQTR 2010 - THETA 17th edition*, Proceedings vol.1, pag. 94-100.

Quick Software Installation Guide for Release 5.4, dSPACE Release, (2007)

Analyzing the Influence of Learning Styles on Students' Behavior in a Social Learning Environment [§]

Elvira Popescu*, Florin Leon**

** Department of Computers and Information Technology, University of Craiova, Craiova, Romania (e-mail: popescu_elvira@software.ucv.ro)*
*** Department of Computer Science and Engineering, Technical University "Gheorghe Asachi" of Iaşi, Iaşi, Romania (e-mail: fleon@cs.tuiasi.ro)*

Abstract: Learning style is one of the students' characteristics which play an important role in learning, referring to the individual manner in which they approach a learning task. Many studies have investigated the relationship between students' learning styles and their behavior in learning management systems or other traditional educational systems. With the increased adoption of Web 2.0 tools in instructional settings, it is interesting to also explore the influence of learning styles on students' usage patterns of these social media tools. Hence, this paper focuses on students' behavior in a social learning environment which integrates four Web 2.0 tools (wiki, blog, microblogging tool, social bookmarking tool); students' learning styles are categorized according to Felder-Silverman model. The analysis is based on typical machine learning algorithms for classification, association rule induction and feature selection. The investigation includes 3 scenarios: i) the analysis of the number of student actions with each social media tool); ii) the dominant tool corresponding to a learning style; and iii) the temporal evolution of the number of actions and their category. Results show that learning styles have a limited influence on the students' level of interaction with each of the four social media tools considered.

Keywords: social media, Web 2.0 tools, social learning environment, learning styles, behavioral patterns, machine learning, classification, association rule induction, feature selection

1. INTRODUCTION

Even if not yet part of the educational technology mainstream, Web 2.0 tools "have reached a high level of maturity and have been increasingly adopted in educational practices worldwide" (Jeremic et al., 2013). This overarching term refers to various applications built on the Web 2.0 infrastructure, such as blog, wiki, social bookmarking tool, social networking service, microblogging tool, media sharing service etc., all of which are also known as social media tools. These technologies can be used to foster communication and collaboration between learners and help create online learning networks. Some practical ways in which selected Web 2.0 tools can support teaching and learning are synthesized in (Conole and Alevizou, 2010), (Homola and Kubincova, 2009), (Orehovacki et al., 2012). Overall, studies report a general positive impact of social media tools on learning, leading to an increase in the effectiveness of the learning process and especially in the learner motivation and satisfaction (Popescu, 2013).

In this context, it is interesting to investigate the relationship between students' individual differences and their preference and behavior toward the social media tools; this would help identify the success factors of using Web 2.0 tools in educational settings. In particular, in this paper we focus on learning style as one of the individual differences that play an important role in learning, according to educational psychologists (Popescu, 2009).

Learning style refers to the individual manner in which a person approaches a learning task. For example, some learners prefer graphical representations and remember best what they see, others prefer audio materials and remember best what they hear, while others prefer text and remember best what they read. There are students who like to be first presented with the definitions followed by examples, while others prefer abstract concepts to be first illustrated by a concrete, practical example. Similarly, some students learn easier when confronted with hands-on experiences, while others prefer traditional lectures and need time to think things through. Some students prefer to work in groups, others learn better alone. These are just a few examples of the many different preferences related to perception modality,

[§] This is an extended version of the conference paper: Florin Leon, Elvira Popescu, *Exploring the Relationships between Students' Learning Styles and Social Media Use in Educational Settings*, Proceedings of the ICSTCC 2013, pp. 657-662, 2013.

processing and organizing information, reasoning, social aspects etc., all of which can be included in the learning style concept (Popescu, 2009).

During the last decades, many learning style models have been proposed, which differ in the learning theories they are based on, the number and the description of the dimensions they include. For the current study, we focus on one of the most popular models in technology-enhanced learning (Derntl and Graf, 2009), namely the Felder-Silverman learning style model (FSLSM) (Felder and Silverman, 1988). According to FSLSM, learners are characterized by their preferences on four dimensions: *active* versus *reflective*; *sensing* versus *intuitive*; *visual* versus *verbal*; *sequential* versus *global*. *Active* students learn by trying things out and enjoy collaborative working, while *reflective* students like to think about the material first and prefer working alone. *Sensing* learners have a preference toward facts and details and they tend to be practical and careful, whereas *intuitive* learners prefer abstract material, they like to innovate, to discover possibilities and relationships. *Visual* learners remember best what they see (pictures, diagrams, schemas etc.) while *verbal* learners get more out of words, either spoken or written. *Sequential* learners tend to gain understanding in linear steps, while *global* learners learn in large leaps, they are fuzzy about the details of the subject but are able to make rapid connections between subjects. It should be noted that these learning styles are seen as tendencies and not fixed, rigid labels.

Up to now, researchers have investigated the relations between students' learning styles and their behavior in traditional e-learning systems (learning management systems, educational hypermedia systems). Examples include rule-based approaches (Graf et al., 2009; Popescu, 2009), Bayesian networks (Garcia et al., 2007), decision trees (Cha et al., 2006; Ozpolat and Akar, 2009), neural networks (Villaverde, 2006) or reinforcement learning algorithms (Dorca et al., 2013) for dynamic identification of students' learning styles. However, students' behavior in the emerging social learning environments (which is the focus of this paper) has been far less explored.

The rest of the paper is structured as follows: section 2 provides an overview of the few related works which investigate the relationship between students' learning styles and their preference or behavior toward social media tools. Section 3 describes the context of our study and the approach used for data collection. Section 4 offers a brief overview of the machine learning algorithms used for analyzing the data. Section 5 details the analysis process and the results obtained. Some discussions and conclusions are included in section 6.

2. RELATED WORK

Saeed and Yang (2008) reported on one of the first studies which explored the correlations between students' learning style and their preferences toward Web 2.0 tools. FSLSM was used and the associated Index of Learning Styles (ILS) questionnaire (Soloman and Felder, 1998) was applied at the beginning of the course in order to identify students' styles. Learners' preferences toward Web 2.0 tools were elicited by means of another dedicated questionnaire; students were asked to rate various tools (including blog, wiki, podcast, vodcast but also email and Blackboard LMS) on a scale from 1 to 5, for various learning activities (e.g., reviewing lectures, submitting group projects, having group discussions, etc.). 89 students enrolled in a Web programming course responded to both questionnaires and were included in the study. Pearson correlation was applied and a few significant relationships were discovered: i) *intuitive* learners (who, according to FSLSM, prefer discovering possibilities and relationships and are always ready to try out new things) preferred blogs; ii) *sensing* learners preferred email (a more traditional communication tool, in line with their more careful and detail-oriented nature); iii) *visual* learners preferred vodcasts (not surprisingly, taking into account their preference toward pictures, diagrams, flow charts etc.); iv) *sequential* learners preferred podcasts (since they tend to gain understanding in linear steps and follow logical stepwise paths, so they could run the sequence of lectures at their own pace over and over again to get a better understanding of the course content). No correlations were found for the *active/reflective* dimension.

The authors also performed a second study (Saeed et al., 2009), in which they analyzed the effects of cognitive style (*adaptors* versus *innovators*) (Kirton, 1976) on learner acceptance of blogs and podcasts. The context of study was again the Web programming course, in which they included blogs and podcasts as support tools. Kirton's Adaption-Innovation inventory was used to identify students' cognitive style and a dedicated questionnaire was used to elicit students' perceptions regarding ease-of-use and usefulness of the Web 2.0 tools. 187 students filled in the two questionnaires and were included in the study. The results showed that *innovator* students are more likely to perceive blogs and podcasts as useful and easy-to-use as compared to *adaptor* students. Furthermore, *innovators* perceive podcasts as more useful, but less easy-to-use than blogs.

Derntl and Graf (2009) investigated the effects of the learning style on the blogging behavior of students in an undergraduate course on software architectures and web technologies. 77 students were enrolled in the course but only 74 of them filled in the ILS questionnaire (for identifying learners' FSLSM dimensions) and were included in the study. Students were asked to use blogs as a kind of personal journal, including insights and remarks on the tasks, problems encountered and solutions found, reflections on the project and teamwork, etc.; however, the blogging activity did not count toward the students' grade. The blogging activities were integrated into the course learning management system and various student actions were recorded in a log file. Rank correlation analysis was used in order to find relationships between the students' learning style and these blogging actions. No significant results were found regarding: i) the number of

visits to the blogging environment; ii) the frequency of reading others' blogs; iii) the preference for using links from the 20 recent blog postings. Some significant correlations were found for the *active/reflective* dimension: i) *active* students tend to post more frequently to their blogs than *reflective* students; ii) *reflective* students' ratio of reading other blog postings vs. posting to their own blogs is significantly higher than that of *active* students; iii) *active* students use charts displaying the number of postings and peer rating more often than *reflective* students. One significant correlation was found for the *sequential/global* dimension as well: *sequential* learners tend to write longer posts than *global* learners. Overall, the results show that the blogging behavior is only slightly influenced by the learning style, at least from a quantitative point of view; the blog content needs to be further analyzed in order to take into account the quality of students' contributions as well.

Lau and Lee (2010) analyzed the influence of learning style and competence level on students' perceptions regarding the utility of various e-learning services, tools and content. The VAK learning style model is used, categorizing students as *visual*, *auditory* or *kinesthetic* (or a combination thereof). The learning style was identified by means of a dedicated inventory, while students' opinions were gauged by means of a Likert-style survey. 31 students participated in the study and filled in both questionnaires. While the opinion survey addressed a variety of issues (e.g., knowledge acquisition services, communication services, performance assessment services, content media type and instructional role), here we summarize only the findings related to Web 2.0 tools. *Visual* and *auditory* students rated wikis as highly useful services, regardless of their competence level; *auditory* students with lower knowledge level also perceived blogs as useful, while *kinesthetic* students favored media sharing services for their online video tutorials. Social networking systems (such as Facebook) were among the top rated communication services for all students, regardless of their learning style. However, it should be mentioned that the paper only provided descriptive results and the statistical significance of the findings was not addressed.

Grekinis (2011) explored the relationship between learning style and blogging performance, in the context of an undergraduate introductory environmental science course. Kolb's learning style model was used, and a dedicated inventory was applied at the beginning of the course to categorize students as *assimilators*, *accomodators*, *convergers* or *divergers* (Kolb, 1984). 70 students were enrolled in the course and consequently participated in the study. During the semester, they were asked to complete 8 blog assignments, consisting of both informational posts and personal reflections on the topics presented in class. Students' blogging performance was evaluated according to several criteria (preparation of blog entry, quality of content, personal reflection, proper citations, use of graphics and multimedia, comments on others' entries), which counted for 40% of the students' final grade. At the end of the semester, a Chi square test was applied to investigate the relationship between the learning style and the grade received for blogging; no significant difference was obtained, so learning styles were not found to influence students' blogging performance.

Several other authors investigated the correlations between learning style and self-reported preference for social media tools used for educational purposes, with various results, e.g.:

- Shahsavar and Tan (2010) reported no significant relationships between students' *field-dependent/field-independent* style (Witkin, 1962) and their attitude toward blogs;
- Chen et al. (2007) reported that *intuitive* students (according to FSLSM) were willing to deliver their knowledge and experience through blogs, as opposed to the students with *visual* preference.

Overall, the reported findings are somewhat contradictory; a few correlations have been found, but they are not consistent throughout the studies. It should be mentioned, however, that various learning style models were involved and different experimental settings were employed. Also, most of the studies were based on student self-reported data, e.g., preference, acceptance or attitude toward social media tools, captured by means of questionnaires. Derntl and Graf's paper (2009) is a notable exception, relying on actual student performance and analysis of behavioral patterns. Our study also explores the relationships between the actual student interaction with the Web 2.0 tools and the FSLSM dimensions. As far as analysis techniques are concerned, statistical correlation tests were the main methods employed in the above studies; in contrast, our approach is based on machine learning algorithms for classification, association rule induction and feature selection. More details regarding the experimental settings are presented in the following section.

3. CONTEXT OF STUDY

The context of our study is a course on "Web Applications' Design" (WAD), delivered to 4th year undergraduate students in Computer Science from the University of Craiova, Romania. A project-based learning (PBL) scenario was used, in which students had to design and implement an authentic Web application (such as a virtual bookstore, an online auction website, a professional social network, an online travel agency), performing all the stages of real-life application development. Due to the complexity of the tasks, the project spanned over the whole semester. Students had to collaborate in teams of 4-5 peers; 45 students were enrolled in the course and 11 such teams were formed (Popescu, 2012).

The project activity was done in a blended mode: there were weekly face-to-face meetings between each team and the instructor (for checking the project progress,

providing feedback and answering questions) and for the rest of the time students had to use social software tools as support for their communication and collaboration activities. More specifically, four Web 2.0 tools were selected by the instructor:

1. Blogger - for documenting the progress of the project (i.e., a kind of "learning diary" - reporting each accomplished activity, describing problems encountered and asking for help, reflecting on their learning experience); publishing ideas, thoughts, interesting project-related findings; communicating with the peers, providing solutions for the peers' problems, critical and constructive feedback, interacting with other teams;

2. MediaWiki - for collaborative writing tasks among the members of a team; gathering and organizing their knowledge and resources regarding the project theme; clearly documenting each stage of the project as well as the final product;

3. Delicious - for storing links to resources of interest for the project (i.e., a kind of "personal knowledge management tool"); sharing discovered bookmarks with peers; tagging and rating the collected resources; checking the resources shared by peers (and especially by own team members);

4. Twitter - for staying connected with peers and posting short news, announcements, questions, and status updates regarding the project.

More details regarding the PBL scenario can be found in (Popescu, 2012).

All student actions on the four social media tools were monitored and recorded by means of a dedicated platform called eMUSE. The platform gathers learner actions from each of the disparate tools, stores them in a local database for further processing (together with a description and an associated timestamp) and presents them to the instructor in suggestive graphical formats. The whole range of functionalities provided by eMUSE can be found in (Popescu, 2014). Figure 1 provides an overview of the data collection mechanism implemented in eMUSE. The technical solution adopted for learner tracking and data collection is accessing the Web 2.0 tools by means of open APIs or Atom/RSS feeds in order to retrieve the students' actions. This integration of content from several external sources to create a new Web application, with added value for the user, is known as *mashup* technique - which is also reflected in the platform name (eMUSE - **e**mpowering **M**ash**U**ps for **S**ocial **E**-learning).

The number of actions performed on each tool was computed for each student after the course, as a quantitative measure of the level of involvement of the student with the Web 2.0 tools. Overall, at the end of the semester about 1700 student actions were stored in the platform database; the distribution of actions over time and over the four tools is illustrated on the right side of Fig. 1 (as it appears to the instructor in eMUSE).

While the students' actions were automatically collected by the eMUSE platform, their learning styles were elicited by means of a dedicated inventory, the Index of Learning Styles questionnaire (ILS) (Soloman and Felder, 1998). ILS consists of 44 questions, each with two possible answers. As a result of the test, the learning style of the student is described on a scale between -11 and +11 (with a step of +/-2) for each FSLSM dimension; e.g., a score of +9 on the *visual/verbal* dimension implies a strong *visual* preference, while a score of -3 implies a mild *verbal* preference. ILS was applied at the beginning of the semester; 42 students filled it in and were therefore included in the analysis, as described next.

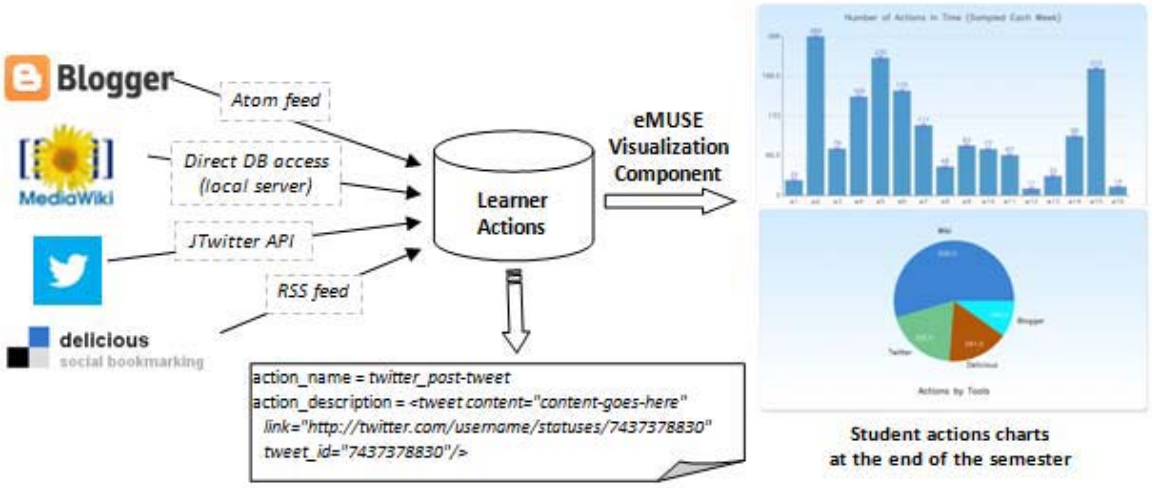

Fig. 1. Data collection mechanism provided by eMUSE social learning environment

4. OVERVIEW OF MACHINE LEARNING ALGORITHMS USED

Our goal was to investigate whether there are dependencies and connections between the actions of the students on the four Web 2.0 tools (as recorded by eMUSE), and their learning styles. To this end, we used typical machine learning algorithms for classification, association rule induction and feature selection. In what follows, we give a brief presentation of the methods used.

Classification is a procedure in which individual instances are placed into groups, or classes, based on quantitative information on one or more of their characteristics, referred to as attributes. Classification is a supervised technique, i.e. the model is built based on a training set of

instances whose classes are known. The information contained in the training set, with instances whose corresponding class labels are known, can be used to classify new, previously unseen instances, based on an explicit or an implicit model (Leon et al., 2010).

For the present analysis, the main goal was to determine a symbolic, explicit model that can be easily interpreted. Also, it was important to use algorithms that belong to different classification paradigms, which can provide different perspectives about the problem at hand. Therefore, we chose a decision tree inducer, *C4.5*, and a generalized instance-based method, *NNGE*.

C4.5 (Quinlan, 1993) generates a decision tree by recursive partitioning of data. The algorithm considers all the possible attribute tests that can split the data within a node and chooses the test that gives the best information gain, i.e. the partitioning that would result in the most homogenous child nodes. It can handle both symbolic and numerical attributes. The algorithm supports tree pruning at the end of the training process, which cuts off some parts of the tree in order to avoid overfitting.

Non-Nested Generalized Exemplar, NNGE (Martin, 1995) is an extension of the classic instance-based learning, where the training instances are simply stored and new ones are classified on the basis of their closeness to their "neighbors" in the training set. The *n* attributes define an *n*-dimensional Euclidean space in which the concepts are represented. NNGE works with generalized exemplars, which can be either hyper-rectangles in the *n*-dimensional space or single training instances, i.e. points, known as "trivial hyper-rectangles". The exemplars are not allowed to nest or overlap, thus reducing overfitting. This is achieved by testing each potential new generalization to ensure that it does not cover any negative examples, and by modifying any generalizations that are later found to do so. The algorithm tries to generalize new examples to their nearest neighbor of the same class, but if this is impossible due to intervening negative examples, no generalization is performed. If a generalization later conflicts with a negative example, it is modified to maintain consistency.

When the dataset has many attributes, some of them may be more important, while others can even be irrelevant to the classification. The relative importance of attributes can be discerned by using *feature selection* algorithms.

One such algorithm is *ReliefF* (Kononenko et al., 1997), whose basic idea is to assess the importance of an attribute according to its ability to distinguish between instances that are close to one another. For an instance, the algorithm searches for its neighbors from the same class and from a different class. Assuming that only one neighbor is used, if the instance and the same-class neighbor have different values for an attribute, this is not desirable because the attribute separates two instances of the same class. Therefore, the quality estimation of that attribute is decreased. If the instance and its other-class neighbor have different values for an attribute, this is desirable since the attribute separates two instances of different classes, and thus contributes to the classification goal. Therefore, the quality estimation of the attribute is increased. The process is repeated for all the instances of the problem. In general, ReliefF searches for *k* nearest neighbors in each class and can handle multi-class problems.

Association rule induction aims at finding regularities in the trends of the data: one tries to find sets of instance values that frequently appear together. Such information is usually expressed in the form of rules. An association rule expresses an association between (sets of) items. However, not every association rule is useful, but only those that are expressive and reliable. Therefore, the standard measures to assess the quality of an association rule are the support and the confidence, both of which are computed from the support of certain item sets (Aflori and Leon, 2004).

In order to extract the rules, the *Apriori* algorithm (Agrawal and Srikant, 1994) was employed. The algorithm is based on the observation that if any given set of attributes *S* is not adequately supported, any superset of *S* will also not be adequately supported. For example, if we know that {*A, B*} is not supported, it follows that {*A, B, C*}, {*A, B, D*}, etc. will also not be supported. The algorithm first determines the support for all single attributes (sets of cardinality 1) in the data set, and deletes all the single attributes that are not adequately supported. Then, for all supported single attributes, it constructs pairs of attributes (sets of cardinality 2). If there are no pairs, it finishes; otherwise it determines the support for the constructed pairs. For all supported pairs of attributes, "candidate" sets of cardinality 3 (triples) are built. Again, if there are no triples, it ends; otherwise it determines the support for the constructed triples. It continues likewise until no more candidate sets can be produced.

Many times, some attributes are symbolic and others are numeric. For example, to apply the classic Apriori algorithm one needs to transform the numerical attributes into symbolic ones as a preprocessing phase. *Discretization* transforms continuous attribute values into a finite number of intervals and associates a discrete value to each of them. One of the simplest methods is the equal-width discretization, which divides the range of values into *k* intervals with equal width. Although practical, discretization can lead to important information loss.

From the implementation point of view, in the present analysis we used *Weka* (Hall et al., 2009), a popular collection of machine learning algorithms.

5. ANALYSIS AND RESULTS

The case studies focused on 3 scenarios: i) the analysis of the number of actions in a category (i.e., with each social media tool); ii) the dominant tool corresponding to a learning style; and iii) the temporal evolution of the number of actions and their category.

5.1 Scenario 1. The Number of Actions

In this scenario, we considered only the total number of actions which a certain student performs on each of the 4 tools. We investigated whether it is possible to predict a student's learning style based on the Web 2.0 tools he/she uses for communication, collaboration and learning support.

We constructed a dataset with each of the 4 descriptors (*sequential/global - SG, active/reflective - AR, sensing*

/intuitive - SI, visual/verbal - VV) as the class and the number of actions for each tool as the inputs.

The learning styles were discretized into 6 categories, based on the scores obtained on the ILS questionnaire: *BigNegative* (-11 and -9), *Negative* (-7 and -5), *SmallNegative* (-3 and -1), *SmallPositive* (+1 and +3), *Positive* (+5 and +7), *BigPositive* (+9 and +11). For example, for the *SG* dimension, a positive value indicates a tendency toward the *sequential* style, while a negative value indicates a *global* style. The number of actions was discretized into 5 categories (*VeryLow, Low, Medium, High, VeryHigh*).

