Baltic Young Scientists Conference

Tallinn, 27.-28.07.2015

Universität Rostock 2015

Herausgeber: Prof. Dr. Clemens Cap
Wissenschaftsverbund „Informations- und Kommunikationstechnologien“ (IuK)

Erstellung der Druckvorlage: André Sandmann

Entwurf des Umschlagbildes: Christine Bräuning

Bezugsmöglichkeiten: Universität Rostock
Institut für Informatik
Frau Jacqueline Tiedemann
Albert-Einstein-Str. 22, Raum 356
18059 Rostock

Universität Rostock
Wissenschaftsverbund IuK
Frau Dr. Christine Bräuning
Albert-Einstein-Str. 22, Raum 364
18059 Rostock

Druck: IT- und Medienzentrum der Universität Rostock

Table of Contents

Preface

Due to the generous support of the German Academic Exchange Service (DAAD - Deutscher Akademischer Austausch Dienst), BaSoTI summer school is going in to its eleventh year, with the associated conference going into its eighth year.

It was in 2005, that the University of Bremen, the University of Lübeck, the International School of New Media at the University of Luebeck (ISNM), and the University of Rostock joined forces for the first Baltic Summer School in Technical Informatics (BaSoTI). Supported by a sponsorship of the German Academic Exchange Service a series of lectures was offered between August 1 and August 14, 2005 at Gediminas Technical University at Vilnius, Lithuania. The goal of the Summer School was to intensify the educational and scientific collaboration of northern German and Baltic Universities at the upper Bachelor and lower Master level.

In continuation of the successful programme, BaSoTI 2 was again held at Vilnius in 2006 and 2009, BaSoTI 3 took place in Riga, Latvia at the Information Systems Management Institute in 2007, BaSoTI 4, BaSoTI 5 and BaSoTI 8 were held at the University of Tartu, BaSoTI 6 took place in Kaunas, Lithuania and BaSoTI 7 and BaSoTI 10 at the Technical University of Riga and BasoTI 11 at the Tallinn University of Technology.

Since BaSoTI 3, the Summer School lectures have been complemented by a one day scientific event. The goal is to give young, aspiring PhD candidates the possibility to learn to give and to survive an academic talk and the ensuing discussion, to get to know the flair and habits of academic publishing and to receive broad feedback from the reviewers and participants. Moreover, the Summer School students would have a chance to participate in what most likely would be their first academic research event.

In 2015, the year of the eleventh anniversary of the BaSoTI the conference was addressed again to young PhD candidates from the Baltic States and the German partner universities especially. Moreover, international, reviewed contributions by researchers were presented and BaSoTI students contributed with short talks on their work.

Clemens H. Cap
Rostock, September 2015.

Programme Committee

Andreas Ahrens (Wismar)
Dennis Boldt (Lübeck)
Clemens Cap (Rostock)
Martin Leucker (Lübeck)
Olaf Manuel Maennel (Tallinn)
Maciej Mühleisen (Hamburg)
Thomas Mundt (Rostock)
Gunnar Piho (Tallinn)
Dennis Pfisterer (Stuttgart)
Peter Sobe (Dresden)

Machine Learning Approximation Techniques Using Dual Trees

Denis Ergashbaev
Universitat Politècnica de Catalunya, Universitat de Barcelona, Universitat Rovira i Virgili
email: denis.ergashbaev@est.fib.upc.edu

Abstract: This paper presents a condensed version of my master's thesis work. It explores a dual-tree framework with underlying kd-tree space partitioning data structure as applied to a particular class of machine learning problems that are collectively referred to as generalized n-body problems. We propose a novel algorithm based on the dual-tree framework to accelerate the task of discovering characterizing boundary points (CBP) – a set of data points defined by geometry rules and representing an optimal interclass boundary under certain notions of robustness. Designed with support for both approximate and exact computations, experimental results confirm superior runtime properties of the algorithm compared to the state-of-the-art solution.

1 Introduction

Many of the machine learning methods – including all-nearest-neighbors problem, range search, kernel density estimation, and two-point correlation – are naively quadratic in the number of data points [GM00]. This time complexity compromises their use in the large-scale machine learning applications thus demanding more efficient solutions to accelerate naive approaches.

One commonly used method to reduce the time complexity of proximity problems is application of space-partitioning data structures, such as kd-trees, and use of branch-and-bound algorithms to reduce the runtime speed [CMR$^+$13]. This idea has been advanced further for a distinct subset of so-called *generalized n-body problems* by application of a dual-tree framework. Initially introduced by Alexander Grey [Gra03], the dual-tree framework has been claimed and theoretically proven to achieve a near-linear performance for a range of n-body problems.

Boosted Geometry-Based Ensembles (Boosted OGE) [Puj10] is a simple ensemble-based classification algorithm that exhibits superior performance compared to some machine-learning techniques, while being commensurate to the kernel methods. The base classifiers of Boosted OGE are built on top of the dataset points that form an optimal boundary between two classes based on a specific notion of robustness and margin [Puj10]. These will be commonly referred to as *characterizing boundary points* (CBPs).

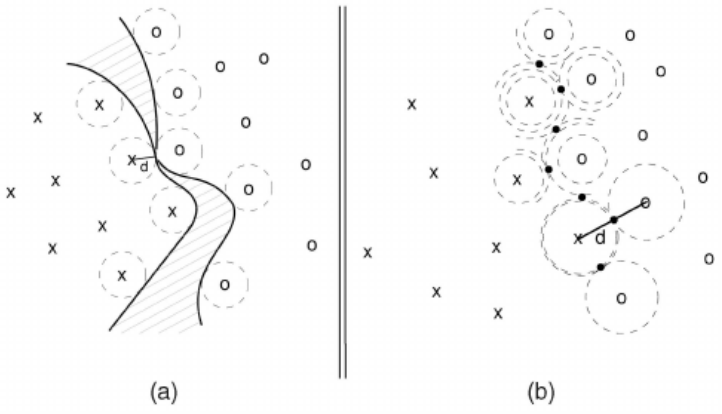


Figure 1: Decision boundary formed by the CBPs and their areas of influence [PM09]

Algorithm 1: $Sokal$. Sokal algorithm for CBP construction

Input: Set of data points $S = \{x_i, y_i\} \in \mathbb{R}^d$ belonging to class $y_i \in \{+1, -1\}$
Output: Set of tuples of indexes $\{(i, j)\}$ that identify the generating data points of CBP
Compute squared Euclidean distance $d(i, j)^2$ from each point x_i to x_j in M
Initialize the set of indexes that define the CBP, $E = \{\}$ **begin**

```
    foreach x_i | y_i = +1 do
        foreach x_j | y_j = -1 do
            foreach x_k | y_k = +1 or y_k = -1 do
                cbp_found = true;
                if x_i ≠ x_k and x_j ≠ x_k then
                    if d(i,j)^2 ≥ d(i,k)^2 + d(j,k)^2 then
                        cbp_found = false;
                        break;
                    end
                end
            end
            if cbp_found then
                E = E ∪ {(x_i, x_j)}
            end
        end
    end
end
```

Consider an illustrative example depicted by Figure 1. The CBPs are the black-colored points that are formed at half the distance between the generating points that belong to opposing classes $\{+1, -1\}$. The set of CBPs forms an optimal geometric boundary between two classes under certain notions of robustness.

Depicted in Algorithm 1, *Sokal* is a state-of-the-art procedure to construct CBPs. Having a $O(n^3)$ complexity in the number of dataset points due to its spacial unawareness and absence of meta-heuristic, it leads to poor scalability of the Boosted OGE with the growth of the dataset size. Furthermore, the number of CBPs raises rapidly as dataset dimension and size increase.

2 Proposal

We develop a novel algorithm based on the dual-tree framework in order to accelerate the task of finding CBPs and therefore contribute to performance improvements of Boosted OGE. Capitalizing on the space-awareness of the underlying data-structures, time savings offered by the divide-and-conquer methodology of the dual-trees, and valuable meta-heuristics implemented in local search we achieve notable speed-ups in the CBP calculation.

Faster CBP Computation Let us define the reference set X_R that holds data points belonging to class -1 and the query set X_Q of points of class $+1$. While the task of computing CBPs is not strictly an n-body problem, it does share one characteristic common to this class of problems: in the worst case, deriving a solution requires comparison of each data point in the reference set X_R to every single data point in the query set Q_R.

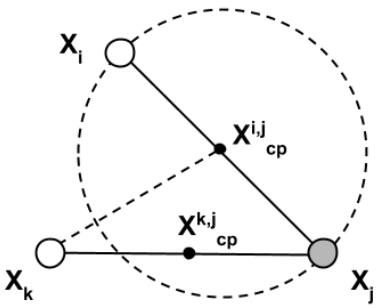


Figure 2: Candidate CBP $x_{cp}^{i,j}$, generating points x_i and x_j, intruding point x_k

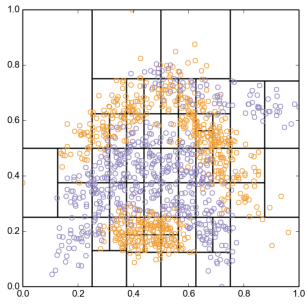


Figure 3: Dual-tree applied to a dataset

Algorithm We construct a tree on the whole dataset (Figure 3) and use the same instance of it as a reference and query tree.

1. Start a depth-first dual-tree traversal comparing reference nodes to the query nodes:
    1.1. At each step a possibility to prune the current pair (N_R and N_Q) is evaluated and the pruning is performed. We have developed three different pruning

strategies:

- *Conservative* – the reduced dataset contains all generating and intruding points such that the exact CBPs are found.
- *Minimum distance* and *Nonadjacent* – approximate strategies that allow computational speed improvements at the expense of precision in finding CBPs.

1.2. If the pair can not be pruned the recursive traversal through the current node descendants is continued until two leafs are reached.

1.3. As soon as both reference and query node are of type leaf node, local search is performed. Its task is to locate the potential generating points i and j as well as potential intruding points k (Figure 2) in an ordered way so that subsequent Sokal algorithm performs only reduced number of real computations.

2. Upon completion of the dual-tree traversal we fill lists of generating and potential intruding points, we subsequently feed them to the baseline Sokal algorithm. Thus, the preprocessing based on dual-trees 1) reduces the search space that the baseline Sokal has to operate on 2) and also orders the points according to metaheuristics that improves effective computations.

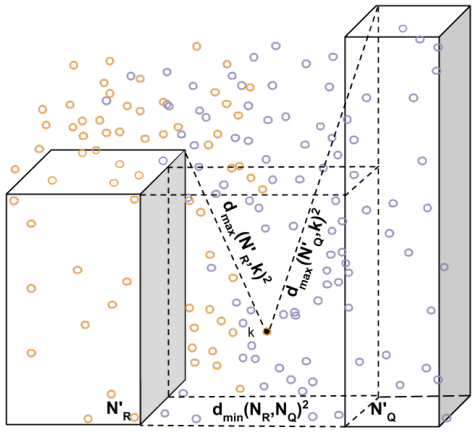


Figure 4: Strict pruning in 3D

Pruning Figure 4 visualizes conservative pruning strategy. It is guided by the Equation 1 that is derived for the batch case from the original Sokal version [PM09].

$$d_{min}(N_R, N_Q)^2 \geq d_{max}(N_R, k)^2 + d_{max}(N_Q, k)^2 \tag{1}$$

Pruning of N_R and N_Q is only allowed if the squared distance between the nodes is greater than the sum of the distances between the nodes and some (intruding) point k between them. Minimum distance and nonadjacent pruning strategies relax this constraint significantly and become of significant value in higher dimensions.

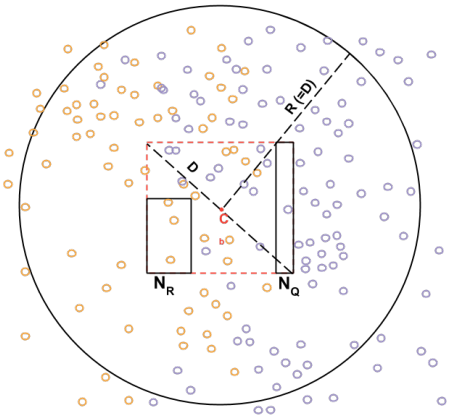


Figure 5: Local Search

Local Search If two nodes were not pruned, we perform a local search – exemplified by Figure 5 – restricting the space of points by a sphere boundary. The boundary has its origin at the centroid C of the bounding hyperrectangle with the radius set to the space diagonal of this hyperrectangle. We execute a k-nearest neighbor search returning only the points of the dual-tree hyperrectangles that are within the boundary. The positive effect of the local search is that by means of nearest-neighbor query the points are ordered in ascending distance from the centroid: the Sokal algorithm is thus able more quickly find intruding points when evaluating generating capacity of nodes N_Q and N_R.

3 Experiments and Results

Table 1 lists the original datasets that have been used in the experiments. In order to explore effects of dimensionality and size but also to ensure that available hardware resources are able to operate on the datasets, these where reduced in dimensions and size.

Figures 6, 7, 8, and 9 summarize experimental results of accelerating the CBP computation:

1. Leaf node size (Figure 6) of the dual-trees affects the performance of our algorithm. The optimal values – in the range of 3% to 7% of the dataset size – correspond to about an order of magnitude performance gain for the 2-dimensional datasets as compared to the Sokal baseline.
2. While pruning, as indicated by Figure 7, contributes significantly to the worst-case computation count in the lower dimensions (and very little to none as dimensions increase therefore demanding a disproportionate growth of the dataset sizes due to the "curse of dimensionality" [Dom12, HTF01]).

Table 1: Datasets

Dataset	Size	Dim
banana	5300	2
EEGEye	14980	14
Example	1014	5
magic	19020	10
ringnorm	7400	20
skin	245057	3
svmguide1	3089	4
transfusion	748	4
Twonorm	7400	20
waveform	5000	21

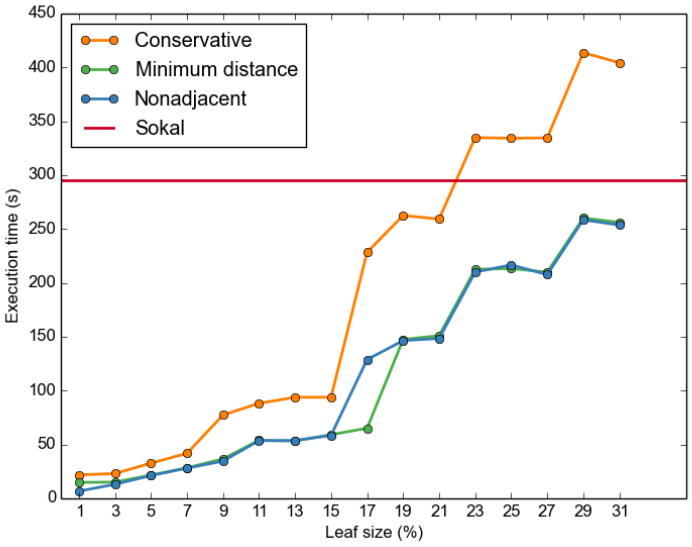


Figure 6: Effect of leaf size (EEGEye 7200x2)

3. Local search is able to dramatically improve the real-case computation count (Figure 8) across the whole spectrum of dimensions.

As Figure 9 illustrates it for a selection of datasets, our algorithm consistently beats the state-of-the-art Sokal solution. The gains are most notable for the datasets of lower dimensions with around 10 to 3 times improvement in speed. A 40% speed boost is achieved on higher dimensional datasets.

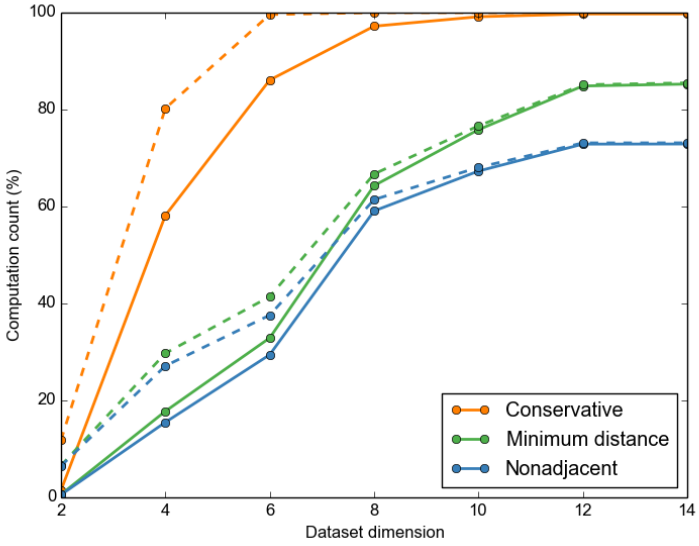


Figure 7: Worst case computation ratio (EEGEye 7200x2)

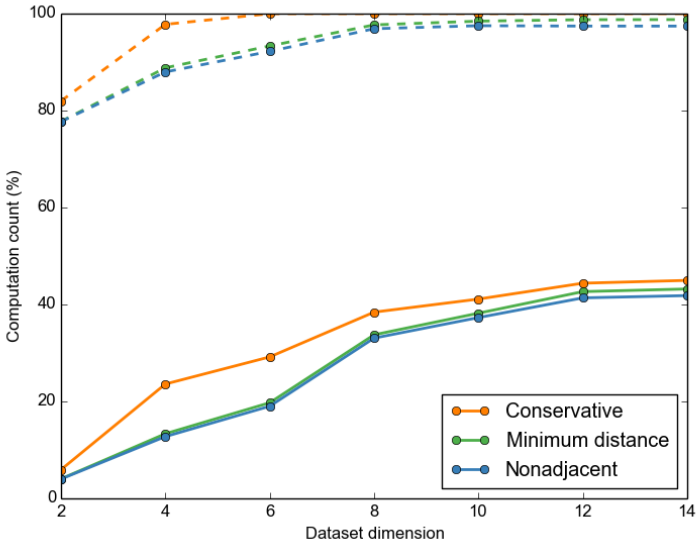


Figure 8: Real computation ratio (EEGEye 7200x2)

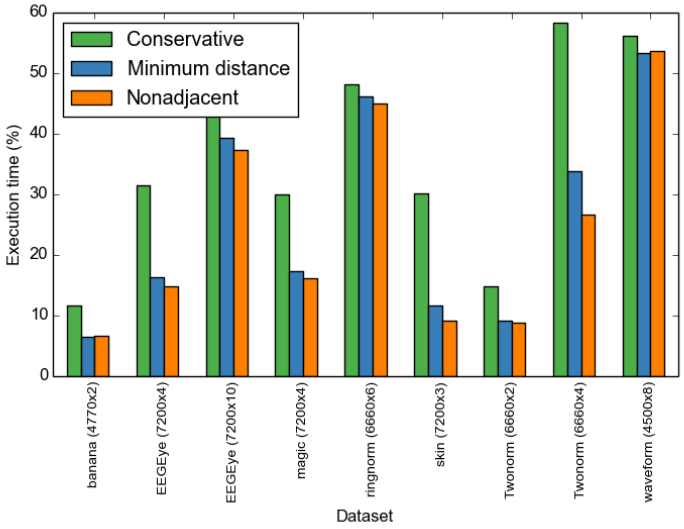


Figure 9: Results summary

4 Conclusions

We have been able to devise a dual-tree based algorithm to deal with the problem of finding characterizing boundary points (CBP), a special case of the Gabriel graph. Owing to the powerful meta-heuristics implemented by the local search and reduction of the dataset brought about by pruning, the developed dual-tree algorithm has been able to both lower worst-case and real-case computation measurements as compared to the baseline Sokal solution.

Based on the experimental evaluation, we have been able to achieve about an order of magnitude performance increases on the two-dimensional datasets. Gains in efficiency were also evident in higher dimensions ranging from 70% to 50% improvement of the state-of-the-art method.

Future Work. This work has opened up a number of areas that further research should address.

The proposed dual-tree algorithm is superior to Sokal in performance, but is inevitably more complex as it requires parameterizing the node size. Experimentally shown to be an important factor for the dual-tree performance and a non-trivial task, a stricter methodology or a set of general observations should be developed to determine optimal values for

the leaf sizes. This would inevitably depend on the implicit dimensionality of the datasets and their sizes and is likely to be specific to the chosen dual-tree architecture.

More awareness about the required size of the datasets in respect to their implicit dimension is needed to be able to conclude beforehand about the expected performance gain of the dual-tree algorithms compared to Sokal.

References

[CMR+13] Ryan R. Curtin, William B. March, Parikshit Ram, David V. Anderson, Alexander G. Gray, and Charles Lee Isbell Jr. Tree-Independent Dual-Tree Algorithms. *CoRR*, abs/1304.4327, 2013.

[Dom12] Pedro Domingos. A Few Useful Things to Know About Machine Learning. *Commun. ACM*, 55(10):78–87, October 2012.

[Erg] Denis Ergashbaev. Machine Learning Approximation Techniques Using Dual Trees.

[GM00] Alexander Gray and Andrew Moore. 'N-Body' Problems in Statistical Learning. In *Advances in Neural Information Processing Systems 13*, pages 521–527. MIT Press, 2000.

[Gra03] Alexander G. Gray. *Bringing Tractability to Generalized N-Body Problems in Statistical and Scientific Computation*. PhD thesis, Carnegie Mellon University, April 2003.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[PM09] Oriol Pujol and David Masip. Geometry-Based Ensembles: Toward a Structural Characterization of the Classification Boundary. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):1140–1146, 2009.

[Puj10] Oriol Pujol. Boosted Geometry-Based Ensembles. In Neamat El Gayar, Josef Kittler, and Fabio Roli, editors, *MCS*, volume 5997 of *Lecture Notes in Computer Science*, pages 195–204. Springer, 2010.

On session-based HTTP flood mitigation

Dmytro Piatkivskyi, Slobodan Petrovic
Gjøvik University College
email: dmytro.piatkivskyi@gmail.com, slobodan.petrovic@hig.no

Abstract: HTTP flood attacks is a threat to the modern IT dependent society. It is a weapon in a cyber war and the loss due to these attacks is tremendous. The number of HTTP flood attacks is growing every year which pushes us to finding new methods to deal with them. An HTTP flood sends massive, but legitimate-looking HTTP traffic to a victim from multiple sources. A single request which is a part of an HTTP flood attack does not exhibit any characteristics by which it can be detected. The only way to detect attack requests is to group them and deal with groups. One way of grouping requests is by HTTP sessions. The problem is that HTTP is a sessionless protocol. Thus, the definition of the session is left up to designers of the mitigation system. In this paper we show that the chosen definition affects the performance of the end system to a large extent. We search for such HTTP session definition which facilitates the best detection performance of the attack requests and thus leads to a more efficient HTTP flood mitigation.

1 Introduction

An HTTP flood is a type of Distributed Denial of Service attack, which runs on the application layer of the OSI model. Such attacks are conducted by simply sending many legitimate looking requests to a web server, exhausting all its resources. Having all the resources exhausted, a web server stops serving legitimate requests at some point violating the availability property. The task for a defender is to make sure that legitimate users still get the requested pages within a reasonable period of time.

The paper describes an attempt to improve the efficiency of HTTP flood detection (mitigation) systems [1]. The main focus is to detect as many different sophisticated attacks as possible taking into consideration flash events. An enhanced mitigation system would never completely eliminate the threat, but it may render many known attacks inefficient. In this way, an attacker would be forced to spend more resources and money to reach his or her goals making it eventually unattractive activity.

The paper is organized as follows. It starts with an overview of HTTP flood mitigation solutions. Then, having defined an HTTP session in Section 3, the influence of session parameters choice is studied in Section 4. Finally, conclusions are given in Section 5.

[1] Hereafter we use terms ”detection” and ”mitigation” interchangeably

2 Mitigation techniques

2.1 Machine learning techniques

First papers that described using machine learning for traffic segregation were written on search robots detection [PPLC06] [TK02]. Tan et al. [TK02] used C4.5 decision tree algorithm to classify and hence distinguish robot and human browsing activity.

One of the first approaches to HTTP flood mitigation was by modeling user browsing behavior, mostly with Hidden semi-Markov Model (HsMM) to describe the browsing behavior of users [YSz] [JXKb] [YSz09]. The technique is based on document popularity and transmission probabilities between pages. It is effective as far as an attacker does not mimic user behavior, which is quite possible. It also suffers from high computational complexity, which makes it unsuitable for online detection. The idea of using transmission probabilities between pages to detect bots was further studied in [CKC] [OM]. The main reasoning behind is that a bot is not aware of probability with which a link on a page is clicked on.

Intuitively, clustering seems to be a proper family of algorithms for attack and normal traffic segregation [CKC] [JZHX]. The clustering can first be done on normal traffic to see normal usage patterns. After this, any deviations from defined clusters are treated as attack sessions. Jie et al. [JZHX] introduced a way to interpret such deviations. They defined a trust value which takes into account the proximity to the nearest cluster and importance of that cluster. Paper [CKC] describes hierarchical clustering applied to solve the problem of HTTP floods.

In [RSS] Support Vector Machine (SVM) was used to classify traffic and detect attacks. Oikonomou et al. [OM] reached very low False Positive rates with decision trees.

2.2 Other approaches

Since the goal of HTTP floods is to send as many requests as possible while remaining undetected, it is reasonable to limit the rates at which the connections are accepted [RSU$^+$09] [HIKC]. In [RSU$^+$09], Ranjan et al. presented a scheme where a suspicion measure is assigned to each session according to session inter-arrival times, request inter-arrival times, and session workload profile. Further, the requests are scheduled according to its session suspicion measures. If a server queue is full, requests from the most suspicious sessions are dropped.

