SNIPER FIRE LOCALIZATION USING WIRELESS SENSOR NETWORKS AND GENETIC ALGORITHM BASED DATA FUSION

Patrick Kuckertz, Junaid Ansari, Janne Riihijärvi and Petri Mähönen
Department of Wireless Networks, RWTH Aachen University, Aachen, Germany

ABSTRACT
This paper presents the design, implementation and performance evaluation of a wireless sensor network based sniper fire localization system. Time-of-arrival (TOA) measurements from various sensor nodes are fused together for estimating various parameters of a supersonic projectile, including its trajectory and speed. Our system uses a genetic algorithm based data fusion framework. The system uses a novel approach for the selection of TOA measurements. We obtained very low error bounds both in the trajectory and velocity estimation results. We believe that the presented comprehensive design details of different components of the system, the developed simulation tool and the validation of results from real shooting experiments will be useful to the community.

INTRODUCTION
Snipers pose a potential threat to diplomats in many parts of the world. In many of the scenarios, a particular area or building needs to be guarded and protected against snipers. We provide a wireless sensor network (WSN) based solution for locating snipers targeting a particular building situated inside the protection area. For estimating the trajectory of a fast moving bullet (a supersonic projectile) coming towards a building, we propose to deploy many inexpensive wireless sensor nodes around the building. These nodes are equipped with a broadband microphone based customized analog circuit board, which is capable of detecting the shockwave generated by the supersonic projectile. All the nodes must be time synchronized with each other so that all the individual events are analyzed at the common reference time. Fig. 1 shows the configuration of the WSN based system. The individual sensor nodes perform time-of-arrival (TOA) measurements and send their individual detection information to a gateway node, attached to a PC. The PC performs fusion of the TOA measurements to determine the trajectory estimation parameters using a genetic algorithm (GA).

When a bullet is fired, it generates a shockwave and a muzzle blast. Our system does not aim to rely on the muzzle blast because its range is limited to the proximity of the sniper’s physical location and it requires that the sniper is very near to the sensor nodes, which might not be the case. Furthermore, most of the present day weapons are equipped with a silencer to suppress the muzzle blast. It may be noted that when a bullet approaches a building (surrounded by sensor nodes), only the nodes lying between the sniper and the building receive the TOA measurements. The nodes lying behind the building (shadowed nodes) do not receive TOA of the shockwave and hence will not be considered for localization.

Figure 1: Wireless sensor network based trajectory estimation system for a supersonic projectile. A customized circuit board attached to each sensor node detects the shockwave generated by the flying supersonic projectile. The sensor nodes transmit the detection information to a gateway node, which is attached to a PC, where data fusion algorithm for computing the trajectory of the projectile is implemented.

The rest of the paper includes the related work, the mathematical formulation for TOA calculation at a sensor node, the overall embedded system architecture. Later on, we comprehensively describe various components of the genetic algorithm based TOA fusion framework and discuss
the results. Finally, we conclude the article and propose some future enhancements to the system.

RELATED WORK

Acoustic localization has been used in many systems both actively and passively. The problem of weapon fire localization has always been important to military, border guards and security agencies. In this context, WSN based systems relying on acoustic signature detection are the most promising ones.

N. Levanon [1] describes a system, which is used to determine the hit position of a supersonic projectile at a virtual plane. Acoustic transducers are placed in the ultimate vicinity of the target plane to measure the TOA of the shockwave. With known position of the transducers, two hit coordinates \((x, h)\) on the virtual vertical target plane can be estimated, in addition to the speed and the horizontal angle of the projectile.

M. Cannella et al. [2] describe a system with six different TOA values detected by nine pressure transducer sensors placed at different spatial locations to measure the direction and speed of a supersonic projectile.

In WSNs, the most concrete research has been carried out by the researchers at Vanderbilt University (V. U.). They developed a series of systems relying on the muzzle blast as well as the shockwave [3]. They used frequency analysis using DSP based solutions for separating shockwave signal from the muzzle blast. For shooting inside a sensor field, 8-10 measurements are used to compute the position of the shooter by applying the method of trilateration. Muzzle blast cannot be relied upon for localized shots fired outside the sensor field. The shockwave of the bullet is used in such cases to compute TOA at different locations and later on fused to estimate the trajectory of the projectile.

Leđeczi et al. [3] argue that the system may be used in urban warfare scenarios, yet the system lacks the self localization scheme. Furthermore, the underlying details of the selection of nodes, feature space boundaries and GA implementation are not precisely given in the references cited. The system is evaluated in two scenarios: when the bullet goes through the sensor field and when it flies near the edge of the sensor field.

We also evaluated the performance of our system in the above mentioned two cases showing that our system yielded accurate performance although the spatial expanse used by our system is smaller. We believe that this is because of improved chromosome selection (both initial and intermediate) and effective elimination of unfit TOA measurements in the GA. (Please refer to later sections for details.) It may be noted that we could not perform one to one comparison because not all the necessary information from V. U. system is available. Our system uses an analog shockwave detection circuit, which not only makes the system much cheaper but also saves the energy consumed in sampling and detecting acoustic pattern of the shockwave signal as done in a DSP board.