A part of the decision tree induced by *C4.5* for the *SG* style is presented next. In parentheses, the number of correctly classified and incorrectly classified instances are noted, respectively. Thus, a resulting rule, although inexact (14 correctly classified instances and 9 exceptions) is, e.g.: "If the number of *Wiki* actions is *VeryLow* and the number of *Delicious* actions is *VeryLow* and the number of *Twitter* actions is *VeryLow*, then *SG* is *SmallPositive*."

```
Wiki = VeryLow
|   Delicious = VeryLow
|   |   Twitter = VeryLow ⇒ SG = SmallPositive (14/9)
|   |   Twitter = Low ⇒ SG = SmallNegative (2/1)
|   |   Twitter = Medium ⇒ SG = SmallNegative (1)
...
Wiki = Low ⇒ SG = SmallPositive (4/2)
Wiki = Medium ⇒ SG = SmallNegative (4/2)
Wiki = High ⇒ SG = BigPositive (1)
Wiki = VeryHigh ⇒ SG = SmallNegative (2/1)
```

As *C4.5* uses the information gain criterion to split data, and *Wiki* (i.e., the number of actions performed on the wiki) corresponds to the first split, it implies that *Wiki* is the most important factor to describe the *SG* model.

The rules (generalized exemplars) provided by *NNGE* are of the following form (in parentheses, at the end, the number of instances covered by that rule is indicated):

```
Blogger in {VeryLow} and Delicious in {VeryLow} and Twitter in
{VeryLow} and Wiki in {VeryLow,Low} ⇒ SG = SmallPositive (3)
```

The algorithms were also applied for the other learning style dimensions. The resulting decision trees are quite large, which shows that there are no compact rules to describe the learning style depending on the discretized number of actions.

Also, there are many rules provided by *NNGE*, with many single instances which cannot be included into a generalized exemplar.

Table 1 presents the error rates and some descriptors of each model. The unpruned version of the *C4.5* algorithm was used, because it consistently provided better results for our learning problems. For the decision trees, the number of leaves was used as an indicator of the tree complexity. In terms of the number of rules given by *NNGE*, the number of generalized exemplars or hyper-rectangles (*H*) and the number of individual instances or singles (*S*) are mentioned.

Table 1. Classification performance for the learning styles as a function of the discretized number of actions

	SG	**AR**	**SI**	**VV**
C4.5 error	47.61 %	47.62 %	35.71 %	33.33 %
C4.5 no. leaves	17	17	17	21
NNGE error	23.81 %	26.19 %	28.57 %	33.33 %
NNGE no. rules	9H / 9S	8H / 11S	6H / 9S	3H / 18S
Most relevant attribute	Wiki	Delicious	Twitter	Wiki

The *ReliefF* feature selection algorithm was applied to identify the most relevant attributes for each classification problem. The results refer only to the algorithm performance on the training sets. Since these errors are very large, there is no point in further applying cross-validation to test the generalization capability. It is clear that the models do not capture the data well.

Beside classification, the *Apriori* algorithm was applied to find association rules. First, we analyzed only the relationships between the number of actions. In the parentheses, the confidence of the rule is indicated. The rule $A \Rightarrow B$ has confidence c if $c\%$ of the transactions in the dataset that contain A also contain B. The first 3 high-confidence rules found are the following:

```
Twitter=Low ⇒ Wiki=VeryLow (1)
Blogger=VeryLow, Twitter=Low ⇒ Wiki=VeryLow (1)
Twitter=VeryLow, Wiki=VeryLow ⇒ Blogger=VeryLow (0.94)
```

For example, one can interpret the first rule as: "When the number of *Twitter* actions is *Low*, the number of *Wiki* actions is always *Low*."

Then, the dataset with the number of actions and the learning styles was analyzed. The first 5 high-confidence rules are in this case:

```
Delicious=VeryLow, Sl=BigPositive ⇒ Twitter=VeryLow (1)
Twitter=VeryLow, Wiki=VeryLow ⇒ Blogger=VeryLow (0.94)
Delicious=VeryLow, Twitter=VeryLow ⇒ Blogger=VeryLow (0.94)
Delicious=VeryLow, Twitter=VeryLow, Wiki=VeryLow ⇒
Blogger=VeryLow (0.93)
Blogger=VeryLow, VV=BigPositive ⇒ Wiki=VeryLow (0.92)
```

However, it seems that these rules do not provide any clear, useful causal relationships. We could only infer that the students with a low level of activity on one tool tend to have a weak performance on other tools as well.

Beside the investigations presented above, the following ones were also attempted: i) the percentages of actions out of the total number of actions, for a student, instead of the actual number of actions; ii) discretization with 3 classes instead of 5; and iii) the comparison of learning styles within student teams. The results were not better than before.

The high errors on the training set in case of classification

can be explained, in part, by the discretization process, where the equal-width interval method may not have captured the trends in the data in a flexible enough manner. However, it is possible to apply *C4.5* and *NNGE* on the unprocessed numerical inputs as well. We performed the same analysis on numerical data, where the inputs were the number of actions for each tool, as percentages out of the total number of actions of a student. Since no information is lost in the inputs, the training set errors are much lower, as shown in Table 2.

Table 2. Classification performance for the learning styles as a function of the relative number of actions

	SG	AR	SI	VV
C4.5 error (TS)	17.64 %	29.41 %	14.71 %	17.65 %
C4.5 no. leaves (TS)	9	8	9	8
C4.5 error (CV)	64.71 %	91.18 %	61.76 %	70.59 %
NNGE error (TS)	0 %	0 %	0 %	0 %
NNGE no. rules (TS)	9H / 6S	9H / 11S	9H / 5S	9H / 5S
NNGE error (CV)	76.47 %	85.29 %	61.76 %	58.82 %
Most relevant attribute	Blogger	Delicious	Twitter	Twitter

By performing 10-fold cross-validation, the error rates become very high. Even if the algorithms, especially the instance-based *NNGE*, can exactly capture the data, the models do not generalize well.

One can notice that the most relevant attribute has changed compared to the corresponding value for *SG* and *VV* in Table 1 (i.e. *Wiki*). The resulting decision tree for *SG* is listed below. The most relevant attribute given by *C4.5*'s information gain criterion (*Twitter*) is now different from the value given by *ReliefF* (*Blogger*). These changes also suggest that there are no stable trends to be identified in the dataset.

```
Twitter <= 3.45
|   Twitter <= 0 ⇒ SG = Negative (3)
|   Twitter > 0 ⇒ SG = BigNegative (2/1)
Twitter > 3.45
|   Delicious <= 10.19
|   |   Delicious <= 0 ⇒ SG = SmallNegative (7/1)
|   |   Delicious > 0
|   |   |   Delicious <= 9.09 ⇒ SG = SmallPositive (2)
|   |   |   Delicious > 9.09 ⇒ SG = SmallNegative (2/1)
|   Delicious > 10.19
|   |   Delicious <= 25.81
|   |   |   Twitter <= 27.78 ⇒ SG = SmallPositive (5)
|   |   |   Twitter > 27.78 ⇒ SG = Negative (3)
|   |   Delicious > 25.81
|   |   |   Twitter <= 15.85 ⇒ SG = SmallNegative (2)
|   |   |   Twitter > 15.85 ⇒ SG = SmallPositive (8/3)
...
```

We present next the hyper-rectangles with the largest number of instances for each learning style, as provided by *NNGE*:

```
4.55<=Blogger<=25 and 0<=Delicious<=10.19 and 12.1<=Twitter<=50
and 25<=Wiki<=77.27 ⇒ SG = SmallNegative (6)
0<=Blogger<=7.32 and 0<=Delicious<=23.33 and
6.67<=Twitter<=18.18 and 70<=Wiki<=77.27 ⇒ AR = SmallNegative
(4)
15.09<=Blogger<=55.56 and 8.89<=Delicious<=42.11 and
3.45<=Twitter<=38.64 and 0<=Wiki<=46.55⇒ SI = Positive (6)
4.26<=Blogger<=22.22 and 13.79<=Delicious<=40.91 and
17.58<=Twitter<=44.83 and 0<=Wiki<=60.44⇒ VV = BigPositive (8)
```

Unfortunately, it seems to be difficult to gain specific insights from these rules that are in line with the theoretical assumptions. Even worse results are obtained when considering the direct relation between an individual tool and a learning style. No single preference toward a learning tool is directly correlated to a learning style.

5.2 Scenario 2. The Dominant Tool

In the second scenario, we tried the opposite approach: to see which communication tool is dominant, and then which one is second-dominant, depending on the learning style. Again, the error on the training set is low, but the cross-validation error is much higher, as shown in Table 3.

Table 3. Classification performance for the dominant use of a tool as a function of the learning styles

	First Dominant	Second Dominant
C4.5 error (TS)	18.75 %	9.38 %
C4.5 no. leaves (TS)	16	24
C4.5 error (CV)	56.25 %	46.88 %
NNGE error (TS)	6.25 %	0 %
NNGE no. rules (TS)	7H / 7S	9H / 5S
NNGE error (CV)	50 %	46.88 %
Most relevant attribute	SI	*Dominant*, then AR

The *NNGE* rules with most instances, for each tool, are the following:

```
SG in {BigNegative, Negative, BigPositive} and AR in {SmallNegative,
SmallPositive, Positive, BigPositive} and SI in {Negative,
SmallNegative, Positive, BigPositive} and VV in {Negative,
SmallNegative, Positive, BigPositive} ⇒ Dominant = Wiki (6)
SG in {Negative, SmallPositive} and AR in {Negative, SmallNegative,
BigPositive} and SI in {SmallPositive} and VV in {Positive, BigPositive}
⇒ Dominant = Twitter (3)
SG in {SmallPositive} and AR in {Positive} and SI in {SmallPositive} and
VV in {SmallPositive} ⇒ Dominant = Delicious (1)
SG in {SmallPositive} and AR in {SmallNegative} and SI in {Positive}
and VV in {BigPositive} ⇒ Dominant = Blogger (1)
```

Some sections of the decision trees for the dominant tool are given next:

```
SI = SmallPositive
|   AR = SmallPositive ⇒ Dominant = Delicious (2/1)
|   AR = Positive ⇒ Dominant = Wiki (4/1)
...
SI = Positive
|   SG = SmallNegative ⇒ Dominant = Wiki (4/2)
|   SG = SmallPositive ⇒ Dominant = Wiki (6/2)
...
SI = BigPositive ⇒ Dominant = Wiki (7)
```

as well as for the decision tree of the second-dominant tool:

```
Dominant = Blogger ⇒ SecondDominant = Wiki (1)
Dominant = Delicious
|   SI = SmallPositive  ⇒ SecondDominant = Wiki (2)
|   SI = Positive ⇒ SecondDominant = Twitter (2)
...
Dominant = Twitter  ⇒ SecondDominant = Delicious (5/2)
Dominant = Wiki
|   AR = SmallNegative ⇒ SecondDominant = Twitter (7/1)
|   AR = SmallPositive
|   |   SG = SmallNegative ⇒ SecondDominant = Twitter (2)
|   |   SG = SmallPositive ⇒ SecondDominant = Twitter (2)
|   AR = Positive
|   |   VV = Positive ⇒ SecondDominant = Twitter (2)
|   |   VV = BigPositive ⇒ SecondDominant = Delicious (3)
|   AR = BigPositive ⇒ SecondDominant = Blogger (2)
...
```

The first high-confidence association rules provided by *Apriori* are the following:

```
SI=BigPositive ⇒ Dominant=Wiki (1)
AR=SmallNegative, SI=Positive, Dominant=Wiki ⇒
SecondDominant=Twitter (1)
SecondDominant=Blogger ⇒ Dominant=Wiki (1)
VV=BigPositive, SecondDominant=Twitter ⇒
SecvGlobal=SmallPositive (1)
AR=SmallPositive, SecondDominant=Twitter ⇒ Dominant=Wiki (1)
```

An interesting finding is the first association rule, which states that the dominant tool for a highly *sensing* student is the wiki. Indeed, wiki contributions are generally based on facts and practical aspects, and are more elaborate, requiring attention to details and careful writing, requirements which are in line with the *sensing* students' nature.

Overall, the classification rules are rather complex and their interpretation is not obvious. The fact that the dominant tool greatly influences the choice of the second-dominant one is also a valid result, but it does not bring additional information about the actual problem, i.e., the relationship between the tools and the learning styles.

5.3 Scenario 3. The Evolution in Time

In this scenario, we analyzed the action data taking time explicitly into account. As a pre-processing step, we discretized the period of data collection into 5 equal intervals, and we eliminated the students with only two or three actions. Then, we computed: i) the total number of actions in each time interval; and ii) for each tool, the percentage of its total number of actions in that time interval.

Figure 2 presents the cumulative number of actions for two students, one with a strong *sequential* preference (*SG* = 9) and one with a strong *global* preference (*SG* = -9). If the time intervals are denoted as *I1*, *I2*, ... *I5*, the corresponding rules for the two graphs are:

```
I1=17 and I2=34 and I3=32 and I4=0 and I5=16 ⇒ SG = BigPositive (1)
I1=1 and I2=18 and I3=13 and I4=5 and I5=21 ⇒ SG = BigNegative (1)
```

when using the actual number of actions and:

```
I1=17.17 and I2=34.34 and I3=32.32 and I4=0 and I5=16.16 ⇒ SG =
BigPositive (1)
I1=1.72 and I2=31.03 and I3=22.41 and I4=8.62 and I5=36.21 ⇒ SG =
BigNegative (1)
```

when using percents.

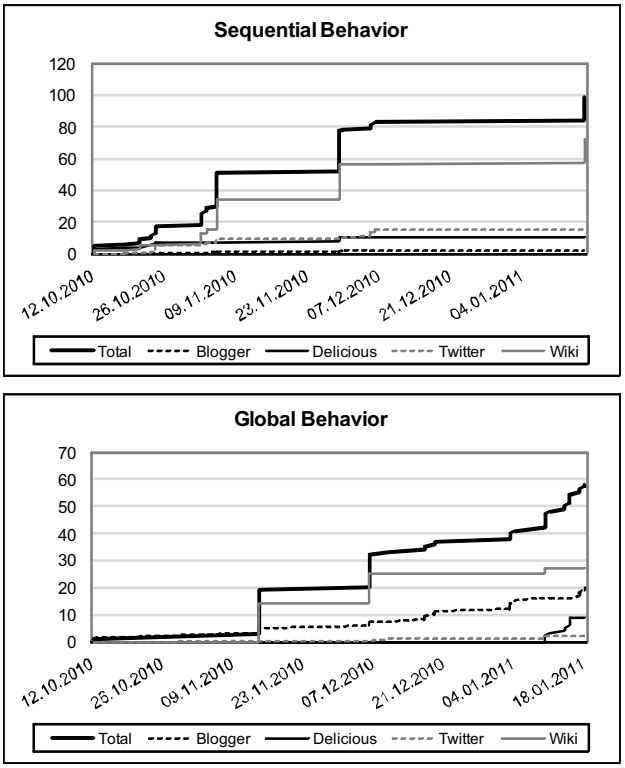

Fig. 2. The difference in behavior between a typically *Sequential* student and a typically *Global* one

These partial results can be considered to be in line with the theoretical description of the *SG* learning style: *sequential* students tend to gain understanding in linear steps, at a more constant pace, while *global* students learn in large leaps and they contribute more toward the end of the semester, once they get the big picture.

However, these partial results are not confirmed by the general model. Table 4 shows both the training set and the cross-validation errors, and it is clear that, again, there is no general model to be captured in this way. Similar results were obtained when considering the relations between individual tools and learning styles.

Finally, association rules were generated for all the available data: the total number of actions in the 5 time intervals (*Total1*, *Total2*,..., *Total5*), the number of actions corresponding to the 4 tools in the 5 time intervals (*Blogger1*, *Blogger2*,..., *Wiki5*) and the 4 learning style dimensions.

The high-confidence rules with the highest support are displayed below, including class association rules, where the learning styles are always in the right hand side of the rule:

Total3=VeryLow, Blogger1=VeryLow, Blogger3=VeryLow, Blogger5=VeryLow, Twitter1=VeryLow, Twitter2=VeryLow ⇒ SG=SmallPositive (1)
Twitter2=VeryLow, Twitter5=VeryLow, Wiki3=VeryLow ⇒ AR=Positive (1)
Blogger3=VeryLow, Blogger5=Low ⇒ SI=Positive (1)
Total3=VeryLow, Blogger2=VeryLow, Wiki2=VeryLow ⇒ VV=BigPositive (1)

and general association rules:

Delicious2=VeryLow ⇒ Total4=VeryLow (1)
Total3=VeryLow ⇒ Total4=VeryLow (1)
Wiki4=VeryLow ⇒ Total4=VeryLow (1)
Wiki4=VeryLow ⇒ Delicious2=VeryLow (1)
Delicious2=VeryLow, Wiki4=VeryLow ⇒ Total4=VeryLow (1)

Table 4. Classification performance for the learning style as a function of the number of actions in the 5 time intervals

	SG	AR	SI	VV
C4.5 error (TS)	23.52 %	26.47 %	11.76 %	17.65 %
C4.5 no. leaves (TS)	8	7	9	7
C4.5 error (CV)	73.53 %	73.53 %	55.88 %	64.71 %
NNGE error (TS)	0 %	0 %	0 %	0 %
NNGE no. rules (TS)	10H / 7S	9H / 6S	8H / 6S	8H / 10S
NNGE error (CV)	76.47 %	73.53 %	55.88 %	55.88 %
Most relevant attribute	I3	I5	I3	I3

As an overall assessment of the obtained results, the rules are quite complex and hard to interpret, especially when they involve a combination of tools. Also, similar conditions appear for different classes (e.g. *Negative* vs. *Positive*) and different conditions appear for the same class or similar classes (e.g. *SmallPositive* and *Positive*). Some potential causes for these inconclusive results are provided in the next section.

5. DISCUSSION AND CONCLUSION

In this paper we investigated whether there are dependencies and relationships between the students' learning style (according to FSLSM) and their preference toward certain Web 2.0 tools used as learning support instruments (according to the number of actions performed on that tool). The results showed that the learning styles have a limited influence on the students' level of activity involving each of the four tools (Blogger, MediaWiki, Delicious, Twitter). In what follows, we provide some possible explanations for these results.

We only took into consideration the students' actions which involve an active interaction or a contribution to the social media tools (e.g., posting or commenting on a blog, but not reading a blog; adding a bookmark on Delicious, but not browsing the peers' bookmarks etc.). Thus we did not completely capture the preference for a certain tool; for example, a *reflective* student may have spent a lot of time reading his/her colleagues' blog posts or analyzing their contributions on the wiki, but this type of interaction was not measured. This limitation comes from the type of student actions that can be collected from the social media tools, as they are provided by means of feeds or APIs.

Also, we did not take into account the content of the students' contributions and the learning tasks they refer to. For example, a *sensing* and an *intuitive* student may have had the same number of actions on the blog, but the former may have posted mainly facts, source code and practical examples, while the latter may have contributed with theoretical aspects and innovative ideas. Furthermore, the recorded actions refer to various types of learning activities: creating content (*blog_post-entry, wiki_revise-page, wiki_upload-file*), social interactions (*delicious_add-friend-to-network*), organizing content (*delicious_post-bookmark*), communication and feedback (*blog_post-comment, twitter_post-tweet*) (Popescu, 2014), but in our current analysis we did not discriminate among them; this is a limitation that we plan to address in our future work.

Finally, it may be that the learning styles do not influence the students' preference and behavior toward the four Web 2.0 tools. This could be explained by students' flexibility to accommodate a wide variety of emerging social media applications into their learning environment, without being limited to a particular tool (Saeed and Yang, 2008). This also suggests that our instructional scenario and the tools it relies on are not biased toward any particular learning style. Indeed, these findings are in line with another analysis that we conducted taking into consideration the students' course grades, which showed that FSLSM dimensions have weak or no correlations with the student performance (Giovannella et al., 2013).

Nevertheless, further studies are needed to fully understand the relationships between learning style and preference toward social media tools in educational contexts. These will involve additional action types (as described above), more analysis algorithms (including also a qualitative dimension), as well as a larger number of students.

ACKNOWLEDGMENT

This work was partially supported by the grant number 15C/2014, awarded in the internal grant competition of the University of Craiova.

REFERENCES

Aflori, C. and Leon, F. (2004). Efficient distributed data mining using intelligent agents. In *Proc. 8th International Symposium on Automatic Control and Computer Science*, Iasi.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proc. VLDB 1994*, pp. 487-499.

Cha, H.J., Kim, Y.S., Park, S.H., Yoon, T.B., Jung, Y.M.,

and Lee J.H. (2006). Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In *Proc. ITS 06,* pp. 513-524, LNCS 4053, Springer.

Chen, L.C., Chu, P.Y, Lin, C.C., and Cheng, Y.M. (2007). The effects of knowledge sources and learning styles of the elders on their motivations to use weblog tools for lifelong learning. In *Proc. ICALT 2007*, pp. 737-738.

Conole, G. and Alevizou, P. (2010). *A literature review of the use of Web 2.0 tools in Higher Education*. The Open University, UK.

Derntl, M. and Graf, S. (2009). Impact of learning styles on student blogging behavior. In *Proc. ICALT 2009*, pp. 369-373, IEEE CS Press.

Dorca, F.A., Lima, L.V., Fernandes, M.A., and Lopes, C.R. (2013). Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis. *Expert Systems with Applications*, 40(6), pp. 2092-2101.

Felder, R.M. and Silverman, L.K. (1988). Learning and teaching styles in engineering education. *Engineering Education*, 78(7), preceded by a preface in 2002: www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf.

Garcia, P., Amandi, A., Schiaffino, S., and Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education*, 49(3), pp. 794-808.

Giovannella, C., Popescu, E., and Scaccia, F. (2013). A PCA study of student performance indicators in a Web 2.0-based learning environment. In *Proc. ICALT 2013*, pp. 33-35.

Graf, S., Kinshuk, and Liu, T.C. (2009). Supporting teachers in identifying students' learning styles in learning management systems: An automatic student modelling approach. *Educational Technology & Society*, 12(4), pp. 3-14.

Grekinis, D. (2011). Self-reported perceptions and learning styles of undergraduate students using blogs in an environmental science course. In *Proc. ICERI 2011*, pp. 1445-1451.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1), pp. 10-18.

Homola, M. and Kubincova, Z. (2009). Taking advantage of Web 2.0 in organized education (a survey). In: *Proc. ICL 2009*, pp. 741-752.

Jeremic, Z., Jovanovic, J., and Gasevic, D. (2013). Personal learning environments on the social semantic Web. *Semantic Web*, 4(1), pp. 23-51.

Kirton, M.J. (1976). Adaptors and innovators: a description and a measure. *Applied Psychology*, 61, pp. 622-629.

Kolb, D. (1984). *Experiential learning*. Englewood Cliffs, Prentice Hall.

Kononenko, I., Simec, E., and Robnik-Sikonja, M. (1997). Overcoming the myopia of inductive learning algorithms with ReliefF. *Applied Intelligence*, 7, pp. 39-55.

Lau S.B.Y. and Lee, C.S. (2010). Contextualising e-learning services and content for computing course in higher education based on learning style and competency level. *Int. J. Learning Technology*, 5(3), pp. 211–242.

Leon, F., Lisa, C., and Curteanu, S. (2010). Prediction of the liquid crystalline property using different classification methods. *Molecular Crystals and Liquid Crystals*, 518, pp. 129-148.

Martin, B. (1995). *Instance-based learning: nearest neighbor with generalization*. Master of Science Thesis, University of Waikato, New Zealand.