Another way to characterize user access behavior is through web-page clicking ratio [JXKb] [JXKa] which expresses page popularity. The approach assumes that all users access the same (so called ”hot”) pages. ConnectionScore scheme [BD12] measures various statistical properties of users and the traffic they generate and compare these properties to the model. If there are significant deviations, the users are blocked.

Paper [SIYL06] does not even attempt to distinguish a DOS attack request from the le-

gitimate ones. Instead, it schedules an incoming request according to the workload of the session it belongs to. This way, the aggressive users will be penalized and good clients will get more resources. Papers [OM] [GCD] suggest using deception techniques. The idea is to embed invisible objects with hyperlinks into a page and mark those users who requested such links as bots.

The very promising and effective approach against application layer floods is CAPTCHA puzzles [KKJB05] [MSC+03]. These are challenge-response tests that tell humans and computers apart.

Works [JZHX] [WVB+06] [NPDD12] implement currency method where a request is served only after a client pays for it. For example, in [JZHX] when a server is overloaded it drops the session connections and asks to retry immediately. Paper [NPDD12] describes a scheme where the client's machine is supposed to solve a puzzle sent by the requested server. To solve a puzzle of defined complexity, a client must pay with CPU time.

3 An HTTP session definition

3.1 Attempts to define an HTTP session

HTTP is a sessionless protocol [New00]. Nevertheless, the notion of an HTTP session is used quite often. Divergence in the definition of an HTTP session in many papers inspired scientists to study what actually an HTTP session is [MDG+09]. Often, notion of a session is defined prior to conducting the actual work pertaining HTTP such as workload characterization [MAFM99], traffic analysis [QLC05] [Coo00], user behavior characterization, etc. Most approaches to HTTP flood mitigation also require HTTP session to be defined [RSU+09] [CC04]. In this section, we first discuss the definitions of other authors and then give our own on which the later reasonings are based.

It is trivial to define session so that a human understands it. According to [MAFM99], a session is a sequence of requests of different types made by a single client during a single visit to a web site. This definition is clear and brings no confusion. But the problem is that in technical sense it is almost impossible to determine a session. It is so due to many factors such as the fact that some users may share the same IP address and not all user requests are sent to the server (some responses might be cached in a user's web-browser), etc. In [CC04], the logical access sequences and the physical access sequences are differentiated. According to this paper, the physical access sequences are those that are actually requested and therefore logged in the access log of a web server. At the same time, the logical access sequences are those that correspond to the user's actions and what the user actually clicks on. Further, paper [CC04] states that it is infeasible to detect anomalous sessions based on logical access sequences due to the fact that it is impossible to track user's clicks. Hence, researchers focused on retrieving a session from mainly web server's logs or incoming traffic analysis.

According to [Coo00], a session can be identified by the referrer field taking into account that referrer of current request of a session must match one of the URLs previously re-

quested in this session. Paper [QLC05] introduced the concept of referrer tree, which was used to aggregate requests into sessions using the timeout as well. An improved algorithm for constructing the referrer tree was described in [MDG+09], which was agnostic to timeouts. This algorithm segments a click stream into logical sessions based on referrer information. Every segment corresponds to a session.

A quite opposite approach was used in [RSU+09] where authors took HTTP/1.1 persistent connections as a base for session-oriented connections. This approach does not reflect the human understandable definition of a session since it lasts as long as the TCP socket is kept opened. For example, for the Apache 2.2 web server the default connection timeout is 5 seconds.

3.2 The proposed definition

Most HTTP flood mitigation systems, [JXKb] [CKC] to name a few, are session-based. But HTTP is a sessionless protocol [New00], which means that we need to define what a session is in order to implement a session-based flood detection. Most papers use an HTTP session definition based solely on a timeout.

Values for the timeout vary to a large extent from a minute [MDG+09] to 30 minutes [JXKb] [CKC] [CP95]. Paper [MDG+09] states that the timeout is chosen arbitrarily and there is no justification for any specific value. Besides, the choice of the timeout changes all the relevant statistic. In spite of this, we have decided to use the session definition by a timeout. Moreover, in our experiment we seek for the proper value of the timeout for HTTP flood detection, i.e. such timeout that makes the best statistic of a session for detection.

A very important requirement for an HTTP flood detection system is to handle traffic efficiently in real time. Obviously, it is a bad idea to use the given HTTP session definition in real time systems. There are many questions to answer. Should we wait until a session is complete (N seconds since the last request received), in order to decide on this session? Or should we calculate a session's statistic having received a sufficient number of requests? What should the number be? If we would have defined the proper sufficient number, there is an open question left whether requests in a session get obsolete. A session defined simply by a timeout can last for a very long time. Does it mean that requests received hours (or minutes) ago become obsolete and should not be counted into the session statistic? We tried to answer all these questions at once, introducing another parameter to the session definition, which is the "last M requests". This parameter is introduced specifically for online detection. A session statistic may change with time. Having a very long session with many requests, a new request would not change statistic much even if the request is an outlier. The statistic of recent requests is more important in a real time detection system. Moreover, counting only last M requests decreases usage of CPU and memory resources.

Another implementation trick pertaining the M parameter is that sessions that consist of less than M requests are not considered at all. Indeed, HTTP flood bots send massive traffic to a target. And if there is not enough requests in a session to make robust statistic,

then such a session is not an attack session.

Thus, we give the following HTTP session definition:

> An HTTP session is a sequence of last M HTTP requests received from one user while the time difference between two consecutive requests is less than a timeout N.

In this way, we defined an HTTP session by two parameters, namely timeout N and last M requests. These two parameters are further investigated in the next section.

4 Session parameters choice

We now study how the choice of the timeout N and the number of last M requests affect the detection accuracy of HTTP flood mitigation systems. For that we generate a dataset and group the requests from the dataset into sessions following the proposed definition of an HTTP session. Then we implement the selected features and calculate those features for each session. The set of features makes a session statistic. At this step, we have a list of session features along with a class label. Then we run machine learning algorithms on the list of session features and we analyze the performance of the algorithms.

4.1 The dataset

The dataset we have generated for the experiment consists of attack and background traffic. As background traffic, we have chosen the publicly available 1998 Football World Cup dataset, which represents a flash event occurred during FIFA World cup in 1998. Based on the background traffic, attack traffic has been generated that consists of many different types of HTTP flood. To generate the dataset we proposed a model of HTTP flood. The attack is modeled by three parameters: request rate, ON/OFF periods and page distribution. Thus, an attack can be described by a three-tuple

$$\mathcal{A} = \{\mathcal{R}, \mathcal{O}N\_OFF, \mathcal{D}\}, \text{ where}$$

$\mathcal{R} = \{r | r$ is a request rate$\}$,
$\mathcal{O}N\_OFF = \{< on, off > | on$ is a period of time of being active,
off is a period of time of being inactive $\}$,
$\mathcal{D} = \{d | d$ is distribution of requested pages$\}$.
The following values of attack model parameters were chosen:

$\mathcal{R} = \{0.05, 0.1, 0.5, 1.0, 5.0\}$,
$\mathcal{O}N\_OFF = \{< 1, 9 >, < 5, 5 >, < 9, 1 >, < 10, 90 >, < 50, 50 >, < 90, 10 >\}$,
$\mathcal{D} = \{$uniform distribution, distribution by popularity, distribution by size$\}$.

4.2 Selected features

The following five features were selected: Request rate, Popularity of accessed objects, Uptime to downtime proportion, Diversity of accessed pages, Average size of objects

4.3 Analysis of an HTTP session definition

We conduct the experiment in two steps. First, we fix parameter M and run the simulated system with different values of parameter N. We range N parameter from 30 to 1000 seconds. We believe that lower value than 30 seconds of N would segment flow of requests into very small sessions. Such small (in terms of number of requests it consists of) sessions would have weak and not stable statistic. On the contrary, a higher value than 1000 seconds does not bring great difference into session statistic, but extends the period of time during which we keep the session alive. In a real time system, a session queue needs to be maintained until the timeout is reached since the last request was received. That means, the longer the session timeout is, the longer it needs to be in memory.

J48, MultilayerPerceptron and ThresholdSelector algorithms were chosen for the experiment.

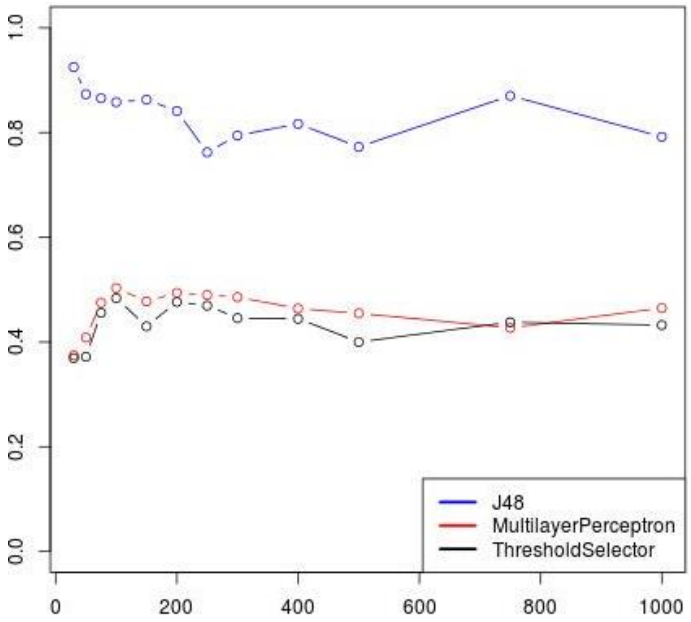


Figure 1: Dependency of detection accuracy on N parameter

We do not consider True Negatives (TN) and False Positives (FP) in our experiment, because all the legitimate sessions have to be detected as legitimate due to our policy. It

means that FP are not accepted at all and TN has to be as close to 1 as possible. Although it is usually not possible to achieve TN = 1, in all our experiments it is equal to 0.99 or higher.

Figure 1 shows that detection accuracy does not change much with the N parameter. We see a better performance of J48 algorithms with small values of N (less than 50). But at the same time, Multilayer Perceptron and Threshold Selector perform badly with small values of N. Moreover, we believe that J48 performs well with small values of N because of overfitting. Since starting from $N = 200$ there are only minor deviations and the detection accuracy is about the same, we conclude that the most common choice of $N = 300$ is appropriate for HTTP flood detection as well. In all further experiments we use $N = 300$.

The second step of the experiment is analysis of the parameter M of the session definition. For that, we fix parameter N equal to 300 and run the system simulation with different values of the parameter M. We range the parameter M in the same way as the parameter N - from 30 to 1000 requests. The reasoning behind is that less than 30 requests in a session results in a weak and unstable session statistic. Moreover, even if a server serves 30 attack, requests that will not harm much. Besides, in flooding attacks, there will be definitely more than 30 requests. On the other hand, too high value of M parameter means late detection, since we need to collect M requests before we calculate a session statistic. It also increases the cost of maintaining a session queue and re-calculating the statistic. We have chosen the same values for M as for N. The algorithms are the same as well.

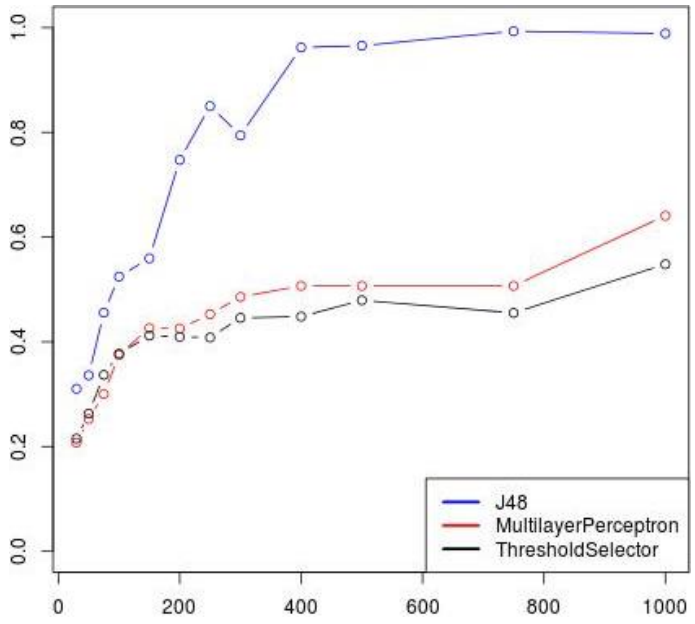


Figure 2: Dependency of detection accuracy on M parameter

From the Figure 2 we see that detection accuracy increases with M up the to point $M = 400$. Further, the detection accuracy does not change much. There is also a slight per-

formance improvement at the point $M = 1000$ for Multilayer Perceptron and Threshold Selector algorithms. The manual investigation shows that it is due to introducing False Positives, which are very undesirable. To take a closer look, ROC curves were built (see Figure 3). Surprisingly, the algorithm has worse performance for $M = 1000$ than for $M = 400$. This confirms that high values of the parameter M might even decrease the detection accuracy of an HTTP flood detection system. This confirms our choice of the proper value of M, which is 400 requests.

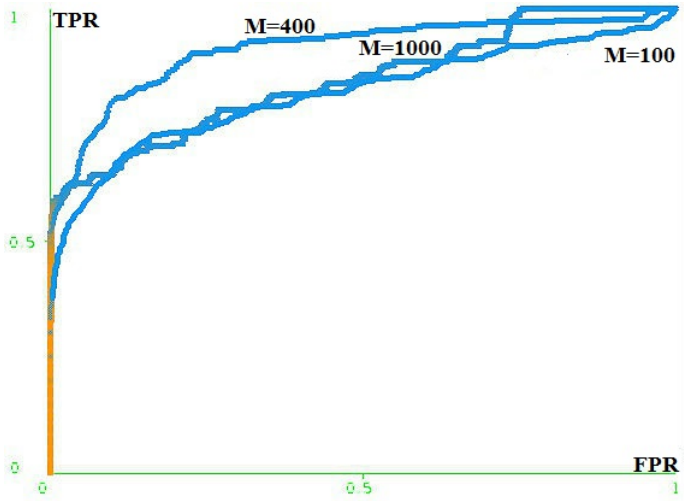


Figure 3: ROC curves for the Multilayer Perceptron algorithm

4.4 Analysis of detection rate by attack type

In order to analyze detection accuracy by attack type, the detection performances of Multilayer Perceptron algorithm for $M = 300$ and $M = 750$ have been examined. Simple comparison of detection accuracies revealed some tendencies. Thus, having normal request rate only volumetric attacks (attacks distributed by size) can be detected. This means that volumetric attacks are the easiest to detect. At the same time, attacks that guess popularity of pages are the most difficult to detect. In addition, it has been concluded that the ON/OFF parameter of the proposed attack model does not influence the detection accuracy. It also means it does not influence an attacker's ability to stay stealth, which means that there is no need to model it.

In order to investigate how detection accuracy changes with the parameter M for each attack type, we compare performances of the Multilayer Perceptron algorithm with $M = 300$ and $M = 750$.

Interestingly enough, attacks that simulate uniform distribution are worse detected using a higher value of the parameter M (see Tables 1 and 2). But, what is more noteworthy is

that attacks that simulate popularity distribution are better detected with $M = 750$ (see Tables 3 and 4). We believe that these two types of HTTP flood are detected, in general, due to three features, namely *request rate*, *average popularity of objects* and *diversity of objects*. They are not detected by *average size of object* feature, because their average size of objects in a session is equal to those of legitimate sessions. Further, as it was concluded in this section earlier, ON/OFF periods do not influence detection much. And, consequently, *uptime/downtime proportion* (*ON/OFF proportion*) feature does not detect any attack.

Attack type	Not detected sessions	Detected sessions
uniform_0.5_1_9	1341	293
uniform_1_1_9	406	2895
uniform_0.5_5_5	1182	299
uniform_1_5_5	922	2406

Table 1: Detection performance of Multilayer Perceptron for $M = 300$

Attack type	Not detected sessions	Detected sessions
uniform_0.5_1_9	1184	0
uniform_1_1_9	2851	0
uniform_0.5_5_5	1031	0
uniform_1_5_5	2878	0

Table 2: Detection performance of Multilayer Perceptron for $M = 750$

Comparing the results of attacks of the same rate, we exclude influence of *request rate* feature, which leaves us only two features. In this way we conclude that difference in detection we observe in this experiment is due to performance of *average popularity of objects* and *diversity of objects* features. In other words, the value of the M parameter that is equal to 300 facilitates the features *average popularity of objects* and *diversity of objects* to detect attacks that simulate uniform distribution. On the other hand, the value of the M parameter that is equal to 750 facilitates the features *average popularity of objects* and *diversity of objects* to detect attacks that simulate popularity distribution.

Attack type	Not detected sessions	Detected sessions
bypopularity_1_1_9	3324	0
bypopularity_5_1_9	533	17168
bypopularity_1_10_90	3313	0
bypopularity_5_10_90	49	17619

Table 3: Detection performance of Multilayer Perceptron for $M = 300$

Attack type	Not detected session points	Detected sessions
bypopularity_1_1_9	2378	496
bypopularity_5_1_9	275	16976
bypopularity_1_10_90	2286	577
bypopularity_5_10_90	105	17113

Table 4: Detection performance of Multilayer Perceptron for $M = 750$

All the discussion above leads to two most important conclusions:

1. Each feature reaches its best utility with different values of parameter M. It means that the proper value of the parameter M should be defined for each feature independently.

2. Two different values of the parameter M facilitate detection of different attacks with the same feature. It suggests using multiple instances of the same feature calculated for different values of parameter M.

5 Conclusions

In this research, HTTP session has been defined by two parameters, namely timeout N and number of last M requests. The experimental analysis showed that the choice of the timeout N does not influence the detection accuracy of HTTP flood attacks as far as it is high enough (more than 200 seconds). That confirmed the common practice found in the literature of using timeout of 300 seconds. Oppositely, it has been shown that the number of last requests M does influence the detection rate. It has also been experimentally determined that the best performance is reached with $M = 400$ requests.

It has been identified that different features reach their maximum utility with different values of M. Moreover, different values of the parameter M facilitate detection of different types of HTTP flood, even by the same feature. This results in a necessity to conduct further analysis for each feature and each type of HTTP flood.

References

[BD12] Hakem Beitollahi and Geert Deconinck. Tackling Application-layer DDoS Attacks. *Procedia Computer Science*, 10(0):432–441, 2012.

[CC04] Sanghyun Cho and Sungdeok Cha. SAD: web session anomaly detection based on parameter estimation. *Computers & Security*, 23(4):312 – 319, 2004.

[CKC] Ye Chengxu, Zheng Kesong, and She Chuyu. Application layer ddos detection using clustering analysis. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 1038–1041. no false positive rate stated.

[Coo00] Robert Walker Cooley. Web Usage Mining: Discovery and Application of Interestin Patterns from Web Data, 2000.

[CP95] Lara D. Catledge and James E. Pitkow. Characterizing browsing strategies in the World-Wide web. *Computer Networks and {ISDN} Systems*, 27(6):1065 – 1073, 1995. Proceedings of the Third International World-Wide Web Conference.

[GCD] D. Gavrilis, I. Chatzis, and E. Dermatas. Flash Crowd Detection Using Decoy Hyperlinks. In *Networking, Sensing and Control, 2007 IEEE International Conference on*, pages 466–470.

[HIKC] Liu Huey-Ing and Chang Kuo-Chao. Defending systems Against Tilt DDoS attacks. In *Telecommunication Systems, Services, and Applications (TSSA), 2011 6th International Conference on*, pages 22–27.

[JXKa] Wang Jin, Yang Xiaolong, and Long Keping. A new relative entropy based app-DDoS detection method. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 966–968.

[JXKb] Wang Jin, Yang Xiaolong, and Long Keping. Web DDoS Detection Schemes Based on Measuring User's Access Behavior with Large Deviation. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5.

[JZHX] Yu Jie, Li Zhoujun, Chen Huowang, and Chen Xiaoming. A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks. In *Networking and Services, 2007. ICNS. Third International Conference on*, pages 54–54.

[KKJB05] Srikanth Kandula, Dina Katabi, Matthias Jacob, and Arthur Berger. Botz-4-sale: surviving organized DDoS attacks that mimic flash crowds, 2005.

[MAFM99] Daniel A. Menascé, Virgilio A. F. Almeida, Rodrigo Fonseca, and Marco A. Mendes. A Methodology for Workload Characterization of E-commerce Sites. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, pages 119–128, New York, NY, USA, 1999. ACM.

[MDG+09] Mark Meiss, John Duncan, Bruno Gon, J. Ramasco, and Filippo Menczer. What's in a session: tracking individual behavior on the web, 2009.

[MSC+03] William G. Morein, Angelos Stavrou, Debra L. Cook, Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. Using graphic turing tests to counter automated DDoS attacks against web servers, 2003.

[New00] Jan Newmarch. HTTP Session Management, August 2000.

[NPDD12] Raju Neyyan, Ancy Paul, Mayank Deshwal, and Amit Deshmukh. Article: Game Theory based Defense Mechanism against Flooding Attack using Puzzle. *IJCA Proceedings on Emerging Trends in Computer Science and Information Technology (ETCSIT2012) etcsit1001*, ETCSIT(5):6–10, April 2012. Published by Foundation of Computer Science, New York, USA.

[OM] G. Oikonomou and J. Mirkovic. Modeling Human Behavior for Defense Against Flash-Crowd Attacks. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1–6.

[PPLC06] KyoungSoo Park, Vivek S. Pai, Kang-Won Lee, and Seraphin Calo. Securing web service by automatic robot detection, 2006.

[QLC05] Feng Qiu, Zhenyu Liu, and Junghoo Cho. Analysis of user web traffic with a focus on search activities. *In Proc. International Workshop on the Web and Databases*, pages 103–108, 2005.

[RSS] A. Ramamoorthi, T. Subbulakshmi, and S. M. Shalinie. Real time detection and classification of DDoS attacks using enhanced SVM with string kernels. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 91–96.

[RSU+09] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly. DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks. *Networking, IEEE/ACM Transactions on*, 17(1):26–39, 2009. asymmetric attack is low rate attack, the detection is based on statistical properties of normal user profiles.

[SIYL06] Mudhakar Srivatsa, Arun Iyengar, Jian Yin, and Ling Liu. *A Middleware System for Protecting Against Application Level Denial of Service Attacks*, volume 4290 of *Lecture Notes in Computer Science*, chapter 14, pages 260–280. Springer Berlin Heidelberg, 2006. good threat model, many references in related work, could be worth looking at them.

[TK02] Pang-Ning Tan and Vipin Kumar. Discovery of Web Robot Sessions Based on their Navigational Patterns. *Data Mining and Knowledge Discovery*, 6(1):9–35, 2002.

[WVB+06] Michael Walfish, Mythili Vutukuru, Hari Balakrishnan, David Karger, and Scott Shenker. DDoS Defense by Offense. In *ACM SIGCOMM 2006*, Pisa, Italy, September 2006.

[YSz] Xie Yi and Yu Shun-zheng. A Novel Model for Detecting Application Layer DDoS Attacks. In *Computer and Computational Sciences, 2006. IMSCCS '06. First International Multi-Symposiums on*, volume 2, pages 56–63.

[YSz09] Xie Yi and Yu Shun-zheng. Monitoring the Application-Layer DDoS Attacks for Popular Websites. *Networking, IEEE/ACM Transactions on*, 17(1):15–25, 2009.

Challenges and Opportunities of Olfactory Interaction in Automotive Context

Dmitrijs Dmitrenko
Automotive Software Development
BFFT Gesellschaft fuer Fahrzeugtechnik mbH
Im Gewerbepark C30, 93053 Regensburg, Germany
e-mail: dmitrijs.dmitrenko@bfft.de

Abstract: This paper addresses the core challenges of olfactory interaction in automotive context and the opportunities it provides for the driver-vehicle and the vehicle-outside-world interaction. It discusses the drawbacks of interaction paradigms implemented in modern cars and the benefits of applying the sense of smell as the novel Human Machine Interface (HMI) approach.

1 Introduction

Board computers of modern cars are programmed to provide the driver with the information on almost everything related to the driving experience, starting from the warnings about the changes in outside temperature and ending with suggestions upon switching off certain electronic devices to extend the battery life, or reminders to make a brake in case of long driving. However, such messages are mostly communicated to the driver using text on the screen or other visual effects (e.g. blinking lamps). This is clear, since human vision and perception of visual signals is an essential part of the driving experience. Cars equipped with infotainment and driving assistance systems have several hundreds of informative and warning messages that can be potentiality shown to the driver. Even after prioritizing them, the driver can easily get overwhelmed with the flow of information. Some messages can get replaced by sounds or haptic feedback, but there are enough limitations applied to these signals as well. Continuous sound warnings and artificial vibrations might turn the driving process into a nightmare. Nevertheless, there is one human sense, which is being almost totally neglected in the automotive HMI approaches - the sense of smell. The goal of this paper is to explore the problems of olfactory interaction in automotive scope, define possible approaches, generate ideas for a prototype that can be implemented in a real vehicle and discuss use cases that visualize these ideas.

2 Related Work

Presentation of information in visual form has always been considered extremely important in the art and science [Sch00]. Not for nothing people say "one look is worth a thousand words". However, not everyone realizes how informative and emotionally rich an olfactory experience can be [BD95, HE96, WL06]. This has been demonstrated in such scents emitting prototypes as the Smelling Screen [MYI13], the oPhone [Edw14] and the Mist Shine [Yam14], where smells are delivered to the user as an informative message.

Nevertheless, smells can be used to stimulate the interaction with some system as well. There are several approaches of applying human perception of scents for interaction purposes. Brewster and Danas claim the efficiency of smell in photo-tagging [BMM06]. Another example can be found in John Cater's firefighters' simulator [ZE99], where odors make virtual environment more realistic.