Finally we note that techniques based on infrared-imagery and on thermal signatures of the bullet have also been suggested but they have been found to have range constraints and to suffer from false alarms [4] [5].

CALCULATION OF TOA AT A SENSOR NODE

![Figure 2: A supersonic projectile hits the virtual plane, vertical to its direction. With a given model containing all information about a projectile's trajectory and velocity, the TOA value can be calculated.](image)

A shockwave generated by a supersonic projectile travels with the velocity of sound. This shockwave spreads conically by Mach angle given by

\[
\alpha_M = \sinh \left( \frac{v_{\text{sound}}}{v_{\text{projectile}}} \right).
\] (1)

Fig. 2 illustrates the path of the first shockwave that arrives at a sensor node at point \(X_m\). The measured TOA at the sensor node consists of two time components \(t_1\) and \(t_2\), representing the time for the projectile to reach point \(B_d\) and the duration of the shockwave to travel from \(B_d\) to the transducer, respectively. The position \(B_d\) indicates the point on the projectile’s trajectory from which the shockwave spreads, reaching the sensor node first. The distance from point \(B_d\) to the virtual panel going through the sensor node depends only on the relation of the speed of sound and
the velocity of the projectile as given by

\[ B_d X_s = \frac{d}{\sqrt{\left(\frac{v_{projectile}}{v_{sound}}\right)^2 - 1}}, \]  

(2)

where \( d \) is the shortest distance between the projectile and the sensor node [1].

We considered a 5.6 mm HK G36 projectile with a weight of 4.0 g. We noticed from the specifications that the projectile suffers some deceleration and has crosswind effects. In order to minimize these effects, we suggest that the spatial expanse of the sensor nodes should be limited. The effects of temperature and humidity are negligible and can safely be ignored.

**EMBEDDED SYSTEM ARCHITECTURE**

Fig. 1 shows the overall architecture of the system. The system is composed of many off-the-shelf Crossbow Inc.’s MICA2 sensor nodes. A customized broadband microphone based detection circuit, attached to each MICA2 node, is used to reliably detect the TOA of the shockwave on each node in the sensor field. Fig. 3 shows a customized acoustic signal detection board that triggers an interrupt on the node upon detecting a shockwave. The circuit basically consists of an electret microphone cartridge, an amplifier, a comparator and a timer. This approach removes the need for sampling the acoustic signal at a very high rate for detecting the short duration impulse signal of the shockwave. We found it impractical to detect a shockwave using MICA2 node’s ADC due to the low supported sampling rate.

Figure 3: Customized circuit board for detecting shock wave signal that can be attached to MICA2 node using the 51-pin expansion connector.

In our prototype setup, the nodes need to be hand-placed at known positions because the nodes do not use a self-localization scheme. This does not put a constraint in scenarios where a protection area has to be guarded against snipers and self-localization capability can anyways be added if needed, although this comes at a cost of increased complexity, node price and energy consumption.

The nodes transmit the TOA values to the gateway node upon detecting a shockwave. The reported times of the shockwave event by different sensor nodes must be at the common reference time scale. Our system uses a MAC time-stamping protocol known as Elapsed Time of Arrival (ETA) [6]. Using this protocol, all the nodes use a time-stamping method with the time of the gateway node as the reference time. We found a mean time-stamping error of approx. 2.5 \( \mu s \) in a single hop scenario. For these measurements two nodes have been attached to one customized circuit board, so that the interrupt on both nodes is triggered at the same time. Since the velocity of the projectile is higher than the velocity of the sound, the nodes receive the TOA of the shockwave before the muzzle blast.

The gateway PC, acting as a data fusion center, contains information about the positions of the sensor nodes and has enough processing power for applying fusion algorithm on the collected data at the gateway node. The fusion center searches for a consistent subset of nodes in the set of all the received TOA values of a particular shockwave event. Using the consistent subset of nodes, a two-stage genetic-algorithm is applied to determine the trajectory and velocity of the projectile with high degree of reliability. The system is able to effectively suppress the positioning errors of nodes and even a few hall errors.

The embedded software application, running on sensor nodes was developed in TinyOS, which is an open-source operating system specially designed for the resource constrained wireless sensor nodes. The gateway PC application, running a two-stage genetic algorithm for trajectory estimation of the projectile, was developed in C#.NET.

**DATA FUSION AND TRAJECTORY ESTIMATION**

As described before, all the nodes transmit the computed TOA measurements to the gateway PC which performs fusion of the collected data to estimate the trajectory of the projectile. In the following, we describe the individual processes involved.