Orehovacki, T., Bubas, G., and Kovacic, A. (2012). Taxonomy of Web 2.0 Applications with Educational Potential. In: *Transformation in Teaching: Social Media Strategies in Higher Education*, pp. 43-72, Informing Science Press.

Ozpolat, E. and Akar, G.B. (2009). Automatic detection of learning styles for an e-learning system. *Computers & Education*, 53(2), pp. 355-367.

Popescu, E. (2009). Diagnosing students' learning style in an educational hypermedia system. In *Cognitive and Emotional Processes in Web-based Education: Integrating Human Factors and Personalization,* pp. 187–208, Advances in Web-Based Learning Book Series, IGI Global.

Popescu, E. (2012). Project-based learning with eMUSE: an experience report. In *Proc. ICWL 2012*, pp. 41-50, LNCS 7558, Springer.

Popescu, E. (2013). *Social learning environments.* Sitech, Craiova.

Popescu, E. (2014). Providing collaborative learning support with social media in an integrated environment. *World Wide Web Journal*, 17(2), Springer, pp. 199-212.

Quinlan, R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Mateo, CA.

Saeed, N. and Yang, Y. (2008). Using learning styles and preferences to incorporate emerging e-learning tools in teaching. In *Proc. ICALT 2008*, pp. 967-971, IEEE CS Press.

Saeed, N., Yang, Y., and Sinnappan, S. (2009). Effects of cognitive style on user acceptance of blogs and podcasts. In *Proc. ICALT 2009*, pp. 293-297.

Shahsavar, Z. and Tan, B.H. (2010). The influence of students' cognitive styles on their attitudes toward using blogs. In *Proc. E-Learn 2010*, pp. 2158-2164.

Soloman, B. and Felder, R.M. (1998). *Index of learning styles questionnaire*. Available at: www.engr.ncsu.edu/learningstyles/ilsweb.html.

Villaverde, J.E., Godoy, D., and Amandi, A. (2006). Learning styles' recognition in e-learning environments with feed-forward neural networks. *Journal of Computer Assisted Learning*, 22(3), pp. 197-206.

Witkin, H.A. (1962). *Psychological differentiation: Studies of development*. Wiley.

Necessity of the implementation of advanced control algorithms in complex processes for energy industry

Ion Marian Popescu*

** Department of Automation, Energy and Environment, University "Constantin Brâncuși" of Târgu Jiu, Târgu Jiu, Romania (e-mail: marian@utgjiu.ro)*
Postdoctoral Researcher at University of Craiova, Craiova, Romania (e-mail: pmarian@automation.ucv.ro)

Abstract: In this paper is presented implementation of a real process control system "rotary inverted pendulum", and the possibility of extending advanced control algorithms in the thermoenergetic processes. Implemented algorithms are based on a state feedback control structure and design was created using LQR (Linear Quadratic Regulator), and Lyapunov direct method, thus making a comparison between the two methods. Rotary inversed pendulum is an underactuated process, nonlinear process, unstable, and has characteristics that we find in a complex process such as thermoenergetic group.

Keywords: LQR, Lyapunov, state feedback, power plant

I. PROCESS DESCRIPTION FOR ROTARY INVERTED PENDULUM

Reverse rotating pendulum is a process widely used in control systems to illustrate the idea of automatic control technology. Being a underactuated, nonlinear, unstable, with strong real-time constraints, it is very useful in testing new developments in nonlinear systems. As the construction [2], is composed of an arm driven by a DC motor rotating in a horizontal plane and is attached to the pendulum itself rotating in a vertical plane perpendicular to the drive arm, as in *Fig.1*

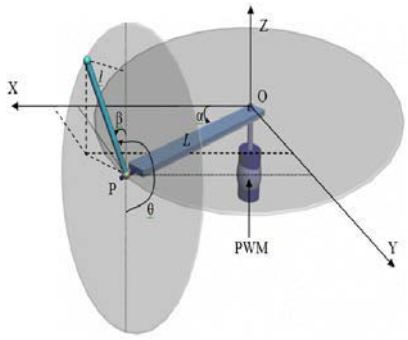

Fig.1. Pendulum in the vertical_Up position

The motion is initiated by rotating the arm from the fixed end by a DC motor powered by a variable voltage, PWM modulator. At the point O and the point P are placed two potentiometric sensors that will provide a voltage proportional to the angle α - the angle of rotation of the arm in a horizontal plane and β - the angle of rotation in the vertical plane and perpendicular to the arm of the pendulum. The components are shown in *Fig.2.*

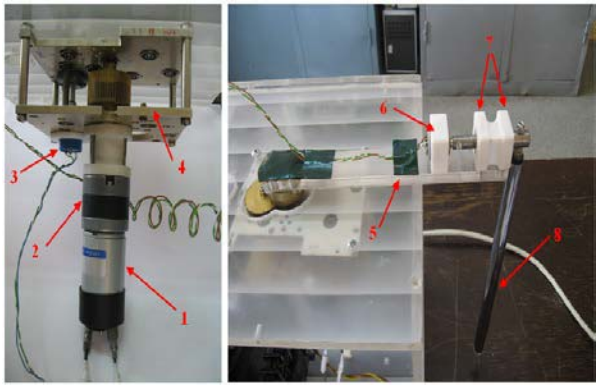

Fig.2. The structure of the physical system: 1-D.C. engine, 2-gearbox, 3-position potentiometer arm, 4-motor-potentiometer mechanism connecting arm, 5-actuating arm, 6-potentiometer position pendulum, 7 bearings, 8-Pendulum

II. MATHEMATICAL MODEL OF THE PENDULUM

Nonlinear equations of the pendulum in the vertical_Up position taken from [1], [2] are:

$$\begin{cases} (J_b + m_p L^2)\ddot{\alpha} + m_p l^2 \ddot{\alpha} \sin^2\beta - m_p Ll\ddot{\beta}\cos\beta + 2m_p l^2 \dot{\alpha}\dot{\beta}\sin\beta\cos\beta + \\ \qquad + m_p Ll\dot{\beta}^2 \sin\beta = \dfrac{K_t}{R_a}u - \dfrac{K_t K_b}{R_a}\dot{\alpha} - C_b\dot{\alpha} \\ -m_p Ll\ddot{\alpha}\cos\beta + (J_p + m_p l^2)\ddot{\beta} - m_p l^2 \dot{\alpha}^2 \sin\beta\cos\beta - m_p gl\sin\beta = -C_p\dot{\beta} \end{cases} \quad (1)$$

Relations (1), which represents the equations for the arm and pendulum in the vertical_Up position will be linearized around a stationary operating point α_0, β_0, characterized by:

$$\alpha(t) = \alpha_0 + \Delta\alpha(t)\,; \quad \beta(t) = \beta_0 + \Delta\beta(t) \quad (2)$$

„This work was supported by the strategic grant POSDRU/159/1.5/S/133255, Project ID 133255 (2014), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007–2013."

Thus $\alpha_0 = \alpha_{ref}$ is the reference for the system and $\beta_0 = 0$ angle is the angle represented by the vertical passing through the anchorage of the pendulum. Choose the state variables as:

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \begin{bmatrix} \Delta\dot{\beta} & \Delta\beta & \Delta\dot{\alpha} & \Delta\alpha \end{bmatrix}^T \quad (3)$$

Linearized Ecuations are:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{fa}{b^2-ea} & \frac{ha}{b^2-ea} & -\frac{cb}{b^2-ea} & 0 \\ 1 & 0 & 0 & 0 \\ -\frac{fb}{b^2-ea} & -\frac{hb}{b^2-ea} & \frac{ce}{b^2-ea} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \frac{db}{b^2-ea} \\ 0 \\ -\frac{de}{b^2-ea} \\ 0 \end{bmatrix} u$$

where: $a = (J_b + m_p L^2)$; $b = -m_p Ll$; $c = C_b + \frac{K_t K_b}{R_a}$;

$$d = \frac{K_t}{R_a};\ e = J_p + m_p l^2;\ f = C_p;\ h = -m_p gl \quad (4)$$

III. Data Acquisition and Command System as Distributed System

Acquisition and control system that manages the system "rotary inverted pendulum" is shown in *Fig.3*.

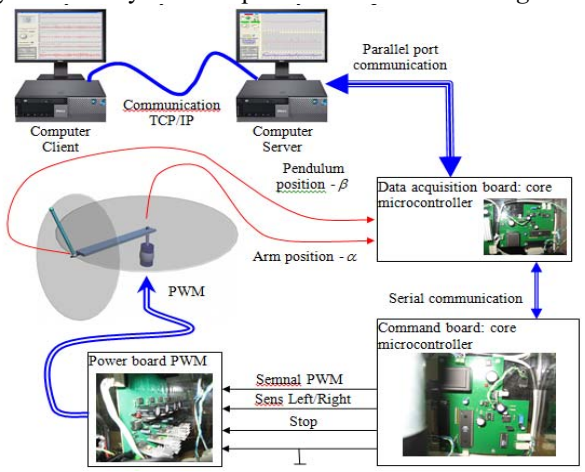

Fig.3. Data acquisition and command system

Because of the necessity for very small sampling times, realization of the parallel operations, the designed system processes signals across multiple processors. The structure of the data acquisition and distributed control has the following components:

-Computer Server: It is a standard PC computer, which implements real-time control algorithm. The sampling period was $T_e = 0.005$ seconds.

-Computer Client: It is another standard PC that connects to the computer server and communicates with it using an interrupt based on TCP/IP protocol implemented in the development LabWindows/CVI. Sampling period is $T_{com\_client} = 20 \cdot T_e$.

-Data Acquisition board with microcontroller core: It is performed around a core AT89S52 microcontroller equipped with an analog to digital converter with two channels on 12bit, ADS7841P for reading the two potentiometric transducers: arm position and pendulum position. Communication with the server is done on the computer parallel port on a dedicated protocol. The main task of this board is to manage the conversion of two channels of the converter and transmission, serial this time to the second microcontroller board, the command received from the server.

-Command board with microcontroller core: It made all around a core microcontroller AT89S52, and mainly serves to form the PWM signal, and transmit it to the force board PWM, that acts engine. It communicates on serial interrupts. with the first board microcontroller which it provides PWM command value.

IV. The Desisgn Using LQR Control Algorithm

Starting from the state equations linearized around a stationary operating point $\Delta\dot{\beta} = 0$; $\Delta\beta = 0$; $\Delta\dot{\alpha} = 0$; $\Delta\alpha = \alpha_0$ for the pendul_Up system of equation (4) the implementation of the state feedback control structure is shown in *Fig.4*.

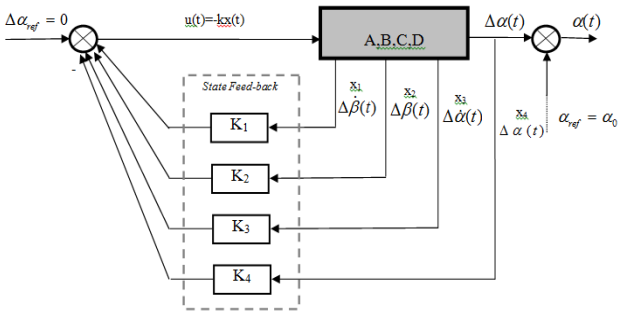

Fig.4.State feedback

Using Matlab simulation environment using lqr function (Linear Quadratic Regulator). is obtained the command:

$$u = -kx = -k_1 x_1 - k_2 x_2 - k_3 x_3 - k_4 x_4 = -k_1 \Delta\dot{\alpha} - k_2 \Delta\alpha - k_3 \Delta\dot{\beta} - k_4 \Delta\beta$$

where

k1=43.77107964964838; k2 =243.8487809616433

k3 = -13.83637129582518; k4 = -7.07106781187112

The response is shown in *Fig.5*.

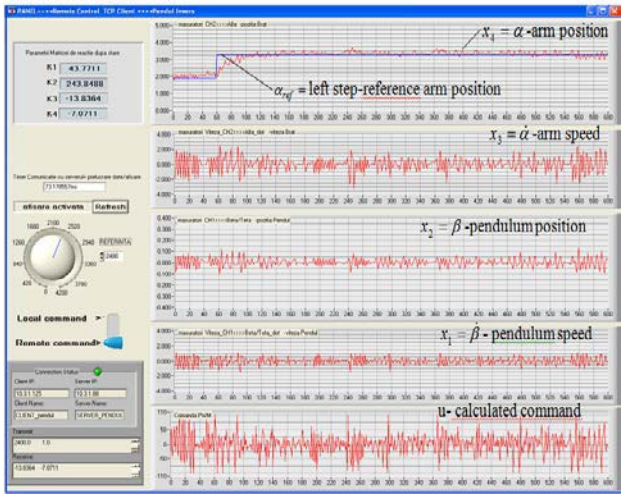

Fig.5. Real system response to step-variation for α_{ref}

V. The Design Control Algorithm by the Direct Method Lyapunov

The general approach used in the study of stability of nonlinear control systems in sense Lyapunov stability

theory introduced in the late 19th century by Alexander Mikhailovich Lyapunov Russian mathematician. In an attempt to increase the stability for the inverted pendulum system, there will be a design using the second Lyapunov method. The method involves the assumption of a Lyapunov function candidate and then designing a control law that lead to the transformation function Lyapunov function candidate in a real. A first step in approaching design nonlinear control algorithm using input-output linearization method by reaction and then stabilize by the second Lyapunov method.

The basic idea of approximation [5], is the transformation of algebraic nonlinear system dynamics into linear dynamics or partially. The major difference compared to conventional linearization around a stationary point, feedback linearization is achieved by exact transformation and connection type state feedback linearization to the entire dynamics of the system. We start from the nonlinear equations (1) we rearrange them in a matrix-vector form:

$$\begin{bmatrix} (J_b + m_p L^2 + m_p l^2 \sin^2\beta) & (-m_p Ll\cos\beta) \\ (-m_p Ll\cos\beta) & (J_p + m_p l^2) \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} +$$

$$\begin{bmatrix} (\frac{K_t K_b}{R_a} + C_b) & (2m_p l^2 \dot{\alpha}\sin\beta\cos\beta + m_p Ll\dot{\beta}\sin\beta) \\ (-m_p l^2 \dot{\alpha}\sin\beta\cos\beta) & C_p \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 \\ (-m_p gl\sin\beta) \end{bmatrix} = \begin{bmatrix} \frac{K_t}{R_a} \\ 0 \end{bmatrix} u$$

$$\Leftrightarrow \quad M\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + N\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + P = Gu \tag{5}$$

$$\Rightarrow \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} = -M^{-1}N\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} - M^{-1}P + M^{-1}Gu \tag{6}$$

where:

$$M^{-1} = \frac{1}{\det(M)}\begin{bmatrix} (J_p + m_p l^2) & (m_p Ll\cos\beta) \\ (m_p Ll\cos\beta) & (J_b + m_p L^2 + m_p l^2\sin^2\beta) \end{bmatrix}$$

The basic idea for Input-output linearization [5], [6] is to find a direct relationship between the control signal and the output of the system and we derive successively output signal to control signal appears directly in the equation:

$$\dot{y} = \dot{\beta}\,; \quad \ddot{y} = \ddot{\beta} = \frac{1}{\det(M)}(b + \mathrm{m_p Llcos}\beta\frac{\mathrm{K_t}}{\mathrm{R_a}}u)\,.$$

Is observed that the entry *"u"* is directly output to the 2nd derivative of the output, which means that the system has relative degree 2. Thus we linearize system (1) for Input-Output by reaction imposing a virtual command v of the form:

$$v = \ddot{y} = \ddot{\beta} = \frac{1}{\det(M)}(b + \mathrm{m_p Llcos}\beta\frac{\mathrm{K_t}}{\mathrm{R_a}}u) \tag{7}$$

Thus the real command applied to the actuator will be of the form:

$$u = \frac{\det(M)v - b}{\frac{K_t}{R_a}m_p Ll\cos\beta} \tag{8}$$

The virtual command v can be elegantly designed by pole placement technique, with available input-output linear system $v->y$, but it should be mentioned that such linearized dynamics is of order 2. Original nonlinear system. has a dynamic of order 4, which means that some of the dynamics of the system is not observable [5] the input-output linearization. This internal dynamic can not be influenced directly by command v from equation (8). In the design of the control law for the system of order 2, we seek a the virtual command type $v(t) = v_1(t) = -kx = [-k_0 \quad -k_1][x_3 \quad x_4]^T$ which stabilizes the subsystem with state vector: $[x_3 \quad x_4]^T = [\beta \quad \dot{\beta}]^T$:

$$\frac{d}{dt}\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_1 = A_2\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} - B_2 k\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \tag{9}$$

Using Matlab simulation environment using lqr function (Linear Quadratic Regulator). is obtained:

k0 =7.34846922834955; k1 =12.83343050227409

The mathematical model of the process after input-output linearization of the reaction becomes:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{\mathrm{C_p}}{m_p Ll}x_4 - \frac{\mathrm{g}}{L}x_3 + \frac{J_p + m_p l^2}{m_p Ll}v \\ x_4 \\ v \end{bmatrix} \tag{10}$$

VI. Asymptotic Stabilization using the direct method Lyapunov for the "system of order 2"

Second linearization of the dynamic model (10) will be used for design. We consider that this subsystem, through The virtual command " $v = v_1$ " is asymptotically stabilized and proposes an energy function of the form (11). We will try to prove that this function [3], [4], [5] is a Lyapunov function. By selectiion, $V_1 > 0 \quad \forall (x_3, x_4) \neq (0,0)$.

$$V_1 = \frac{1}{2}x_3^2 + \frac{1}{2}(x_4 + ax_3)^2 \tag{11}$$

We still have the Lyapunov function derivative relationship:

$$\dot{V}_1 = \nabla V_1 \cdot \dot{x} = \begin{bmatrix} \frac{\partial V_1}{\partial x_3} & \frac{\partial V_1}{\partial x_4} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -k_0 & -k_1 \end{bmatrix}\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \tag{12}$$

we obtain:

$$\dot{V}_1 = -ax_3^2 - b(x_4 + ax_3)^2 + x_4^2(a + b - k_1) + \\ + ax_3^2(1 + ab - k_0) + x_3 x_4(a(a + b - k_1) + (1 + ab - k_0)) \tag{13}$$

In (13) we impose equality of terms $k_1 = a + b$ and $k_0 = 1 + ab$, we get the expression of Lyapunov function derivative in the form: $\dot{V}_1 = -ax_3^2 - b(x_4 + ax_3)^2 < 0$, where parameters $a, b > 0$ depend on the tunning parameters $a = a(k_0, k_1)$, $b = b(k_0, k_1)$.

The virtual command is:

$$v = v_1 = -k_0 x_3 - k_1 x_4 = -(1 + ab)x_3 - (a + b)x_4 \tag{14}$$

VII. Design by Lyapunov direct method for the "system of order 3"

Asymptotic stabilization approach based on Partial feedback linearization where "zero dynamic" is unstable, it can be considered a simple control problem as a matter of standard design through Lyapunov direct method [3], [5]. The advantage is that the system has already stabilized dynamic part of through linearization and thus simplifies the design method. Also, the method can be extended iterative to increase the complexity of the system.

We consider the subsystem of order 2 stabilized and try to use Lyapunov function found V_1, to build a new Lyapunov function, so that stabilize the speed of the arm $\dot{\alpha}$, taking in consideration the the system of order 3.
Because Lyapunov function is a function of energy can be considered intuitively that it takes extra energy, so try to stabilize the whole system of order 3. The new Lyapunov function for the system of order 3 is proposed by [4], [5]:

$$V_2 = V_1 + \frac{1}{2}\lim_{t\to\infty}\tilde{x}_2^2(t) \tag{15}$$

where $\tilde{x}_2(t)$ - is the first component of the solution by putting the order 3 system and initial condition $x_0 = [x_2 \quad x_3 \quad x_4]^T$. Using Matlab simulation environment [1] is obtained:

$$\lim_{t\to\infty}\tilde{x}_2(t) = [1.0000 \quad -70.6882 \quad -6.67475]x_0 =$$
$$\Rightarrow \lim_{t\to\infty}\tilde{x}_2(t) = [1 \quad r \quad p]\cdot[x_2 \quad x_3 \quad x_4]^T = x_2 + rx_3 + px_4 \tag{16}$$

where $r = r(a,b)$, $p = p(a,b)$ are parameters which depend on the design of the matrix of eigenvectors of the matrix A_3, which in their turn, depends on the tunning parameters a,b, which in their turn, depend on the nonlinear control parameters "v_1", that is, the parameters k_0, k_1. New Lyapunov function becomes:

$$V_2 = V_1 + \frac{1}{2}\lim_{t\to\infty}\tilde{x}_2^2(t) = V_1 + \frac{1}{2}(x_2 + rx_3 + px_4)^2 \tag{17}$$

$$\Rightarrow V_2 = \frac{1}{2}x_3^2 + \frac{1}{2}(x_4 + ax_3)^2 + \frac{1}{2}(x_2 + rx_3 + px_4)^2 > 0 \tag{18}$$

$\forall \quad (x_2, x_3, x_4) \neq (0,0,0)$

Further, to achieve asymptotic stability of signal $\dot{\alpha}$, must have $\dot{V}_2 < 0 \quad \forall \quad (x_2, x_3, x_4) \neq (0,0,0)$, which means that we need to add virtual command $v = v_1(x_3, x_4)$, an additional term that takes into account the state x_2, that in the form $v = v_1(x_3, x_4) + v_2(x_2, x_3, x_4) \Rightarrow v = v(x_2, x_3, x_4)$.
Using this new form to virtual command v we have:

$$\frac{d}{dt}\begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{C_p}{m_p L l}x_4 - \frac{g}{L}x_3 + \frac{J_p + m_p l^2}{m_p L l}v \\ x_4 \\ v \end{bmatrix} \tag{19}$$

Equation (19) written in the matrix form using vector notation: $w = \frac{(J_p + m_p l^2)}{m_p L l}$; $B_3 = [w \quad 0 \quad 1]^T$;

$x = [x_2, x_3, x_4]^T$

$$\Rightarrow \frac{d}{dt}\begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & m & n \\ 0 & 0 & 1 \\ 0 & s & q \end{bmatrix}\begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} w \\ 0 \\ 1 \end{bmatrix}v_2 \Rightarrow \dot{x} = A_3 x + B_3 v_2 \tag{20}$$

Furthermore, considering the Lyapunov function in equation (18) we have for the new virtual control $v = v_1(x_3, x_4) + v_2(x_2, x_3, x_4)$ applied to the system (19):

$$\dot{V}_2\Big|_{v=v(x_3,x_4)+v_2} = \nabla V_2 \cdot \dot{x} = \nabla V_2 \cdot A_3 x + \nabla V_2 \cdot B_3 v_2 \tag{21}$$

Due to the fact that the system is of order 3 is fully stabilized only through control $v_2(x_2, x_3, x_4)$, which is actually a correction for the command $v = v_1(x_3, x_4)$ what stabilizes the system. of order 2, we propose a control law of the form $v_2 = -k_3 \nabla V_2 \cdot B_3$ in the idea that the 2nd term in equation (21) is negative: $\nabla V_2 \cdot B_3 v_2 = -k_3(\nabla V_2 \cdot B_3)^2$.
The k_3 parameter was added as a percentage of the total command. By processing the proposed command:

$$v_2 = -k_3 \nabla V_2 \cdot B_3 = -k_3\begin{bmatrix} \frac{\partial V_2}{\partial x_2} & \frac{\partial V_2}{\partial x_3} & \frac{\partial V_2}{\partial x_4} \end{bmatrix}\begin{bmatrix} w \\ 0 \\ 1 \end{bmatrix} \tag{22}$$

We obtain the command:

$$v_2 = c_1 x_2 + c_2 x_3 + c_3 x_4 = -k_3 C_1 x_2 - k_3 C_2 x_3 - k_3 C_3 x_4 \tag{23}$$

where $c_1 = -k_3(w + p) = -k_3 C_1$;
$c_2 = -k_3(a + rw + rp) = -k_3 C_2$;
$c_3 = -k_3(pw + 1 + p^2) = -k_3 C_3$ (24)

The c_1, c_2, c_3 parameters depend on the design parameters of the system, the parameters for tunnig k_0, k_1, by parameters r, p, a, b. Designed total command until this step is:

$$v = -(1 + ab)x_3 - (a + b)x_4 - k_3 C_1 x_2 - k_3 C_2 x_3 - k_3 C_3 x_4 \tag{25}$$

Structure for the "system of order 3" is shown in Fig. 6.