As the field emerged, more and more commercial smell generators started to appear for both indoor and outdoor solutions [Sen], as well as for special applications, like refreshing air on cruise ships or in exhibitions [Sce10]. The latter has been proposed even for the application within the car, however, the only example provided by its designers comes from the automotive fair, where the scent generator has been used to boost the presentation effect of the new car, without actually integrating the smell emitter inside the vehicle.

Applying odors as information transmitters is very efficient, since people perceive smells with a high degree of intensity and are able to interpret them as a mood stimulator, characteristic of some process or as a warning [Dal96]. Nevertheless, comparing to visual signals, olfactory signals are not that easy to identify and name. In order to use smells as informative signals, it is extremely important to define which smell corresponds to what informative message. This can be done defining an olfactory language, as described in [OG15].

The intensity of smells can even define the way we move around. Quercia et al. invented "smelly maps" [QSAM15] capable of suggesting a walking route, which would be the most attractive one from the olfactory perspective. This idea could surely be applied in the automotive contex, namely for navigation purposes.

What it comes to the approaches already proposed in the field of adaptive automotive user interfaces, it is important to mention the concepts claimed by Nasoz et al. [NLV10] and the researchers from the INEMAS [INE15] project. Both solutions are based on the recognition of driver's emotional state, extracting such features of his or her body as heart rate, body temperature, facial expressions and voice parameters. Based on the recognized mood the interface inside the car would adapt to its user. Nevertheless, both of these strategies are completely neglecting the sense of smell. Taking this sense into account might be the next step towards percepting the driver as a human being.

3 Classification of Interaction Techniques

In order to classify the olfactory interaction techniques to be applied in the automotive HMI system, it is important to localize the circumstances and constraints its development has to deal with. There are thousands of smells exposed by the environment around the car, which it could recognize. There are odors the vehicle can release to send information to the driver. Finally, such a closed and limited room as the space inside the cabin is a perfect environment for the distribution of smells brought in by the driver. The possible flow of odors in such interaction scenarios is displayed on the image below (see Figure 1).

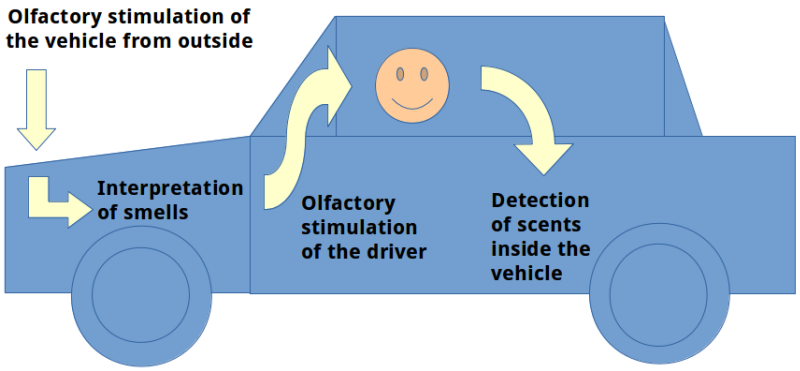


Figure 1: Olfactory stimulation in and outside the vehicle

As we drive, we pass by different locations with different smells. We might find them informative (smell of pizza tells us there might be an Italian restaurant near by), pleasant (e.g. smell of the ocean), disgusting (e.g. passing by a pig farm) or warning (smell of something burning triggers awareness of danger). If the car could "smell" these scents, it could react in an appropriate way. In this case we are talking about the **car-outside-world interaction**. Such an interaction technique can help detecting unpleasant smells in the environment the car is currently passing through and stop them from entering the vehicle's air condition. In this case the car could for instance stop sucking the fresh air from outside and circulate the air available inside the cabin for a couple of minutes. Same approach could be applied for enjoyable smells. In case of passing by a field with freshly cut grass, the olfactory system could enhance this smell making the air inside the car even more refreshing. The challenge behind this is however to develop an artificial nose the car could "breathe" with, like the ones described by eNose [eNo14].

Modern driving assistance systems receive data from so many devices both inside and outside the car (e.g. car engine data, satellite data, temperature data). This data can be partially delivered to the driver by smells as well. I call this the **car-driver interaction technique**. In this interaction paradigm smells can support or even replace many peaces of information brought to the user. This can shrink the amount of visual information and relieve the person holding the steering wheel. Scents can warn him or her (e.g. a smell

of rain, if bad weather has been reported), motivating the driver to slow down, suggest making a brake in case of a long driving (e.g. applying a smell of coffee) or deliver a smell stored in the user's phone book in case of an incoming call. The challenge behind this implementation is to choose the right smell for every single peace of data brought to the driver by the olfactory system.

Finally, the car can recognize smells inside the cabin. If the artificial nose for instance identifies the scent of sweat, this might mean the driver is in stress. Calm music could be turned on by the car's infotainment system in this moment, in order to help the driver relax. Such a way of interaction is to classify as the **cabin-smell-response**. What's challenging about this approach, is the correct sensing and interpretation of smells. Application of artificial intelligence might be necessary for processing the decision tasks here.

4 Use Cases of Olfactory Interaction

In order to present practical examples of how the different interaction techniques can be implemented, this paper discusses the scenarios of uses cases for three different olfactory experiences taking place in real life driving situations. These use cases form a basis for the creation of the first interaction prototype.

In the first use case (see Figure 2) Oliver is driving from Munich to Berlin to visit his friend during the long weekend. The journey starts on Friday. It is going to be a long ride, so he decides to make a short day at work and leave right after the lunch. Oliver had a good lunch, but after 2,5 hours of driving he gets a little bit hungry and feels like having a good coffee with a peace of cake. Since he's an occasional driver on this route, he doesn't know where the bakeries are placed next to the highway. Oliver doesn't want to stop and start searching for the place using his smart phone. He wants to spend as little time on his way as possible. In this moment he activates the "bakery search assistant" on the touch screen of the board computer and the system starts searching for a bakery itself. Oliver is an attentive driver and doesn't want to get disrupted by any additional textual messages. In a few minutes the system detects a bakery just 5 km away from the current location and exposes the smell of cinnamon to let Oliver know the desired spot is nearby. Now Oliver only needs to press the activation button next to the gearshift and the system turns on the navigation tool, which guides Oliver to the bakery.

Figure 2: Oliver is feeling hungry during the long driving

In use case two Sophia is challenging her driving assistance system in the scope of olfactory enjoyable route leading to her office (see Figure 3). Sophia lives in an industrial city. Factory districts of it are well known for unpleasant smells. As Sophia has to drive throughout the whole city to her job every day, she wants to avoid bad odors on her way and is ready to take a longer, but more pleasant path. She lets the driving assistance system of her car analyze the "smelly map" of her city and find the sequence of streets that smell the best. Thursday evening Sophia was out with friends and wakes up late on Friday morning. Suddenly she realizes, she has an appointment starting very soon. Sophia is in a hurry, so she chooses "the fastest route" in her car's navigation system. The fastest route leads her through the highway on the suburb of the city. "Smelly maps" have no data about this area. On her way she passes by a dairy farm. As the car gets out of the coverage of the "smelly map", the car's odor's sensor gets activated. The sensor detects a smell which is classified as "stinky". In this moment the car's air conditioning system locks the flow of the air from the outside environment and starts circulating the air already available inside of it, injecting some refreshing fragrance. Sophia gets to the office fast while still enjoying a very good olfactory experience.

Figure 3: Sophia is having a great olfatory experience while driving through the area of bad smell

The third use case pictures Jack driving to the job interview (see Figure 4). He wants to get this job very much and is very nervous. Jack's heart is beating fast and he starts to sweat. The car detects the smell of the sweat. As a response, the car cools the air temperature inside the cabin, starts playing relaxing music and exposing relaxing odors. Jack starts calming himself down and arrives to the appointment much more unbent and self confident. He leaves a good impression on his potential employee.

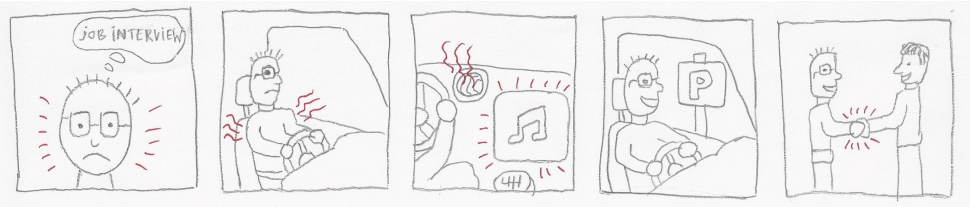


Figure 4: Jack is feeling nervous when driving to a job interview

As we can see, use cases introduced here point out different issues and practices, but all of them indicate how olfaction can amend the pleasure and comfort of driving.

5 Discussion

As we can see, smell is an ultimate interaction tool with an enormous emotional potential, due to what it can bring HMI to a brand new level. There is quite some research already done in this field, but barely any to almost no implementations in the automotive context.

Surely, spreading smells inside the cabin could enhance the pleasure of driving and even serve as an information delivery mechanism. Still we have to be aware of the constraints set by the cabin of the car, which is not that easy to ventilate (ventilation needs time). A smell is relatively easy to generate and spread, but hard to eliminate or neutralize [BKLP04]. This problem could be solved applying an "olfactory white" [WSY+12] a "white" smell generated by mixing many different scents in same proportions, which results in different odors actually smelling same. However, it's still a question, if the "white" smell can "hide" any other smell, or just a certain set of smells, as well as how it influences our perception of new smells entering the air. Another solution would be to use powerful air circulation, yet no previous work could be found on this matter. Finally, one could simply introduce time-outs for smell distribution in order to prevent overlapping of different scents.

Of course, all people are different and everyone of us has his or her own awareness of smells, that is why extensive surveys on olfactory experiences, like [OTH14] are essential. Only based on the opinion of a large group of people one can make a decision about what odors to classify as pleasant or undesirable in driving process.

Finally it is important to mention that so far nobody knows exactly how a human is going to respond to olfactory signals while driving. A psychological research might be necessary on this matter. Another important aspect is setting up an olfactory interface within the car in the most efficient way. The questions like where to place the scent emitter and how to remove the smell after the interaction are still open.

6 Future Work

The next step to take now is to elaborate on the designed use cases and to develop a working prototype of a vehicle olfactory system. For this purpose it would be necessary to create a hard- and software set-up imitating the interior of the car. The next step is to find or create own programmable scent generator. Every certain scent this devise is able to emit needs to be assigned to a particular system message the car would normaly convey to the driver in the form of a visual signal or a sound. These messages would then trigger the distribution of different smells. An important challenge on this point is the priority management of the "smelly" signals and decision algorithm on when and which signals can be brought to the driver by odors, in the way they don't colide with each other and the interaction as well as driving process stays comfortable. For the decision on which scent and when to apply, I would rely on the existing surveys on smell, complementing them with my own user studies.

7 Conclusions

As we can see, there are many problems one would need to deal with on the way of introducing an olfactory interface inside the vehicle. Nevertheless, emitting and processing smells during the driving process would lead to a revolution of automotive HMI applications. Suggestions on interaction techniques claimed in this paper would not only make driving more enjoyable, but make modern cars even smarter. An extensive user study is necessary though to make sure the novel approach works in its every aspect.

Acknowledgements

The author would like to thank Dr Marianna Obrist from the University of Sussex for the motivation to write this paper and for the subject-specific support. Special thanks also to Dr Christine Braeuning for encouraging the author to apply for this conference.

References

[BD95] W. Barfield and E. Danas. Comments on the Use of Olfactory Displays for Virtual Environments. *Presence*, 1995.

[BKLP04] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2004.

[BMM06] S. Brewster, D. McGookin, and C. Miller. Olfoto: designing a smell-based interaction. *Proc. CHI*, 2006.

[Dal96] Pamela Dalton. Odor Perception and Beliefs about Risk. *Oxford University Press*, 1996.

[Edw14] D. Edwards. oPhone DUO Project. Technical report, 2014.

[eNo14] eNose Technology. Technical report, The eNose Company, 2014.

[HE96] R.S. Herz and T. Engen. Odor memory: review and analysis. *Psy. Bulletin & Review*, 1996.

[INE15] Grundlagen interaktions- und emotionssensitiver Assistenzsysteme (INEMAS). Technical report, KU Eichstaett-Ingolstadt, 2015.

[MYI13] H. Matsukura, T. Yoneda, and H. Ishida. Smelling Screen: Development and Evaluation of an Olfactory Display System for Presenting a Virtual Odor Source. *IEEE TVCG*, 2013.

[NLV10] Fatma Nasoz, Christine L. Lisetti, and Athanasios V. Vasilakos. Affectively intelligent and adaptive car interfaces. *Elsevier*, 2010.

[OG15] Jonas K. Olofsson and Jay A. Gottfried. The muted sense: neurocognitive limitations of olfactory language. *Trends in Cognitive Sciences*, 2015.

[OTH14] M. Obrist, A. Tuch, and K. Hornbk. Opportunities for Odor: Experiences with Smell and Implications for Technology. *In Proceedings CHI14, SIGCHI Conference on Human Factors in Computing Systems*, 2014.

[QSAM15] Daniele Quercia, Rossano Schifanella, Luca Maria Aiello, and Kate McLean. Smelly Maps: The Digital Life of Urban Smellscapes. 2015.

[Sce10] Scent Generator. Technical report, Biophysical Human Research Institute, Ljubljana, 2010.

[Sch00] H. Schumann. *Visualisierung: Grundlagen und allgemeine Methoden*. Springer, 2000.

[Sen] Smell Generator. Technical report, Sensoryco.TS.

[WL06] J. Willander and M. Larsson. Smell your way back to childhood: Autobiographical odor memory. *Psy. Bulletin & Review*, 2006.

[WSY+12] T. Weiss, K. Snitz, A. Yablonka, R. Khan, D. Gafsou, E. Schneidman, and N. Sobel. Perceptual convergence of multi-component mixtures in olfaction implies an olfactory white. 2012.

[Yam14] T. Yamada. Mist Shine Prototype. Technical report, 2014.

[ZE99] M. Zybura and G. A. Eskeland. Olfaction for Virtual Reality. *University of Washington*, 1999.

Utilisation of ℓ-Diversity and Differential Privacy in the Anonymisation of Network Traces

Shankar Lal
Aalto University, Finland
shankar.lal@aalto.fi

Ian Oliver, Yoan Miche
Security Research
Nokia Neworks, Finland
first.last@nokia.com

Abstract: Noise addition for anonymisation is a known technique for increasing the privacy of a data sets. However this technique is often presented as individual and independent, or, just stated as techniques to be applied. This increases the danger of misapplication of these techniques and a resulting anonymised data set that is open to relatively easy re-identification or reconstruction. To better understand the application of these techniques we demonstrate their application to a specific domain - that of network trace anonymisation.

1 Introduction

Privacy and especially anonymisation of data sets is a hot topic. It, therefore, comes as no surprise that data sets such as network traces which contain large amounts of sensitive information about the behavior of users on a network require such treatment.

Techniques such as suppression, hashing and encryption of fields suffice to a point. In the case of suppression information is lost, while in hashing or encryption of data, the information content is transformed, from say an IP address which identifies a particular machine, into just some kind of generic identifier. In many cases a pattern of behavior is still recognizable, for example, hashing source and target IP addresses still reveals a unique pattern of communication if not the precise identities [1] [2].

More advanced techniques such as κ-anonymity [3], ℓ-diversity [4] and differential privacy [5] (amongst others) have been developed; κ-anonymity in particular has been successfully used with medical data. These techniques are now being recommended, if not mandated, to be used in the process of anonymisation.

In this paper, we present techniques for anonymising network traces that preserve some degree of statistical properties such that some meaningful analysis can still be made. Working in this specific domain means that we can carefully tailor techniques such as differential privacy such that a reasonable degree of privacy is assured.

2 Network trace files

A network trace file contains sensitive fields such as source and destination IP addresses, protocol type, packet lengths etc. Some of these can further act as quasi identifiers whose combination can lead to the identification of an individual despite seemingly identifying fields being removed or anonymised [6]. The source/destination IP address and time-stamp field can disclose who is talking to whom and also provide proof that communication existed between the parties in certain period of time. Protocol field is crucial in the sense that certain protocols can identify the nature of traffic.

Packet length field refers to the total length of an IP packet which includes payload and header information. This field is also very important from the security point of view, since certain security incidents have fixed packet length for example some network worms i.e. Slammer worm and Nachi worm have fixed packet length of 404 bytes and 92 bytes respectively [16] [10]. The packet length field is also vital in the sense that major transport protocols like TCP and UDP, mostly have packets of larger length e.g. around 1500 bytes. The other management protocols for example ICMP, DNS etc. have packet lengths mostly fewer than 200 bytes. Therefore due to this structure of the fields, it, sometime, can be easy to guess the protocol type by checking its packet length.

3 Overview of anonymisation techniques

Data anonymity can not be ensured by employing any single anonymisation technique as each technique has its advantages and disadvantages. All data sets are required to be processed through any combination of the techniques presented here and the many variations thereof [7].

3.1 Differential Privacy

The notion of adding noise and randomized values to the data in a controlled manner is known as differential privacy and provides a technique suited to continuous data sets such as location data, or in our case, data such as packet length and time-stamp data.

Consider two neighboring data sets D1, D2. The neighboring data sets are the data sets which differ only from one entry, one row or one record. They produce output S when mechanism $\mathcal{K}$ which satisfies ϵ-differential privacy, is applied. The mechanism $\mathcal{K}$ can be a randomized function to add jitter to some data field, fulfills the condition about the information disclosure related to any individual. Differential privacy algorithm states that probability of data set D1 producing output S is nearly the same as the probability of data set D2 producing same output.

Dwork's [5] definition of differential privacy is following:

A mechanism K satisfies ϵ-*differential privacy* if for all pairs of adjacent databases D' and D'', and all $S \subseteq Range(\mathcal{K})$,

$$Pr[\mathcal{K}(D') \in S] \leq e^{\epsilon} \times Pr[\mathcal{K}(D'') \in S] \quad (1)$$

Here, ϵ is known as privacy parameter. The ϵ-value corresponds to the strength of the privacy, smaller value of ϵ usually returns better privacy. Differential privacy uses Laplace noise which is calculated by following formula Δf / ϵ.

Where Δf is the sensitivity of the function and defined as the maximum amount the outputs that can be perturbed by adding or removing some records from the data sets. This measures how much output can be altered if a response is either included or excluded from the result. To get an idea about the value of sensitivity, if a data set is queered that 'how many rows have certain property, will yield sensitivity value of 1'. In our network trace, we first generate query that "what is the value of packet length/time-stamp field" which gives us sensitivity value as 1 and then we add Laplace noise to the queried field using suitable ϵ-value.

3.2 ℓ-diversity

The technique of ℓ-diversity involves partitioning fields within the data set such that within each group there exists a balanced representation [8]. This addresses a number of weaknesses in the κ-anonymity techniques such as homogeneity attacks [9].

The discrete field in network trace such as protocol is also sensitive and need to be anonymised. For example, some specific protocols like BitTorrent used for file sharing or UDP mostly used for video streaming, can possibly identify the nature of traffic. Machanavajjhala et al. [4] define ℓ-diversity principle as:

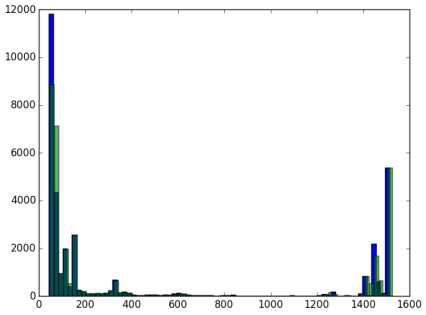
"A q-block is ℓ-diverse if contains at least ℓ "well-represented" values for the sensitive attribute S. A table is ℓ-diverse if every q-block is ℓ-diverse."

4 Implementation of anonymisation techniques

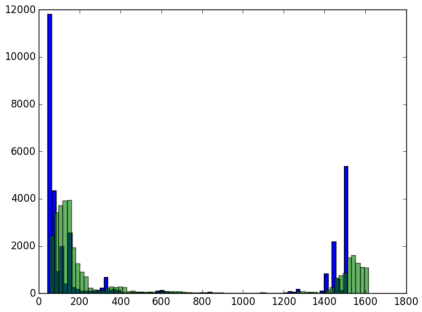
In this section we apply the Differential Privacy and ℓ-diversity techniques to a network trace along with guidelines on how the techniques are to be applied in this domain.

4.1 Differential Privacy over Continuous Fields

Differential privacy is suitable for continuous numerical fields so we add random Laplace noise to time-stamp and packet length field in our network trace. We have tried range of different values of ϵ to obfuscate data fields but for the sake of simplicity, we present the result based on two ϵ values in this paper. We first select ϵ value as 0.01 and plot histogram of both original and obfuscated data of time-stamp and packets length field and then we again plot the same histogram with ϵ value as 0.1 and compare both distributions. In figure 1 and 2, blue bars represents the original distribution of the data and green bars represent distribution of obfuscated data.



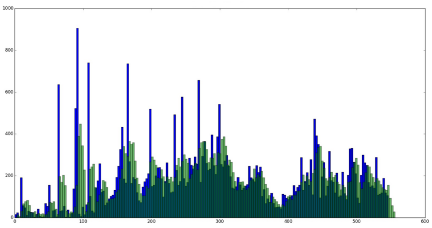
(a) Length Distribution with $\epsilon = 0.1$



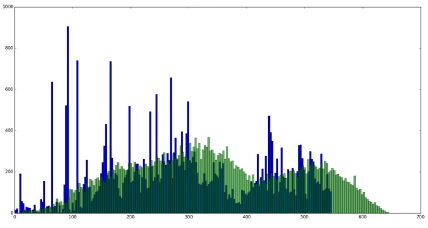
(b) Length Distribution with $\epsilon = 0.01$

Figure 1: Packet Length Distributions

From the figure 1(b) and 2(b), we can infer that noise magnitude with $\epsilon = 0.01$ has heavily modified the data and the obfuscated data does not follow the distribution of the original data any more. This type of noise destroys the useful features of the data and makes statistical analysis useless. While, as seen in figure 1(a) and 2(a), noise addition with $\epsilon = 0.1$ produce the identical distribution as the original one and implies that even if individual records of the data are perturbed but overall distribution of the data is preserved.



(a) Packet Frequency with Noise, $\epsilon = 0.1$



(b) Packet Frequency with Noise, $\epsilon = 0.01$

Figure 2: Time-stamp Distribution

The box plot in figure 3 compares the statistical parameters of original data and set of obfuscated data with different ϵ-value. It can be seen in figure 3 that obfuscated packet length field with ϵ-values 0.1 maintains almost similar features of the box plot such as Minimum, Maximum, Median, Q1 and Q3 values. In our experiment, it turned out that ϵ value of 0.1 is the most suitable for our network trace.

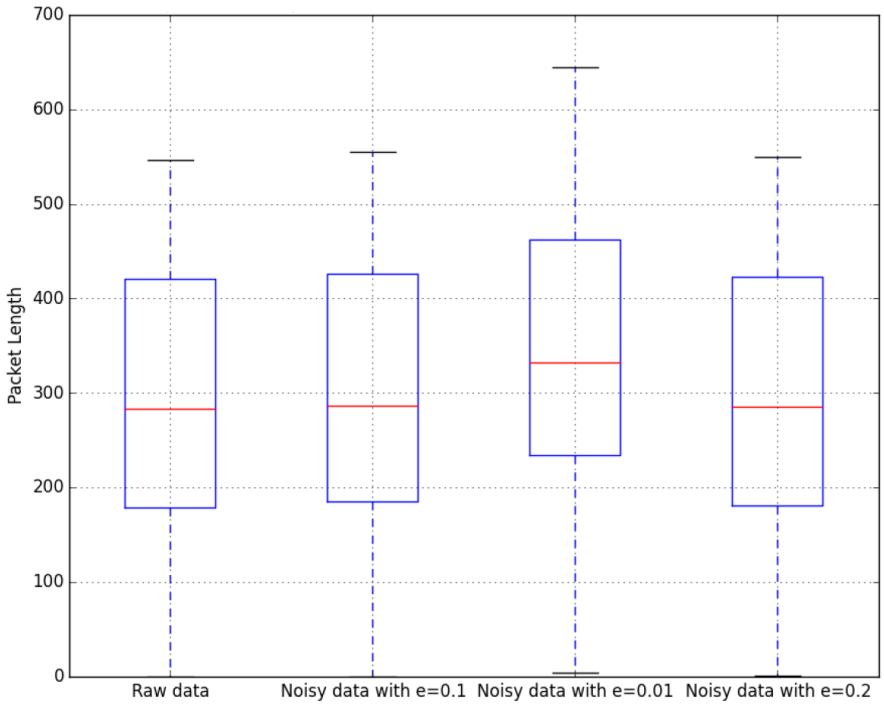


Figure 3: Distribution Spread of Packet Length

4.2 ℓ-Diversity over Discrete Fields

Protocol field can be grouped in a family equivalence class, where family name represents the characteristics of its members. To create the protocol equivalence classes, we first examine the type of protocols present in the network trace and then we group protocols of similar functionalities and put them in their respective equivalence class. In order to obfuscate the protocol field, we can replace it with its equivalence class name. The benefit of doing so is to avoid any inference attack which might occur if original protocol field is exposed.

Our anonymised network trace consist of 5 equivalence classes namely Transport protocols, Management protocols, Security protocols, Mobile network protocols and other protocols. Each equivalence class contains protocols with similar functionalities for example major transport protocols such as TCP and UDP are placed in Transport protocol equivalence class. Although, replacing the protocol field with its equivalence class ruins some amount of data but still provides enough information about the types of protocol, being anonymised, this is actually the trade-off between privacy and data utility.

After replacing the original protocol field with its respective equivalence class, the percentage of each equivalence class present in anonymised trace can easily be calculated and plotted using pie chart as shown in Figure 4. Figure 5 presents one sample of 5-diverse network trace with each block of data containing 5 diverse values of equivalence class.

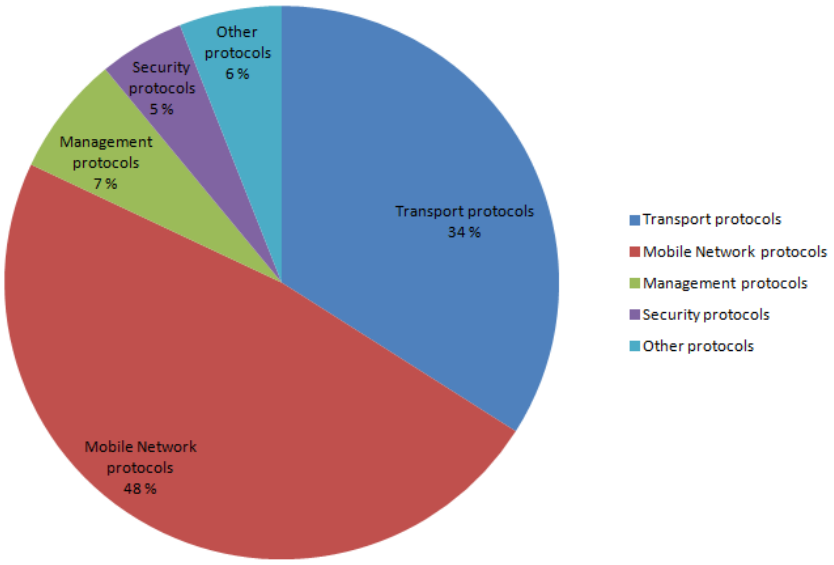
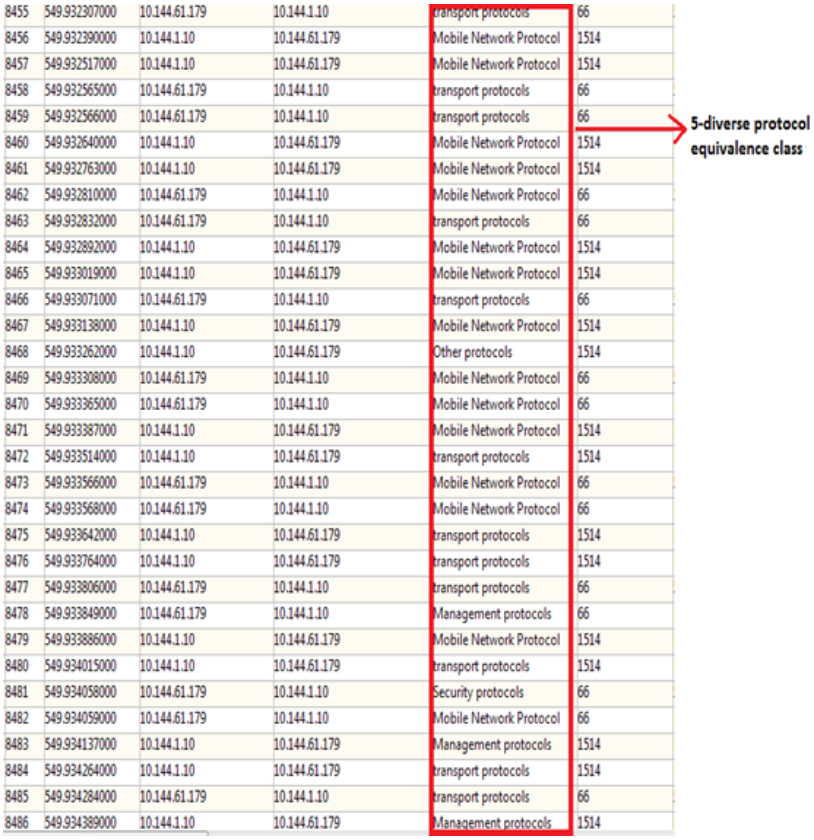


Figure 4: Protocol Equivalence Class Distribution

4.3 IP address anonymisation

There exist a number of ways to anonymise IP addresses. Traditionally, IP address are anonymised by using hashing methods or converting them to some real number. The problem with these methods is that they do not provide any means to carry out statistical analysis over anonymised IP addresses.
We tried two different methods to anonymise IP addresses. First method suppresses the last octet of the IP address while keeping other octets as intact e.g. 192.168.1.$*$. This technique



8455	549.932307000	10.144.61.179	10.144.1.10	transport protocols	66
8456	549.932390000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8457	549.932517000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8458	549.932565000	10.144.61.179	10.144.1.10	transport protocols	66
8459	549.932566000	10.144.61.179	10.144.1.10	transport protocols	66
8460	549.932640000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8461	549.932763000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8462	549.932810000	10.144.61.179	10.144.1.10	Mobile Network Protocol	66
8463	549.932832000	10.144.61.179	10.144.1.10	transport protocols	66
8464	549.932892000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8465	549.933019000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8466	549.933071000	10.144.61.179	10.144.1.10	transport protocols	66
8467	549.933138000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8468	549.933262000	10.144.1.10	10.144.61.179	Other protocols	1514
8469	549.933308000	10.144.61.179	10.144.1.10	Mobile Network Protocol	66
8470	549.933365000	10.144.61.179	10.144.1.10	Mobile Network Protocol	66
8471	549.933387000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8472	549.933514000	10.144.1.10	10.144.61.179	transport protocols	1514
8473	549.933566000	10.144.61.179	10.144.1.10	Mobile Network Protocol	66
8474	549.933568000	10.144.61.179	10.144.1.10	Mobile Network Protocol	66
8475	549.933642000	10.144.1.10	10.144.61.179	transport protocols	1514
8476	549.933764000	10.144.1.10	10.144.61.179	transport protocols	1514
8477	549.933806000	10.144.61.179	10.144.1.10	transport protocols	66
8478	549.933849000	10.144.61.179	10.144.1.10	Management protocols	66
8479	549.933886000	10.144.1.10	10.144.61.179	Mobile Network Protocol	1514
8480	549.934015000	10.144.1.10	10.144.61.179	transport protocols	1514
8481	549.934058000	10.144.61.179	10.144.1.10	Security protocols	66
8482	549.934059000	10.144.61.179	10.144.1.10	Mobile Network Protocol	66
8483	549.934137000	10.144.1.10	10.144.61.179	Management protocols	1514
8484	549.934264000	10.144.1.10	10.144.61.179	transport protocols	1514
8485	549.934284000	10.144.61.179	10.144.1.10	transport protocols	66
8486	549.934389000	10.144.1.10	10.144.61.179	Management protocols	1514

Figure 5: A sample of 5-diverse anonymised network trace

ensures that user who generated the data packet cannot be traced back while on the other hand provides information about the network topology which might be useful in certain analysis. In the second method, we replaced IP addresses with their corresponding class type for example IP address 192.168.1.10 is replaced by Class C and so on. Although, this technique ruins the information about the network ID and subnet mask but still provides some knowledge about IP address class and range of the addresses.

5 Clustering analysis of obfuscated data

In this section, we experiment with the obfuscated data obtained after applying above anonymisation techniques to observe its usefulness. This experiment uses packet flow records calculated from obfuscated data which become available for clustering.

Flow statistics [11], [12] are a set of measurements of the traffic data. Analyzing the traffic at the flow level (and not at the packet level) allows for several types of analysis, typically usage-based charging, network anomaly detection, identification of the heavy users, etc. In practice, and for on-line systems and analysis, there is usually a mandatory sampling of the packets, as direct measurement at the flow level on a high-speed link is not possible in terms of CPU requirements, as well as memory and storage matters. Ideally, one would want to use all packet data to compute the flows for higher precision of the calculated statistics [13], [14]. In the case of this paper, this is actually possible, as we only consider off-line network traces for the experiment presented. We have used a NetMate-flowcalc based approach, the software is named *flowtbag*, which is specifically designed for off-line flow statistics extraction.

5.1 Practicalities about data processing

The overall methodology for this clustering-based analysis is described on Figure 6, and in more details in the following.

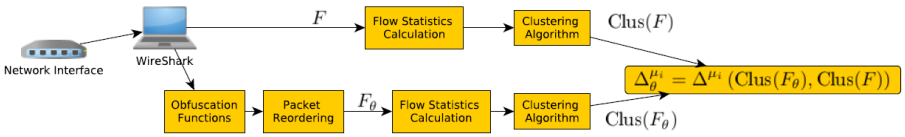


Figure 6: Overall block diagram of the data processing flow used.

Given a certain network traffic interface G, the traffic passing through this interface over a certain period of time Δt is sent to a computer running WireShark (latest development version) and dumping the traffic G(Δt) to a PCAP file F.

This PCAP data file F is then sent directly to flow statistics extraction, which will compute features that are directly usable for clustering. The resulting clustering is denoted Clus(F).

The very same file F is also run through a set of functions, each of which is parametrized by θ, and will obfuscate some of the data, leading to the file Fθ. The θ parameter controls the amount of obfuscation applied to the traffic data. The Fθ file then needs to be re-ordered, if some noise has been applied to the time field of the PCAP records, as this will have put the packets out of order, and therefore rendered this data unusable for computing the flow statistics. The obfuscated data Fθ is re-ordered using the development version (1.99) of WireShark 2, which allows for doing this directly on the PCAP file (using the associated reorderpcap tool).

The re-ordered obfuscated PCAP file is then sent through the very same clustering as the original file (non-obfuscated). This results in a certain clustering of the data which we will denote Clus(Fθ).

5.2 Remarks

Using flowtbag, on a Linux based computer with 6GB RAM, and i5-4300U@1.9GHz CPU, we obtained the following processing speeds for a PCAP network trace:

- 14:7 secs to extract about 106124 packets for flow statistics;
- About 8200 packets/sec;
- About 0:00012 sec/packet.

Figure 7 (a) and (b) shows the effect of noise and ϵ-value on the flow records available for clustering. It can be noted, from Figures 7 (a) that the number of available flow statistics records for analysis depends heavily on the noise value used. Indeed, if the introduced noise in the time-stamp of the packets is too large, the flow statistics, which are directly based on this, become impossible to compute for a large amount of packets.

As expected, from Figure 7 (b), the dependency of the number of available flow records to the ϵ value is almost non-existent, compared to that of the noise.

In order to have meaningful results for the clustering part, it is thus necessary to have reasonable values for both the noise amount and ϵ. In the following experiment, ϵ is varied to observe its influence on the clustering performed.

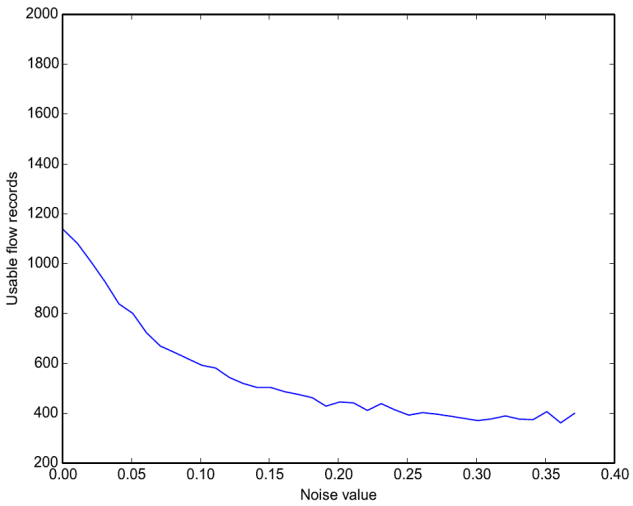


Figure 7: Overall block diagram of the data processing flow used.

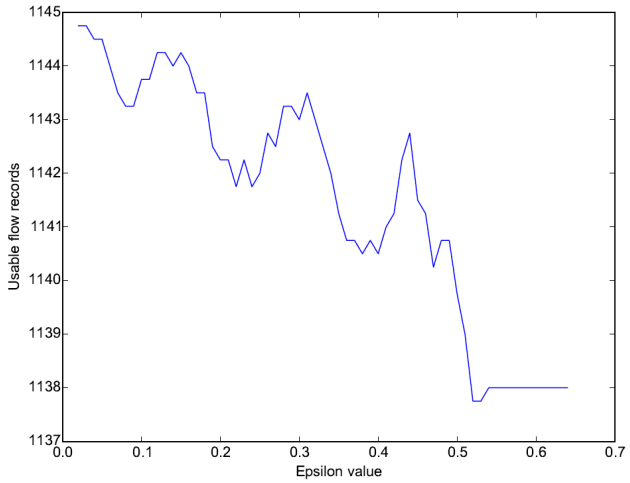


Figure 8: Overall block diagram of the data processing flow used.

6 Discussion

In this paper, we have presented anonymisation techniques that can be applied specifically in the network trace domain for the practice of anonymisation. Specifically we have emphasized on differential privacy and ℓ-diversity as these are lightly used for any anonymisation (with respect to privacy) and also being techniques that are being promoted by the privacy community.

One of the dangers of any type of anonymisation techniques is that techniques are either applied to single fields, ignoring the presence of functional dependencies and quasi-identifiers, or are applied without context to the semantics domain of the data. In this paper we have shown the application of differential privacy and ℓ-diversity to obfuscate data in the network trace domain.

As plain statistical analysis is just one mechanism for understanding the underlying data, so machine learning provides a more sophisticated manner in which re-identification might be made. While this work is still at a relatively early stage and understanding of how differential privacy and other forms of noise addition techniques effect some analysis such as clustering, will become critical to preserve privacy.

Acknowledgements

This paper was partially funded by the TEKES CyberTrust Programme.

References

[1] C. C. Aggarwal and P. S. Yu, *A general survey of privacy-preserving data mining models and algorithms, in Privacy-Preserving Data Mining, ser. The Kluwer International Series on Advances in Database Systems, C. C. Aggarwal, P. S. Yu, and A. K. Elmagarmid, Eds. Springer US, 2008, vol. 34, pp. 1152.*

[2] P. Langendorfer, M. Maaser, K. Piotrowski, and S. Peter, *"Privacy Enhancing Techniques: A Survey and Classification." Information Science Reference, 2008.*

[3] L. Sweeney, *"κ-anonymity: A model for protecting privacy" Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol. 10, no. 5, pp. 557570, Oct. 2002.*

[4] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, *"ℓ-diversity: Privacy beyond κ-anonymity, 2013 IEEE 29th International Conference on Data Engineering (ICDE), vol. 0, p. 24, 2006.*

[5] C. Dwork, *"Differential privacy: A survey of results, in Theory and Applications of Models of Computation," ser. Lecture Notes in Computer Science, vol. 4978. Springer Verlag, April 2008, pp. 119.*

[6] R. Motwani and Y. Xu, *Efficient algorithms for masking and finding quasi-identifiers, in Proceedings of the Conference on Very Large Data Bases (VLDB), 2007.*

[7] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, *Unique in the Crowd: The privacy bounds of human mobility, Scientific Reports, vol. 3, March 2013.*

[8] N. Li and T. Li, *t-closeness: Privacy beyond κ-anonymity and ℓ-diversity, in In Proc. of IEEE 23rd International Conference on Data Engineering (ICDE07*

[9] A. Friedman, R. Wolff, and A. Schuster, *Providing κ-anonymity in data mining, The VLDB Journal, vol. 17, no. 4, pp. 789804, Jul. 2008.*

[10] W. Yurcik, C. Woolam, G. Hellings, L. Khan, and B. M. Thuraisingham, *Toward trusted sharing of network packet traces using anonymization: Single-field privacy/analysis tradeoffs, CoRR, vol. abs/0710.3979, 2007.*

[11] S. Handelman, S. Stibler, N. Brownlee, and G. Ruth, *RFC 2724: RTFM: New attributes for traffic flow measurement, 1999. [Online]. Available: http://tools.ietf.org/html/rfc2724*

[12] N. Brownlee, *Network management and realtime traffic flow measurement,Journal of Network and Systems Management, vol. 6, no. 2, pp. 223228, 1998.*

[13] C. Estan and G. Varghese, *New directions in traffic measurement and accounting, SIGCOMM Comput. Commun. Rev., vol. 32, no. 4, pp. 323336, Aug. 2002. [Online]. Available: http://doi.acm.org/10.1145/964725.633056*

[14] N. Duffield, C. Lund, and M. Thorup, *Properties and prediction of flowstatistics from sampled packet streams, in In Proc. ACM SIGCOMM Internet Measurement Workshop, 2002, pp. 159171.*

[15] S. Lloyd, *Least squares quantization in pcm, Information Theory, IEEE Transactions on, vol. 28, no. 2, pp. 129137, Mar 1982*

[16] Chen, Thomas M. *"Intrusion detection for viruses and worms." IEC Annual Review of Communications 57 (2004).*

On the Regularization of Chattering Executions in Real Time Simulation of Hybrid Systems

Ayman ALJARBOUH
INRIA/IRISA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France
email: ayman.aljarbouh@inria.fr

Abstract: In this paper we present a new method to perform the higher order sliding modes analysis of trajectories of hybrid systems with chattering behavior. This method improves our previous work [AC15] as it modifies numerical simulation algorithms to make them compute the higher order terms of the normal unit vectors of the systems dynamics whenever the 1^{st} order sliding mode theory cannot be applied. Such modification does not affect the generality of our previous contribution in [AC15]. Our algorithm is general enough to handle both chattering on a single $\mathbb{R}^{n-1}$ switching manifold (i.e. chattering between two dynamics) as well as chattering on the intersection of finitely many intersected $\mathbb{R}^{n-1}$ switching manifolds. In this last case, we show by a special hierarchical application of convex combinations, that unique solutions can be found in general cases when the switching function takes the form of finitely many intersecting manifolds so that an efficient numerical treatment of the sliding motion constrained on the entire discontinuity region (including the switching intersection) is guaranteed. Illustrations of the techniques developed in this article are given on representative examples.

1 Introduction

Because of their heterogeneous composition, the word hybrid is attached to dynamical systems which contain state variables that are capable of evolving continuously (flowing) and/or evolving discontinuously (jumping) [CGST07]. That is, the presence of two different behaviors, continuous and discrete, is the cause of heterogeneity. Systems of this type are common in embedded computation, robotics, mechatronics, avionics, and process control [ZJLS01] [CGST07]. Hybrid systems also arise naturally in control systems where the value of a control variable may jump or whenever the laws of physics are dicontinuous. Typically, the continuous dynamics of the system in the different operation modes are described by sets of ordinary differential equations or differential-algebraic equations. The changing between different operation modes is modeled by discrete transitions resulting in switching between sets of equations describing each operation mode [LA09].
However, the interaction of continuous-time and discrete-time dynamics emerging from its components and/or their interconnection may lead to *chattering executions* [ZYM08]. Similar behavior appears in variable structure control systems and in relay control systems [JBÅ02]. Chattering executions can be defined as solutions to the system having infinitely many discrete transitions in finite time. Although physical systems do not show chattering behavior, models of real systems may be chattering due to modeling over-

abstraction. Physically, chattering behaviour occurs if equal thresholds for the transition conditions of different modes are given and the system starts to oscillate around them. On the other hand, numerical errors may lead to numerical chattering as transition conditions may be satisfied due to local errors. The numerical solution of a hybrid system exhibiting chattering behavior requires high computational costs as small step-sizes are required to restart the integration after each mode change. In the worst case, the numerical integration breaks down, as it does not proceed in time, but chatters between modes. The chattering behavior has to be treated in an appropriate way to ensure that the numerical integration terminates in a reasonable time. To deal with chattering executions of hybrid systems one needs to detect regions on the switching manifold, on which chattering can occur, and force the solution trajectory to slide on the manifold in these regions [dBBC$^+$08] [WKH14] [GST11]. An additional mode, the so-called sliding mode, can be inserted into the hybrid system to represent the dynamics during sliding, and thus, replaces the chattering. The sliding mode became the principle operation mode in so-called variable structures systems. A variable structure system consists of a set of continuous subsystems with control actions are discontinuous functions of the system state, disturbances (if they are accessible for measurement), and reference inputs. Filippovs differential inclusion method (the so-called equivalent dynamics) [Fil88] [BHJ13] is a method that was developed by Filippov to define the system dynamics on the switching surface in such a way that the state trajectory moves along the surface. In this method, regularizations of the solution trajectories on both sides in a small neighborhood around the surface are used to determine the average velocity on the surface. Another approach called equivalent control was presented by Utkin [Utk92]. For linear control, this is an identical approximation to the equivalent dynamics. However, nonlinear control may derive different behaviors since the true system behavior near the sliding surface can be attributed to hysteresis phenomena. The method of equivalent dynamics derives sliding behavior closer to the true dynamics than the method of equivalent control in these situations. In some cases, where system behavior is not well-behaved near the switching surface, i.e. the case of a saturated high gain amplifier, when system variable values tends to infinity close to the discontinuity, equivalent control may generate better approximations. The reason is that there are no higher order hysteresis effects, therefore, modeling with equivalent dynamics, which assumes hysteresis, results in the generation of deviant behaviors. However, the computation of the equivalent dynamics turns of to be difficult whenever the systems chatters between more than two dynamics or modes of operations, a scenario which may appears in control applications whenever there are multiple discontinuous control variables. Indeed, the computation of equivalent dynamics is a challenging task in special classes of hybrid systems where the data constraints in the system do not allow to determine the existence of chattering execution using the 1^{st} order theory of sliding modes.

As an extension to our previous work in [AC15], we propose in this aricle an adequate technique to detect the chattering set "on the fly " in real time simulation of hybrid systems using the higher order theory of sliding modes, and therefore circumvent it by appropriately regularization the execution of the system beyond the limit time of the infinitely fast discrete transitions. Our approach is based on mixing compile-time transformations of hybrid programs (generating what is necessary to compute the smooth equivalent dynamics), the decision at run-time of the necessary and sufficient conditions for entering and exiting

a sliding mode, and the computation, at run-time, of the smooth equivalent dynamics.
The rest of this article is organized as follows: In Section 2, we present our formalism of hyrbid systems and hybrid solution trajectory as well as the chattering execution. Then, in Section 3, we explain how we could detect and regularize the chattering execution for the most general case when the chattering set belongs to the intersection of p transversally intersected $\mathbb{R}^{n-1}$ switching manifolds in at least p dimensions for any finite (positive) integer p. Section 4 presents the higher order sliding mode analysis with application to control problems with relay feedback. Finally, the simulation results and conclusion of the study are given in Sections 5 and 6 respectively.

2 Preliminaries

In this section we provide a brief introduction to hybrid systems, their executions, and chattering behavior.

Definition 1. (Hybrid Dynamical System) Define a hybrid system as a tuple

$$\mathcal{H} = (Q, D, E, G, R, F)$$

where

- $Q = \{1, ..., M\} \subset \mathbb{N}$ is a finite set of *discrete states*,
- $D = \{D_q\}_{q \in Q}$ is a set of *domains (or invariants)*, where D_q, a compact subset of $\mathbb{R}^n$, describes the conditions that the continuous state x has to satisfy at the discrete state $q \in Q$,
- $E \subset Q \times Q$ is a set of *discrete transitions (or edges)*, which define the connection beween states by identifying the pairs (q, q'), where for each $e = (q, q') \in E$ we denote its source $s(e) = q$ and its target $t(e) = q'$,
- $G = \{G_e\}_{e \in E}$ is a set of *guards*, where $G_e \subseteq D_{s(e)}$,
- $R = \{\phi_e\}_{e \in E}$ is a set of *reset maps*, where for each $e = (q, q') \in E$, $R_e : G_e \subseteq D_{s(e)} \to D_{t(e)}$,
- $F = \{f_q\}_{q \in Q}$ is a set of *vector fields*, where for each $q \in Q$, $f_q : D_q \to \mathbb{R}^n$ is Lipschitz on $\mathbb{R}^n$ and describes through a differential equation ODE the continuous evolution of the continuous state variables in $q \in Q$. The solution to the ODE is denoted by $x_i(t)$, where $x_i(t_0) = x_0$.

Definition 2. (Execution of a Hybrid System) An *execution* or *hybrid trajectory* of a hybrid system is a tuple

$$\chi = (\tau, \xi, \rho)$$

where

- $\tau = \{\tau_i\}_{i\in\mathbb{N}}$ such that $0 = \tau_0 \leq \tau_1 \leq \cdots \leq \tau_i \leq \cdots$ is a set of *events (or switching) times*.
- $\xi = \{\xi_i\}_{i\in\mathbb{N}}$ is a set of *initial conditions* with $\xi_i \in D_q$ for some $q \in Q$.
- $\eta = \{\eta_i\}_{i\in\mathbb{N}}$ with $\eta_i \in E$ is a *hybrid edge sequence*. An execution χ must satisfy the conditions for $i \in \mathbb{N}$

  1. $\xi_i = x_{s(\eta_i)}(\tau_i)$
  2. $\tau_{i+1} = min\{t \geq \tau_i : x_{s(\eta_i)}(t) \in G_{\eta_i}\}$
  3. $s(\eta_{i+1}) = t(\eta_i)$
  4. $\xi_{i+1} = R_{\eta_i}(x_{s(\eta_i)}(\tau_{i+1}))$

  The first and second conditions say that an event must occur at time τ_{i+1}. The thrid condition says that the discrete evolution map must evolve in a way that is consistent with the edges. The fourth condition says that the initial conditions must be in the image of the guards under the reset maps. We also require that the flow must stay in the domain D_{η_i} (i.e. $x_{s(\eta_i)}(t) \in D_{s(\eta_i)}$) for all time in $[\tau_i, \tau_{i+1}]$.