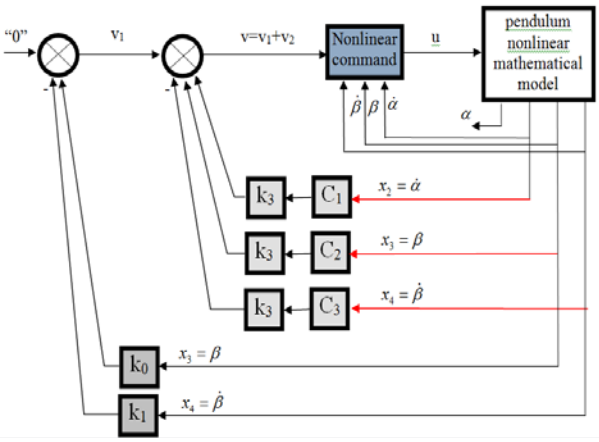

Fig.6. The structure of the command in this phase

VIII. Design by Lyapunov Direct Method for the "System of Order 4"

In this step of processing. the subsystem of order 3 can be considered stabilized and try to use the Lyapunov function V_2, to build a new Lyapunov function, so that stabilize and position the arm α, taking in consideration the the system of order 4. The method is applied recursively as a system of order 3. The proposed new Lyapunov function for the system of order 4 is:

$$V_3 = V_2 + \frac{1}{2}\lim_{t\to\infty}\tilde{x}_1^2 \tag{25}$$

where $\tilde{x}_1(t)$ - is the first component of the solution system the 4th order, with initial condition $x_0 = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$.

After processing in Matlab [1], we have:

$$\lim_{t\to\infty}\tilde{x}_1(t)=[1.0000,\ 2.6842,\ -72.9898,\ -8.2842]x_0$$

$$\Rightarrow \lim_{t\to\infty}\tilde{x}_1(t)=\begin{bmatrix}1 & r_1 & p_1 & q_1\end{bmatrix}\cdot\begin{bmatrix}x_1 & x_2 & x_3 & x_4\end{bmatrix}^T = \quad (26)$$

$$= x_1 + r_1x_2 + p_1x_3 + q_1x_4$$

where r_1, p_1, q_1 depend on the matrix of eigenvectors of the matrix A_0, which in their turn depends on the parameters for tunnig a,b,r,p, which in their turn depend on the parameters of the first linearized command *"v"*, that is k_0, k_1. We have:

$$V_3=\frac{1}{2}x_3^2+\frac{1}{2}(x_4+ax_3)^2+\frac{1}{2}(x_2+rx_3+px_4)^2+ \quad (27)$$

$$+\frac{1}{2}(x_1+r_1x_2+p_1x_3+q_1x_4)^2>0 \quad \forall(x_1,x_2,x_3,x_4)\neq(0,0,0,0)$$

To obtain asymptotic stability and for arm position α, must have $\dot{V}_3<0 \quad \forall \quad (x_1,x_2,x_3,x_4)\neq(0,0,0,0)$ which means that we try to add to virtual command $v=v_1(x_3,x_4)+v_2(x_2,x_3,x_4)$ an additional term that takes into account the state x_1, that in the form:

$$v=v_1(x_3,x_4)+v_2(x_2,x_3,x_4)+v_3\Rightarrow v=v(x_1,x_2,x_3,x_4) \quad (28)$$

New form for virtual command v is:

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\0 & wc_1 & m+wc_2 & n+wc_3\\0 & 0 & 0 & 1\\0 & c_1 & s+c_2 & q+c_3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}+\begin{bmatrix}0\\\frac{J_p+m_pl^2}{m_pLl}\\0\\1\end{bmatrix}v_3=$$

$$=A_0\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}+\begin{bmatrix}0\\w\\0\\1\end{bmatrix}v_3 \quad \Leftrightarrow \quad \dot{x}=A_0x+B_0v_3 \quad (29)$$

The virtual command $v=v_1(x_3,x_4)+v_2(x_2,x_3,x_4)+v_3$:

$$\dot{V}_3\Big|_{v=v_1(x_3,x_4)+v_2(x_2,x_3,x_4)+v_3}=\nabla V_3\cdot\dot{x}=\nabla V_3\cdot A_0x+\nabla V_3\cdot B_0v_3 \quad (30)$$

The 4th order system is fully stabilized only by command v_3 which is a correction for the command $v=v_1+v_2$ which stabilizes the system of order 3. We propose a control law of the form $v_3=-k_4\nabla V_3\cdot B_0$ in the idea that the 2nd term in equation (30) is negative: $\nabla V_3\cdot B_0v_3=-k_4(\nabla V_3\cdot B_0)^2$.

The k_4 parameter was added as a percentage and we have:

$$v_3=-k_4\nabla V_3\cdot B_0=-k_4[(wr_1+q_1)x_1+(w+wr_1^2+p+r_1q_1)x_2+$$

$$+(wr+wr_1p_1+a+rp+p_1q_1)x_3+(wp+wr_1q_1+1+p^2+q_1^2)x_4]$$

We note $d_1=-k_4(wr_1+q_1)=-k_4D_1$

$$d_2=-k_4(w+wr_1^2+p+r_1q_1)=-k_4D_2$$

$$d_3=-k_4(wr+wr_1p_1+a+rp+p_1q_1)=-k_4D_3$$

$$d_4=-k_4(wp+wr_1q_1+1+p^2+q_1^2)=-k_4D_4$$

We obtain additional virtual command:

$$v_3=-k_4D_1x_1-k_4D_2x_2-k_4D_3x_3-k_4D_4x_4 \quad (31)$$

The d_1,d_1,d_2,d_3 parameters depend on the design parameters of the system, but also the parameters for tunning k_0,k_1 by parameters. r,p,a,b. The structure of the order 4 designed system is shown in *Fig. 7*

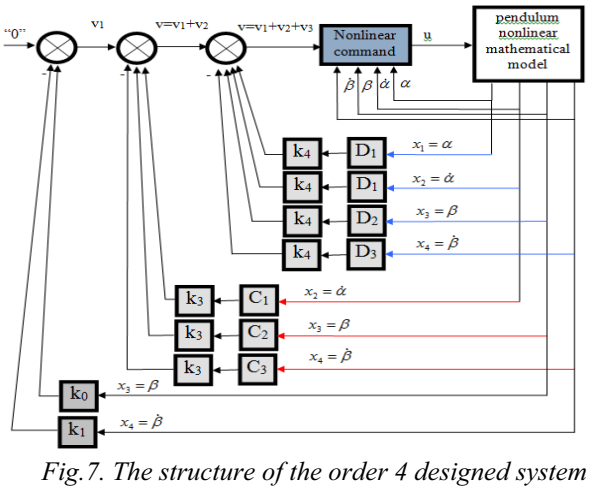

Fig.7. The structure of the order 4 designed system

IX. COMPARISON THE RESPONSES DESIGN USING LQR METHOD AND LYAPUNOV DIRECT METHOD

In Fig.8. and Fig.9. is compares the inverted pendulum system responses, by implementing the two structures designed by Lyapunov direct method respectively LQR method.

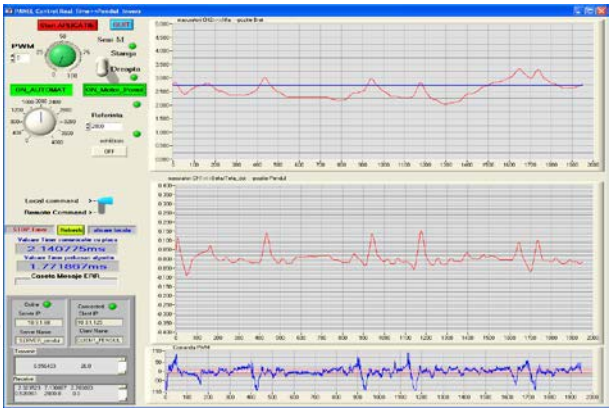

Fig.8. Real system response "inverted pendulum" designed by Lyapunov direct method (Server)

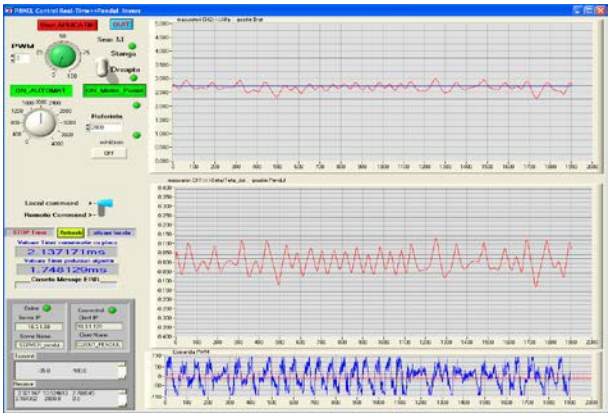

Fig.9. Real system response "inverted pendulum" designed by LQR method (Server)

Graphs are performed in real time on the server that implements the control algorithm. The sampling period is $T_e = 0.005$ seconds, and the responses are displayed on a period of 10 seconds. The first graph of each figure can see the size of the arm position α, that is better stabilized in the case of LQR method, but the oscillations are higher. In the graph of each of the 2nd figures, we can see the position of the pendulum which is very oscilating, with large amplitudes for LQR method compared to the method which creates a Lyapunov stability better in vertical_Up position of the pendulum. At the same time, it can be seen that the effort LQR control method is higher, in the sense that the command have a strong oscillating compared with the method of Lyapunov, when the command have less oscillating, and thereby control design is able to find much better equilibrium point of the pendulum.

X. APLICATIONS IN TERMOELECTRIC ENERGY PRODUCTION

As is expected, the "green energy" represented by wind and photovoltaic energy, has priority in the energy system in Romania. Thermoelectric power generation from coal has to compensate dynamically power consumption when green energy is no longer available as shown in *Fig.10.* [7]

Basically, high dynamic variation for the green energy requires the corresponding dynamics in the operation of a coal power plants, which currently is not implemented.

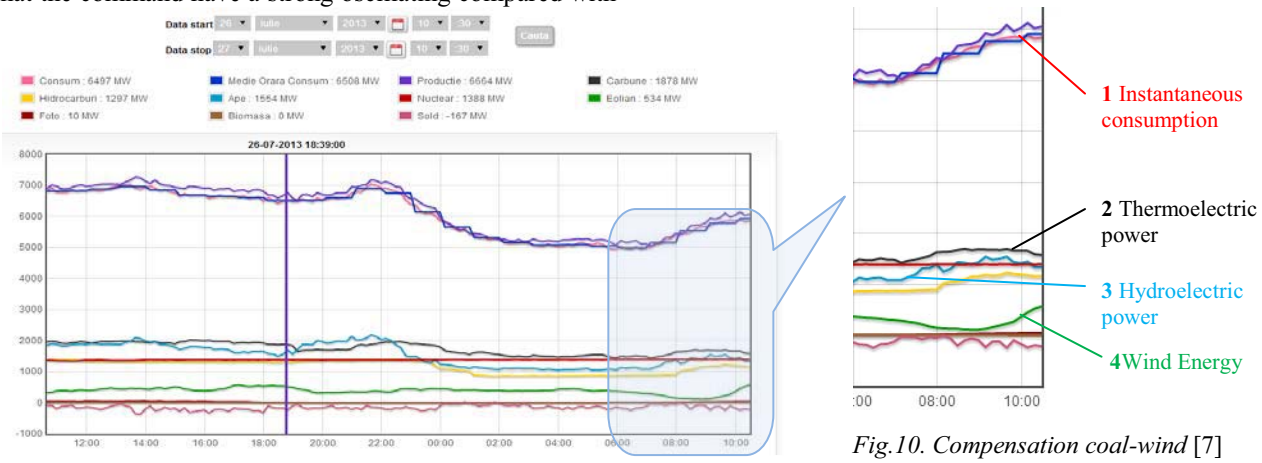

Fig.10. Compensation coal-wind [7]

Instantaneous consumption (1-red), must be supported by thermoelectric power (2-black),. hydropower (3-blue),. wind energy (4-green). In the period presented in the *Fig.10* was drought, hydroelectric energy produced was limited (3-blue) and compensation was made to power plants (2 black). To this, add the fact that wind energy (4-green), which is a priority in the system significantly decreased, which means that the dynamics of the operation of power plants must be even stronger. It can be said simplistic as ***"the wind blows -> thermoelectric power debited decreases***" and ***"the wind stops -> debited thermoelectric power increases."*** This new high dynamic for the power plants, leading to low efficiency operation and to increase the price per MW. Also appears a phenomenon of "stress" in the mechanical installation, that generates wear and additional costs. In the thermal group from CEO (Oltenia Energy Complex - Power plant Rovinari) distributed automation system implements control loops with the PID laws (Proportional Integrator Derivative), plus feed-forward corrections. This classical approach is taken because experimental tunning facilities for a law PID,. which add complexity of the process. The modeling process is very complex and difficult to do in these circumstances evolve the design of control algorithms is difficult. However, due to the higher dynamics of operation ***"impose"*** classics control algorithms redesign and bringing them into the advanced control algorithms. Example with inverted pendulum presented, induce the existence of this idea in the sense that the nonlinear control algorithm designed, succeed in find very accurate stationary operating point and also increases the attraction around this point. This means a robust evolution of the system and that was very important, leading to a shock free operation. The complex process, which is a thermal energy group, can be analyzed in depth, by distributed automation system which shows online a number of 9000 measurement points, and the history of evolution for the process for 6 years ago since the system was implemented. This is an idea that will be studied and I think the results will not delay to appear.

REFERENCES

[1] Popescu Ion Marian, Runceanu Adrian, Alina Dincă, "Control system for rotary inverted pendulum designed by LQR method and Lyapunov direct", *Proc. S G E M 2 0 1 2, Section Informatics*, Bulgaria, Vol. 3, DOI: 10.5593/sgem2012, ISSN 1314-2704, pp. 65-72.

[2] Popescu Ion Marian, "Modelarea matematica a procesului pendul invers rotativ", *Analele Universităţii "Constantin Brâncuşi" Tg. Jiu*, Seria Inginerie, Nr.2/2012, http://www.utgjiu.ro/revista/ing/pdf/2012-2/16_Marian%20POPESCU.pdf

[3] H. Bouzaouache, N. B. Braiek, "On the stability analysis of nonlinear systems using polynomial Lyapunov functions", *Mathematics and Computers in Simulation*, Vol. 76, Issues 5-6, 7 January 2008, pp. 316-329

[4] C.J.A. van Kats, "Nonlinear control of a Furuta Rotary inverted pendulum", *DCT report nr.2004.69, 2004, TU/e Bachelor Final Project Report*, Coach: Dr. Ir. L. Moreau, Supervisor: Prof. Dr. Ir. H. Nijmeijer, Eindhoven University of Technology, Division Dynamical Systems Design

[5] J.J. Slotine, W. Li, *Aplied nonlinear control*, Pretince-Hall, 1991, ISBN: 0-13040-0890-5

[6] B. Srinivasan, P. Huguenin, D. Bonvin, "Global stabilization of an inverted pendulum-Control strategy and experimental verification", *Automatica*, Vol. 45, Issue 1, January 2009, Pages 265-269

[7] http://www.transelectrica.ro/widget/web/tel/sen-grafic/-/SENGrafic_WAR_SENGraficportlet?display=CARB

Methods to Reduce the Harmonics Generated in the Asynchronous Motor Fed by Power Converter

Răzvan Prejbeanu*

**RELOC S.A., 109, Bd. DECEBAL, 200746*
CRAIOVA, ROMANIA (e-mail: razvan.prejbeanu@ relocsa.ro).

Abstract: This paper presents new control principles of the induction motors supplied from static converters and determines the coefficients that are needed to control the electrical drives through the analysis of the output voltage deformation and the system stability. The new proposed method is verified by simulation in Matlab/Simulink which is used to draw the diagrams of the stator current, of the rotor speed and of the electromagnetic torque. This method is possibly to be validated experimentally using the tests stand, which allows finding the necessary parameters for dimensioning the electrical drives and the accepted value of the stator current harmonics.

Keywords: converter, harmonics, space vector, asynchronous machine, voltage phasor.

1. INTRODUCTION

In the middle of 1980s a type of pulse wide modulation (PWM) called "modulation type vector space" (SVM) was proposed; it was considered to offer significant advantages comparing to the natural PWM and to the regular sampling in terms of performance, simplicity of implementation and ratio of the maximum transfer [1].

The PWM vector space modulation is a sophisticated method that offers advantages especially in the digital applications when compared to the traditional sinusoidal PWM modulation; these advantages refer to:
- a higher use of the DC link voltage;
- a lower total harmonic distortion (THD).

1.1 Vectorial space

A common way to represent the phase voltages A, B, C is the vectorial space (SV) pattern [2,5]. The three arms of the three phases inverter can be connected to the motor phases and the inverter entrance - to the positive or negative DC voltage terminal. Taking into account that only one switch per arm should be closed, it is obvious that eight different stages are possible. SVM principle is based on this fact: there are only eight possible switching combinations for a three-phase inverter. Spatial vectors are similar to the phasors and are defined by amplitude and angle. It is important to note that spatial vectors are not phasors. Vectors are used to represent a sinusoidal variation in time, while spatial vectors - to represent three different spatial entities time correlated. If the sum of these three different spatial entities is zero and they are spatially separated by a 120º - angle, these entities can be expressed in a single vector space (relation1):

$$\left(\underline{u} = V_{an}(t)e^{j0} + V_{bn}(t)e^{j\frac{2}{3}} + V_{cn}(t)e^{-j\frac{2}{3}} \right) \quad (1)$$

2. TECHNICAL CONSIDERATIONS

It is possible to associate a reference vector to each of the eight stages. In order to generate a rotating field, the inverter must be passed in six of the eight conditions. This operating mode is called *six-step mode*. Basic states of the inverter switches are shown in Fig. 1.

Two of these stages (SV_0 and SV_7) correspond to “zero” stages and are called "zero vectors" (they are considered to be the center of the hexagon), while the other six stages can be considered stationary vectors that can form d-q plane, as shown in Fig. 2. The two vectors are zero because, in these stages, the voltage applied to the motor windings is zero up to the midpoint of each arm which is connected to the ground (GND) or the voltage bus.

Zero vectors situated in the middle of the hexagon can be used to adjust the space vector magnitude (Fig. 3). Each vector has a length of 2/3 of the inverter voltage for each switching period ΔT/2. Each vector of the space is selected for a period which is less than a half of the tact period.

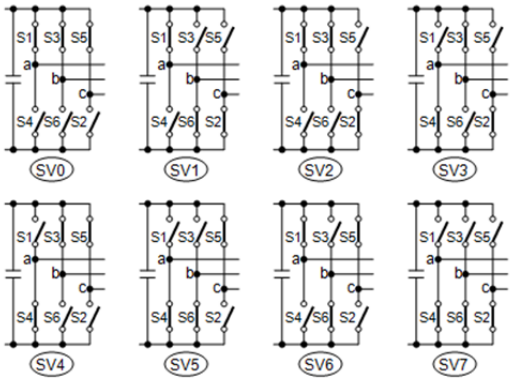

Fig. 1. Eight possible combinations of the arms of an inverter

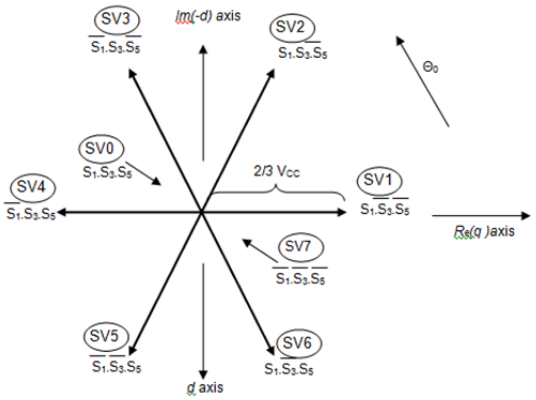

Fig. 2. Location of the eight standing voltage vectors in d-q plane

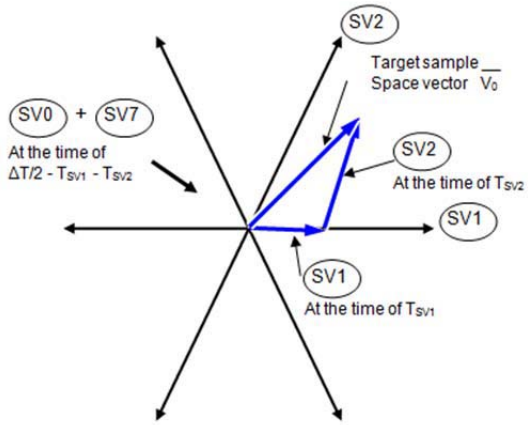

Fig. 3. Creating an arbitrary target phasor as a geometrical sum of two close vectors from the vectorial plan

It can be seen that SVM is a regular sampling intrinsic process because, in essence, it is equal with the sum of two spatial volt-second vectors divided by a half of tact period (relation 2):

$$\frac{T_{SV1}}{(\Delta T/2)}\overline{SV_1} + \frac{T_{SV2}}{(\Delta T/2)}\overline{SV_2} = \overline{V_0} \qquad (2)$$

There are two ways not to apply a voltage to the motor. The first way is to simply connect all three phases to the negative potential of the inverter; this situation will be called "inverter 7 stage" and the corresponding switching pattern is (-,-,-). The second way is to connect all three phases to the positive potential of the inverter; this situation will be called "inverter 0 stage" and the corresponding switching pattern is (+, +, +). In order to approximate the voltage $\underline{u_S}$ during the period of the PWM clock, the pulses should be used in the sequence shown in Fig. 4.

This control uses a symmetrical modulation of space vector or a center alignment in order to reduce THD. Switching pulse sequences shown in Fig. 4 are according to Fig. 5 where both stages as modules and space vector values to the voltage intermediate circuit are represented. The six pulses are symmetrically generated related to PWM clock impulse period [3].

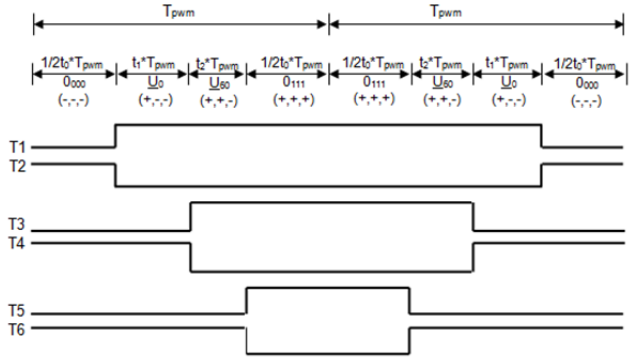

Fig. 4. Modulation of the symmetrical space vector

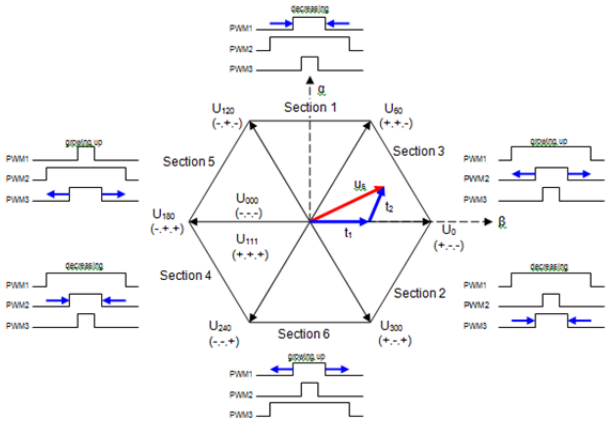

Fig. 5. Inverter voltage steps diagram

If the modulation index exceeding $\frac{\sqrt{3}}{2}$ (0.866), t_0 can become a negative value (depending on the angle). Because it is not possible to apply one of the zero vectors for a negative time, the maximum index modulation for the spatial vector modulation is about 0.866 (except it works in overmodulation). Graphically, the amplitude of the reference vector must be entirely within the hexagon for the space vector modulation $\underline{u_S}$ to work properly. Each of the vectors U_0, U_{60}, etc in the diagram represents the six stages of the voltage developed by the voltage inverter, if "zero vectors" 0_{000} and 0_{111} are in the origin. In each of these stages the inverter transistors are in steady state. In order to develop a sine wave at the motor, a switching pattern should be designed; it produces a voltage not only to the six stable stages of the vector, but also one which represents the transitions between these stages. This means that effectively a continuously rotating vector U_{out} is produced which makes a smooth transition without problems from stage to stage. SV_{PWM} aims to mediate adjacent vectors to each sector.

3. CONTROL STRATEGY

The number of pulses at the output side of the PWM inverter (which feeds with voltage and frequency the asynchronous motors) is important because it has as results the occurrence of harmonics at the output that feeds the induction machine. If the PWM output pulse number is low, the fundamental amplitude is reduced; in this case the number and the effects

of the harmonics is strong. If the number of pulses is increased, the higher harmonics become stronger. These effects are demonstrated by measuring currents in induction machine. The model shown below is necessary to simulate the effects introduced by the six order harmonics in the dynamic of the induction machine fed by a PWM inverter system.

The basic model for induction machine is rewritten [1]:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_s + \left(\frac{p}{\omega_b}\right)X_s & \left(\frac{\omega_a}{\omega_b}\right)X_s & \left(\frac{p}{\omega_b}\right)X_m & \left(\frac{\omega_a}{\omega_b}\right)X_m \\ -\left(\frac{\omega_a}{\omega_b}\right)X_s & r_s + \left(\frac{p}{\omega_b}\right)X_s & -\left(\frac{\omega_a}{\omega_b}\right)X_m & \left(\frac{p}{\omega_b}\right)X_m \\ \left(\frac{p}{\omega_b}\right)X_m & \left(\frac{\omega_a - \omega_r}{\omega_b}\right)X_m & r_r + \left(\frac{p}{\omega_b}\right)X_r' & \left(\frac{\omega_a - \omega_r}{\omega_b}\right)X_r' \\ -\left(\frac{\omega_a - \omega_r}{\omega_b}\right)X_m & \left(\frac{p}{\omega_b}\right)X_m & -\left(\frac{\omega_a - \omega_r}{\omega_b}\right)X' & r_r' + \left(\frac{p}{\omega_b}\right)X_r' \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr}' \\ i_{dr}' \end{bmatrix} \quad (3)$$

with:

$$f_R = \frac{\omega_a}{\omega_b} \quad (4)$$

and

$$sf_R = \frac{\omega_a - \omega_r}{\omega_b} \quad (5)$$

from where slipping s:

$$s = \frac{\omega_a - \omega_r}{\omega_a} \quad (6)$$

Considering that the harmonics of an order higher than 12 are insignificant, the actual values of the different variables of the asynchronous motor can be written as a sum of the constant harmonic components and the 6th harmonic (relation 7):

$$\left.\begin{aligned} T_e &= T_{e0} + T_{e6} \\ \omega_r &= \omega_{r0} + \omega_{r6} \\ v_{qs} &= v_{qs0} + v_{qs6} \\ v_{ds} &= v_{ds0} + v_{ds6} \\ i_{qs} &= i_{qs0} + i_{qs6} \\ i_{ds} &= i_{ds0} + i_{ds6} \\ i_{qr}' &= i_{qr0}' + i_{qr6}' \\ i_{dr}' &= i_{dr0}' + i_{dr6}' \end{aligned}\right\} \quad (7)$$

Considering only the effects of 6th harmonic, relation 3 becomes:

$$\begin{bmatrix} v_{qs6} \\ v_{ds6} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_s + \left(\frac{p}{\omega_b}\right)X_s & f_R X_s & \left(\frac{p}{\omega_b}\right)X_m & f_R X_m \\ -f_R X_s & r_s + \left(\frac{p}{\omega_b}\right)X_s & -f_R X_m & \left(\frac{p}{\omega_b}\right)X_m \\ \left(\frac{p}{\omega_b}\right)X_m & f_R s_0 X_m & r_r' + \left(\frac{p}{\omega_b}\right)X_r' & f_R s_0 X_r' \\ -f_R s_0 X_m & \left(\frac{p}{\omega_b}\right)X_m & -f_R s_0 X' & r_r' + \left(\frac{p}{\omega_b}\right)X_r' \end{bmatrix} \begin{bmatrix} i_{qs6} \\ i_{ds6} \\ i_{qr6}' \\ i_{dr6}' \end{bmatrix} \quad (8)$$

The fundamental frequency of the output signal is an integer multiple of the modulation frequency and is symmetrical on both sides of the cycle. In addition, in order to have real symmetry in each of the two half-cycle, the output signal must be equal to the number of pulses in each half-cycle of the PWM output. This means that the clock should have the fundamental frequency (relation 9):

$$f_e = 2{,}0(m + 0{,}5)\, f_m, \quad (9)$$

where "m" is an odd number of pulses required in each half cycle of the PWM output and "f_m" is the frequency modulator.

(4.1)

Equation (8) is empirical and the hypothesis that this fact significantly reduces the amplitude of the harmonics may be accepted. Required number of pulses in the PWM output depends on the following factors:

- limits switching devices;
- harmonic content;
- DC link circuit use;
- properly control U/f.

Considering the equivalent circuit of the induction motor at the harmonic k_{TH} (Fig. 6), the steady harmonic current k_{TH} can be written:

$$I_k = \frac{V_h}{h\left(X_{Is} + X_{Ir}'\right)} \quad (10)$$

In steady state harmonic I_b may be defined as:

$$I_b = \sqrt{I_5^2 + I_7^2 + I_{11}^2 + I_{13}^2 + \ldots + I_{n5}^2} = \sqrt{\sum_{h=5}^{n} I_h^2} \quad (11)$$

The total losses can be written:

$$P_{total} = 3\left\{\left(I_{is}^2 + I_b^2\right)r_s + \left(I_{ir}^2 + I_b^2\right)r_r'\right\} \quad (12)$$

The presence of harmonics also determines increased core loss, although they are lower than P_{total}.

Asynchronous motor parameters (figure 6) are supposed to be constant at small temperature variations. The amplitude of the harmonic losses depends on the harmonic content of voltage and current in the motor. A large harmonic voltage at low frequencies may cause the increase of losses and can reduce motor efficiency.

(4.4)

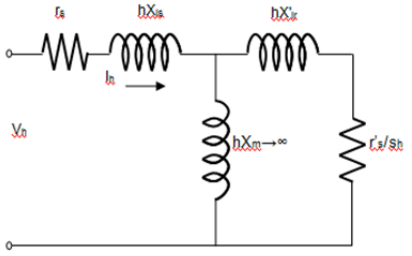

Fig. 6. Equivalent "h" harmonic circuit phase of the asynchronous motor (modified [1])

Steady harmonic torque is developed by the interaction between the stator flux harmonics and the rotor flux.

Steady harmonics torque value is very low in comparison to the nominal torque because:

- the amount of harmonic flow current and the harmonic flux are small in comparison to the fundamental values;
- the torque produced by the direct sequence of the harmonic components is partially canceled by the harmonic torque that rotates in the opposite direction.

The low torque of the 5th harmonic opposes to the torque of the 7th harmonic. A similar torque opposition occurs between the 11th and 13th harmonics. The effects of the high order harmonics may be ignored because they are very small in size. The overall effects of harmonics are insignificant in steady torque.

4. SIMULATION, RESULTS AND DISCUSSION

For the calculation of the harmonics a Matlab-Simulink was used [12]; the analysis was performed using the Fourier series decomposition.

The voltage harmonics of the inverter output PWM voltage modulation were studied, calculated and measured in three conditions:

- sinusoidal modulation produced by the intersection of a triangular signal with a sinusoidal modulating signal,
- vector space modulation system,
- optimal modulation system.

Every periodic function $f(t)$ of period T can be represented by trigonometric Fourier series (relation 13):

$$f(t) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty}\left[a_k \sin(k\omega t) + b_k \cos(k\omega t)\right], \quad (13)$$

where $\omega = \frac{2\pi}{T}$ is fundamental pulsation and $f = 1/T$ is the fundamental frequency.

The Fourier series can be written as:

$$f(t) = f_0 + \sum_{k=1}^{\infty} f_k \sin(k\omega t + \Phi_k), \quad (14)$$

where:

$$f_0 = \frac{a_0}{2}, f_k = \sqrt{a_k^2 + b_k^2}, \Phi_k = arctg\left(\frac{b_k}{a_k}\right). \quad (15)$$

Simulink block [12] which carries out the calculation of the Fourier decomposition is shown in Fig. 7.

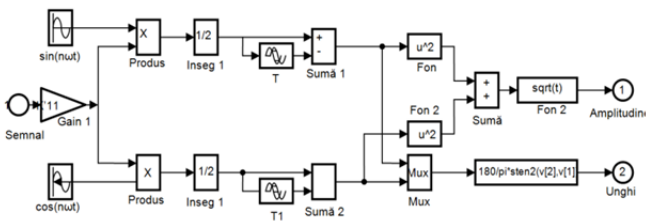

Fig. 7. Simulink block of Fourier series

Samples of all three PWM signals using specific control techniques were recorded experimentally, using the tests bench and a data acquisition system. Acquisitions were made for different frequencies and for various factors modulation. The waveforms of the sine/vector wave PWM signal for various values of the modulation factor are described below.

Fourier decomposition of two PWM signals, one obtained from a sinusoidal PWM control, the other - from a vector PWM control, was used [10].

In Table 1 the magnitudes and the phases of the two analyzed signals are presented.

Table 1. Magnitudes and phases of the two analyzed signals

Modulation factor	Sinusoidal PWM									Vectorial PWM								
	Magnitude					Phase				Magnitude					Phase			
	Am1	Am3	Am5	Am7	Am9	1	3	5	7	9	Am1	Am3	Am5	Am7	Am9	1	3	5	7	9
0.4	0.2004	0.0188	0.0190	0.0206	0.0223	5.1890	92.1053	93.8564	95.8331	97.6619	0.2320	0.0504	0.0065	0.0076	0.0067	1.1997	5.5335	47.3681	43.4969	58.6845
0.6	0.3004	0.0188	0.0195	0.0207	0.0223	3.4988	92.5498	94.0270	95.7207	97.6962	0.3478	0.0746	0.0070	0.0070	0.0062	0.8027	3.7368	46.5767	44.4928	78.7307
0.8	0.4001	0.0187	0.0196	0.0203	0.0223	2.6235	92.9549	93.8040	96.577	97.8126	0.4638	0.0985	0.0074	0.0066	0.0061	0.6070	2.7479	44.2437	47.4844	101.9488
1	0.5000	0.0188	0.0194	0.0207	0.0223	2.0123	92.0754	94.2198	96.3212	97.6420	0.5795	0.1223	0.0078	0.0065	0.0069	0.4887	2.2531	43.5570	51.8660	121.7256

The real signal acquired by the oscilloscope is shown in Fig. 8. Fig. 9 shows the analyzed signal shape generated by an inverter and the fundamental harmonic PWM signal acquired.

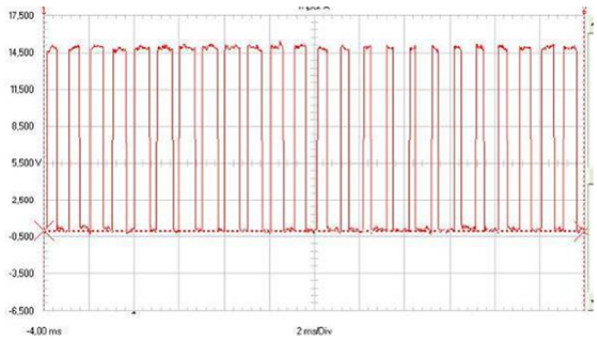

Fig. 8. Signal shape acquired by oscilloscope

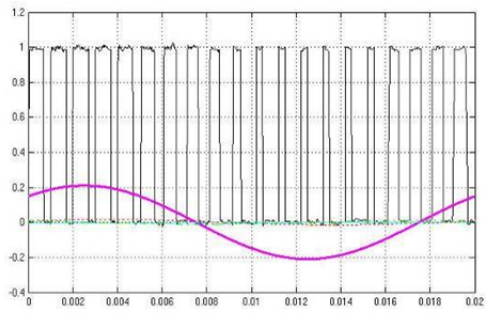

Fig. 9. Fundamental harmonic signal shape

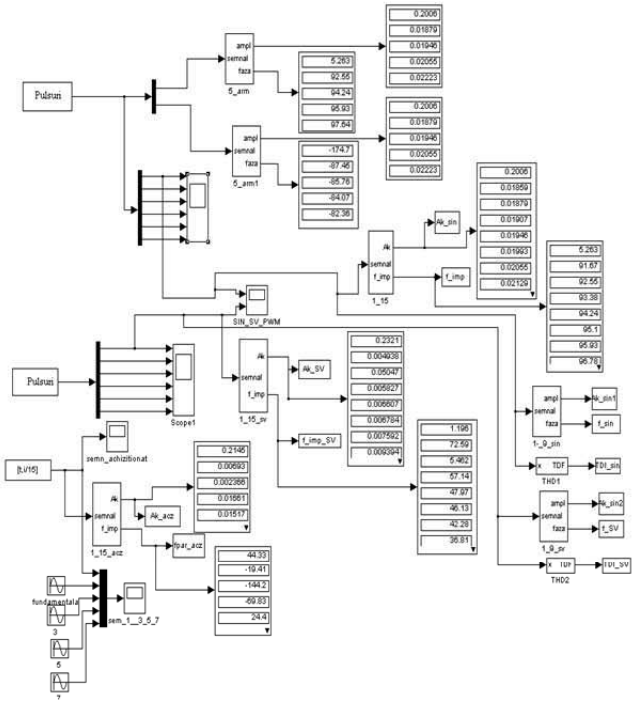

Fig. 10. Full diagram simulation

Full Simulink diagram is shown in Fig. 10. Using the simulation schematic in Fig. 10 the pulses acquired for oscilloscope are processed, calculating the harmonics amplitudes and phases. Finally, THD is calculated for the three acquired and generated signals.

Fig. 11 shows the harmonic decomposition of the analyzed signals. The values of total harmonic coefficients in the three analyzed situations are:
TDI = 3.61 for the PWM sinusoidal modulation;
TDI = 3.49 for the space vector PWM modulation without the third harmonic cancellation;
TDI = 3.29 optimized PWM acquired signal.

When the asynchronous motor is fed by a variable voltage converter [7,11], the electromagnetic torque, the rotor speed and the line current have a ripple component due to PWM control from the output voltage of the converter [9]. As shown in Figs. 12, 13 and 14, all parameters corresponding to the switching frequency show a ripple (detailed in Fig. 12 b).

A major influence on the total harmonics, meaning their increase when the motor is fed by a converter, is noted by acquiring two current signals from the same three-phase motor fed from a network and from a power converter.

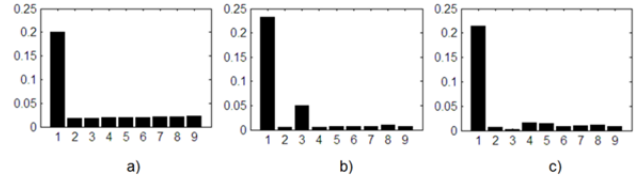

Fig. 11. Harmonic spectrums corresponding to the sinusoidal modulation (a); vector modulation (b); optimal modulated acquired signal (c)

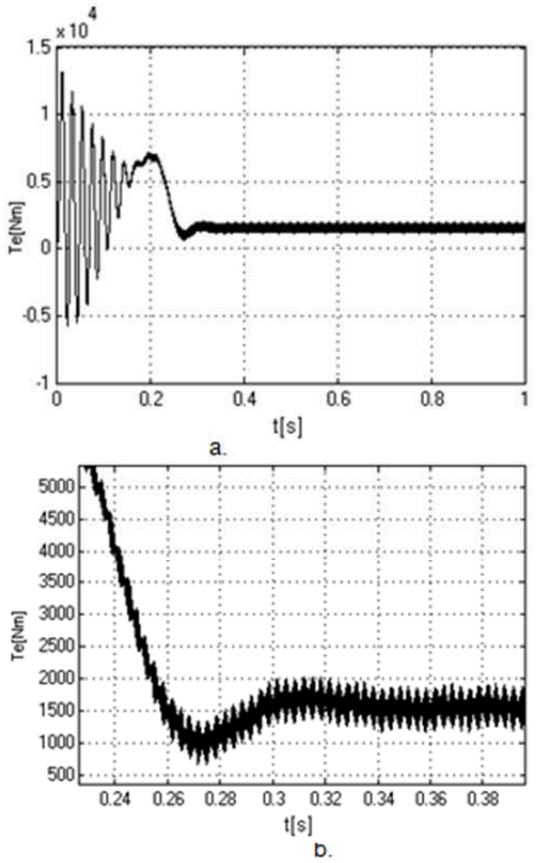

Fig. 12. The electromagnetic torque given by induction motor powered by a PWM inverter (a); detail of the torque area stabilization (b)

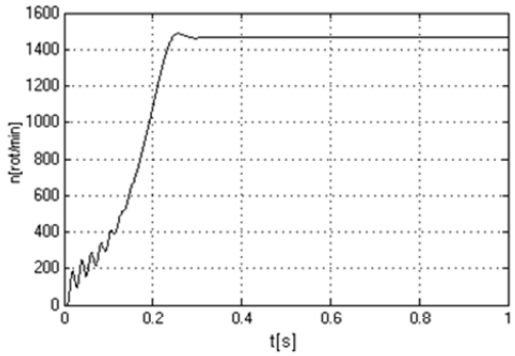

Fig. 13. The rotor speed of the asynchronous motor fed by a converter

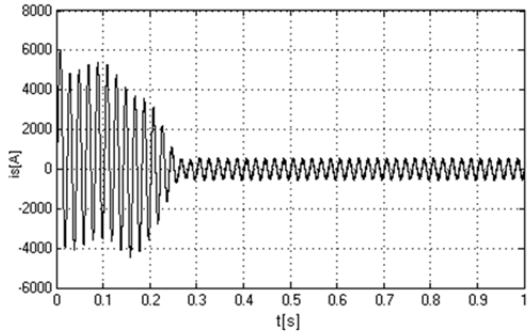

Fig. 14. The current of motor stator fed from a converter

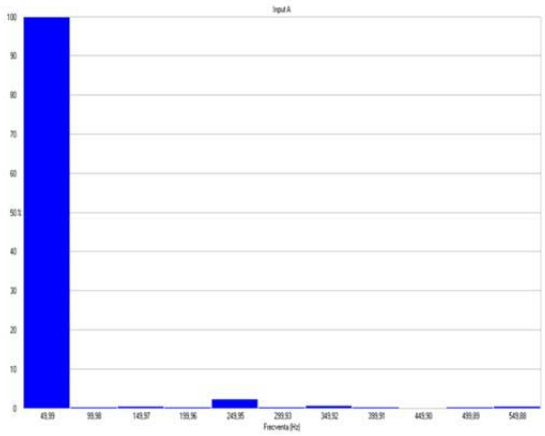

Fig. 15. Fourier decomposition of the phase current with network supplying

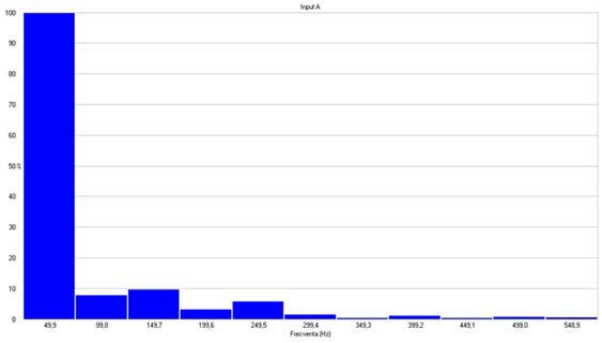

Fig. 16. Fourier decomposition of the phase current with power converter supplying

The harmonic content, shown in Figs. 15 and 16 for the current obtained from the motor fed by the network versus converter, is determined doing the Fourier analysis of the two signals from current.

The total harmonics signal obtained from the analysis shown in figure 15 considering the first 11 harmonics is $THD_{SIN} = 2.48\ \%$; the signal shown in figure 16 is $THD_{CON} = 25.5\ \%$.

5. CONCLUSIONS

The optimal control determines the number of required pulses for the inverter in order to the output values of the fundamental amplitude to be the suitable ones and the number of harmonics to be reduced. Thermal effects occur because of the presence of harmonics of voltage and frequency in the voltage output of converters. The harmonic torques produce effects which can be stationary or pulsed. The required number of pulses of inverter output side depends on the switching limits of the devices used and on the type of control required harmonic content. High order harmonics are generally filtered by the asynchronous motor; this phenomenon is quantified by measuring and calculation of the phase current and total distortion coefficient.

Low order harmonics produce negative effects: limitation or reduction of the functionality of the asynchronous motor.

REFERENCES

[1] Abdulrahman, S.M. – Kettleborough, J.G., Smith, I.R. (1993). Fast Calculation of Harmonic Torque Pulsations in a VSI Induction Motor Drive. *IEEE Transactions on Industrial Electronics*, vol. 40, no. 5, p. 561.

[2] Appelbaum, J., Khan, I.A., Fuchs, E.F., White, J.C. (1987). Optimization of three-phase induction motor design, part 2: The efficiency and cost of an optimal design. *IEEE Transactions on Energy Conversion*, EC-2, No. 3.

[3] Adams, R., Fox., R.S. (1973). Several Modulation Technique for a Pulse Width Modulated Inverter. *IEEE Transactions on Industrial Applications*, vol. IA-8, no. 5, p. 636.

[4] Bose, B.K. (1986). *Power Electronics and AC Drives*, Prentice-Hall, Englewood Cliffs, New Jersey, USA.