Definition 3. (Lie Derivative) Assume the flow map f_q is analytic in its second argument, the *Lie derivatives* $\mathcal{L}^k_{f_q} g_q : \mathbb{R}^n \to \mathbb{R}^n$ of a function g_q, also analytic in its second argument, along f_q, for $k > 0$, is defined by:

$$\mathcal{L}^k_{f_q} g_q(x(t)) = \left(\frac{\partial \mathcal{L}^{k-1}_{f_q} g_q(x(t))}{\partial x(t)}\right) \cdot f_q(x(t)) \tag{1}$$

with

$$\mathcal{L}^0_{f_q} g_q(x(t)) = g_q(x(t)) \tag{2}$$

Definition 4. (Pointwise relative degree)
We define the relative degree $n_q(x) : \mathbb{R}^n \to \mathbb{N}$ by:

$$n_q(x) = k \; if \bigwedge_{j<k} \mathcal{L}^j_{f_q} g_q(x) = 0 \wedge \mathcal{L}^k_{f_q} g_q(x) \neq 0 \tag{3}$$

Definition 5. (Chattering Hybrid System)
A hybrid system $\mathcal{H}$ is chattering if for some execution χ of $\mathcal{H}$ there exist finite constants τ_∞ and C such that

$$\lim_{i\to\infty} \tau_i = \sum_{i=0}^{\infty} (\tau_{i+1} - \tau_i) = \tau_\infty \tag{4}$$

$$\forall i \geq C : \;\; \tau_{i+1} - \tau_i = 0 \tag{5}$$

3 Robust Detection and Regularization of Chattering Executions

Following our contribution in [AC15], we consider a hybrid automaton $\mathcal{H}$ with a finite set of discrete states $q \in Q$ with transverse invariants where the state space is split into 2^p open convex regions (sub-domains) $C_q \in \mathbb{R}^n$, $q = 1, ...2^p$, and p switching manifold $\gamma_j(x) \in \mathbb{R}^{n-1}$, $j = 1, 2, ..., p$ by the intersection of p transversally intersected $\mathbb{R}^{n-1}$ switching manifolds Γ_j defined as the zeros of a set of scalar functions $\gamma_j(x)$ for $j = 1, 2, ..., p$,

$$\Gamma_j = \{x \in \mathbb{R}^n : \ \gamma_j(x) = 0 \, ; \quad j = 1, 2, ..., p\} \tag{6}$$

The zero crossing in opposite directions defines the switching between two adjacent flow sets. We will assume that all γ_j are assumed to be analytic in their second arguments so the normal unit vector $\perp_j$ for each one of the intersected switching manifolds Γ_j is well defined. Moreover, $\perp_j$ are linearly independent for all the $\mathbb{R}^{(n-r)}$ intersections where $r \in \{2, 3, ..., n\}$.

We use the multi-valued function $\alpha_j(x)$ such that the convex set is given for all $\Gamma_j|_{j=1,2,...,p}$ and $C_i|_{i=1,2,...,2^p}$ by:

$$\dot{x} \in \sum_{i=1}^{2^p} \left((\prod_{j=1}^{p} \frac{1 + 2\Psi_{j,i} \cdot \alpha_j(x) - \Psi_{j,i}}{2}) \cdot f_i(x) \right) \tag{7}$$

$$\alpha_j(x) = \left\{ \begin{array}{cc} 1 & for \ \ \gamma_j(x) > 0 \\ [0, 1] & for \ \ \gamma_j(x) = 0 \\ 0 & for \ \ \gamma_j(x) < 0 \end{array} \right\} \tag{8}$$

$$\sum_{i=1}^{2^p} \left(\prod_{j=1}^{p} \frac{1 + 2\Psi_{j,i} \cdot \alpha_j - \Psi_{j,i}}{2} \right) = 1 \tag{9}$$

where $\Psi_{j,i}$ gives the sign of the switching function $\gamma_j(x)$ in the domain D_i. Equations (7), (8), and (9) yield in

$$\dot{x} \in (1 - \alpha_j) \cdot \sum_{i=1}^{2^p} (R_1 \cdot f_i(x)) + \alpha_j \cdot \sum_{i=1}^{2^p} (R_2 \cdot f_i(x)) \tag{10}$$

$$R_1 = \prod_{k=1; k \neq j; \Psi_{k,i} = -1}^{p} \left(\frac{1 + 2\Psi_{k,i} \cdot \alpha_k - \Psi_{k,i}}{2} \right) \tag{11}$$

$$R_2 = \prod_{k=1; k \neq j; \Psi_{k,i} = 1}^{p} \left(\frac{1 + 2\Psi_{k,i} \cdot \alpha_k - \Psi_{k,i}}{2} \right) \tag{12}$$

Define a matrix F of the normal projections $f_i^{\perp_j}(x)$ for $j = 1, 2, ..., p$ and $i = 1, 2, ..., 2^p$

as

$$F = \begin{Bmatrix} f_1^{\perp_1}(x) & f_1^{\perp_2}(x) & \cdots & f_1^{\perp_p}(x) \\ f_2^{\perp_1}(x) & f_2^{\perp_2}(x) & \cdots & f_2^{\perp_p}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ f_{2^p}^{\perp_1}(x) & f_{2^p}^{\perp_2}(x) & \cdots & f_{2^p}^{\perp_p}(x) \end{Bmatrix} \tag{13}$$

where

$$f_i^{\perp_j}(x) = \mathcal{L}_{f_i}\gamma_j(x) = \left(\frac{\partial \gamma_j(x)}{\partial x}\right)^T \cdot f_i(x) \tag{14}$$

In agreement with the sign matrix Ψ, the attractive chattering on any $\mathbb{R}^{(n-r)}$ switching manifold for $r = 2, 3, ..., n$ can be easily observed by checking the signs of the matrices F and Ψ.

Lemma 1:

The sufficient condition for having an attractive chattering on any switching intersection in the system's state space requires a nodal attractivity towards the intersection itself, for all the flow maps f_i in the $\mathbb{R}^n$ regions C_i associated to this intersection. That is, the following constraint should be satisfied

$$\forall i, j: \;\; sgn(f_i^{\perp_j}(x)) = -sgn(\Psi_{j,i}) \tag{15}$$

To keep the solution trajectory in a sliding motion on the intersection as long as the attractive chattering condition is satisfied we impose

$$\forall j = 1, ...p: \;\; \sum_{i=1}^{2^p}\left(\prod_{j=1}^{p} \frac{1 + 2\Psi_{j,i} \cdot \alpha_j - \Psi_{j,i}}{2}\right) \cdot f_i^{\perp_j}(x) = 0 \tag{16}$$

so that

$$\alpha_j = \frac{W_1}{W_1 - W_2} \tag{17}$$

$$W_1 = \sum_{i=1}^{2^p}\left(\prod_{k=1;k\neq j;\Psi_{k,i}=-1}^{p} \frac{1 + 2\Psi_{k,i} \cdot \alpha_k - \Psi_{k,i}}{2}\right) \cdot f_i^{\perp_j} \tag{18}$$

$$W_2 = \sum_{i=1}^{2^p}\left(\prod_{k=1;k\neq j;\Psi_{k,i}=1}^{p} \frac{1 + 2\Psi_{k,i} \cdot \alpha_k - \Psi_{k,i}}{2}\right) \cdot f_i^{\perp_j} \tag{19}$$

For all $\alpha_k \in (0, 1)$, the product term in (18) (respectively (19)) takes always a value in (0,1) since it is always a product of $(1 - \alpha_k)$ (respectively α_k). It holds always that $W_1 > 0 \wedge W_2 < 0$ as long as an attractive chattering takes place at $x \in \left(\bigcup_{j=1}^{p} \Gamma_j\right) \cap C$, where C is the entire flow set in the system phase space (i.e. $C = \bigcup_{q\in Q}\{D_q\}$). This gives us a hypercube convex hull of sign coordiantes (± 1, ± 1, $\cdots$, ± 1) with an edge of length 2 and $\mathbb{R}^n$ volume 2^p. Therefore, a solution to the fixed point non-linear problem

(16) exists. However, the uniqueness of the solution is no guaranteed. To deal with non-uniqueness on the intersection−on which the attractive chattering occurs−we propose to give an equivalent to the product term in (16) so that the sliding parameters are given in term of a rational function of coefficients κ_i:

$$\prod_{j=1}^{p} \frac{1 + 2\Psi_{j,i} \cdot \alpha_j - \Psi_{j,i}}{2} = \frac{\kappa_i}{\sum_{k=1}^{2^p} \kappa_k} \tag{20}$$

where

$$\kappa_i = \frac{\left(\prod_{l=1;l\neq i}^{2^p}(\Omega_l)\right)^{\frac{1}{2^p-1}}}{\left(\prod_{l=1;l\neq i}^{2^p}(\Omega_l)\right)^{\frac{1}{2^p-1}} - (\Omega_i)} \tag{21}$$

$$\Omega_i = [(b_i)_1\ (b_i)_2\ \cdots\ (b_i)_p] \cdot \begin{bmatrix} \mathcal{L}_{f_i}\gamma_1(x) \\ \mathcal{L}_{f_i}\gamma_2(x) \\ \vdots \\ \mathcal{L}_{f_i}\gamma_p(x) \end{bmatrix} \tag{22}$$

$$\Omega_l = [(b_l)_1\ (b_l)_2\ \cdots\ (b_l)_p] \cdot \begin{bmatrix} \mathcal{L}_{f_l}\gamma_1(x) \\ \mathcal{L}_{f_l}\gamma_2(x) \\ \vdots \\ \mathcal{L}_{f_l}\gamma_p(x) \end{bmatrix} \tag{23}$$

The vectors b_n for $n = i, l$ are given as sign permutations of coordinates $[\pm 1, \pm 1, \cdots, \pm 1]^T$ under the constraint:

$$sgn(b_n)_j = \left\{ \begin{array}{c} sgn(\mathcal{L}_{f_n}\gamma_j(x))\ \ for\ \ n = 1 \\ -sgn(\mathcal{L}_{f_n}\gamma_j(x))\ \ for\ \ n = 2, 3, ..., 2^p \end{array} \right\} \tag{24}$$

where $j = 1, 2, ..., p$ and p is the number of the intersected $\mathbb{R}^{(n-1)}$ switching manifolds. This gives always $\Omega_1 > 0$ and $\Omega_n < 0$ for all $n \in \{2, 3, .., 2^p\}$ which is exactly what we want. Another advantage of using the signs constraint in (24) is that $\kappa_j = 0$ for all $j \neq i$ when $\kappa_i = 1$ for a given index $i \in \{1, 2, ..., 2^p\}$, this allows us to detect when a switching regime of different dimension has been reached by the solution trajectory, and then, to select the appropriate vector fields on this regime. Moreover, the parameter κ_i takes always a value $0 \leq \kappa_i \leq 1$ for $i = 1, 2, ..., 2^p$, yields in $\sum_{i=1}^{2^p} \left(\frac{\kappa_i}{\sum_{k=1}^{2^p} \kappa_k}\right) = 1$, which is consistent with the approach of Filippov differential inclusion.

However, special structures in control problems may lead to very complicated situations. In particular, in linear control problems with relay feedback, the existence of 1^{st} order sliding modes can simply be determined from studying the normal projections $f_1^{\perp}(x(t)) = CAx + CB$ and $f_2^{\perp}(x(t)) = CAx - CB$ close to hyper switching manifold Γ. We see that depending on the value CB we can decide whether we should expect to have 1^{st} order sliding modes or not. Roughly speaking, if the data in the system are given such that $CB = 0$, that is, the sliding region Γ_s vanishes so that $f_1^{\perp}(x(t)) = f_2^{\perp}(x(t)) = 0$ then it necessary to deal with higher order conditions for both f_1 and f_2. We provide in the following the necessary and sufficient condition for the existence of multiple fast switches in linear control problems with relay feedback using higher order sliding modes analysis.

4 Sliding with Higher Order Conditions: Application to Control Problems

Relay feedback exists in a lot of control application such that automatic tuning of PID controllers, modeling of quantization errors in digital control, and the analysis of sigma-delta converters, friction models. By simply replacing the controller by a relay, measuring the amplitude and frequency of the possible oscillation to derive the controller parameters, a robust control design method is obtained. The relay feedback system consists of a dynamical system and a sign function connected in feedback. The sign function leads to a discontinuous differential equation. The problem given as a linear system with relay feedback is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= C^T x(t) \\ u(t) &= -sgn(y(t)) = \left\{ \begin{array}{c} 1 \;\; for \;\; y(t) < 0 \\ -1 \;\; for \;\; y(t) > 0 \end{array} \right\} \end{aligned} \tag{25}$$

The sign function is discontinuous at $y(t) = 0$, therefore, we have a hybrid system of two discrete states q_1 and q_2 where the phase space of the system is split by a single hyper switching surface $\Gamma = \{x \in \mathbb{R}^n : \gamma(x(t)) = 0\}$ into two domain: $D_1 = \{x \in \mathbb{R}^n : \gamma(x(t)) \leq 0\}$ and $D_2 = \{x \in \mathbb{R}^n : \gamma(x(t)) \geq 0\}$ so that opposed zero crossing of the switching function γ defines the switching from q_1 to q_2 and vice-versa. In our notation, we have $\gamma = C^T x(t)$ and $\Gamma = \Gamma_1 \cap \Gamma_2 = \{x \in \mathbb{R}^n : C^T x(t) = 0\}$. It is assumed that $\frac{\partial \gamma(x(t))}{\partial x(t)} = C^T \neq 0$ for all $x \in \Gamma$.

The system dynamics, including the sliding dynamics $f_s = \frac{1-\delta(x)}{2} \cdot f_1(x) + \frac{1+\delta(x)}{2} \cdot f_2(x)$ with $\delta(x) \in [-1, 1]$ on the sliding surface $\Gamma_s \subset \Gamma$, are given by

$$\dot{x} = \left\{ \begin{array}{ll} f_1(x(t)) = Ax(t) + \;\; B & for \;\; x(t) \in D_1 \\ f_s(x(t)) = Ax(t) - \delta B & for \;\; x(t) \in \Gamma_s \\ f_2(x(t)) = Ax(t) - \;\; B & for \;\; x(t) \in D_2 \end{array} \right\} \tag{26}$$

Suppose that $x_m = x(t_m) \in \Gamma$, so the 1^{st} order Lie derivatives $f_1^{\perp}(x_m)$ and $f_2^{\perp}(x_m)$ are given by

$$f_1^{\perp}(x_m) = \frac{\partial \gamma(x_m)}{\partial x_m} \cdot f_1(x_m) = C^T A x_m + C^T B \tag{27}$$

$$f_2^{\perp}(x_m) = \frac{\partial \gamma(x_m)}{\partial x_m} \cdot f_2(x_m) = C^T A x_m - C^T B \tag{28}$$

We have to consider the following two cases:

Case I. $C^T B \neq 0$: For which an attractive sliding motion takes place

1. at all $x_m \in \Gamma$ (i.e. $C^T x_m = 0$) if and only if $f_1^{\perp}(x_m) > 0 \wedge f_2^{\perp}(x_m) < 0$ satisfied by the constraint $|C^T A x_m| < C^T B$. Let's denote ζ_1 and ζ_2 to the two boundaries of the sliding surface Γ_s. These two boundaries are defined explicitely then by

$$\zeta_1 = \{x_m \in \Gamma : C^T A x_m = -C^T B\}; \quad \zeta_2 = \{x_m \in \Gamma : C^T A x_m = \;\; C^T B\}$$

2. at ζ_1 if and only if $\exists k > 1 \in \mathbb{N}$ such that $\left(\bigwedge_{i=0}^{1} f_1^{\perp^{(i)}}(x(t_m)) = 0 \wedge f_1^{\perp^{(k)}}(x(t_m)) > 0\right)$ satisfied by the constraint

   $$(C^T x_m = 0) \wedge (C^T A x_m = -C^T B) \wedge (\exists k > 1 \in \mathbb{N} : C^T A^k x_m > -C^T A^{k-1} B)$$

   Note that, since $C^T A x_m = -C^T B$ then $C^T A x_m < C^T B$ (i.e. $f_2^{\perp}(x_m) < 0$), and therefore, the sufficient condition for attractive chattering is already violated.

3. at ζ_2 if and only if $\exists k > 1 \in \mathbb{N}$ such that $\left(\bigwedge_{i=0}^{1} f_2^{\perp^{(i)}}(x(t_m)) = 0 \wedge f_2^{\perp^{(k)}}(x(t_m)) < 0\right)$ satisfied by the constraint

   $$(C^T x_m = 0) \ \wedge \ (C^T A x_m = C^T B) \ \wedge \ (\exists k > 1 \in \mathbb{N} : C^T A^k x_m < C^T A^{k-1} B)$$

   Similarly, since $C^T A x_m = C^T B$ then $C^T A x_m > -C^T B$ (i.e. $f_1^{\perp}(x_m) > 0$), and therefore, the sufficient condition for attractive chattering is already violated.

Case II. $C^T B = 0$: This case is considerably more complicated to decide whether we should expect to leave Γ or to slide on it by the standard 1^{st} order theory of sliding concept since we have from (27) and (28): $f_1^{\perp}(x_m) = f_2^{\perp}(x_m) = C^T A x_m$. One way to treat this special case is to consider the higher order conditions (i.e. the higher order norms of the projections of both f_1 and f_2 normal onto Γ). It is clear to realize that it is not allowed to have neither transversality nor attractive sliding motion on Γ unless $f_1^{\perp}(x_m) = f_2^{\perp}(x_m) = 0$ satisfied by the constraint $C^T A x_m = 0$. This constraint on the system dynamics represents the necessary (but not sufficient) condition for the existence of 2^{nd} order transversality/sliding on Γ.
Recalling (1), (2), and (25) with $x_m = x(t_m) \in \Gamma$, the 2^{nd} order normal projections $f_1^{\perp^{(2)}}(x_m)$ and $f_2^{\perp^{(2)}}(x_m)$ are given by

$$f_1^{\perp^{(2)}}(x_m) = \frac{\partial f_1^{\perp}(x_m)}{\partial x_m} \cdot f_1(x_m) = C^T A^2 x_m + C^T A B \tag{29}$$

$$f_2^{\perp^{(2)}}(x_m) = \frac{\partial f_2^{\perp}(x_m)}{\partial x_m} \cdot f_2(x_m) = C^T A^2 x_m - C^T A B \tag{30}$$

Similarly, we should consider the following two cases:

Case I. $C^T A B \neq 0$: For which a 2^{nd} order attractive sliding motion takes place

1. at all $x_m \in \Gamma$ (i.e. $C^T x_m = 0$) if and only if $f_1^{\perp^{(2)}}(x(t_m)) > 0 \wedge f_2^{\perp^{(2)}}(x(t_m)) < 0$ satisfied by the constraint $|C^T A^2 x_m| < C^T A B$. In this case, the two boundaries are defined explicitely then by

   $$\zeta_1 = \{x_m \in \Gamma : C^T A x_m = 0 \ \wedge \ C^T A^2 x_m = -C^T A B\}$$

   $$\zeta_2 = \{x_m \in \Gamma : C^T A x_m = 0 \ \wedge \ C^T A^2 x_m = \ \ C^T A B\}$$

2. at ζ_1 if and only if $\exists k > 2 \in \mathbb{N}$ such that $\left(\bigwedge_{i=0}^{2} f_1^{\perp^{(i)}}(x(t_m)) = 0 \wedge f_1^{\perp^{(k)}}(x(t_m)) > 0\right)$ satisfied by the two constraints

$$(C^T x_m = 0) \ \wedge \ (C^T A x_m = 0) \ \wedge \ (C^T A^2 x_m = -C^T AB)$$

$$\exists k > 2 \in \mathbb{N} : C^T A^k x_m > -C^T A^{k-1} B$$

3. at ζ_2 if and only if $\exists k > 2 \in \mathbb{N}$ such that $\left(\bigwedge_{i=0}^{2} f_2^{\perp^{(i)}}(x(t_m)) = 0 \wedge f_2^{\perp^{(k)}}(x(t_m)) < 0\right)$ satisfied by the two constraints

$$(C^T x_m = 0) \ \wedge \ (C^T A x_m = 0) \ \wedge \ (C^T A^2 x_m = C^T AB)$$

$$\exists k > 2 \in \mathbb{N} : C^T A^k x_m < C^T A^{k-1} B$$

Case II. $C^T B = 0$: for which we have from (29) and (30): $f_1^{\perp^{(2)}}(x_m) = f_2^{\perp^{(2)}}(x_m) = C^T A^2 x_m$, the case in which we should consider the 3^{rd} conditions to determine whether we should stay on the sliding surface Γ_s or leave it.
Lemma: In the k^{th} order sliding modes analysis of control problems with relay feedback with $k > 0 \in \mathbb{N}$, the multi-valued sliding parameter $\delta(x)$ is given by

$$\delta(x) = \frac{C \cdot A^k \cdot x}{C \cdot A^{k-1} \cdot B} \tag{31}$$

We summarize in Table 1 the general case of higher order conditions analysis for the existence of an attractive sliding motion on $\Gamma_s \subset \Gamma$. The constraints on the data as well as on the dynamics are reported under the heading "Data"and "Dynamics", respectively. In Tables 2 and 3 we summarize the higher order conditions for the staying conditions at the tangential exit points ζ_1 and ζ_2 respectively.

Table 1 - Higher-order conditions for the existence of sliding on $\Gamma_s \subset \Gamma$

Order	Data	Dynamics	$\delta(x)$	Sliding on $\Gamma_s \subset \Gamma$
1	$CB \neq 0$	$Cx(t) = 0$	$CAx(t)/CB$	$\lvert\delta(x)\rvert < 1$
2	$CB = 0 \wedge$ $CAB \neq 0$	$Cx(t) = 0 \wedge$ $CAx(t) = 0$	$CA^2x(t)/CAB$	$\lvert\delta(x)\rvert < 1$
3	$CB = 0 \wedge$ $CAB = 0 \wedge$ $CA^2B \neq 0$	$Cx(t) = 0 \wedge$ $CAx(t) = 0 \wedge$ $CA^2x(t) = 0$	$CA^3x(t)/CA^2B$	$\lvert\delta(x)\rvert < 1$
...	...	...	...	...
k	$\bigwedge_{i=0}^{k-2} CA^iB =$ 0 $\wedge CA^{k-1}B \neq 0$	$\bigwedge_{i=0}^{k-1} CA^ix =$ 0	$CA^kx(t)/CA^{k-1}B$	$\lvert\delta(x)\rvert < 1$

Table 2 - Higher-order conditions for the existence of sliding on $\zeta_1 \in \Gamma$

Order	Data	Dynamics	$\delta(x)$	Sliding on $\zeta_1 \in \Gamma$
1	$CB \neq 0$	$Cx(t) = 0$	$CAx(t)/CB$	$\delta(x) = -1 \wedge \exists k > 1 : \frac{CA^k x(t) >}{-CA^{k-1}B}$
2	$CB = 0 \wedge$ $CAB \neq 0$	$Cx(t) = 0 \wedge$ $CAx(t) = 0$	$CA^2x(t)/CAB$	$\delta(x) = -1 \wedge \exists k > 1 : \frac{CA^k x(t) >}{-CA^{k-1}B}$
3	$CB = 0 \wedge$ $CAB = 0 \wedge$ $CA^2B \neq 0$	$Cx(t) = 0 \wedge$ $CAx(t) = 0 \wedge$ $CA^2x(t) = 0$	$CA^3x(t)/CA^2B$	$\delta(x) = -1 \wedge \exists k > 1 : \frac{CA^k x(t) >}{-CA^{k-1}B}$
...	...	...	...	...
k	$\bigwedge_{i=0}^{k-2} CA^iB = 0$ $\wedge CA^{k-1}B \neq 0$	$\bigwedge_{i=0}^{k-1} CA^ix = 0$	$CA^kx(t)/CA^{k-1}B$	$\delta(x) = -1 \wedge \exists k > 1 : \frac{CA^k x(t) >}{-CA^{k-1}B}$

Table 3 - Higher-order conditions for the existence of sliding on $\zeta_2 \in \Gamma$

Order	Data	Dynamics	$\delta(x)$	Sliding on $\zeta_2 \in \Gamma$
1	$CB \neq 0$	$Cx(t) = 0$	$CAx(t)/CB$	$\delta(x) = 1 \wedge \exists k > 1 : \frac{CA^k x(t) <}{CA^{k-1}B}$
2	$CB = 0 \wedge$ $CAB \neq 0$	$Cx(t) = 0 \wedge$ $CAx(t) = 0$	$CA^2x(t)/CAB$	$\delta(x) = 1 \wedge \exists k > 1 : \frac{CA^k x(t) <}{CA^{k-1}B}$
3	$CB = 0 \wedge$ $CAB = 0 \wedge$ $CA^2B \neq 0$	$Cx(t) = 0 \wedge$ $CAx(t) = 0 \wedge$ $CA^2x(t) = 0$	$CA^3x(t)/CA^2B$	$\delta(x) = 1 \wedge \exists k > 1 : \frac{CA^k x(t) <}{CA^{k-1}B}$
...	...	...	...	...
k	$\bigwedge_{i=0}^{k-2} CA^iB = 0$ $\wedge CA^{k-1}B \neq 0$	$\bigwedge_{i=0}^{k-1} CA^ix = 0$	$CA^kx(t)/CA^{k-1}B$	$\delta(x) = 1 \wedge \exists k > 1 : \frac{CA^k x(t) <}{CA^{k-1}B}$

5 Simulation Results

In this section we carry out a series of simulation tests to illustrate the performance as well as the efficiency of the approaches developed and presented in this paper.