[5] Câmpeanu A., Rădulescu, M., Prejbeanu, R, Cauţil I., Câmpeanu T. (2010). Predetermination of Dynamical Stresses of Asynchronous Traction Motors. In: *Proc. of XIX International Conference on Electrical Machines ICEM*, Rome.

[6] Kelemen Á., Imecs, Mária (1991). *Vector Control of AC Drives. Volume 1: Vector Control of Induction Machine Drives*, OMIKK Publisher, Budapest.

[7] Lee, J.Y., Sun. Y.Y (1986). Adaptive Harmonic Control in PWM Inverters with Fluctuating Input Voltage. *IEEE Transactions on Industrial Electronics*, vol. IE-33, no. 1.

[8] Lipo, T. A., Krause, P.C., Jordon, H.E. (1969). Harmonic Torque and Speed Pulsations in a rectifier-inverter Induction Motor Drive. *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 5.

[9] Narayanan, G., Ranganathan, V.T. (1999). Synchronized PWM Strategies Based on Space Vector Approach. Part 1: Principles of Waveform Generation. *IEE Proceedings on Electrical Power Applications*, vol. 146, no. 3.

[10] Williamson, A.C. (1982). The Effects of System Harmonics upon Machines, *International Journal of Electrical Engineering Education*, vol. 19.

[11] Zach, F.C., Thiel, F.A. (1984). *Pulse Width Modulated (PWM) Inverters for Efficiency Optimal Control of ac Drives. Switching Angles and Efficiency/Loss Profiles*. In: Control in Power Electronics and Electrical Drives, Ed. R. Zwicky, Pergamon Press.

[12] *** (1992). *Matlab User's Guide*, The Math Works, Inc. Natick, Massachusetts, USA.

[13] *** (1999). *Texas Instruments, Application Report SPRA524*, USA.

The Stability Analysis of a Lateral-Directional Control System for GAM-Admire Theoretical Aircraft

Claudia Alice State *

* *Doctoral School of Control Engineering and Computers, University of Craiova, A.I.Cuza, 13, Craiova, RO-200585, Romania (e-mail: cldstate@ yahoo.com).*

Abstract:
This paper deals with the results of an investigation that was conducted in order to determine an improved mathematical model, for the lateral-directional dynamic of the GAM (Generic Aerodynamic Model) Admire theoretical aircraft. The considered system is of fourth order, expressing the lateral-directional flight dynamic of the mentioned model. In the stability analysis of the pilot-aircraft system, the human operator is supposed to control the airplane along its longitudinal and vertical axis of rotation, by means of the control surfaces - ailerons and rudder. The dynamic stability of the theoretical GAM-Admire airplane was determined by studying the linearized equations of motion around the equilibrium point. The stationary point was obtained by solving the associated homogeneous nonlinear system and the analytical research was completed by numerical computations. Several modifications of the original GAM-Admire model were conducted in order to eliminate the initial instability of the pilot-aircraft system. The closed-loop control system has been simulated using *MATLAB/Simulink* software package and the results obtained during the simulations under this approach confirm that the automatic control system has a good behavior and the stability is also achieved in the presence of saturation.

Keywords: Lateral-directional stability, Admire, Control system, Linearization, Feedback stabilization, Hurwitz polynomial

1. INTRODUCTION

Developed by Saab AB, the Generic Aerodata Model (GAM) is a theoretical model of a small fighter aircraft, which was used by the Aeronautical Research Institute of Sweden in order to construct a mathematical model for a single-seated fighter with a delta-canard configuration (Admire) (see [Backstrőm (May 1997)] and [L. S. Forssell and Johansson (August 2001)]). The original GAM-Admire system has twelve states (details for this model can be found in [Forssel and Nilsson (2005)]).

The airplane studied in this investigation of lateral-directional dynamic stability is the theoretical GAM-Admire aircraft. The symbolic and numeric computations are performed here for the simplified GAM-Admire model.

The analyzed lateral-directional flight control system consists of an aileron (stabilizer for lateral control) and a rudder (stabilizer for directional control) in order to satisfy the requirements that contribute to the lateral-directional dynamic stabilization.

This paper is organized in five sections as follows. The nonlinear differential equations of the open-loop system, used for describing the lateral-directional motion of the GAM-Admire aircraft, are presented in Section 2. Preliminary results from this aircraft model were first presented in [State (2009)]. The feedback control system for the aircraft lateral-directional dynamic is constructed and the equilibrium point is determined in the same section, by solving analytically the initial nonlinear system. In Section 3, by applying the linearization procedure to the fourth-order nonlinear system, the linearized model around the equilibrium point is obtained. The eigenvalues of the Jacobian matrix (evaluated at the nominal point) of the linearized system, for the mentioned lateral-directional dynamic, are computed and commented in a separated subsection. In Section 4, techniques from the time and frequency domain are presented and applied to both open and closed-loop control systems, in order to determine the linear stability properties of the equilibrium, by the sign of real part of eigenvalues of the Jacobian matrix, and to investigate the stability of the 4th degree polynomials with real coefficients, by using the Routh-Hurwitz stability criterion [Cook (2007)]. Also, the closed-loop control system is analyzed through numerical simulations. Finally, in the Section 5 are presented the main conclusions regarding the obtained results and some perspectives for further research are highlighted.

⋆ This work has been partly supported by the Research Project CNCSIS ID-95.

2. THE SIMPLIFIED PILOT-AIRCRAFT SYSTEM FOR THE LATERAL-DIRECTIONAL DYNAMIC

The simplified lateral-directional GAM-Admire model used here was obtained taking into account [Balint and Balint (2011)] and [Ioniţă et al. (2008)].

2.1 Equations of the lateral-directional GAM-Admire aircraft model

From [Bates and Hagström (2007)], [State (2010)] and [State (2009)], the differential equations of the lateral-directional GAM-Admire aircraft model, with constant speed, are expressed as follows:

$$\begin{cases} \dot{\phi} = p + (q_0 \sin\phi + r\cos\phi) tg\theta_0 \\ \dot{p} = l_p p + l_{\delta_a}\delta_a + l_\beta \beta + (l_r - i_1 q_0)\, r + l_{\delta_r}\delta_r \\ \dot{\beta} = y_\beta \beta - r + y_{\delta_r}\delta_r + \dfrac{g}{V_0}\sin\phi\cos\theta_0 + \alpha_0 p + y_{\delta_a}\delta_a \\ \dot{r} = n_\beta \beta + n_r(\beta) r + n_{\delta_r}\delta_r + (n_p(\beta) + i_3 q_0)\, p + n_{\delta_a}\delta_a \end{cases} \quad (1)$$

For the above system, the state vector x consists of:

- ϕ - roll angle of the aircraft [rad]
- p - roll rate [rad/sec]
- β - sideslip angle [rad]
- r - yaw rate [rad/sec]

and the control input vector u consists of the aileron (δ_a) and rudder (δ_r) positions.

The nonlinear system (1) is analyzed and solved considering the conditions derived from the longitudinal ADMIRE steady-state [Ioniţă et al. (2008)]:

- $\alpha_0 \simeq 0.2[rad]$
- $q_0 = 0$
- $\theta_0 = 0$

This leads to the following first-order nonlinear dynamic system:

$$\begin{cases} \dot{\phi} = p \\ \dot{p} = l_p p + l_{\delta_a}\delta_a + l_\beta \beta + l_r r + l_{\delta_r}\delta_r \\ \dot{\beta} = y_\beta \beta - r + y_{\delta_r}\delta_r + \dfrac{g}{V_0}\sin\phi + \alpha_0 p + y_{\delta_a}\delta_a \\ \dot{r} = n_\beta \beta + n_r(\beta) r + n_{\delta_r}\delta_r + n_p(\beta) p + n_{\delta_a}\delta_a \end{cases} \quad (2)$$

where

$$n_r(\beta) = -a_3(1.07\beta^2 + 0.005) \quad (3)$$

$$n_p(\beta) = a_3(2.865\alpha_0 + 0.3)\beta^2 \quad (4)$$

Table 1. Coefficients for the simplified lateral-directional Admire model

Symbol/Quantity	Symbol/Quantity	Symbol/Quantity
$V_0 = 84.5\frac{m}{s}$	$l_\beta = -6.962$	$n_\beta = 1.674$
$g = 9.81\frac{m}{s^2}$	$l_p = -19.928$	$n_{\delta_a} = 0.941$
$y_\beta = -0.195$	$l_r = 6.247$	$n_{\delta_r} = -1.561$
$y_{\delta_a} = 0.03$	$l_{\delta_a} = 13.665$	$a_1 = 11.96$
$y_{\delta_r} = 0.045$	$l_{\delta_r} = 1.708$	$a_3 = 18.45$

The parameters values for the simplified lateral-directional Admire model are given in Table 1.

The stationary point is evaluated by solving the following system:

$$\begin{cases} \bar{p} = 0 \\ l_p\bar{p} + l_\beta\bar{\beta} + l_r\bar{r} + l_{\delta_a}\bar{\delta}_a + l_{\delta_r}\bar{\delta}_r = 0 \\ \dfrac{g}{V_0}\sin\bar{\phi} + \alpha_0 p + y_\beta\bar{\beta} - \bar{r} + y_{\delta_a}\bar{\delta}_a + y_{\delta_r}\bar{\delta}_r = 0 \\ n_\beta\bar{\beta} + n_p(\bar{\beta})\bar{p} + n_r(\beta)\bar{r} + n_{\delta_a}\bar{\delta}_a + n_{\delta_r}\bar{\delta}_r = 0 \end{cases} \quad (5)$$

The unique solution of the system (5) is determined to be zero, provided that $\bar{\delta}_a = \bar{\delta}_r = 0$.

2.2 The nonlinear model of the closed-loop pilot-aircraft system

Consider the following relations for the input signals to the system controller, in order to introduce the feedback control for improving system dynamic performance:

$$\delta_a = -(k_\phi\phi + k_p p) + \delta_{a_c} \quad (6)$$

$$\delta_r = -(k_\beta\beta + k_r r) + \delta_{r_i} \quad (7)$$

The equations (6) and (7) are introduced for the lateral-directional dynamics of the nonlinear system (1), aiming to obtain the mathematical model of the closed-loop pilot-aircraft system.

The global gains, $k_\phi = 0.277$, $k_p = 0.521$, $k_\beta = 4.93$ and $k_r = -7.673$, can be seen as contributions both from a human operator and, also, from an automatic pilot system.

Starting from the system (2), and taking into account the relations (6) and (7), the following system of differential equations is determined:

$$\begin{cases} \dot{\phi} &= p \\ \dot{p} &= -l_{\delta_a}k_\phi\phi + (l_p - l_{\delta_a}k_p)\, p + l_{\delta_a}\delta_{a_c} \\ &\quad + (l_\beta - l_{\delta_r}k_\beta)\,\beta + (l_r - l_{\delta_r}k_r)\, r + l_r\delta_{r_i} \\ \dot{\beta} &= (y_\beta - y_{\delta_r}k_\beta)\,\beta - (1 + y_{\delta_r}k_r)\, r + y_{\delta_r}\delta_{r_i} \\ &\quad + \left(\dfrac{g}{V_0}\dfrac{\sin\phi}{\phi} - y_{\delta_a}k_\phi\right)\phi + (\alpha_0 - y_{\delta_a}k_p)p + y_{\delta_a}\delta_{a_c} \\ \dot{r} &= (n_\beta - n_{\delta_r}k_\beta)\,\beta + (n_r(\beta) - n_{\delta_r}k_r)\, r + n_{\delta_r}\delta_{r_i} \\ &\quad - n_{\delta_a}k_\phi\phi + (n_p(\beta) - n_{\delta_a}k_p)\, p + n_{\delta_a}\delta_{a_c} \end{cases} \quad (8)$$

For the stability analysis of the resulting nonlinear system (8), it is considered $\delta_{a_c} = 0$, $\delta_{r_i} = 0$, and the steady state of the closed-loop system is again the zero equilibrium.

2.3 The steady-state solution for the lateral-directional pilot-aircraft system

In this section, the stationary point for the 4th order nonlinear system is determined as follows.

In steady state, from the first equation of the associated homogeneous system, results that $\bar{p} = 0$. By making use of the approximation $\sin\bar{\phi} \approx \bar{\phi}$, the system (8) becomes:

$$\begin{cases} -l_{\delta_a}k_\phi\bar{\phi} + (l_\beta - l_{\delta_r}k_\beta)\,\bar{\beta} + (l_r - l_{\delta_r}k_r)\,\bar{r} = 0 \\ \left(\dfrac{g}{V_0} - y_{\delta_a}k_\phi\right)\bar{\phi} + (y_\beta - y_{\delta_r}k_\beta)\,\bar{\beta} - (1 + y_{\delta_r}k_r)\,\bar{r} = 0 \\ -n_{\delta_a}k_\phi\bar{\phi} + (n_\beta - n_{\delta_r}k_\beta)\,\bar{\beta} + (n_r(\beta) - n_{\delta_r}k_r)\,\bar{r} = 0 \end{cases} \quad (9)$$

From the first equation of the system (9) is obtained:

$$\bar{\phi} = \frac{1}{l_{\delta_a}k_\phi}\left[(l_\beta - l_{\delta_r}k_\beta)\,\bar{\beta} + (l_r - l_{\delta_r}k_r)\,\bar{r}\right] \quad (10)$$

Replacing relation (10) into the second and third equations of the system (9), results a system of two equation depending on β and r, as follows:

$$\begin{cases} \left(k_1 + \dfrac{k_2 k_4}{l_{\delta_a} k_\phi}\right)\bar{\beta} + \left(\dfrac{k_4 k_6}{l_{\delta_a} k_\phi} - k_5\right)\bar{r} = 0 \\ \left(k_3 - \dfrac{n_{\delta_a}}{l_{\delta_a}} k_2\right)\bar{\beta} + \left(k_7 - \dfrac{n_{\delta_a}}{l_{\delta_a}} k_6\right)\bar{r} = 0 \end{cases} \tag{11}$$

where the following notations have been made:

$$\begin{aligned} k_1 &= y_\beta - y_{\delta_r} k_\beta \\ k_2 &= l_\beta - l_{\delta_r} k_\beta \\ k_3 &= n_\beta - n_{\delta_r} k_\beta \\ k_4 &= \frac{g}{V_0} - y_{\delta_a} k_\phi \\ k_5 &= 1 + y_{\delta_r} k_r \\ k_6 &= l_r - l_{\delta_r} k_r \\ k_7 &= n_r(\bar{\beta}) - n_{\delta_r} k_r \end{aligned} \tag{12}$$

with

$$n_r(\bar{\beta}) = -a_3(c_2\bar{\beta}^2 + c_0) \tag{13}$$

From the first equation of the system (11) it follows that

$$\bar{r} = \frac{k_1 l_{\delta_a} k_\phi + k_2 k_4}{k_5 l_{\delta_a} k_\phi - k_4 k_6}\bar{\beta} \tag{14}$$

Substituting relation (14) into the second equation of the system (11), results a 3rd degree equation in one variable (β):

$$\bar{\beta}\left[\bar{\beta}^2 + \frac{k_\phi l_{\delta_a}(k_1\rho_1 - k_5\rho_2) + k_4(k_2\rho_1 + k_6\rho_2)}{a_3 c_2 l_{\delta_a}(k_1 l_{\delta_a} k_\phi + k_2 k_4)}\right] = 0 \tag{15}$$

where the following notations were employed:

$$\rho_1 = a_3 c_0 l_{\delta_a} + n_{\delta_r} k_r l_{\delta_a} + n_{\delta_a} k_6 \tag{16}$$

$$\rho_2 = k_3 l_{\delta_a} - n_{\delta_a} k_2 \tag{17}$$

From the above equation is derived that

$$\bar{\beta}_0 = 0 \tag{18}$$

Table 2. Elements of the free term with their signs for the 2nd degree equation of (15)

Element/Sign	Element/Sign
$k_1 < 0$	$k_\phi l_{\delta_a} > 0$
$k_2 < 0$	$\rho_1 > 0$
$k_4 > 0$	$\rho_2 > 0$
$k_5 > 0$	$\lvert k_2\rho_1\rvert - \lvert k_6\rho_2\rvert > 0$
$k_6 > 0$	$a_3 c_2 l_{\delta_a} > 0$

Considering the previous table, the free term of the 2nd degree equation from the parenthesis of (15) has a positive sign.

From the above considerations results that the closed-loop system does not admit other solutions different from the trivial one.

3. THE LINEARIZATION PROCEDURE APPLIED TO THE ROLL-YAW DYNAMICS

In this section, the linearization of system (2), used for describing the lateral-directional motion of the GAM-ADMIRE aircraft, is performed for the purpose to determine the stability of both closed and open-loop control systems. The obtained stationary point, from the previous section, is used in the linearization process to approximate the first-order nonlinear dynamic system.

3.1 Linearization around the equilibrium point for the open-loop system

In order to investigate the stability of the trivial equilibrium point, the following system of equations after the first approximation is considered, instead of the original nonlinear one:

The linear differential model, formulated in state-space, is:

$$\Delta\dot{x} = \frac{\partial f}{\partial x}(\bar{x})\Delta x + \frac{\partial f}{\partial u}(\bar{u})\Delta u \tag{19}$$

where

- $\Delta x = [\,\Delta\phi\ \Delta p\ \Delta\beta\ \Delta r\,]^T$ is the state vector;
- $\Delta u = \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix}$ represents the control vector;
- the vector function $f(x,u)$ is given by

$$f(x,u) = \begin{bmatrix} p \\ l_p p + l_{\delta_a}\delta_a + l_\beta\beta + l_r r + l_{\delta_r}\delta_r \\ y_\beta\beta - r + y_{\delta_r}\delta_r + \dfrac{g}{V_0}\sin\phi + \alpha_0 p + y_{\delta_a}\delta_a \\ n_\beta\beta + n_r(\beta)r + n_{\delta_r}\delta_r + n_p(\beta)p + n_{\delta_a}\delta_a \end{bmatrix} \tag{20}$$

To assess the stability of the open-loop system, the Jacobian matrix of the column vector function f and the characteristic polynomial, $P(\lambda) = |\lambda I - A|$, have to be determined.

The state-space model consists of the matrices A and B that contain dimensional stability and control derivatives as defined below:

$$A = \frac{\partial f}{\partial x}(\bar{x}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & l_p & l_\beta & l_r \\ \dfrac{g}{V_0}\cos\bar{\phi} & \alpha_0 & y_\beta & -1 \\ 0 & n_p(\bar{\beta}) & N_\beta & n_r(\bar{\beta}) \end{bmatrix} \tag{21}$$

where

$$l_\beta = a_1\left(0.896\alpha_0^2 - 0.47\alpha_0 - 0.04\right) \tag{22}$$

$$l_r = a_1\left(0.344\alpha_0 + 0.02\right) \tag{23}$$

$$N_\beta = n_\beta + 2a_3\left[(2.865\alpha_0 + 0.3)\,\bar{p} - 1.07\bar{r}\right]\bar{\beta} \tag{24}$$

and the input matrix

$$B = \frac{\partial f}{\partial u}(\bar{u}) = \begin{bmatrix} 0 & 0 \\ l_{\delta_a} & l_{\delta_r} \\ y_{\delta_a} & y_{\delta_r} \\ n_{\delta_a} & n_{\delta_r} \end{bmatrix} \tag{25}$$

The linearized model around the stationary point, obtained from (19), (21) and (25), is:

$$\begin{cases} \Delta\dot{\phi} = \Delta p \\ \Delta\dot{p} = l_p\Delta p + l_\beta\Delta\beta + l_r\Delta r + l_{\delta_a}\Delta\delta_a + l_{\delta_r}\Delta\delta_r \\ \Delta\dot{\beta} = \dfrac{g}{V_0}\cos\bar{\phi}\Delta\phi + y_\beta\Delta\beta - \Delta r + y_{\delta_a}\Delta\delta_a + y_{\delta_r}\Delta\delta_r \\ \Delta\dot{r} = n_p(\beta)\Delta p + N_\beta\Delta\beta + n_r(\beta)\Delta r + n_{\delta_a}\Delta\delta_a + n_{\delta_r}\Delta\delta_r \end{cases} \tag{26}$$

For the linearized system (26), the following relation occurs (for $\bar{\phi} = 0$):

$$cos\bar{\phi} = 1 \tag{27}$$

3.2 Theoretical considerations for linear stability

The matrix $A = \overline{\frac{\partial f}{\partial x}}$ defines the first order (linear) approximation of the system around the stationary solution $\bar{x}$.

The stability conditions for the homogeneous linearized system with constant coefficients is formulated in the terms of the eigenvalues of the Jacobian matrix ([Vrabie (2011)], pp.167-173, and [Chicone (1999)], pp.174-177).

Theorem 1. A linear homogeneous system with constant coefficients is *asymptotically stable* if and only if all eigenvalues of its matrix have negative real parts.

Remark 2. The matrix A that satisfies the Theorem 1 is called *Hurwitzian* if all the roots of the characteristic equation $\det(sI - A) = 0$ have negative real parts.

Theorem 3. (*Linear system stability*). A linear system of differential equations is *asymptotically stable* if and only if its matrix A is Hurwitzian. If some roots of the characteristic equation $\det(sI - A) = 0$ have strictly negative real part, and the remaining roots are purely imaginary, but simple, then the system is *stable*. If there is at last one root of the characteristic equation $\det(sI - A) = 0$, with strictly positive real part, then the zero solution of the system is *unstable*.

Regarding the roots of the characteristic equation $\det(sI - A) = 0$, there is the Hurwitz stability criterion that gives all necessary and sufficient conditions (which must be satisfied by the coefficients of the characteristic polynomial), for all its roots have negative real part.

In the following, for the stability analysis of the linearized system, the positions in the complex plane of the eigenvalues and the roots of the characteristic polynomial associated with the matrix A, are investigated.

3.3 Numerical considerations for the open-loop linearized system

For evaluating the Jacobian matrices around the equilibrium point, the numerical values specified in Table 1, are replaced. Thus, the matrix A of the open-loop system has the following values for the lateral-directional dynamics of the aircraft:

$$A = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -19.9277 & -6.9617 & 6.2465 \\ 0.1161 & 0.1970 & -0.1949 & -1.0000 \\ 0 & 0 & 1.6736 & -0.0923 \end{bmatrix} \tag{28}$$

The input matrix B is defined to have the following values:

$$B = \begin{bmatrix} 0 & 0 \\ 13.6647 & 1.7081 \\ -0.0296 & 0.0448 \\ 0.9410 & -1.5610 \end{bmatrix} \tag{29}$$

The eigenvalues obtained from the matrix A are:

$$\lambda = \begin{bmatrix} -19.8551 \\ -0.1971 + 1.2723i \\ -0.1971 - 1.2723i \\ 0.0346 \end{bmatrix} \tag{30}$$

This means that the linearized system is open-loop unstable (one of its eigenvalues is in the right-half plane).

4. STABILITY ANALYSIS OF THE PILOT-AIRCRAFT SYSTEM. FEEDBACK STABILIZATION

In this section, the stability of the linearized system around the equilibrium point is studied.

4.1 Routh-Hurwith criterion for stability of the open-loop system

The algebraic Routh-Hurwitz criterion (presented in [Cook (2007)], pp. 227) allows to appreciate the stability of the system on the base of the coefficients of the characteristic equation $det(sI - A) = 0$, without actually solving the equation.