Example 1. Consider the following system with relay feedback

$$\dot{x} = Ax + Bu; \qquad y = Cx; \qquad u = -sgn(y) \tag{32}$$

$$A = \begin{bmatrix} -3 & 1 & 0 \\ -3 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}; \qquad B = \begin{bmatrix} 1 \\ -2\beta \\ \beta^2 \end{bmatrix}; \tag{33}$$

$$C = [1 \;\; 0 \;\; 0]; \;\; x = (x_1, ..., x_n)^T \in \mathbb{R}^n \tag{34}$$

Depending on the value of CB, a classification of the directions of the trajectories divide the switch plane into two or three regions. In this example we have $CB > 0$, therefore, there exist 1^{st} order sliding modes. Figure 1 shows a $2D$ plot of the simulation of this system with $\beta = 0.5$ and $x_0 = [0.5\ 3\ 0.1]^T$.

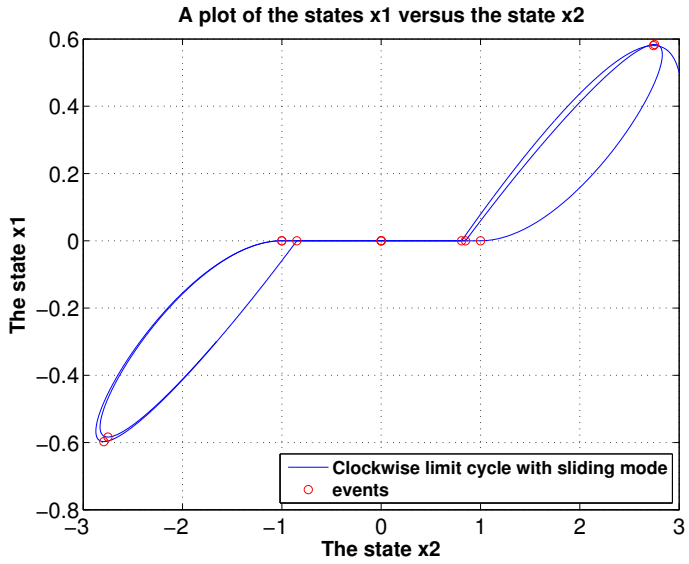


Figure 1: A clockwise limit cycle of the system with $\beta = 0.5$, and $x_0 = [0.5\ 3\ 0.1]^T$.

As it is demonstrated in Figure 1, the exit from the sliding surface is tangential at the exit points ζ_1 and ζ_2. For a simulation time of 20 seconds: (i) six relay switches have been recorded with three sliding segments, (ii) 4 detections of tangential crossing outside the hyper switching plane Γ have been recorded. Such detection of the tangential crossings gives precise information whether the gradient of the continuous time behavior of the systems trajectory is directed or not towards the switching plane. A $3D$ plot of the simulation of this system with the same initial conditions is illustrated in Figure 2.

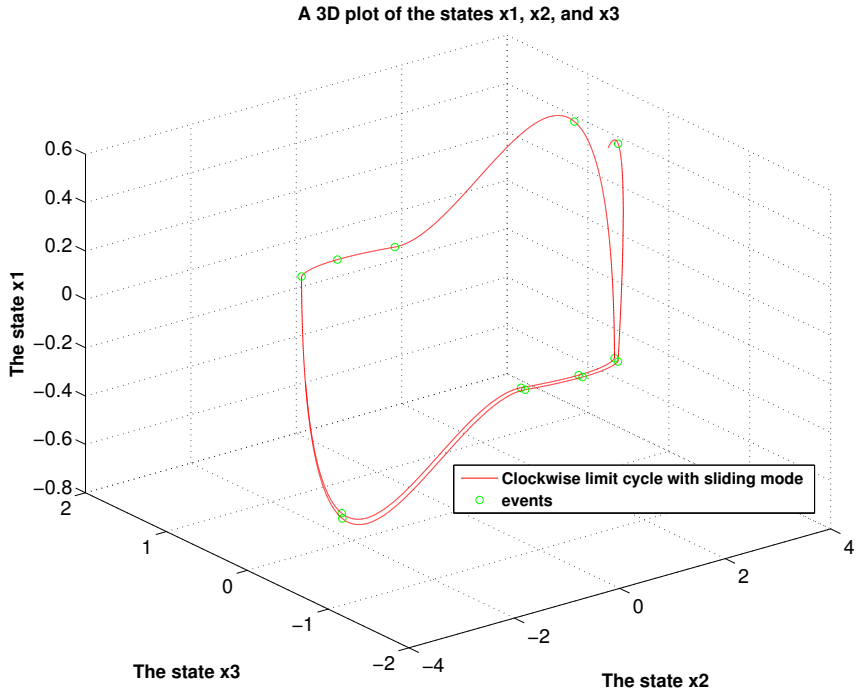


Figure 2: A $3D$ orbit of x_1 versus x_2 and x_3 with $\beta = 0.5$, and $x_0 = [0.5\ 3\ 0.1]^T$.

Example 2. Consider the following system with relay feedback

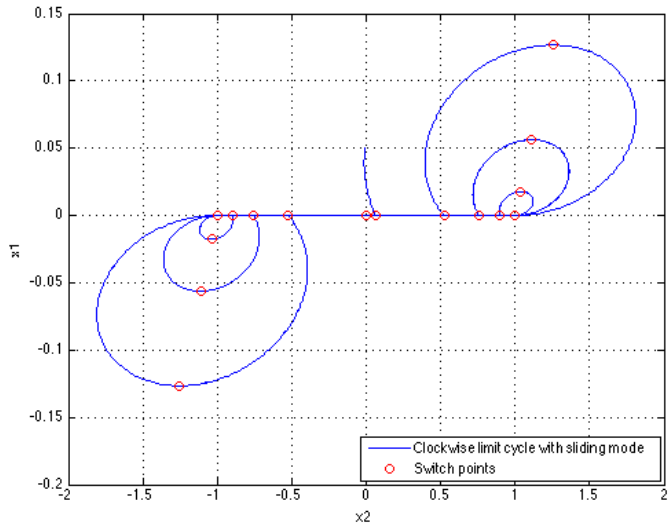
$$\dot{x} = Ax + Bu; \qquad y = Cx; \qquad u = -sgn(y) \tag{35}$$

where

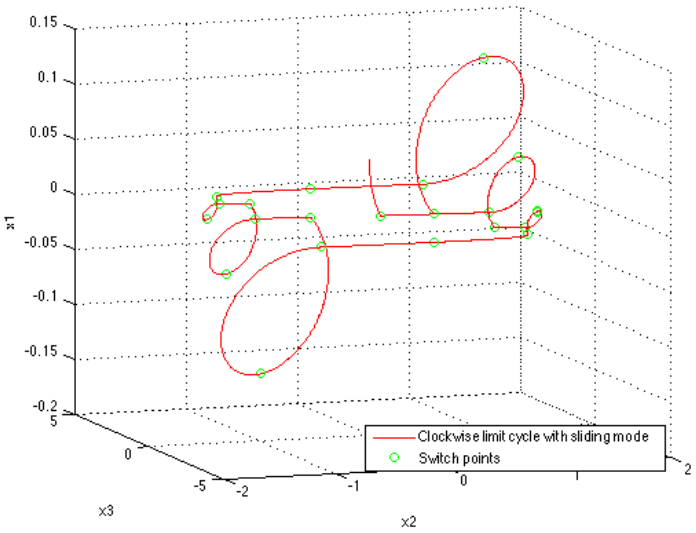
$$A = \begin{bmatrix} -(2ab+1) & 0 & 1 \\ -(2ab+b^2) & 0 & 1 \\ -b^2 & 0 & 0 \end{bmatrix}; \qquad B = \begin{bmatrix} d \\ -2s \\ 1 \end{bmatrix}; \tag{36}$$

$$C = [1\ \ 0\ \ 0]; \ \ x = (x_1, ..., x_n)^T \in \mathbb{R}^n \tag{37}$$

With this data set of B, a first-order attractive sliding is expected on the switch plane $Cx = 0$ as long as d is positive. Figure 3 shows the clockwise trajectories of the system with the parameters $a = 0.05$, $b = 10$, $d = 1$, $s = -2$ and initial conditions $x_0 = [0.05\ -0.01\ 0.1]^T$. In a simulation time of 20 seconds, 68 events were detected including 20 tangential crossings outside the switching plane.



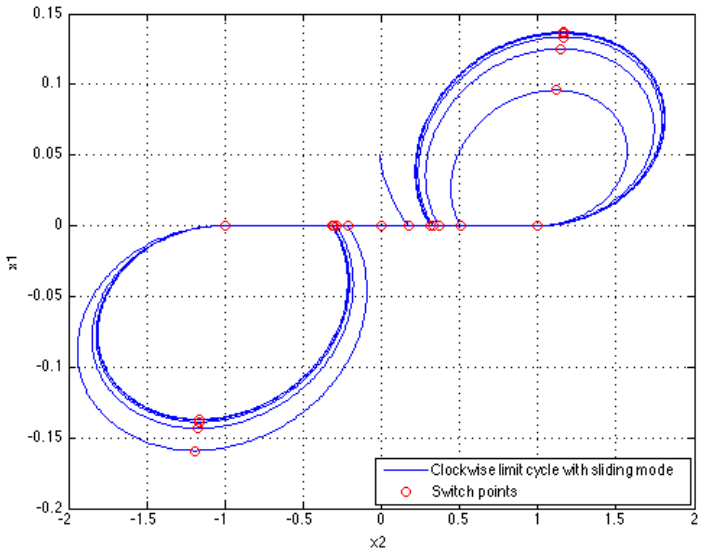
(a)



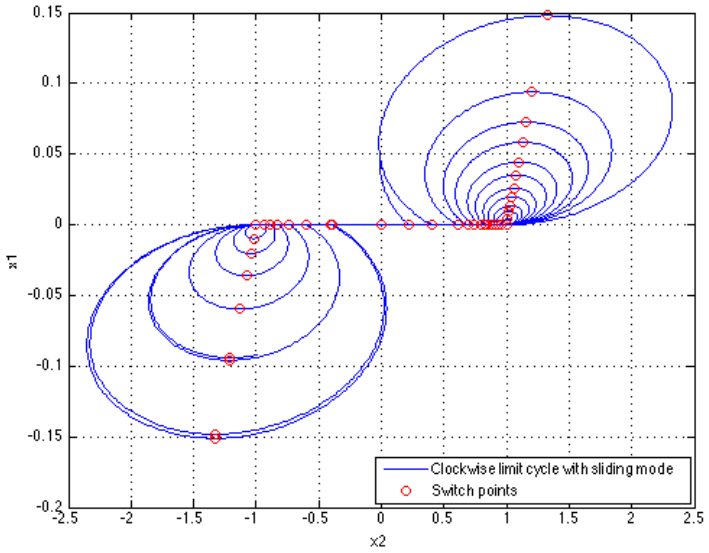
(b)

Figure 3: Simulation of the system (35) for $a = 0.05$, $b = 10$, $d = 1$, $s = -2$, and $x_0 = [0.05\ -0.01\ 0.1]^T$: (a). The evolution of x_2 versus x_1. (b). A $3D$ view of the Periodic orbit of x_1 versus x_2 and x_3.

Two other dynamical scenarios were simulated for this system. It has been observed that increasing b and decreasing s reduces the number of the sliding segments (Figure 4 (a)). Large enough b and small enough s will result in a transversality switching (Figure 4 (b)).



(a)



(b)

Figure 4: Clockwise trajectories of the system (35): (a). The evolution of x_2 versus x_1 for $a = 0.01$, $b = 10$, $d = 1$, $s = -3$, and $x_0 = [0.05 \ -0,01 \ 0.1]^T$. (b). The evolution of x_2 versus x_1 for $a = 0.01$, $b = 10$, $d = 1$, $s = -3$, and $x_0 = [0.05 \ -0,01 \ 0.1]^T$.

Example 3. Consider again the system in Example 2. Setting $d = 0$ yields in $CB = 0$, and therefore, applying the higher order analysis implies that multiple fast switches exist only if $CAB > 0$. In Figure 5, with the set of parameters $a = 0.03$, $b = 5$, $d = 0$, $s = -10$, we have $CAB = -2s = 20 > 0$. Note that, the sliding motion takes place on the order switching plane $\Gamma^{(2)}$ which is given by

$$\Gamma^{(2)} = \{x \in \mathbb{R}^n : \ \gamma^{(2)}(x) = -1.3x_1 + x_2 = 0\} \tag{38}$$

On the plane x_1, a transversality switching is observed. The trajectory converges to a limit cycle. It was recorded that the bigger is the parameters a and b the faster is the convergence to the limit cycle (see Figure 6).

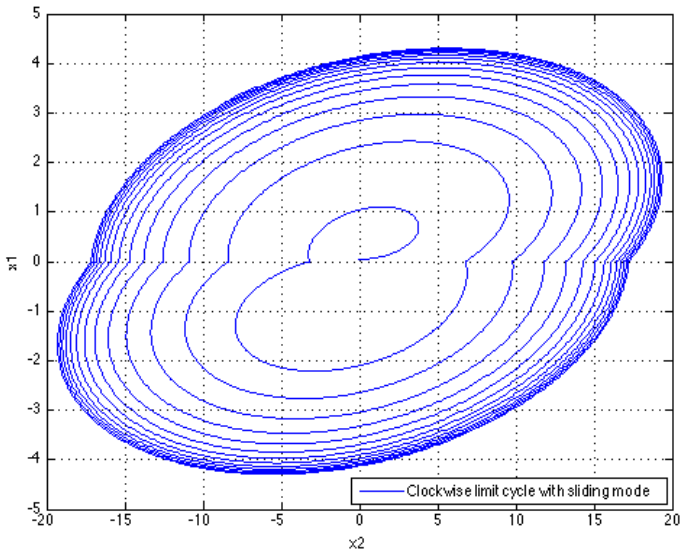


Figure 5: Clockwise trajectories of the system (35) with a 2^{nd} order sliding mode simulation for $a = 0.03$, $b = 5$, $d = 0$, $s = -10$, and $x_0 = [0.05 \ -0,01 \ 0.1]^T$: The evolution of x_2 versus x_1.

6 Conclusions

In this paper we presented a new computational framewrork for the purpose of a robust detection of the chattering behavior "on the fly" in real-time simulation of hybrid systems as well as the treatment of chattering behavior during the numerical simulation using the higher order sliding mode simulation. The main objective of the proposed regularization technique is to switch between the transversality modes and the sliding modes simulation automatically as well as integrating each particular state appropriately and localize the structural changes in the system in an accurate way. The method presented in this paper

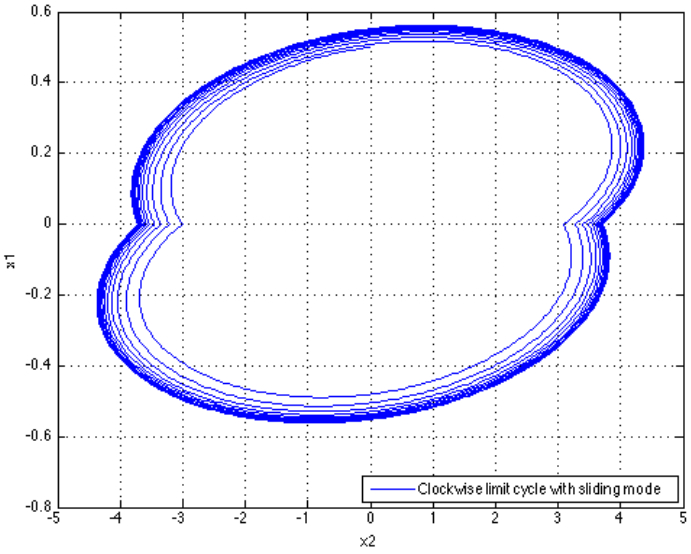


Figure 6: Clockwise trajectories of the system (35) with a 2^{nd} order sliding mode simulation for $a = 0.07$, $b = 8$, $d = 0$, $s = -10$, and $x_0 = [0.05\ -0,01\ 0.1]^T$: The evolution of x_2 versus x_1.

makes use of the higher order sliding modes analysis in the appliactions when the 1^{st} order analysis cannot be applied. Finally, the simulation results - reported here on a set of representative examples - showed that our approach is efficient and precise enough to provide a chattering bath avoidance, to perform a special numerical treatment of the constrained motion along the discontinuity surface, as well as its robustness in achieving an accurate detection and localization of all the switch points.

Acknowledgements

This work was supported by the ITEA2/MODRIO project under contract N° 6892, and the ARED grant of the Brittany Regional Council.

References

[AC15] Ayman ALJARBOUH and Benoit CAILLAUD. Robust Simulation for Hybrid Systems: Chattering Bath Avoidance. *LiU Electronic Press*, October 2015.

[BHJ13] Martin Biák, Tomáš Hanus, and Drahoslava Janovská. Some applications of Filippov's dynamical systems. *Journal of Computational and Applied Mathematics*, 2013.

[CGST07] Chaohong Cai, Rafal Goebel, Ricardo Sanfelice, and Andrew Teel. Hybrid systems: limit sets and zero dynamics with a view toward output regulation. *Springer-Verilag*, 2007.

[dBBC+08] Mario di Bernardo, Chris Budd, Alan Champneys, Piotr Kowalczyk, Arne Nordmark, Gerard Olivar Tost, and Petri Piiroinen. Bifurcations in Nonsmooth Dynamical Systems. *Industrial and Applied Mathematics*, 2008.

[Fil88] Aleksei Fedorovich Filippov. Differential Equations with Discontinuous Right-Hand Sides. *Mathematics and its Applications*, Kluwer Academic, 1988.

[GST11] Marcel Guardia, Tere Seara, and Marco Antonio Teixeira. Generic bifurcations of low codimension of planar Filippov Systems. *Journal of Differential Equations*, 2011.

[JBÅ02] Karl Henrik Johansson, Andrey Barabanov, and Karl Johan Åström. Limit Cycles With Chattering in Relay Feedback Systems. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 47(9):1073–1096, September 2002.

[LA09] Hai Lin and Panos J. Antsaklis. Hybrid Dynamical Systems: Stability and Stabilization. *The Control Systems Handbook, Second Edition: Control System Advanced Methods. CRC Press*, 2009.

[Utk92] Vadim Utkin. Sliding Mode in Control and Optimization. *Springer, Berlin*, 1992.

[WKH14] Daniel Weiss, Tassilo Küpper, and Hany Hosham. Invariant manifolds for nonsmooth systems with sliding mode. *Mathematics and Computers in Simulation*, 2014.

[ZJLS01] Jun Zhang, Karl Henrik Johansson, John Lygeros, and Shankar Sastry. Zeno hybrid systems. *International Journal of Robust and Nonlinear Control*, 2001.

[ZYM08] Fu Zhang, Murali Yeddanapudi, and Pieter Mosterman. Zero-Crossing Location and Detection Algorithms For Hybrid System Simulation. *IFAC World Congress*, 2008.

Simulation of Incident Responses for O&G Cyber Security

Petro Bondarenko
Gjøvik University College
email: bondarenko-peter@ukr.net

Abstract: Oil and gas drilling process is treated as an entire industrial complex vulnerable to cyber and human related threats due to the automation of the main industrial operations on one hand, and the complexity of the well's stability maintenance on the other hand. The few basic rules of offshore drilling, which are not to allow a blowout, to keep the well properly protected, maintain staff security, and to safeguard the automated systems of the drilling rig from different threats comprise the main task of our paper. The main problem on which paper was focused on is how to secure the automated production stages from such threats as industrial espionage targeted and distributed sabotage, other APT's. An automated system in a drilling rig could be a target of such attacks. A lot is known about the steps and mistakes which lead to an incident, but the simulation model development is a real challenge, as it gives the possibility to test the security measures efficiency, measures to mitigate the consequences of an incident and develop the recovery plan activity. The main result and contribution of the research is the development of incident responses and prevention strategies for the offshore drilling industry.

1 Introduction

The article under review is dedicated to the important issue of the offshore drilling automated systems security provision, that is, how to secure drilling process at all the stages of oil and gas (O&G) production. Offshore drilling is a highly competitive industry, in which safety and security concerns are paramount. An incident on a drillingrig can not only compromise the rig, the well or the reservoir; it could lead to environmental problems, significant financial loses and, in a worst case scenerio, casualties.

Recently reported cyber attacks - such as Shamoon [htt12] and Stuxnet [Zet11], show that in cases where safety regulations are neglected, consequences are severe.

The article offers a model to simulate cyber security incident responses on drilling rigs. The simulation is a high degree approximation to the real case in order to get the valid means to response the incident. The model foresees the main decision-making steps and responses to attacks. The article is based on the Master's paper [Bon14], which investigates the potential consequences of a cyber attack in a drilling rig employing automation and control systems.

The paper is organized as follows. It starts with Section 2 which deals with the potential attack's overview in order to define the most dangerous. Section 3 contains the main results of the case study scenario that serves as the basis for the model development. Section

4 represents the model of incident responses itself and highlights the decisions making process in charge of cyber security in the drilling rig in response to a cyber attack; the subsections provide the reader with all the model's nodes and strategies/responses to the attack. Conclusions are given in Section 5 and contain the main contribution of the research conducted.

2 Potential attacks

2.1 General attack's overview

As it has already been mentioned, the main goal on the research was to develop the simulation model the drilling rig personnel may use to check the state of the cyber security. That is why the recent cases reported were analyzed in order to define the most vulnerable drilling equipment elements, on the one hand, and the most widespread reasons for compromising the information system of a probable target. The case study reviled that the following reasons turned out to be the most widespread: industrial espionage (cyber espionage as well) and sabotage of different types.

Sabotage is understood as a hostile activity aimed at blocking or stopping the competitor's business. The rise of the Internet and the spread of social networks have opened new ways of industrial and targeted espionage. According to the specialists' estimations, 50000 enterprises a day are subjected to a cyber attack, and this rate is doubled every year. The cyber espionage is generally treated as state-supported, as the attackers are able to get access to personal and commercial data. The analysis of cyber attacks should be based on deep and thorough knowledge of the networks algorithms and principles of their functioning. [CS13] [Mah13]

It goes without saying that cyber threats to the O&G production industry are becoming more and more urgent nowadays. The attacks on the companies are represented by a lot of forms, from a single breach in order to get the necessary or sensitive information to the attempts to disrupt the companies' physical activity. It is worth mentioning that these threats are gradually becoming more sophisticated and experienced: they have evolved from lone hackers trying to breach the system and steal information into state-sponsored and corporation-supported teams of professionals with almost unlimited resources and time. Nowadays it is no longer a question of a hackers' attack on the drilling rig or company in a single state, it is the issue of global threat. [CS13]

2.2 Industrial espionage

Industrial espionage is obtaining confidential information illegally and unethically. It is an attempt to gain access to information about company's plans, products, clients or targeted secrets. In most cases, accessing trade secrets is illegal. In case it is the sensitive information which is the target, access to private information and company's records is obtained

with the help of an insider. But if such an insider is bribed, black-mailed or threatened, it may be concerned to be the case of espionage. Hacking into the computer system of the targeted company for getting private information is another widespread industrial espionage practice. [Kab05]

The information obtained may be sold to other companies or used for black-mailing. It may be used in order to affect stock prices. For example, such a scheme was used with Shamoon [Kab05] in order to affect the Saudi Aramco shares and the production activity. In some cases, the presence of a spy may be traced by the industrial espionage software, programs which give the entire information about the targeted company computer users. Such programs allow one to hack into private files of another computer user or get the keystrokes or even the system code. [Kab05]

The main problem with the industrial espionage is that the companies do not want to admit the fact that their sensitive information was stolen. Also, in case of a trial the stolen information is usually made public during the hearing in court. Here is the dilemma as any sensitive information is valuable until it is kept in secret that is, its confidentiality is its price. Any disclosure during a trial lessens its value as the production secrets and know-hows become public. It is especially important for the offshore drilling as any know-how may reduce the production costs. Therefore this information as well as the incident response plans' details are highly valued by competitors. [Rus13]

2.3 Targeted sabotage

In reference to offshore drilling industry, targeted sabotage can be defined as a complex of actions aimed at preventing the production activity of an oil producing company by means of interfering into its automated computer systems with the aim of stopping or preventing oil production.

As an example of targeted sabotage, it is worth mentioning Stuxnet and Night Dragon. Night Dragon was detected by McAfee [Bon13]. The investigation showed that Night Dragon was coordinated, covert and targeted campaign by hackers from China who tried to get sensitive data from five main Western energy companies within 2008-2011. The approximate damage cannot be estimated, but some companies have faced losses at auctions.

As for Stuxnet [Byr12], there were some speculations that it was developed to attack industrial Programmable Logic Controllers (PLCs) by the Israeli and US agencies in order to block the nuclear programs of Iran. This virus was predominantly distributed through Microsoft Windows products aiming at Siemens industrial control systems. Stuxnet compromised Iranian PLCs, collecting information on industrial systems and causing the fast-spinning centrifuges to tear themselves apart. The main peculiarity of Stuxnet was that for the first time it contained a programmable logic controller rootkit. The worm that was initially spread was almost undetected, but it hit only Siemens SCADA system which allowed a reprogramming of the infected devises. Now it is clear that the target was the Iranian Uranium Enrichment, and five Iranian organizations were reached by means of different variants of Stuxnet. [Byr12]

2.4 Distributed sabotage

Distributed sabotage of offshore drilling industry is a complex of actions directed against a particular company or state using any available resources and technologies.