For the nonlinear system (8), considering the feedback zero ($k_\phi = k_p = k_\beta = k_r = 0$), the open-loop stability is analysed for the following 4th order system:

$$\begin{cases} \dot{\phi} = p \\ \dot{p} = l_p p + l_\beta \beta + l_r r \\ \dot{\beta} = y_\beta \beta - r + \dfrac{g}{V_0}\phi + \alpha_0 p \\ \dot{r} = n_\beta \beta - a_3 c_0 r \end{cases} \tag{31}$$

The fourth-degree polynomial equation for the open-loop system (31) is:

$$P(s) = t_4 s^4 + t_3 s^3 + t_2 s^2 + t_1 s + t_0 = 0 \tag{32}$$

where the following notations were made:

- $t_4 = 1$
- $t_3 = a_3 c_0 - y_\beta - l_p$
- $t_2 = n_\beta - a_3 c_0 y_\beta - l_p(a_3 c_0 - y_\beta) - \alpha_0 l_\beta$
- $t_1 = l_p(a_3 c_0 - n_\beta) - \alpha_0(l_\beta a_3 c_0 + n_\beta l_r) - \frac{g}{V_0} n_\beta l_r$
- $t_0 = -\frac{g}{V_0} l_\beta a_3 c_0$

For the fourth-order characteristic polynomial (32), the Routh-Hurwith criterion is mainly used to determine the stability property for the open-loop system.

The necessary but not sufficient condition for stability is that all the polynomial coefficients must be strictly positive (equivalently all of them must be strictly negative, with no sign change).

$$t_i > 0, i = \overline{0,4} \tag{33}$$

For the 4th order system (31), the Routh-Hurwits criterion gives the following sufficient conditions of the stability:

$$t_2 t_3 - t_1 t_4 > 0 \tag{34}$$

$$t_1(t_2 t_3 - t_1 t_4) - t_0 t_3^2 > 0 \tag{35}$$

Substituting numerically the aerodynamic constants of the system (31), it can be verified that the coefficients of the fourth-order degree polynomial $P(s)$ doesn't satisfy $t_0 > 0$ (the necessary condition of the stability is not fulfilled).

4.2 The BIBO stability analysis of the linearised model around the unique equilibrium

The state-space model of nonlinear multivariable system consists of:

(1) the state equation, $\dot{x} = f(x,u)$, representing the vector form of the difference equations set;
(2) the output equation, $y = g(x)$, representing the vector form of the algebraic equations.

To get the characteristic polynomial for the closed-loop system, we consider the state equations of linearized time-invariant system as

$$\Delta\dot{x} = A\Delta x + B\Delta u \tag{36}$$

where the Jacobian matrix of the open loop system is denoted by A.

By linearization, the nonlinear model $y = g(x)$ is replaced by the linear model $\Delta y = K\Delta x$.

To construct the linear dynamic system, represented by the input-state-output model:

$$\begin{cases} \Delta\dot{x} = A\Delta x + B\Delta u \\ \Delta y = K\Delta x \end{cases} \tag{37}$$

it is obtained

$$\Delta u = -K\Delta x \tag{38}$$

where the feedback gain matrix is

$$K = \begin{bmatrix} k_\phi & k_p & 0 & 0 \\ 0 & 0 & k_\beta & k_r \end{bmatrix} \tag{39}$$

The relation (38) is substituted into the first equation of the system (37), and it is determined

$$\Delta\dot{x} = (A - BK)\,\Delta x \tag{40}$$

The characteristic polynomial for the closed-loop system is

$$Q(s) = |sI_4 - J| \tag{41}$$

in which the Jacobian matrix of the closed loop system (8), denoted by J, is:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -l_{\delta_a}k_\phi & l_p - l_{\delta_a}k_p & l_\beta - l_{\delta_r}k_\beta & l_r - l_{\delta_r}k_r \\ \dfrac{g}{V_0} - y_{\delta_a}k_\phi & \alpha_0 - y_{\delta_a}k_p & y_\beta - y_{\delta_r}k_\beta & -1 - y_{\delta_r}k_r \\ -n_{\delta_a} & -n_{\delta_a}k_p & n_\beta - n_{\delta_r}k_\beta & -a_3c_0 - n_{\delta_r}k_r \end{bmatrix}. \tag{42}$$

4.3 Numerical considerations for the closed-loop system

The numerical values of the Jacobian matrix J, for the closed-loop system (8), are given below:

$$J = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ -3.7851 & -27.0470 & -15.3825 & 19.3526 \\ 0.1243 & 0.2124 & -0.4159 & -0.6560 \\ -0.2607 & -0.4903 & 9.3693 & -12.0698 \end{bmatrix} \tag{43}$$

The solutions of the characteristic equation, $det(sI - J) = 0$, are the eigenvalues of the matrix J. This gives

$$\lambda = \begin{bmatrix} -26.0346 \\ -0.1528 \\ -0.9653 \\ -12.3800 \end{bmatrix} \tag{44}$$

which are all in the left half-plane.

Therefore, the linearized system around the equilibrium point is closed-loop asymptotically stable.

Developing and substituting numerically the coefficients from relation (41), the following characteristic polynomial of the closed-loop system is obtained:

$$Q(s) = a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0 \tag{45}$$

The numerical values of the polynomial coefficients are: $a_0 = 47.5320$, $a_1 = 366.0329$, $a_2 = 365.4068$, $a_3 = 39.5327$ and $a_4 = 1$.

To provide conditions that are both necessary and sufficient for all roots of the characteristic polynomial $Q(s)$ to be located in the left-half plane, the Routh-Hurwitz analysis involves creating the Routh array from the polynomial coefficients, that has the following structure:

$$\begin{bmatrix} a_4 & a_2 & a_0 \\ a_3 & a_1 & 0 \\ b_1 & b_2 & 0 \\ c_1 & 0 & 0 \\ d_1 & 0 & 0 \end{bmatrix} \tag{46}$$

where the elements from the last two rows, b_1, b_2 and c_1, are defined as follows:

$$b_1 = \frac{a_3a_2 - a_4a_1}{a_3} \tag{47}$$

$$b_2 = a_0 \tag{48}$$

$$c_1 = \frac{b_1a_1 - a_3b_2}{b_1} = \frac{(a_3a_2 - a_4a_1)a_1 - a_3^2a_0}{a_3a_2 - a_4a_1} \tag{49}$$

$$d_1 = b_2 = a_0 \tag{50}$$

The Rouths stability criterion states that the number of roots of the characteristic equation with positive real parts (unstable) is given by to the number of changes of sign in the first column of the array.

Thus, the system stability is ensured if and only is all the elements in the first column of the Routh array have the same sign.

Putting the coefficients of the 4th degree polynomial into the Routhian array, and doing some computations, the following table is obtained:

$$\begin{vmatrix} 1.0000 & 365.4068 & 47.5320 \\ 39.5327 & 366.0329 & 0 \\ 356.1478 & 47.5320 & 0 \\ 360.7568 & 0 & 0 \\ 47.5320 & 0 & 0 \end{vmatrix} \tag{51}$$

Remark 4. The first two lines of the table (51) are formed from the coefficients of the polynomial $Q(s)$, and the next three are calculated according to the relations of (46), from (47) through (50).

By analyzing the relation (51), there are no sign changes in the first column of the array. Therefore, all the roots the characteristic polynomial $Q(s)$ are in the left-half of the s-plane. This means that the linearized dynamical system (2), described by the transfer function $G(s) = \dfrac{H(s)}{Q(s)}$, is stable.

5. MATLAB SIMULATIONS OF THE LATERAL-DIRECTIONAL MOTION

This section describes the graphical outputs of the lateral-directional MATLAB/Simultink model that simulates the flight behaviour of the ADMIRE aircraft during flight maneuvers.

5.1 SIMULINK non-linear system and simulation results

The general representation of the improved control system and the time simulation results for the nonlinear closed-loop system (8) are presented in the following figures:

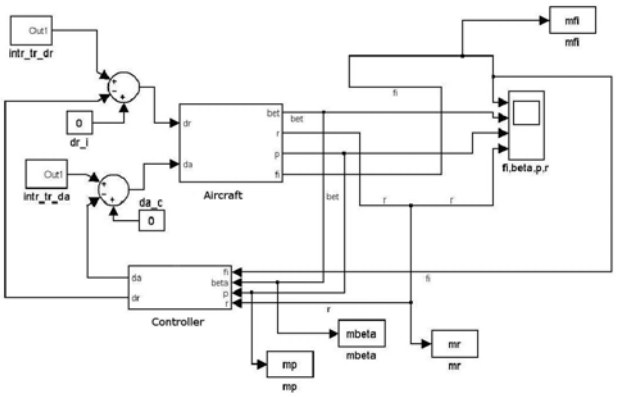

Fig. 1. The simplified block diagram of the nonlinear control system (8)

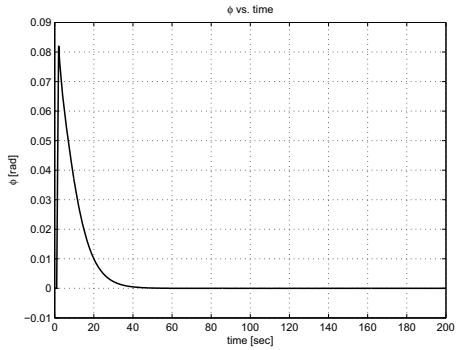

Fig. 2. The graphical representation of the roll angle ϕ

The stability is determined by the behavior of the nonlinear system in the immediate vicinity of the equilibrium point.

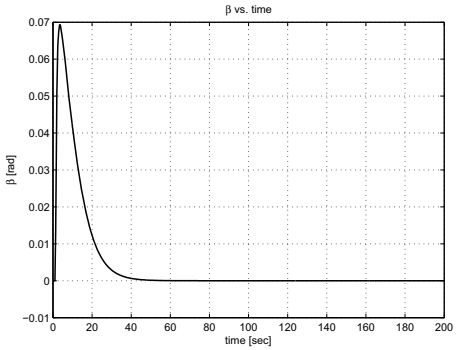

Fig. 3. The graphical representation of the sideslip angle β

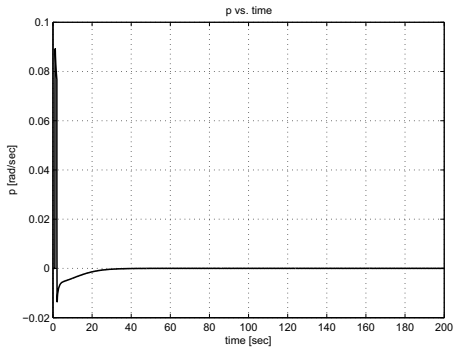

Fig. 4. The graphical representation of the roll rate p

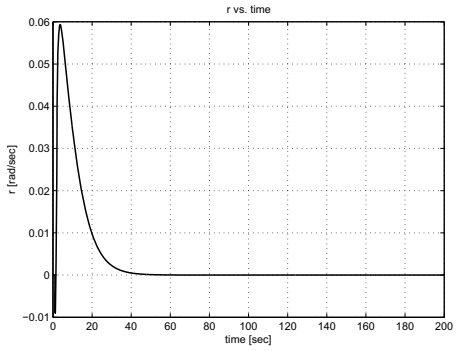

Fig. 5. The graphical representation of the yaw rate r

5.2 SIMULINK linearized system and simulation results

The analysis of simulation results is extended, by considering the control saturation effects, in order to determine the changes in stability.

The Simulink schema of the linearized closed-loop system with saturations is presented as follows:

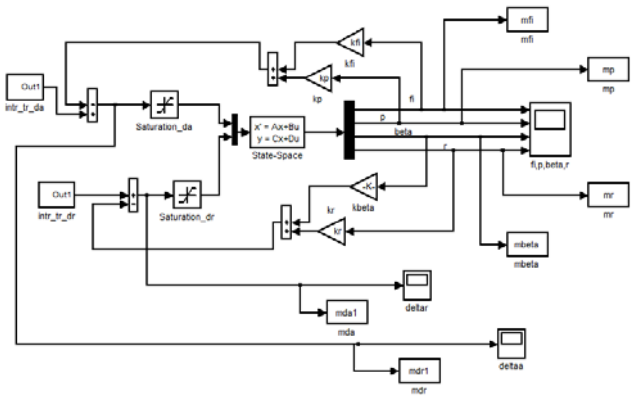

Fig. 6. The block diagram of the linearized system (37), with saturations

In the presence of saturation, the following simulation results were obtained for the linearized Admire model:

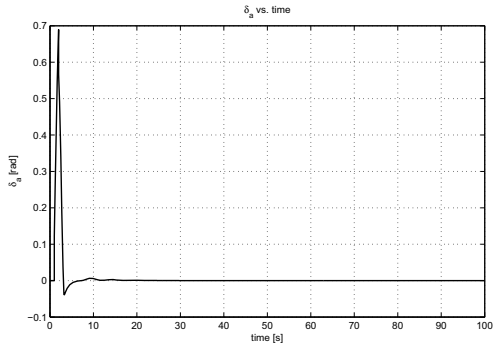

Fig. 7. Aileron deflection δ_a

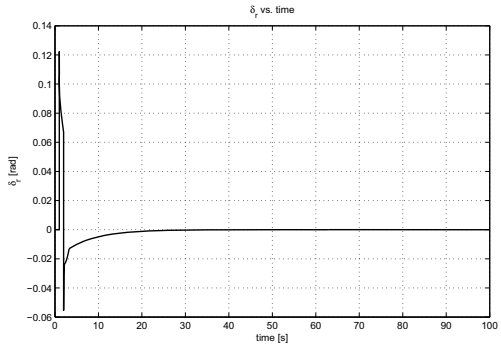

Fig. 8. Rudder deflection δ_r

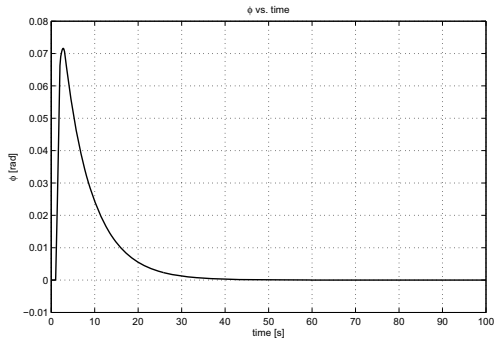

Fig. 9. Time response of the roll angle ϕ, with saturation

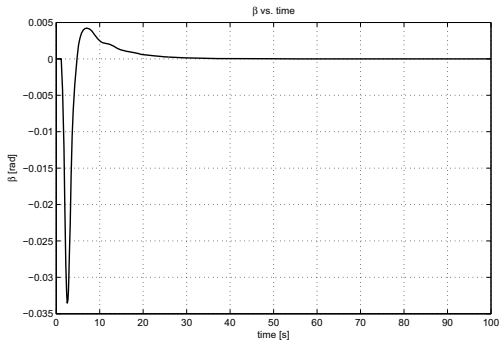

Fig. 10. Time response of sideslip angle β, with saturation

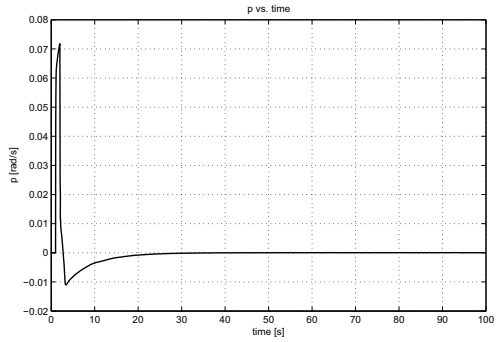

Fig. 11. Time response of the roll rate p, with saturation

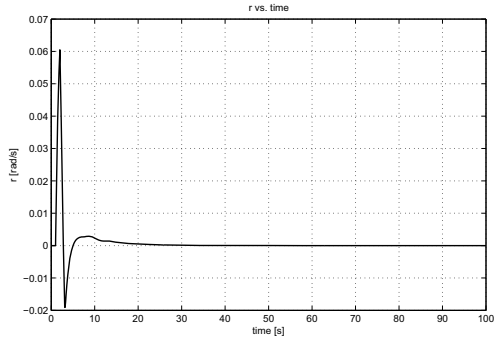

Fig. 12. Time response of the yaw rate r, with saturation

6. CONCLUSION

From the determination of the analytical solution for the lateral-directional control system, results that the equilibrium point of the open-loop system, used to approximate the nonlinear system, coincides with the stationary point of the closed-loop system. Consequently, the system has an unique equilibrium point at the origin.

The initial lateral-directional system considered in this study is open-loop unstable. Accordingly, the usage of the feedback controller was necessary in order to control the dynamical system, to stabilize and improve the performance of the pilot-aircraft system.

The simulation results obtained and presented in this paper demonstrate that the closed-loop control system (with negative feedback) has a good performance even in the presence of saturation.

As a future work, by using a rate saturated actuator, a more realistic mathematical model might be developed to describe the aircraft lateral-directional dynamics.

REFERENCES

Backstrőm, H. (May 1997). Report on the usage of the generic aerodata model. *AAB Aircraft AB, SAAB Military Aircraft, S-58199 Linkping Sweden.*

Balint, A. and Balint, S. (2011). Oscillations susceptibility of an unmanned aircraft whose automatic flight control system fails. *In Advances in Flight Control Systems*, InTech, 1st edition, 275–296.

Bates, D. and Hagström, M. (2007). The admire benchmark aircraft model. In *Nonlinear Analysis and Synthesis Techniques for Aircraft Control. Series: Lecture Notes in Control and Information Sciences*, volume 365, 35–54. Springer-Verlag Berlin and Heidelberg GmbH Co, New York, 1st edition.

Chicone, C. (1999). *Ordinary Differential Equations with Applications.* Texts in Applied Mathematics, vol. 34. Springer, New York.

Cook, M.V. (2007). *Flight Dynamics Principles, Second Edition: A Linear Systems Approach to Aircraft Stability and Control.* Oxford UK; Burlington, MA: Butterworth-Heinemann/Elsevier, UK.

Forssel, L. and Nilsson, U. (2005). *ADMIRE The Aero-Data Model In a Research Environment, Version 4.0, Swedish Defence Re Description.* Report FOI-R–1624–SE ISSN-1650-1942.

Ioniţă, A., Balint, A.M., and Balint, S. (2008). Limit cycle behaviour in aircraft longitudinal terminal phase. *7th International Conference on Mathematical Problems in Engineering*, Aerospace and Sciences, Genova, 276–287.

L. S. Forssell, G. Hovmark, A.H. and Johansson, F. (August 2001). The aero-data model in a research environment (admire) for flight control robustness evaluation. *GARTUER/TP-119-7.*

State, C.A. (2009). The influence of saturation relative to the aileron control in the roll motion. *Proceedings of the XXXIInd Caius Iacob National Conference on Fluid Mechanics and its Technical Applications*, INCAS - National Institute for Aerospace Research "Elie Carafol", Bucharest, 216–223.

State, C.A. (2010). Numerical considerations relative to simulation of aileron-ruder controls in the roll-yaw motion. *INCAS BULLETIN*, 2(1), Bucharest, 81–85 (DOI: 10.13111/2066–8201.2010.2.1.10).

Vrabie, I.I. (2011). *Differential Equations. An introduction to basic results, concepts and applications.* Second Edition, World Scientific, New Jersey-London-Singapore-Beijing-Shanghai-Hong Kong-Taipei-Chennai, xxi+460 pp. ISBN 981-4335-62-2.

Using Variable Precision Rough Sets Theory to Classify Students' Results

Anca Loredana Udristoiu*, Stefan Udristoiu*

**University of Craiova, Faculty of Automation, Computers and Electronics, Romania*
(phone: 40-251-438-198; fax: 40-251-438-198; e-mail: aion@software.ucv.ro).

Abstract: This paper proposes a method for predicting student scholar performance based on the utilization of precision variable rough set theory. The study presents how to deal with an incomplete e-learning system in order to generate interesting rules capable of classification. The variable precision rough set theory is a powerful approach that permits the searching for patterns in an incomplete e-learning database. The results of experiments are very promising and show that the methods based on variable precision rough set theory are very useful for predicting the results of the student during a course activity.

Keywords: Variable precision rough sets, Data mining, Electronic learning.

1. INTRODUCTION

Predicting students' performance is one of the most important and useful applications of educational data mining and its goal is to score or mark the student from student course behavior and activity (Romero et al., 2013). In this study, the variable precision rough set (VPRS) technique is used to offer methods for understanding, processing and modelling data, resolving the limitations of the incomplete e-learning systems.

The classical rough set theory (RST) was discovered by Zdzislaw Pawlak (Pawlak et. al, 1994) and is a powerful mathematical tool for modelling the imperfect and incomplete knowledge (Stepaniuk, 2008; Michalski, 1983). Rough set theory has also excellent results in approximate reasoning (Wang et al., 2012), mathematical logic analysis and reduct (Ren et al., 2011; Zaras et al., 2012; Ke et al., 2011), building of predictive models (Lai et al., 2005), and decision support system (Chaoi et al., 2011; Hossam, 2011; Qu et al., 2009). Many studies have shown that the use of rough set theory formulate a clear decision-making projects and enhance the effectiveness of the research while doing optimization (Zaras et al., 2012). The research related to education of Qu and Wang (Qu et al., 2009) provided a basis of personalized teaching strategies in distance learning website by analysis of reduct and attribute significance. In (Sheu et al., 2013) the study analyzed students' misconception based on rough set theory. Although the rough set theory is rarely used in education, in this study we use its characteristics, which are very suitable for discovering rules useful in educational process.

Due to some incapability of RST to work with noise data, Ziarko (Ziarko, 1993) proposed the variable precision model to deal with noisy data and uncertain information. VPRS model is an extension of the classical RST as a tool for classification of objects. VPRS deals with partial classification by introducing a probability value β. The β represents a bound on the conditional probability of a proportion of objects in a condition class which are classified into the same decision class (Gong et al., 2012).

Ziarko (Ziarko, 1993) considered β as a classification error, defined to be in the interval [0; 0.5) and his model degenerates into the classical rough set model if β= 0. Some researchers have studied VPRS models and got some meaningful results (Beynon, 2001; Liu et al., 2008; Gong et al., 2012).

The purpose of this paper is to combine the VPRS model and incomplete e-learning system. By calculating discernibility matrix based on β -lower approximation, the β-reducts and decision rules capable to mark students' activity are obtained.

We firstly present how we model the information about students 'activity in the e-learning system in Section 2. In Section 3 basic notions of RST and VPRS models and incomplete information system are introduce to facilitate the understanding. Methods for computing the discernibility matrix and discernibility functions for incomplete decision table are given in Section 4. An illustrative example is analyzed in Section 5 to show the feasibility of the proposed approach. Results and comparison are summarized in Section 6.

2. STUDENT REPRESENTATION AND DISCRETIZATION

We have collected data from on-line course activity provided by Moodle (Cole, 2005) that is one of the most widely used open source learning management system. In fact, we have used the following data based on student 'Database' course activity (Romero et al., 2013; Udristoiu et al., 2014):

-Nassignment – number of assignments taken;
-Nquiz - number of quiz taken;
-Nquiz_p - number of quiz passed;
-Nquiz_f - number of quiz failed;

-Nmessages - number of messages sent to the chat;
-Nmessages_ap - number of messages sent to the teacher;
-Nposts - number of messages sent to the forum;
-Nread - number or forum messages read;
-Total_time_assignment- total time spent on assignment;
-Total_time_quiz – total time used in quizzes;
-Total_time_forum- total time used in forum;
-Mark- final mark the student obtained in the course.