The Shamoon and Stuxnet attacks [htt12] [Zet11] are the bright examples of how cyber attacks may go beyond their original tasks and become a real threat to the oil and gas production industry. Soon after Stuxnet attacked the target in Iran, it also infected the Chevron's computer systems but without any significant damage to them. Two weeks later after the attack on Saudi Aramco, Shamoon attacked the Qatari natural-gas firm disabling its website and e-mail servers. It is the evidence of the fact that O&G producing companies have become targets for the cyber attacks which are aimed at stopping their functioning. On the other hand, such attacks are likely to be rare. These target public relations facilities (companies' websites) or business activity (by altering e-mail servers). It is explained by the fact that it is difficult to upset the critical physical infrastructure which is naturally harder to penetrate. But it doesn't mean that SCADA systems are not vulnerable. [BT13] [Byr12]

The fact that the originally single centralized supervisory systems have evolved into a decentralized interconnected networks means that they are easier to be compromised. In case of a severe attack the cyber criminals may compromise the SCADA system causing pumping stations to shut down, or even worse, to cause the destruction or malfunction of the equipment. Any serious interference of the kind may lead to a severe environmental damage, commercial losses or other consequences. Speaking of the possible ways to secure computer systems, one could recommend establishing reserved servers to keep backup data there in case of a massive cyber attack or damaging of hard drives. [BT13] [Byr12]

3 Case study

The case study is served as a real case example for the model of incident response development and is based on the recently reported incidents. It provided us with the possibility of defining the typical components of the attack scenario. (Table 1)

Scenario of our research problem is a targeted sabotage cyber attack. The attacker uses a USB stick containing the malicious code that can influence the PLC that controls one of the most critical and vulnerable pieces of drilling equipment, the mud pump. When the mud pump PLC meets some criteria the new code - updated with malicious code - will command the actuators in such a way that the mud pump can cause an abnormal and dangerous level of pressure. Low pressure can cause kicks or blowouts, so it is very dangerous and risky. High pressure can cause drilling mud to enter the reservoir, blocking the circulation of hydrocarbons in the parts of the reservoir near to the well, or injecting drilling mud or fluid that can cause a quick or even widespread deterioration of the reservoir.

Attack options	Description
1.Attack scenario	Targeted Sabotage
2.Attacker	Terrorist in collaboration with insider
3.Tool	Insider locally on the site with usb stick with tailored attack tools and PLC update package (hacker needs to update the code in PLC to do this attack)
4.Vulnerability	Lack of scanning of USB, lack of proper procedures for ”human” security, lack of logical monitoring for the events happening in the system
5.Action	Modify (remove and insert)
6.Target	Controller(PLC) of mud system
7.Unauthorized Result	Be able to control the behaviour of mud system which is critical to the drilling process
8.Objective	Monetary losses, casualties, environmental disaster and publicity to the terrorist organization

Table 1: **Case study scenario** [Bon14]

4 Incident response model

This Section proposes a model to simulate and assess incident handling decision-making through an Influence Diagram model [GP08] (Figure 1). The model is exemplified through a generalized example case, based on the targeted sabotage scenario, for understanding both the model approach and some key challenges in modelling drilling cyber security approaches.

The model represents the attack through a series of sequential actions of the attacker, actions with an uncertain result. Each of the nodes represents the likelihood of the attacker to carry out a successful action after the previous - and required - actions have been successful. Since the arcs represent conditional probabilities (e.g., if the likelihood of action 1 is 50% and the likelihood of action 2 is 40%, then the likelihood of a successful action 2 is 20% before knowledge of the result of action 1 and 40% after a successful action 1).

The example case has the following attack actions:

- Action 1. The insider physically inserts a USB Stick with the attack tool in a USB port connected to the rig network.
- Action 2. The attack tool gains logical access to the rig network.
- Action 3. The attack tool spreads over the rig network and over the ICS network until it finds the I/O server in charge of the mud pump PLC.
- Action 4. The attack tool gains access to the I/O server in charge of mud pump PLC.
- Action 5. The attack tool updates the mud pump PLC code with malicious code.
- Action 6. When certain criteria are met, the malicious code makes the PLC manipulate, through the actuators of the PLC, the Mud Pump to increase or decrease the

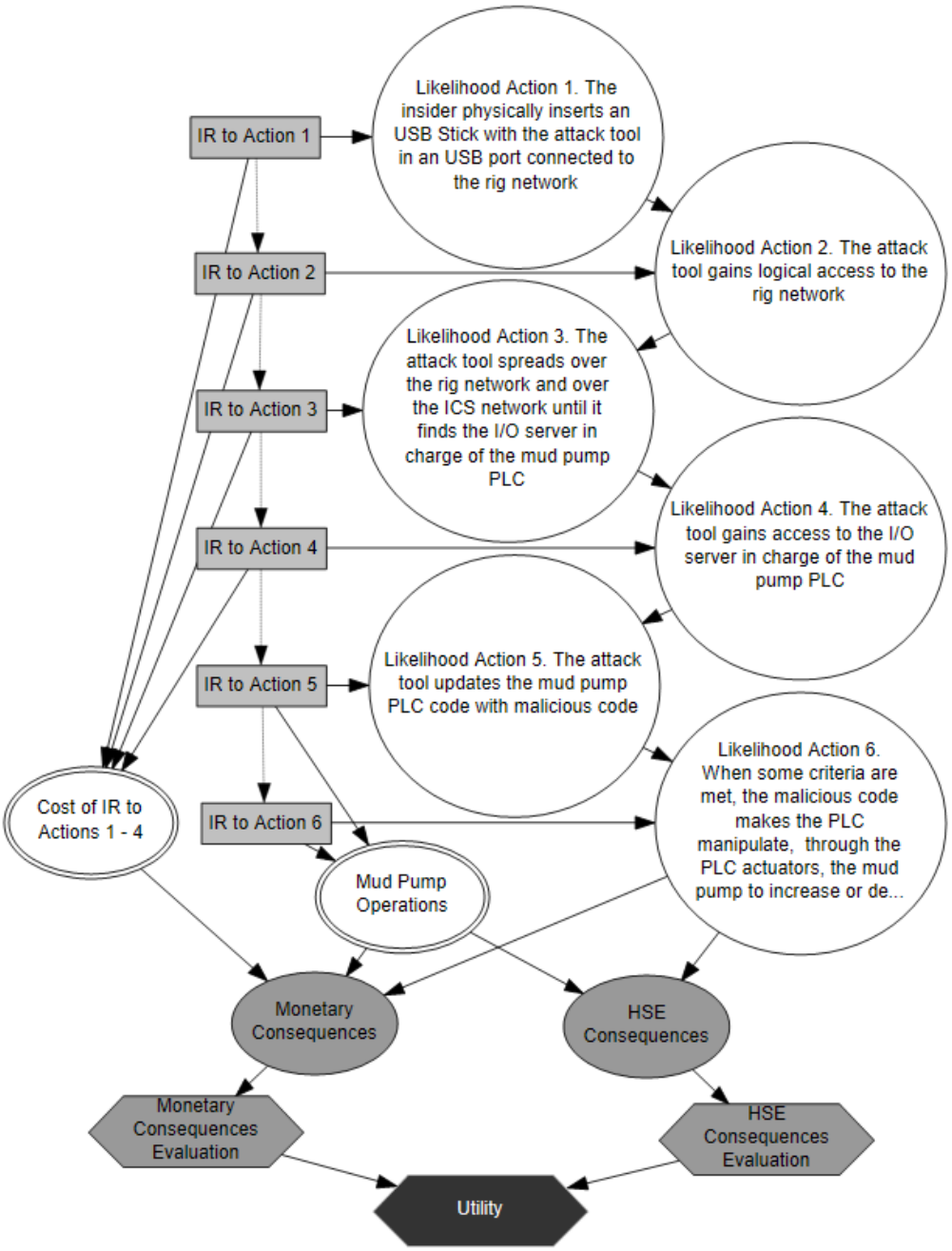


Figure 1: **Model of incident responses** [Bon14]

pressure of the drilling mud to an abnormal level that can cause a kick, a blowout, or other incidents.

4.1 Incident response nodes

In this example model, each of the attacker actions has the option to be handled, i.e., reduce the likelihood of such actions. For simplicity, we assume that the incident response to the action is simultaneous to the action. Specific incident response strategies were discussed that can be applied to the different actions of the attacker, as well as the challenges or cyber security decisions that can be taken in regard to the mud pump.

The responses for the first four actions are cyber security measures within the rig or ICS network to avoid the attacker achieving further stages of his attack. Each of these four nodes has two generic options, one facing the action of the attacker to reduce the likelihood and the option of inaction:

- Response to Action 1: Strengthen the physical security policy or not.
- Response to Action 2: Strengthen device authentication or not.
- Response to Action 3: Increase the difficulty of the spread of the attack tool within the network or not.
- Response to Action 4: Increase the difficulty of the access of the attack tool to the I/O server or not.

For simplicity, it is assumed that each of the response actions cost 75.000 $ due to a 6-th delay, or a cost equivalent to it, in drilling operations (a high-tech drilling rig costs 300.000 $/day).

On the other hand, the response to these last two actions has to be a response more coordinated with the core business function of the rig, drilling. After the attack tool attains the I/O server the eradication of the attack is more problematic.

The mud pump is a piece of equipment that should not be shut down at any time, since the drilling mud, and therefore the well, depends on it. This is modelled by the three options available in the last two decisions nodes:

- Immediately stop the mud pump computerized operation to reconfigure the I/O server (Response to Action 5) or to update the PLC code (Response to Action 6). However, this action also poses a serious risk, since the mud pump should be working to keep the pressure level of the well at a safe and efficient level.
- Stop, during the maintenance of the mud pump, its computerized operation to reconfigure the I/O server or to update the PLC code. Responding at such a moment reduces the risks related to the work of the mud pump, but also increases the risks related to cyber attacks since the mud pump PLC will spend more days hosting the malicious code.

- Continue drilling without solving the problems on the I/O server or PLC and assume such risks until the end of the drilling operations.

This model includes two deterministic nodes employed to simplify the operation (avoiding explosion of variables) of the simulation, but with no additional meaning in the model. The cost schema of the first four responses is represented in the Cost of IR to Actions 1-4 of the deterministic node. The three operational situations of the last two incident responses are represented in the Mud pump operations deterministic node.

4.2 Consequences nodes

Two kinds of consequences are highlighted – the Monetary Consequences node represents all the consequences that could be monetized whereas the HSE Consequences node represents, specially, the loss of human life. Each of the consequence nodes has three final states, high impact level (HIL), medium impact level (MIL) and low impact level (LIL).

These different impact levels have different likelihoods based on the state of the mud pump operations (Mud Pump Operations node) and the likelihood of success of the final action of the attacker modelled in the uncertainty node of action 6. Additionally, the Monetary Consequences node is also affected by the cost of the incident response actions.

In such a way, the evaluation of the incident handling decisions incorporates a typical risk analysis approach within the ID model logic. The last of the uncertainty nodes, representing the final action of the attacker, is equivalent to the likelihood of a threat in a typical risk analysis. On the other hand the uncertainty nodes representing the consequences of the attack (grey ovals) denote the impact level of a threat in a typical risk analysis.

4.3 Evaluation nodes

The utility nodes evaluating consequences order the different impact levels based on the risk-attitudes of those responsible for making decisions. Assigning the value 1 to the most preferred state and 0 to the least preferred, and ordering the rest of the scenarios, assigning values based on a method to capture utility. The expected utility function is a function that assigns a value to the potential final outcomes of the decisions and uncertain states. These values measure the preferences and attitudes toward risk of the decision-makers. This function is useful to solve the problem through the maximization of these expected utility functions. [Aut10] [Lev06]

The outcome of these nodes is the expected utility that the stakeholder will get based on the likelihood of the different potential states of the consequences of uncertainty nodes and the utility that each state provides.

Finally, the black Utility node merges the two evaluations into a final one using criteria, based on the Economics theory of utility, to order the distinct preferences. The outcome of the node is the expected utility that the stakeholder will get due to the decisions they

made and the likelihood of the different uncertain states. [Lev06] The important thing is not the value of the utility but being able to order the different sets of decisions from the most preferred to the least preferred based on this expected utility. The set of decisions that provides the maximum expected utility is the optimal solution to the model.

4.4 Incident Handling Strategies applicable to the Scenario

NIST Cyber security Framework and NIST SP 800-53 [NIS13] Recommended Security Controls provide a way to identify security measures, including incident handling, for different security problems. We employed the Cybersecurity Framework to identify security measures that could be useful in our scenario.

The Cyber security Framework links these measures to different standards, including SP 800-53 [NIS13], which provides specific guidelines in security categorized in different families (such as Incident response - IR -, risk assessment -RA-, etc.). Each family also categorizes the different controls (such as incident handling -IR4-, incident monitoring -IR5-, incident reporting -IR6-, etc.). Based on the reading of these security controls we provide general strategies that might be applicable in a situation similar to our scenario.

4.4.1 Preparation for Incident Handling

This activity pivots on having a contingency plan for information systems, which is part of an overall organizational program aimed at achieving continuity of operations through the maintenance of their essential business functions and the eventual restoration of these system to a known safe state. It is important to identify essential business mission, functions, assets, roles, recovery objectives and restoration priorities.

- Develop, maintain and have available a baseline configuration of the information system (CM-2)
- Have available a minimum functionality configuration of the information system to identify the essential capabilities and their restricted use that supports essential organizational operations (CM-7).
- Have available alternate processing, storage and communication resources and sites to provide capabilities to support essential organization operations (CP).
- Have available incident handling capabilities through the establishment, development and maintenance of incident response policies, plans, training, testing and monitoring (IR).

4.4.2 Responses to Action 1

- Allow only authorized personnel with valid identification to enter locations where they are allowed (PE-2) [NIS13], and verify them before granting access (PE-3) [NIS13].
- Escort personnel without proper clearance or formal access authorization (PE-2) [NIS13].
- Remove individuals from access lists when their access is no longer required (PE-2) [NIS13].
- Check, reconfigure or implement physical security measures to enforce physical access control (PE-3) [NIS13]. For example: 24/7 monitoring, locking casings, testing penetration of the facility.
- Implement security checks to mitigate the risk of exfiltration (PE-3) [NIS13].
- Control physical access to the information system communication lines, test actions such as eavesdropping, disruption of cabling, and check the ability to prevent them (PE-4) [NIS13].
- Control physical access to the information system output devices to prevent unauthorized individuals to obtain such output, and ensure that only authorized and identified individuals obtain output from these devices (PE-5) [NIS13].

4.4.3 Responses to Action 2

- Allow only device authentication before the information system establishes a connection using techniques such as cryptographic authentication or device attestation (i.e. identification and authentication of a device based on its configuration and known operating state) (IA-3) [NIS13].
- Restrict or prohibit external traffic that appears to be spoofing internal addresses (SC-7) [NIS13].
- Limit the number of external network connections to ease monitoring activities (SC-7) [NIS13].

4.4.4 Responses to Action 3

- Use dynamic reconfiguration to increase the difficulty level for attacks or isolate attacks. For example this can be done, by changing router rules or access control lists (IR-4) [NIS13].
- Increase difficulty for attacks or ensure continuity of operations. For example, by dynamic reconfiguration of information flows or rules, graceful degradation, information system shutdown, or use of alternative operating modes of the information system, etc. (IR-4) [NIS13].

- Change information flow policy during certain conditions, such as changes in the operational situation, changes in the risk environment or tolerance, or the detection of potentially adverse events (AC-4) [NIS13].
- Compare security attributes of both the source and destination to find information flows not allowed by the organizations' policies, and block or quarantine them (AC-4) [NIS13].

4.4.5 Common Responses to Action 3 and 4

- Employ a white listing approach (deny-all, permit-by-exception policy) to allow only the execution of authorized software on the information system (CM-7) [NIS13] or to allow only authorized communication traffic (SC-7) [NIS13].
- Employ a blacklisting approach (allow-all, deny-by-exception policy) to prohibit the execution of unauthorized software on the information system (CM-7) [NIS13].
- Monitor, verify and change configuration settings of information system components that affect the security or functionality of the information system in accordance with policies and procedures (CM-6) [NIS13].
- Respond to unauthorized changes to configuration settings (such as baseline configuration) by restoring configuration settings to a safe configuration (e.g., baseline or least functionality) or halting information system processing or system functions (CM-6, CM-3) [NIS13].
- Have a dynamic contingency capability of deploying replacements or new resources in response to security incidents (IR-4) [NIS13].
- Monitor, create, enable, modify, disable, and remove information system accounts in accordance with the organization procedures and operational conditions(SI-7) [NIS13].
- Authorize access to the information system based on valid access authorization and intended use in accordance with conditions set by the organization (AC-2) [NIS13].
- Review abnormal traffic (e.g. extended traffic or access to sensitive areas) or audit actions in order to monitor activity in the system (e.g., who was online, who sent the outgoing messages, who modifies commands, or who download files) (AC-2) [NIS13].
- Foresee the need to terminate credentials or restrict the use of specific shared groups or accounts that exhibit atypical behavior considered a threat (AC-2) [NIS13].
- Analyze changes to the information system prior to their implementation to determine potential security risks, and verify those changes after their implementation, in order to know if the system will and does operate as intended (CM-4) [NIS13].

4.4.6 Common Responses to Action 3, 4 and 5

- Prevent the installation of software and firmware without verification that the component has been approved by the organization (CM-5) [NIS13].
- Implement dual authorization (two qualified individuals) to implement changes in selected components of the system (CM-5) [NIS13].
- Limit, review and reevaluate the privileges of users and libraries to change system components and system-related information (CM-5) [NIS13].
- Coordinate with internal and external stakeholders that could be affected by an incident or that can improve the handling of incidents (IR-4) [NIS13].

4.4.7 Response to Action 6

The solution is to eliminate the malicious code in the PLC by restoring the code to a safe known state.

5 Conclusions

The paper has proved that the offshore O&G industry main concern is the safety and security of drilling operations. As the aim was to develop the simulation incident model, the main aspects of the offshore oil and gas industry were taken into account, which showed that the issue of making this sector safer is becoming more and more urgent. Also, the research proves that the human factor is the first enabler of incidents, corroborating, that for the sake of safety, the future of drilling operations should embrace automation and control systems.

The case study showed that the most critical cyber attacks to the offshore drilling are the ones like Stuxnet and Shamoon as they provide the attacker with the possibility to altering, through system critical data, the behaviour of computerized physical equipment.

The model is based on the assumption that the successful cyber attack may alternate the mud pump PLC instructions to sabotage it, leading to a drop of the pressure level, which, in its turn, may result in the collapse of the well or even a blowout. The advantage of model offered is that it combines and takes into account the attack steps, combine them with the incident response actions, and add an evaluation of the likelihood of these events and potential consequences. The main contribution of the research conducted is the following:

- The model offered gives the possibility to check the readiness of the company to tackle the incident and mitigate its results;
- It presupposes the possibility of working out the variant and invariant incident response plan's components;

- It proves that simulation seems to be an efficient and usable tool for security attacks exploring and handling, it may be a great advantage for the oil and gas industry and a starting point for industrial case studies.
- The paper materials may be used for further investigations in the given field and may serve as a basis for more comprehensive simulation models aimed at the increase of the cyber security awareness;

References

[Aut10] David Autor. Lecture Note 14: Uncertainty, Expected Utility Theory and the Market for Risk. https://www.aaai.org/Papers/JAIR/Vol33/JAIR-3304.pdf, 2010. (Lecture in Massachusetts Institute of Technology 14.03/14.003, Microeconomic Theory and Public Policy).

[Bon13] Petro Bondarenko. Advanced Persistent Threat (APT) Beyond the hype. Project report in IMT4582 Network Security at Gjøvik University College during spring 2013, 2013.

[Bon14] Petro Bondarenko. Simulation of incident response cases for offshore drilling. Master's thesis report paper performed at Gjøvik University College during spring semester 2014, 2014.

[BT13] Christopher Bronk and Eneken Tikk. The Cyber Attack on Saudi Aramco. http://www.iiss.org/en/publications/survival/sections/2013-94b0/survival–global-politics-and-strategy-april-may-2013-b2cc/55-2-08-bronk-and-tikk-ringas-e272, 2013.

[Byr12] Eric Byres. Next Generation Cyber Attacks Target Oil and Gas Scada. http://pipelineandgasjournal.com/next-generation-cyber-attacks-target-oil-and-gas-scada?page=show, 2012.

[CS13] Blake Clayton and Adam Segal. ENERGY BRIEF: Addressing Cyber Threats to Oil and Gas Suppliers. http://www.isn.ethz.ch/Digital-Library/Publications/Detail/?ots591=0c54e3b3-1e9c-be1e-2c24-a6a8c7060233&lng=en&id=166056, 2013.

[GP08] Ya'akov Gal and Avi Pfeffer. Networks of Influence Diagrams: A Formalism for Representing Agents' Beliefs and Decision Making Processess. https://www.aaai.org/Papers/JAIR/Vol33/JAIR-3304.pdf, 2008.

[htt12] http://analysisintelligence.com/. Cyber Attacks in the Spin Cycle: Saudi Aramco and Shamoon. http://analysisintelligence.com/cyber-defense/narrative-of-a-cyber-attack-saudi-aramco-and-shamoon/, 2012.

[Kab05] M. E. Kabay. Industrial Espionage. http://www.mekabay.com/overviews/industrial_espionage.pdf, 2005.

[Lev06] Jonathan Levin. Choice under Uncertainty. http://www.stanford.edu/ jdlevin/Econ%20202/Uncertainty.pdf, 2006.

[Mah13] Shiraz Maher. Eusers Energy Talks. http://www.kcl.ac.uk/sspp/departments/warstudies/research/groups/eucers/energy-talks-report-2.pdf, 2013.

[NIS13] NIST. NIST Special Publication 800-53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf, 2013.

[Rus13] Gary Ruskin. Spooky Business: Corporate Espionage Against Nonprofit Organizations. http://www.corporatepolicy.org/spookybusiness.pdf, 2013.

[Zet11] Kim Zetter. How digital detectives deciphered Stuxnet, the most menacing malware in history. http://arstechnica.com/tech-policy/2011/07/how-digital-detectives-deciphered-stuxnet-the-most-menacing-malware-in-history/5/, 2011.

From 2-way to 1-way Alternating Büchi Automata

Alfredo Maceratesi
Università degli Studi di Roma "La Sapienza"
Universität zu Lübeck
alf.maceratesi@gmail.com

Abstract: This paper is about one step in the translation of linear temporal logic with past (pLTL) formulae into deterministic Moore machines. This step is translating 2-Way Alternating Büchi Automata (2ABA) into 1-Way Alternating Büchi Automata (ABA). Such translations are needed to generate monitors for runtime verification. Such monitors are used to verify the correctness of a program at runtime. The correctness property to verify is given in a temporal logic. To use this logic in runtime verification applications, we need to generate monitors from the specifications. Such a translation starts with generating a 2ABA from the given pLTL expression. The resulting automaton can then be simplified to an ABA expressing the same property. From this ABA one can generate a non deterministic deterministic Büchi Automata and, from these, deterministic monitor.

1 Introduction

One way to reduce risks related to software errors, is the application of *formal verification* methods. It can be done specifying the intended system behaviour in a formal description and verify whether this specification holds for every possible run of the system or not [MH84]. This process is called *model checking* [BBG+94]. Unfortunately, considering all potential executions of a system is difficult, especially in systems which have processes which don't terminate [Tho81]. Using *runtime verification*, we can satisfy this purpose, monitoring the system as it runs [LS11]. Step by step, we can validate whether its current state still satisfies the intended behaviour.
A well known method for defining a system's specification according to discrete time steps is the usage of temporal logics like the *linear temporal logic with past* (pLTL). With this we can describe with logical formulas which properties the system should satisfy and should have satisfied in a certain time during its execution [LPZ85] or using an extension of it Regular Temporal Logic with past [SL10] [LS10]. We therefore can apply pLTL formula for monitoring the system during runtime using the *finite state automaton* (FSA) derived from it [GO03] [Var98]. This can then process the current state of the system at every time step and decide if this state is accepted by the temporal logical formula or not [RS59].
In this paper we will provide a modification of the algorithm found by Piterman and Vardi in [PV03], in which they discussed the translation from 2-way Non-deterministic Büchi Automata to one way, in order to work for 2-way Alternating Büchi Automata.

2 2-Way Alternating Büchi Automaton

Translating pLTL formulas to automata we obtain 2-Way Very-Weak Alternating Automata [GO03]. They are a subset of 2ABA which have been introduced by Kupferman, Piterman and Vardi in [KPV01].

A **2-way Alternating Büchi Automaton** (2ABA) is a *five-tuple* $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ where:

- Q is the set of states,
- Σ is the alphabet,
- $\delta : Q \times \Sigma \to \mathcal{B}^+(Q \times \{-1, 0, 1\})$ is the transition function, where $\mathcal{B}^+(Q \times \{-1, 0, 1\})$ is a positive boolean formula with state and direction,
- $S = \mathcal{B}^+(I)$ where $I \subseteq 2^Q$,
- $F \subseteq Q$ is the set of final states.

Two-way Alternating Automata allow us to pose both existential and universal demands on the suffix and the prefix of the word. Technically, when the reading head of $\mathcal{A}$ is on the i-th position of a word w, it can move to locations $i-1$, i, and $i+1$.
For example, $\delta(q_0, a) = ((q_1, 1) \wedge (q_2, -1)) \vee (q_3, 0)$ means that when the automaton is in state q_0 reading the letter a in location i, it can either send a copy in state q_1 to location $i+1$ and a copy in state q_2 to location $i-1$, or stay in location i in the state q_3.

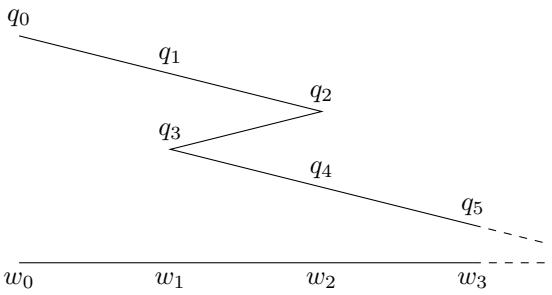
Definition 1. *A **run** of $\mathcal{A}$ on a word $w \in \Sigma^\omega$ is a $(Q \times \mathbb{N})$-labelled tree $\langle T, r \rangle$, where $r(\varepsilon) = (q_0, 0)$ and for all $x \in T$ with $r(x) = (r, k)$, the set $\{(q, \delta) | c \in \mathbb{N}, x \cdot c \in T,$ and $r(x \cdot c) = (q, k + \delta)\}$ satisfies the formula $\delta(r, w_k)$. For a path π, the set $inf(r|\pi)$ is defined as in alternating automata, thus $inf(r|\pi) = s|$ there are infinitely many nodes $x \in \pi$ with $r(x) \in s \times \mathbb{N}$.*

Definition 2. *A run of a 2-way alternating Büchi automaton is **accepting** if all infinite paths π in T have $inf(r|\pi) \cap F \neq \emptyset$.*

3 The translation algorithm

We will explain the algorithm found by Piterman and Vardi in [PV03] based on [She59], adopted for working on 2-Way Alternating Automata.

We will describe a method for the conversion of a 2ABA to an ABA accepting the same language. Since we are working on ω-words, we have to take count of the two possibilities which an accepting run of an ABA must either contain an infinite branch visiting some accepting state infinitely often or ends in a loop visiting at least one accepting state.



$$\pi = (q_0, w_0), (q_1, w_1), (q_2, w_2), (q_3, w_1), (q_4, w_2), (q_5, w_3), \ldots$$

(q_0, w_0) forward tuple $\quad \ldots \quad$ (q_2, w_2) forward tuple $\quad (q_3, w_1)$ backward tuple $\quad \ldots$

q_0, q_1, q_4, q_5 single states

(q_2, q_3) pair state

Figure 1: Example of a Zigzag Sequence

3.1 Anatomy of 2-way runs

We will now analyse the structure of a run a 2ABA $\mathcal{A} = (Q, \Sigma, \delta, S, F)$ without ε-moves. A run of $\mathcal{A}$ over an infinite word w could be seen as a positive boolean formula $\mathcal{B}^+(\pi)$ of sequences $\pi = (q_0, 0), (q_1, i_1), \ldots, (q_m, i_m)$ of states (q_j) and locations (i_j).

Let's start giving some definitions:

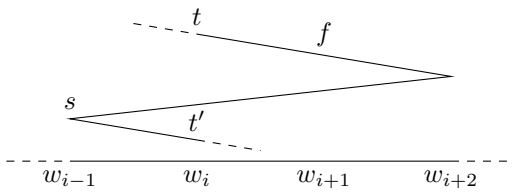
- *forward state* is a state (q_j, i_j) in π if $i_j = i_{j-1} + 1$;
- *backward state* is a state (q_j, i_j) in π if $i_j = i_{j-1} - 1$;
- $(q_0, 0)$ is a forward tuple.

We can then imagine the run π over the time and the states as the Piterman and Vardi's "zigzag diagram" (Figure 1).
Since the ABA we want to achieve is a 1-way automaton, it can read the word only moving forward. In case of 2ABA we could have two or more transitions reading the same letter w_j. The ABA will continue processing the word w from the state of the last occurrence of w_j in the zigzag and it will spawn a new process for each branch remained.
In the example in Figure 1 we have $\delta(q_1, w_1) = (q_2, 1)$ and $\delta(q_3, w_1) = (q_4, 1)$ which read the same letter w_1. The ABA has then to spawn two parallel processes, one has to check that q_1 results in q_3 and the second that the part of the sequence π after q_3 is accepting.

Basing on this concept, we can split the states of the ABA into two different types:

- A *singleton state* $q \in Q$ is a state which represents a part of the sequence that is only looking forward. For each index $i \in \mathbb{N}_0$ of w in a run of the ABA exists exactly one singleton state q.
- If a state $q \in Q$ is a backward state ($(q, -1) \in \delta(q', i)$) we associate it with the index i which was read in the previous state (q'). Therefore, a *pair state* $(t, s) \in Q \times Q$ is a pair of two states t and s for which t is a forward state and s is a backward state both associated to the same index i.

Since 2ABA work with infinite words, we need to assert promises of visiting the acceptance set. We can do it reinforcing the definition of singleton and pair states using the symbols $\top$ and $\bot$. In particular, the state $(t, s, \top)$ with $t, s \in Q$ means that the run segment leading from t to s has to visit at least one accepting state $f \in F$ (Figure 2). The singleton state $(t', \top)$ means that the zigzag segment connecting t' to the previous singleton state t has to visit some state $f \in F$ at least once (Figure 2). Every states with $\bot$ instead of $\top$ do not hold these promises of visiting an accepting state.



If $f \in F$ then $((t, s), \top)$

If $f \in F$ then $(t', \top)$

Figure 2: Explanation of $\top$ and $\bot$ states

We define the state set of the new ABA $\mathcal{A}'$ as $Q' = (Q \cup (Q \times Q)) \times \{\top, \bot\}$, which includes all combinations of singleton state and pair states paired with $\top$ and $\bot$ therefore, in the worst case, the blow-up among the number of states used in the ABA in comparison to the number of states in the 2ABA is quadratic. The positive boolean formula of initial states is $S' = \mathcal{B}^+(I \times \{\bot\})$.
We define the acceptance set of $\mathcal{A}'$ as $F' = (Q \times \{\top\}) \cup (F \times \{\bot\})$. The transition function η will be described in the following two segments. Afterwards, the ABA will be defined as $\mathcal{A}' = (Q', \Sigma, \eta, S', F') = (((Q \cup (Q \times Q)) \times \{\top, \bot\}), \Sigma, \eta, \mathcal{B}^+(I \times \{\bot\}), (Q \times \{\top\}) \cup (F \times \{\bot\}))$.
Using this definition, we can describe an ABA simulating the zigzag run shown in Figure 1 using the Figure 3.

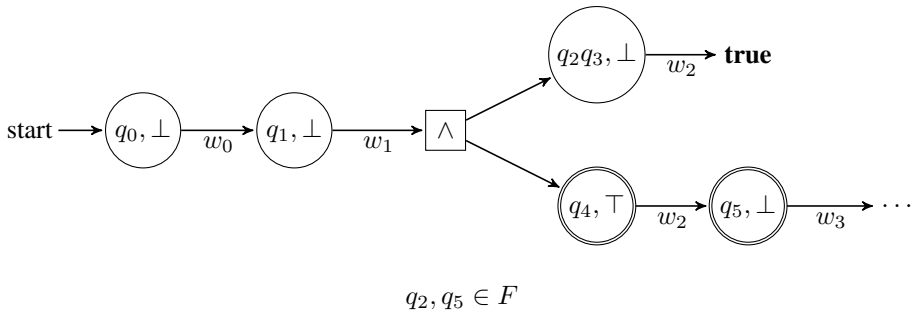


Figure 3: ABA simulating the zigzag in Figure 1

3.2 Transitions at singleton states

We can define the transition function η in two stages. Firstly we need to define it for singleton states. When the ABA is in a singleton state $t \in Q'$ reading the symbol w_i, it will guess all the states of the 2ABA $\mathcal{A}$ that will be reading w_i in the zigzag run. Since every word accepted by a 2ABA can be accepted by a simple run [KPV01], there will be k other states reading w_i, where $0 \leq k \leq n-1 : n = |Q|$. We name these states according to the order they appear during the zigzag run as $s_1, s_2, \ldots, s_k$. We then denote their successors by $t_1, t_2, \ldots, t_k$ and we call t_0 the successor of t.
We call a state $t_c \in \{t_0, t_1, \ldots, t_k\}$ bidirectional iff it has a transition function which contains conjunction among states which lead in both directions, for example $\delta(t_c, b) = (q_0, 1) \wedge (q_1, -1)$ where $b \in \Sigma$ and $\{t_c, q_0, q_1\} \in Q$. We then define another state $\overrightarrow{t_c}$ which have $\delta(t_c, b)$ in disjunctive normal form without the backward direction tuples as transition function, and another state $\overleftarrow{t_c}$ which have $\delta(t_c, b)$ in DNF without forward directions as transition function.

$$\forall b \in \Sigma \text{ let } \delta(t_c, b) = \bigvee_i c_i \text{ be a CNF.}$$

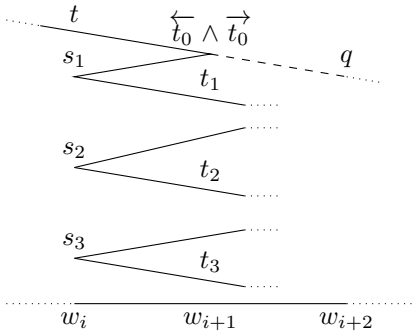
Than for all clauses c_i of $\delta(t_c, b)$ we define a clause $\overrightarrow{c_i}$ and $\overleftarrow{c_i}$ s.t.

$$(q, 1) \in c_i \rightarrow (q, 1) \in \overrightarrow{c_i} \text{ and}$$
$$(q, -1) \in c_i \rightarrow (q, -1) \in \overleftarrow{c_i}.$$

Using this, we now define the following new transitions:

$$\delta(\overrightarrow{t_c}, b) = \bigvee_i \overrightarrow{c_i}$$
$$\delta(\overleftarrow{t_c}, b) = \bigvee_i \overleftarrow{c_i}$$

Figure 4 shows an example for $k = 3$ and $\delta(t_0, w_{i+1}) = (q, 1) \wedge (s_1, -1)$. The ABA has to check if t_j will lead to s_{j+1} for $0 \leq j < k$ and if the singleton state t_k will eventually reach an accepting state reading the rest of the word w.



$$R^t_{w_i} = \{\langle \overleftarrow{t_0}, s_1, t_1, s_2, t_2, s_3, t_3 \rangle \wedge \overrightarrow{t_0}\}$$

$$t_1 \in F$$

$$\eta((t, \bot), w_i) = ((\overleftarrow{t_0}, s_1), \bot) \wedge (\overrightarrow{t_0}, \bot) \wedge ((t_1, s_2), \bot) \wedge ((t_2, s_3), \bot) \wedge (t_3, \top)$$

$$\eta((\overrightarrow{t_0}, \bot), w_{i+1}) = (q, \bot)$$

Figure 4: Example of ABA zigzag run at the singleton state t with $k = 3$ and with the transition with conjunction in different directions

Given a state t and a symbol a of the alphabet Σ we construct the set R^t_a of conjunctions of all possible state sequences which are of length at most $2n - 1$ which do not contain equal states at two even positions (t) or at two odd positions (s), and the states $\overrightarrow{t_i}$ we have already defined. We further demand that the first state t_0 in the sequence be a successor of t when reading a, so $(t_0, 1) \in \delta(t, a)$ and demand the same for all pairs s_j and t_j, so $(t_j, 1) \in \delta(s_j, a)$ must hold for $1 \leq j \leq k$.

Formally:

$$R^t_a = \left\{ \begin{array}{c|l} \langle t_0, s_1, t_1, \ldots, s_g, \overleftarrow{t_g}, \ldots, s_h, \overleftarrow{t_h}, \ldots, s_k, t_k \rangle & 0 \leq g \leq h \leq k < n \\ \wedge & (t_0, 1) \in \delta(t, a) \\ \overrightarrow{t_g} \wedge \cdots \wedge \overrightarrow{t_h} & \forall i < j, s_i \neq s_j \text{ and } t_i \neq t_j \\ & \forall j, (t_j, 1) \in \delta(s_j, a) \\ & t_g, \ldots, t_h \text{ are bidirectional states} \end{array} \right\}$$

In the ABA, the state $t_k \in Q'$ could be labelled either with $\top$ or with $\bot$. The first case means that there is at least one pair state (t_j, s_{j+1}) with $0 \leq j < k$ labelled with $\top$, indicating it guesses to encounter an accepting state while going from t_j to s_{j+1}. Although other pairs might visit an accepting state, they will labelled by $\bot$. The second case means the opposite.

In order to formalize it, we construct the sequence α_k^R including all the labels assigned to the singleton and pair states extracted from R_a^t sequences, in which if the last element is $\top$ there has to be exactly one other element to be $\top$. Otherwise, all elements has to be $\bot$.

$$\alpha_k^R = \left\{ \langle \alpha_0, \ldots, \alpha_k \rangle \in \{\bot, \top\}^{k+1} \;\middle|\; \begin{array}{ll} \text{If } \alpha_k = \top & \text{then } \exists! i : 0 \leq i < k \text{ and } \alpha_i = \top \\ \text{If } \alpha_k = \bot & \text{then } \forall\, 0 \leq i < k \text{ and } \alpha_i = \bot \end{array} \right\}$$

Combining the R_a^t and the α_k^R we are able produce a great part of the ABA transitions, but this is not enough. What we need now is to consider the 2ABA branches ending in infinite loops. To handle this, we have to construct one set of two sequences of pair states. The first one describes the zigzag part which lead from the state t to the first state of the loop. The second one describes the loop itself. We can see an example in Figure 5.

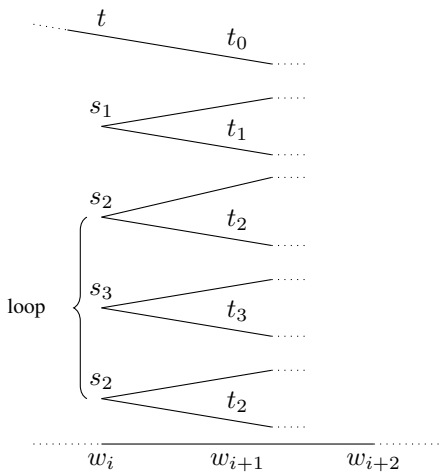
Formally, we construct a set of sequences of two sequences of states L_a^t where:

- The length of both sequences is less than the total number of states n
- The first element of the first sequence t_0^1 is the successor of t when it read the symbol $a \in \Sigma$, so $(t_0^1, 1) \in \delta(t, a)$
- The first state of the second sequence t_0^2 is the successor of the last state of the first sequence s_k^1, so $(t_0^2, 1) \in \delta(s_k^1, a)$
- In both sequences there are not odd equal states and even equal states
- Every t_i states are successors of s_i states, so $(t_i^p, 1) \in \delta(s_i^p, a)$ where $p \in \{1, 2\}$
- The last state of both sequences are equal

$$L_a^t = \left\{ \left\langle \begin{array}{l} \langle t_0^1, s_1^1, \ldots, t_{k-1}^1, s_k^1 \rangle, \\ \langle t_0^2, s_1^2, \ldots, t_{l-1}^2, s_l^2 \rangle \end{array} \right\rangle \;\middle|\; \begin{array}{l} 0 \leq k < n, \quad 0 \leq l < n \\ (t_0^1, 1) \in \delta(t, a), \quad (t_0^2, 1) \in \delta(s_k^1, a) \\ \forall i < j, s_i^1 \neq s_j^1 \text{ and } t_i^1 \neq t_j^1 \\ \forall i < j, s_i^2 \neq s_j^2 \text{ and } t_i^2 \neq t_j^2 \\ \forall i, \forall p, (t_i^p, 1) \in \delta(s_i^p, a) \\ s_k^1 = s_k^2 \end{array} \right\}$$

To be sure an accepting state is visited infinitely often during the loop, at least one of the pair state in the second sequence has to be accepting.
Just like in the normal case, we define our sequences of $\bot$ and $\top$ as follows:

$$\alpha_l^L = \{\langle \alpha_o, \ldots, \alpha_l \rangle \in \{\bot, \top\}^{l+1} | \exists! i : \alpha_i = \top\}$$



$$L^t_{w_i} = \left\langle \begin{array}{l} \langle t_0, s_1, t_1, s_2\rangle, \\ \langle t_2, s_3, t_3, s_2\rangle \end{array} \right\rangle$$

$$\eta((t,\bot), w_i) = (t_0, s_1, \bot) \wedge (t_1, s_2, \bot) \wedge (t_2, s_3, \bot) \wedge (t_3, s_2, \top)$$

Figure 5: Example of ABA transition at the state t guessing an infinite loop starting in s_2 while reading w_i

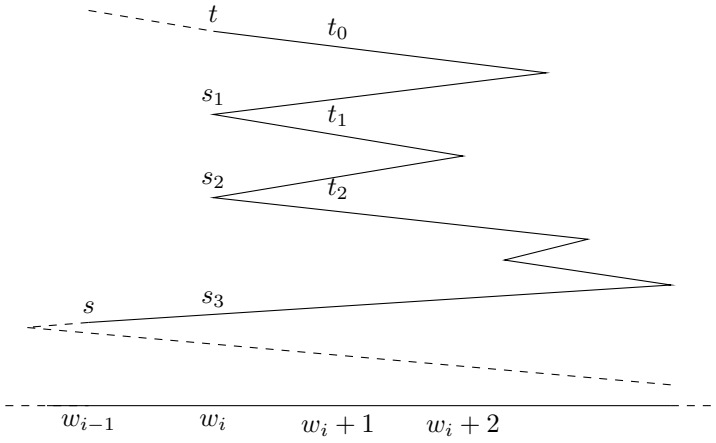
Formally, the transition function of a zigzag branch of the 2ABA run for singleton states chooses a sequence in $R^t_a \cup L^t_a$ and a sequence of $\top$ and $\bot$. It is defined as follows:

$$\eta((t,\bot),a) = \eta((t,\top),a) = \begin{array}{c} \bigvee\limits_{R^t_a, \alpha^R_k} \begin{array}{c} (t_0, s_1, \alpha_0) \wedge \cdots \wedge (t_{k-1}, s_k, \alpha_{k-1}) \wedge (t_k, \alpha_k) \\ \wedge(\overrightarrow{t_g}, \bot) \wedge \cdots \wedge (\overrightarrow{t_h}, \bot) \end{array} \\ \bigvee \\ \bigvee\limits_{L^t_a, \alpha^L_l} \left(\begin{array}{l} (t^1_0, s^1_1, \bot) \wedge \cdots \wedge (t^1_{k-1}, s^1_k, \bot) \wedge \\ (t^2_0, s^2_1, \alpha_0) \wedge \cdots \wedge (t^2_{l-1}, s^2_l, \alpha_l) \end{array} \right) \end{array}$$

3.3 Transitions at pair states

The transitions at pair states are similar to those at singleton states recently explained. When the ABA is in a pair state (t, s) reading the symbol w_i, it has to check if there is a zigzag segment which connects t to s. It guess all the states of the 2ABA between t and s in which w_i will be red which are part of backward tuple followed by a forward tuple. As in the singleton case there will be k other states reading w_i. We name these states according to the order they appear during the zigzag run as $s_1, s_2, \ldots, s_k$. We then denote

their successors by $t_1, t_2, \ldots, t_k$ and we call t_0 the successor of t. The ABA has also to guess the state reading w_i which is part of a backward tuple (s_{k+1}) followed by another backward tuple (s), so $(s, -1) \in \delta(s_{k+1}, w_i)$. Figure 6 shows an example with $k = 2$.



$$R_{w_i}^{(t,s)} = \{\langle t_0, s_1, t_1, s_2, t_2, s_3\rangle\}$$

$$\eta(((t,s), \bot), w_i) = ((t_0, s_1), \bot) \wedge ((t_1, s_2), \bot) \wedge ((t_2, s_3), \bot)$$

Figure 6: Example of AFA transition at the pair state (t, s) with $k = 3$

Given a state t and a symbol of the alphabet a we construct the set $R_a^{t,s}$ of all possible state sequences which are of length at most $2n - 1$ which do not contain equal states at two even positions (t) or at two odd positions (s), like the set R_a^t we described before. We further demand that the first state t_0 in the sequence to be a successor of t when reading a, so $(t_0, 1) \in \delta(t, a)$ and demand the same for all pairs s_j and t_j, so $(t_j, 1) \in \delta(s_j, a)$ must hold for $1 \leq j \leq k$. Finally, we need $(s, -1) \in \delta(s_{k+1}, a)$ to be true.
Formally:

$$R_a^{(t,s)} = \left\{ \langle t_0, s_1, t_1, \ldots, s_k, t_k, s_{k+1}\rangle \;\middle|\; \begin{array}{l} 0 \leq k < n \\ (t_0, 1) \in \delta(t, a) \\ (s, -1) \in \delta(s_{k+1}, a) \\ \forall i < j, s_i \neq s_j \text{ and } t_i \neq t_j \\ \forall i, (t_i, 1) \in \delta(s_i, a) \end{array} \right\}$$

If a pair state (t, s) is labelled with $\top$, we have to check that a visit to an accepting state $f \in F$ occurs on the segment of the zigzag run connecting t to s. If $t \in F$ or $s \in F$, this control is already passed.
If for a pair state (t, s) reading w_i the prediction $(s, -1) \in \delta(t, w_i)$ already holds, the transition function will be evaluated to **true** because the predicted future has been verified for this branch. If, differently, the predicted future is not yet verified, the ABA non-deterministically chooses one of the sequences from $R_a^{(t,s)}$ and continues the process.

We can now define a set of sequences of $\bot$ and $\top$.

$$\alpha^R_{(t,s),k} = \left\{ \langle \alpha_0, \ldots, \alpha_k \rangle \in \{\top, \bot\}^{k+1} \;\middle|\; \begin{array}{ll} \text{If } s \notin F \text{ and } t \notin F & \text{then } \exists! i : \alpha_i = \top \\ \text{Otherwise} & \forall\, 0 \leq i \leq k, \alpha_i = \bot \end{array} \right\}$$

We now can define η for pair states as follows:

$$\eta((t,s),\bot),a) = \begin{cases} \textbf{true} & \text{If } (s,-1) \in \delta(t,a) \\ \bigvee\limits_{R_a^{(t,s)}} (t_0, s_1, \bot) \wedge \cdots \wedge (t_{k-1}, s_k, \bot) & \text{Otherwise} \end{cases}$$

$$\eta((t,s),\top),a) = \begin{cases} \textbf{true} & \text{If } (s,-1) \in \delta(t,a) \\ & \text{and } (s \in F \text{ or } t \in F) \\ \bigvee\limits_{R_a^{(t,s)}, \alpha^R_{(t,s),k}} (t_0, s_1, \alpha_0) \wedge \cdots \wedge (t_{k-1}, s_k, \alpha_k) & \text{Otherwise} \end{cases}$$

Since we defined the transition function η, we can now construct the 1-way ABA.

4 Conclusions

In this paper we discussed the translation from 2-way Alternating Büchi Automata to 1-way. Such translation works with all the automata of this type.
We get a 1-way ABA with $\mathcal{O}(n^2)$ states and, since we are working with ω-words, transitions of exponential size. For that reason we can not use the Globerman and Harel 0-steps construction [GH96] to reduce it to polynomial size.
Next step of this work could be proving that this translation starting from 2-way Very Weak ABA produce an 1-way Very Weak ABA because this could be translated back to LTL.

Acknowledgements

The author would like to thank: university of Lübeck and Prof. Dr. Martin Leucker, Malte Schmitz and Torben Scheffel for supervising this work; the Erasmus+ project for financing it and the DAAD and the Baltic Summer School for giving him the possibility to publish this paper.

References

[BBG+94] Ilan Beer, Shoham Ben-David, Daniel Geist, Raanan Gewirtzman, and Michael Yoeli. Methodology and System for Practical Formal Verification of Reactive Hardware. In *Computer Aided Verification, 6th International Conference, CAV '94, Stanford, California, USA, June 21-23, 1994, Proceedings*, pages 182–193, 1994.

[GH96] Noa Globerman and David Harel. Complexity Results for Two-Way and Multi-Pebble Automata and their Logics. *Theor. Comput. Sci.*, 169(2):161–184, 1996.

[GO03] Paul Gastin and Denis Oddoux. LTL with Past and Two-Way Very-Weak Alternating Automata. In *Mathematical Foundations of Computer Science 2003, 28th International Symposium, MFCS 2003, Bratislava, Slovakia, August 25-29, 2003, Proceedings*, pages 439–448, 2003.

[KPV01] Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. Extended Temporal Logic Revisited. In Kim Guldstrand Larsen and Mogens Nielsen, editors, *CONCUR*, volume 2154 of *Lecture Notes in Computer Science*, pages 519–535. Springer, 2001.

[LPZ85] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck. The Glory of the Past. In *Logics of Programs, Conference, Brooklyn College, June 17-19, 1985, Proceedings*, pages 196–218, 1985.

[LS10] Martin Leucker and César Sánchez. Regular Linear-Time Temporal Logic. In Nicolas Markey and Jef Wijsen, editors, *TIME*, pages 3–5. IEEE Computer Society, 2010.

[LS11] Andreas Bauer 0002, Martin Leucker, and Christian Schallhart. Runtime Verification for LTL and TLTL. *ACM Trans. Softw. Eng. Methodol.*, 20(4):14, 2011.

[MH84] Satoru Miyano and Takeshi Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32(3):321–330, 1984.

[PV03] Nir Piterman and Moshe Y. Vardi. From bidirectionality to alternation. *Theor. Comput. Sci.*, 1-3:295–321, 2003.

[RS59] Michael O. Rabin and D. Scott. Finite Automata and Their Decision Problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[She59] J. C. Shepherdson. The Reduction of Two-way Automata to One-way Automata. *IBM J. Res. Dev.*, 3(2):198–200, April 1959.

[SL10] César Sánchez and Martin Leucker. Regular Linear Temporal Logic with Past. In Gilles Barthe and Manuel Hermenegildo, editors, *Proceedings of the 11th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'10)*, volume 5944 of *Lecture Notes in Computer Science*, page 295–311. Springer, Springer, 2010.

[Tho81] Wolfgang Thomas. A Combinatorial Approach to the Theory of omega-Automata. *Information and Control*, 48(3):261–283, 1981.

[Var98] Moshe Y. Vardi. Reasoning about The Past with Two-Way Automata. In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, pages 628–641, 1998.