Since the data provided by Moodle are structured, they didn't necessitate preparation (Romero et al., 2013). So, we directly discretize them, transforming numerical values into categorical ones for a good interpretation and understanding. We have used the manual method for discretizing all attributes, so the teacher has to specify the cut off points. The mark descriptor has four values: insufficient, if value < 5; average, if value > 5 and < 7; good if value >7 and < 9; excellent if value > 9. The other attributes have the values: LOW, MEDIUM and HIGH (Romero et al., 2013; Udristoiu et al., 2014).

A student is represented in Prolog by means of a term:

student(ListofDescriptors)

where the argument is a list of terms used to specify the student attributes.

The term used to specify the student attributes is of the form:

descriptor(DescriptorName,DescriptorValue)

The model representation of a student is in the following example:

```
student([
descriptor(Nassignment, medium), descriptor(Nquiz, null),
descriptor(Nquiz_p,low),descriptor(Nquiz_f,high),
descriptor(Nmessages,medium),
descriptor(Nmessages_ap, medium),
descriptor(Nposts,low),descriptor(Nread,low),
descriptor(Total_time_assignment, low),
descriptor(Total_time_quiz, null),
descriptor(Total_time_forum, low)]).
```

3. MODELLING OF STUDENT INFORMATION USING ROUGH SETS

3.1 Rough Sets

Rough sets theory is an intelligent mathematical tool and it is based on the concept of approximation space (Pawlak et. al, 1994; Hassanien et al., 2008). In rough sets theory, the notion of information system determines the knowledge representation system.

In this section, we recall some basic definitions from literature (Pawlak et. al, 1994; Hassanien et al., 2008).
Let U denote a finite non-empty set of objects (students) called the universe. Further, let A denote a finite non-empty set of attributes. Every attribute $a \in A$, there is a function a: U → Va, where Va is the set of all possible values of a, to be called the domain of a. A pair IS = (U, A) is an information system. Usually, the specification of an information system can be presented in tabular form. Each subset of attributes $B \subseteq A$ determines a binary *B–indiscernibility* relation *IND(B)* consisting of pairs of objects indiscernible with respect to attributes from B like in (1):

$$\mathrm{IND(B)} = \{(x, y) \in U \times U : \forall a \in B, a(x) = a(y)\} \quad (1)$$

IND(B) is an equivalence relation and determines a partition of U, which is denoted by *U/IND(B).* The set of objects indiscernible with an object $x \in U$ with respect to the attribute set, *B,* is denoted by $I_B(x)$ and is called *B-indiscernibility* class:

$$\mathrm{I_B(x)} = \{y \in U : (x, y) \in \mathrm{IND(B)}\} \quad (2)$$

$$\mathrm{U/IND(B)} = \{\mathrm{I_B(x)} : x \in U\} \quad (3)$$

It is said that a pair AS_B = *(U, IND(B))* is an approximation space for the information system *IS=(U, A)*, where $B \subseteq A$.

The *B-lower approximation X,* $\underline{B}X$, is the union of all equivalence classes in *IND(B)* which are contained by the target set *X.* The lower approximation of *X* is called the positive region of *X* and is noted *POS(X).*

$$\underline{B}X = \bigcup \{ w_i \mid w_i \subseteq X \} \quad (4)$$

The *B-upper approximation* $\overline{B}X$ is the union of all equivalence classes in *IND(B)* which have non-empty intersection with the target set *X.*

$$\overline{B}X = \bigcup \{ w_i \mid w_i \cap X \neq \varnothing \} \quad (5)$$

The accuracy of a rough set is defined as:

$$\alpha_B = \text{cardinality}(\underline{B}X) / \text{cardinality}(\overline{B}X) \quad (6)$$

If the accuracy is equal to 1, then the approximation is perfect.

3.2 Variable precision rough sets

Variable precision rough set (VPRS) extends rough set theory by the relaxation of the subset operator (Ziarko, 1993). So, the objects are classified with an error smaller than a certain pre-defined level.

We recall some definitions from the literature (Ziarko, 1993; Beynon, 2001; Liu et al., 2008; Gong et al., 2012).
Let $X, Y \subseteq U$ be two non- empty subsets of the universe U. The error classification rate of X relative to Y is denoted by e(X, Y), is defined by:

$$e(X,Y) = \begin{cases} 1 - \dfrac{|X \cap Y|}{|X|}, & |X| > 0 \\ 0, & |X| = 0 \end{cases} \quad (7)$$

Let β be a real number within the range $0 \le \beta < 0.5$, X be a subset of the universe U, and B an attribute subset of A. Then, the β-lower and β-upper approximations are:

$$\underline{B}_{\beta}X = \{x \in U / e(x, X) \le \beta\} \quad (8)$$

$$\overline{B}_{\beta}X = \{x \in U / e(x, X) > 1 - \beta\} \quad (9)$$

The accuracy of approximation variable precision (accuracy of variable precision roughness) of any subset $X \subseteq U$ with respect to $B \subseteq A$ is:

$$\alpha_{B_{\beta}} = \frac{\left|\underline{B}_{\beta}X\right|}{\left|\overline{B}_{\beta}X\right|} \quad (10)$$

4. INCOMPLETE E-LEARNING SYSTEM

Due to student's selection of activities, in e-learning systems it may happen that some attribute values for an object are missing. To indicate such a situation, a distinguished value, so-called null value, is usually assigned to those attributes (Gong et al., 2012).

Let *(U, Q, V, f)* be an incomplete information system. For attribute subset $A \subseteq Q$, the tolerance relation R_A is defined as:

$$R_A = \bigcap_{a \in A} \{(x, y) \in U \times U / a(x) = a(y) \text{ or } a(x) = \text{null} \text{ or } a(y) = \text{null}\} \quad (11)$$

The set of all objects possibly indiscernible with x is defined as in (12) and is called tolerance class of x:

$$S_A(x) = \{y \in U / (x, y) \in R_A\} \quad (12)$$

Also, $U / R_A = \{S_A(x_1), S_A(x_2), .., S_A(x_n)\}$.

A subset $A \subseteq Q$ is a reduct of information system iff $U / R_A = U / Q$ and for any $B \subseteq A$, $U / R_B \neq U / Q$.

In our case, the incomplete information system is of the form: *(U, C ∪ D, V, f)* where *C* represents the set of conditional attributes, and *D* the set of decision attributes.

The β-dependency degree between *A* and *D*, where $A \subseteq C$ is defined:

$$\gamma(A, D, \beta) = \frac{\left|POS(A, D, \beta)\right|}{\left|U\right|} \quad (13)$$

where

$POS(A, D, \beta) = \{x / x \in U, e(S_A(x), E) \leq \beta\}, E \in U / D$.

In our study we determined the value of β as in (Gong et al., 2012).

With respect to a subset of conditional attributes A, where $A \subseteq C$, the discernible threshold β can be computed as in (14):

$$\beta = \max(m_1, m_2) \quad (14)$$

where

$$m_1 = 1 - \min\{e(S_A(x), X) / e(S_A(x), X) > 0.5\}$$

$$m_2 = \max\{e(S_A(x), X) / e(S_A(x), X) < 0.5\}$$

4.1 Incomplete decision table, discernibility matrix and discernibility functions

An important notion used in rough set theory is the decision table (Pawlak et. al, 1994).

Incomplete decision table is *DT = (U, C ∪ D, V, f)*, where *C* is a set of distinguished conditional attributes and *D* is a set of decision attributes.

We transform the decision table into discernibility matrix to compute the reducts. Let *DT = (U, C, D)* be the decision table, with *U = {R1, R2, R3, R4, R5, R6, R7, R8}*. By a discernibility matrix of *DT*, denoted *DM(T),* we will mean *nxn* matrix defined as:

$$m_{ij}^{a(R_i)} = \{(a \in C : a(R_i) \neq a(R_j)) \text{ and } a(R_i) \neq \text{null} \text{ and } a(R_j) \neq \text{null} \text{ and } (d(R_i) \neq d(R_j))\} \quad (15)$$

where *i, j=1,2,...,11, R_i, $R_j \in U$.*

The items within each cell of the discernibility matrix, *DM(DT)* are aggregated disjunctively, and the individual cells are then aggregated conjunctively. To compute the reducts of the discernibility matrix we use the following theorems that demonstrate equivalence between reducts and prime implicants of suitable Boolean functions (Stepaniuk, 2008; Qu et al., 2009). For every object $R_i \in U$, the following Boolean function is defined:

$$g_{R_i}(C) = \bigwedge_{R_j \in U} \left(\bigvee_{a \in m_{ij}} a \right) \quad (16)$$

The following conditions are equivalent (Hassanien et al., 2008):

1. $\{a_{i1}, \ldots, a_{in}\}$ is a reduct for the object R_i, *i = 1..n.* and
2. $a_{i1} \wedge a_{i2} \wedge .. \wedge a_{in}$ is a prime implicant of the Boolean function g_{Ri}.

On Boolean expression the absorption Boolean algebra rule is applied. The absorption law is an identity linking a pair of binary operations. For example: $a \vee (a \wedge b) = a \wedge (a \vee b) = a$

From the decision matrix we form a set of Boolean expressions, one expression for each row of the matrix.

4.2 Decision rule extraction using variable precision rough sets models for an incomplete e-learning system

In this paper we present the application of rough set to discover student rules between students' descriptors and mark categories. A rule is represented using a Prolog fact:

rule(Mark, Accuracy, Coverage, ListofStudentDescriptors)

where *Mark*, the head of the rule, is the mark category, *Accuracy* is the rule accuracy computed as in (17), *Coverage* is the rule coverage computed as in (18) the body of the rule, is composed by conjunctions of student descriptors, while *Mark*, the head of the rule, is the mark category.

For a rule *if descriptors then mark*, we may define the support (s) and accuracy (a) of by:

$$a(\text{rule}) = \frac{\text{cardinality}(\text{descriptorSet} \cap \text{markSet})}{\text{cardinality}(\text{descriptorSet})} \quad (17)$$

where the set $descriptorSet \bigcap markSet$ is composed from student descriptors which have a certain *descriptorSet* and a certain *mark*.
The coverage(c) of a rule is defined by:

$$c(\text{rule}) = \frac{\text{cardinality}(\text{descriptorSet} \bigcap \text{markSet})}{\text{cardinality}(\text{markSet})} \qquad (18)$$

The steps of the algorithm for decision rules extraction are:

- Determine the tolerance classes and decision classes.
- Determine the precision parameter β
- Determine the set of β-reducts from the discernibility matrix.
- Construct from the decision matrix a set of Boolean expressions, one expression for each row of the matrix.

The student classification algorithm based on the discovered rules can be resumed as:

- collect all the decision rules in a classifier,
- compute for each rule the support, accuracy and coverage,
- eliminate the rules with the support less than the minimum defined support,
- order the rules by accuracy, than by coverage,
- if a student matches more rules select the first one: a student matches a rule, if all the descriptors, which appear in the body of the rule, are included in the descriptors of the student.

5. CASE STUDY FOR AN INCOMPLETE E-LEARNING SYSTEM

The experiments presented in this paper are realized in the e-leaning system containing 40 students enrolled for *Database* course. From practice, we observe that the students didn't take all course activity. So, this generates an incomplete information system.
In this example, we present the application of variable precision rough set theory in an in complete e-learning system. Let $(U, C \cup D)$ be an e-learning incomplete system, where:

C={Nassigment, Nquiz, Nquiz_p, Nquiz_f, Nmessages, Nmessages_ap, Nposts, Nread, Total_time_assignemt, Total_time_forum, Total_time_quiz} and *D={Mark}*
In the Table 1, the partial e-learning system is presented. For the simplicity of presentation we consider that *C={Nassigment, Nquiz, Nmessages_ap, Nposts}* and *D={Mark}*

Table 1. Partial student information system

U	***Nassignment***	***Nquiz***	***Nmessages_ap***	***Nposts***	***Mark***
R_1	high	low	null	null	average
R_2	high	high	medium	medium	excellent
R_3	null	medium	high	null	average
R_4	low	low	null	medium	insuff
R_5	medium	null	good	good	good
R_6	medium	medium	null	null	average
R_7	high	high	null	medium	good
R_8	high	low	high	null	excellent

1. The tolerance classes are:

$U/C = \{S_C(1), .., S_C(8)\}$, where
$S_C(1)= \{R_1, R_8\}$; $S_C(2)=\{R_2, R_7\}$; $S_C(3)=\{R_3, R_6\}$;
$S_C(4)=\{R_4\}$; $S_C(5)=\{R_5, R_6\}$; $S_C(6)= \{R_5, R_6\}$;
$S_C(7)=\{ R_2, R_7\}$; $S_C(8)=\{ R_1, R_8\}$

$U/D=\{D_{average}, D_{good}, D_{excellent}, D_{insufficient}\}$ where
$D_{average}=\{ R_1, R_3, R_6\}$; $D_{good}=\{ R_5, R_7\}$; $D_{excellent}=\{ R_1, R_8\}$; $D_{insufficient}=\{R_4\}$

2. Determination of the precision parameter β.

$e(S_C(1), D_{average})=1-\frac{1}{2}=\frac{1}{2}$;

$e(S_C(1), D_{good})=1-\frac{0}{2}=1$;

$e(S_C(1), D_{excellent})=1-\frac{2}{2}=0$;

$e(S_C(1), D_{insufficient})=1-\frac{0}{2}=1$;

$e(S_C(2), D_{average})=1-\frac{0}{2}=1$;

$e(S_C(2), D_{good})=1-\frac{1}{2}=\frac{1}{2}$;

$e(S_C(2), D_{excellent})=1-\frac{0}{2}=1$;

$e(S_C(2), D_{insufficient})=1-\frac{0}{2}=1$;

$e(S_C(3), D_{average})=1-\frac{2}{2}=0$;

$e(S_C(3), D_{good})=1-\frac{0}{2}=1$;

$e(S_C(3), D_{excellent})=1-\frac{2}{2}=0$;

$e(S_C(3), D_{insufficient})=1-\frac{0}{2}=1$;

$e(S_C(4), D_{average})=1-\frac{0}{1}=1$;

$e(S_C(4), D_{good})= 1-\frac{0}{1}=1$;

$e(S_C(4), D_{excellent})= 1-\frac{0}{1}=1$;

$e(S_C(4), D_{insufficient})= 1-\frac{0}{1}=1$;

$e(S_C(5), D_{average})=1-\frac{1}{2}=\frac{1}{2}$;

$e(S_C(5), D_{good})= 1-\frac{1}{2}=\frac{1}{2}$;

$e(S_C(5), D_{excellent}) = 1 - \frac{0}{2} = 1$;

$e(S_C(5), D_{insufficient}) = 1 - \frac{0}{2} = 1$;

$e(S_C(6), D_{average}) = 1 - \frac{1}{2} = \frac{1}{2}$;

$e(S_C(6), D_{good}) = 1 - \frac{1}{2} = \frac{1}{2}$;

$e(S_C(6), D_{excellent}) = 1 - \frac{0}{2} = 1$;

$e(S_C(6), D_{insufficient}) = 1 - \frac{0}{2} = 1$;

$e(S_C(7), D_{average}) = 1 - \frac{0}{2} = 1$;

$e(S_C(7), D_{good}) = 1 - \frac{1}{2} = \frac{1}{2}$;

$e(S_C(7), D_{excellent}) = 1 - \frac{0}{2} = 1$;

$e(S_C(7), D_{insufficient}) = 1 - \frac{0}{2} = 1$;

$e(S_C(8), D_{average}) = 1 - \frac{1}{2} = \frac{1}{2}$;

$e(S_C(8), D_{good}) = 1 - \frac{0}{2} = 1$;

$e(S_C(8), D_{excellent}) = 1 - \frac{2}{2} = 0$;

$e(S_C(8), D_{insufficient}) = 1 - \frac{0}{2} = 1$;

So, $\beta = \frac{1}{2}$.

3. The set of β-reducts are:

$\underline{Q}_{\beta}(D_{good}) = \{S_C(2), S_C(5), S_C(6)\}$;

$\underline{Q}_{\beta}(D_{average}) = \{S_C(3), S_C(5), S_C(6)\}$;

$\underline{Q}_{\beta}(D_{excellent}) = \{S_C(1), S_C(3), S_C(8)\}$;

$\underline{Q}_{\beta}(D_{insufficient}) = \{\phi\}$

Based on the discernibility matrix generated as in (15), the following rules are produced:

For the *average* mark, we obtain the following rules based on the table reducts:

- (Nquiz=low) ∧ (Nassignment =high) ∧ (Nquiz=low)
- (Nquiz=medium ∨ Nmessages_ap =high) ∧ (Nmessages_ap =high) ∧ (Nquiz=medium)
- (Nquiz=medium) ∧ (Nassignment=medium) ∧ (Nassignment=medium ∨ Nquiz=medium)

For the *good* mark we obtain the following rules based on the table reducts:

- (Nassignment = high ∨ Nposts=good) ∧ (Nmessages_ap=good)
- (Nquiz=high) ∧ (Nassignment =high ∨ Nquiz=high)

For the *excellent* mark we obtain the following rules based on the table reducts:

- (Nquiz=high) ∧ (Nmessages_ap =high) ∧ (Nassignment=high ∨ Nposts=medium) ∧ (Nassignment =high ∨ Nquiz=high)
- (Nquiz=low) ∧ (Nassignment =high ∨ Nmessage_ap=high) ∧ (Nassignment =high ∨ Nquiz=low) ∧ (Nquiz=low)

By applying the absorption rule on the prime implicants, the following rules are generated:

- Rule 1: Nquiz=low ∧ Nassignment =high →average
- Rule2: Nmessages_ap = high ∧ Nquiz=medium→average
- Rule 3: Nquiz=medium ∧ Nassignment=medium →average
- Rule 4: (Nassignment = high ∨ Nposts=good) ∧ (Nmessages_ap=good) →good
- Rule 5: Nquiz=high →good
- Rule 6: (Nquiz=high) ∧ (Nmessages_ap =high) ∧ (Nassignment=high ∨ Nposts=medium) →excellent
- Rule 7: (Nquiz=low) ∧ (Nassignment =high ∨ Nmessage_ap=high) →excellent

6. EXPERIMENTS AND CONCLUSION

In the experiments realized through this study, two databases are used for the learning and testing process. The database used to learn the correlations between student behaviour and marks, contains information about 40 students, each described by 11 descriptors. For each mark class, the following metrics: accuracy, specificity, and sensitivity. The counted results are presented in Table 2.

Table 2. Results recorded for different marks

Mark	*Accuracy(%)*	*Sensitivity(%)*	*Specificity(%)*
Good	98.7	97.1	75.1
Average	97.7	96.3	73.8
Excellent	97.9	96.4	73.8
Insufficient	96.8	96.5	73.4

In this study, a method based on variable precision rough set theory is proposed and developed to assist the teacher by doing the pre-evaluation of students during a course study. For establishing correlations with the mark, we experimented and selected some descriptors of the student activity in the Moodle system for a "Database" course.

Experimental results show that the proposed system can exactly provide appropriate feedback to the learner and teacher, resulting in increased the learning efficiency and learning performance.

The Prolog language used for representation of students' descriptors and rules makes a simple and flexible integration of our methods with other learning management systems.

In future work, it would be interesting to repeat the analysis, using more data from different types of courses and also to select other student descriptors. It would be also very useful to do experiments using more experts in

order to analyse the obtained rules for discovering interesting relationships.

ACKNOWLEDGMENT

This work was partially supported by the grant number 16C/2014, awarded in the internal grant competition of the University of Craiova.

REFERENCES

Pawlak, Z., Skowron, A (1994). Rough Membership Functions. Advances in the Dempster-Shafer Theory of Evidence pp.251-271. John Wiley and Sons, New York.

Stepaniuk, J (2008). *Rough Granular Computing in Knowledge Discovery and Data Mining*. Springer-Verlag, Germany.

Hassanien, A.E., Abraham A., Peters, J.F. (2008). Rough Sets in Medical Imaging: Foundations and Trends. Computational Intelligence in Medical Imaging: Techniques and Applications, pp. 47-87. CRC Press, USA.

Michalski, R. (1983). A Theory and Methodology of Inductive Learning. *Artificial Intelligence*, vol. 20(2), pp. 111 – 161.

Wang, R., Zhou, M., William, Z. (2012). A New Approach to Establish Variable Consistency Dominance-Based Rough Sets Based on Dominance Matrices. *Proc. International Conference on Intelligent System Design and Engineering Application*, Sanya, Hainan, pp. 48-51.

Ren, Y., Xing, T., Quan, Q., Chen, X. (2011). Attributes Knowledge Reduction and Evaluation Decision of Logistics Centre Location Based on Rough Sets. *Proc. 4th International Conference on Intelligent Computation Technology and Automation*, Shenzhen, 2011, pp. 67-70.

Zaras, K., Marin, J. C., Boudreau-Trude, B. (2012). Dominance- Based Rough Set Approach in Selection of Portfolio of Sustainable Development Projects. *American Journal of Operations Research*, vol. 2(4), pp. 502-508.

Ke, G., Mingwu, L., Yong, F., Xia, Z. (2011). A Hybrid Model of Rough Sets and Shannon Entropy for Building a Foreign Trade Forecasting System. *Proc. 4th International Joint Conference on Computational Sciences and Optimization*, Yunnan, pp. 7-11.

Lai, C. J., Wen, K. L. (2005). Application of Rough Set Approach to Credit Screening Evaluation. *Journal of Quantitative Management*, vol. 12(1), pp. 69-78.

Chao, D., Sulin, P. (2011). The BSC Alarm Management System Based on Rough Set Theory in Mobile Communication. *Proc. 7th International Conference on Computational Intelligence and Security*, Hainan, pp. 1557-1561.

Hossam, A. N. (2011). A Probabilistic Rough Set Approach to Rule Discovery. *International Journal of Advanced Science and Technology*, vol. 30, pp. 25-34.

Qu, Z. Wang, X. (2009). Application of Clustering Algorithm and Rough Set in Distance Education. *Proc. 1st International Workshop on Education Technology and Computer Science*, Wuhan,pp. 489-493.

Sheu, T., Chen, T., Tsai, C., Tzeng, J., Deng, C. and Nagai, M. (2013). Analysis of Students' Misconception Based on Rough Set Theory. *Journal of Intelligent Learning Systems and Applications*, vol. 5(2), pp. 67-83.

Romero, C., Zafra, A., Luna, J.M., Ventura, S.(2013). Association rule mining using genetic programming to provide feedback to instructors from multiple-choice quiz data. *Expert Systems*, vol. 30(2), pp. 162-172.

Udristoiu, A., Udristoiu, S., and Popescu, E. (2014). Predicting Students' Results Using Rough Sets Theory. *Proc. Intelligent Data Engineering and Automated Learning–IDEAL*, Salamanca, Spain, pp. 336-343.

Cole, J. (2005). *Using Moodle*. O'Reilly.

Ziarko, W. (1993). Variable Precision Rough Set Model. *Journal of Computer and System Sciences*, vol. 46(1), pp. 39-59.

Beynon, M. (2001). Reducts within the Variable Precision Rough Sets Model: A Further Investigation. *European Journal of Operational Research*, vol. 134(3), pp. 592-605.

Liu, D., Hu, P., Jiang, C. Z. (2008). The Incremental Learning Methodology of VPRS Based on Complete Information System. *Lecture Notes in Computer Science*, Springer, pp. 276-283.

Gong, Z., Shi, Z., Yao, H. (2012). Variable Precision Rough Set Model For Incomplete Information Systems And Its β –Reducts. *Computing and Informatics*, vol. 31, pp. 1385-1399.

Author Index