**Pre-filtering of the TOA Measurements**

The individual sensor node positions are known before hand at the gateway PC. The maximum distance \( d_{max} \) between any two nodes defines the maximum time \( t_{max} = d_{max}/v_{sound} \), which defines the boundary limit for the set of TOA measurements, describing a given acoustic event. Since the
projectile travels with a supersonic velocity, the shockwave arrives before the muzzle blast at the transducer circuit. The TOA measurements are saved in a sorted list, which is parsed for a set of at least six TOA values in the interval of $t_{\text{max}}$. Since the radio messages from different nodes for a particular acoustic event do not arrive at the same time instant, a delay timer is used to allow more than six TOA values. The maximum distance between any two nodes of the resulting set of nodes is computed and the nodes, which do not adhere to the resulting time range are removed from the subset. Afterwards, the remaining nodes are checked for consistency. The minimum velocity between any two nodes in the subset is also computed and if the velocity is found to be smaller than the speed of sound, both nodes are removed from the subset. For estimating the trajectory of the projectile, the subset is required to have data from at least six nodes after the pre-filtering operation.

### Genetic Algorithm Modelling

We use a continuous genetic algorithm to find the best model for the gathered TOA values at different node positions. We use a chromosome of 7 genes representing the unknown parameters of the model. The first 6 parameters describe the trajectory and velocity of the 7-parameter model: $(X, Y, Z, \alpha_{\text{azimuth}}, \alpha_{\text{elevation}}, v_{\text{projectile}}, S)$, whereas the parameter $S$ describes the subset of nodes taken into consideration for calculation of the fitness function. By calculating the fitness function, error corrections within the genetic algorithm is possible by removing the shortly delayed TOA measurements. The subset of nodes $S$, is represented as a binary string and 1 indicates that the node is part of the subset, whereas 0 shows that the node at this position is not part of the subset.

#### Initial Population

We use a random initial population to generate further population of chromosomes. Each gene of a chromosome must lie within the boundaries as described in Table 1. When a projectile passes by the sensor field, multiple solutions may be obtained, specially when error mitigation by selecting a subset of nodes is involved. This phenomenon is referred to as trajectory inversion problem, described in [3]. Since, we are not able to detect trajectories outside the sensor field reliably, especially when error correction is involved, we propose to limit the spatial expanse to the area of the convex hull formed by the subset of nodes. This approach assumes that the trajectory of the projectile goes through the sensor field. There are several known algorithms available to compute the convex hull of non-sorted points forming a polygon. We use Graham Scan [7] for computing the convex hull because of its computational advantages. Afterwards, we determine if a randomly created point lies within the polygon. If it is outside, the gene is simply discarded. We only use the convex hull in the process of generating the initial population. This allows an accurate estimation of a trajectory at the border of the sensor field.

#### The Fitness Function

The theoretical TOA value of each sensor node is determined by the model. The fitness function calculates the fitness of the gathered TOA measurements to that of the model. The fitness for each chromosome is computed after spawning the initial population.

Since the reference time $t_0$ of the gathered and calculated TOA values is not the same, the calculated values have to be shifted to make them overlap with the measured values. First, the mean of the TOA values in the subset is computed by

$$\text{TOA}_{\text{mean}} = \frac{\sum_{i \in S} \text{TOA}_i}{|S|},$$

where $\text{TOA}_{\text{mean}}$ denotes the mean value of the set of TOA values, $i$ is the index of TOA value and $|S|$ is the number of nodes in the subset. Later on, the measured TOA values are shifted by the mean values to overlap with the measured TOA values as:

$$\text{TOA}_{\text{shift}} = \text{TOA}_{\text{mean,measured}} - \text{TOA}_{\text{mean,calculated}},$$

$$\text{TOA}_{\text{measured,}i} = \text{TOA}_{\text{measured,}i} - \text{TOA}_{\text{shift}}.$$  

The set of nodes taken for the calculation of the fitness criterion is given by

$$i \in \tilde{S} \text{ if } i \in S \text{ or } (\text{TOA}_{\text{measured,}i} - \text{TOA}_{\text{calculated,}i}) < T,$$

where $\tilde{S}$ is the subset of nodes taken for the calculation of the fitness, $S$ is the pre-filtered set of nodes and $T =$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Xmin</td>
<td>Xmax</td>
</tr>
<tr>
<td>Y</td>
<td>Ymin</td>
<td>Ymax</td>
</tr>
<tr>
<td>Z</td>
<td>Zmin</td>
<td>Zmax</td>
</tr>
<tr>
<td>$\alpha_{\text{azimuth}}$</td>
<td>0</td>
<td>360</td>
</tr>
<tr>
<td>$\alpha_{\text{elevation}}$</td>
<td>80</td>
<td>160</td>
</tr>
<tr>
<td>$v_{\text{projectile}}$</td>
<td>$v_{\text{sound}} + 1$</td>
<td>930</td>
</tr>
<tr>
<td>S (Subset of nodes)</td>
<td>7 number of nodes</td>
<td></td>
</tr>
</tbody>
</table>
TOA$_{error} + \text{position}_\text{error}/v_{\text{sound}}$ is the sum of the maximum allowed delay of TOA recordings and the maximum positioning error. Finding an optimized value of the threshold parameter $T$ is very important because a very small value will suppress correct measurements, whereas a very large value does not allow us to filter out delayed TOA recordings within the GA framework. We found that a value of 1 ms is suitable for a projectile going through the middle of the sensor field and 3 ms for a projectile at the border of the field. The underlying rationale is as follows: a projectile going through the middle of the sensor nodes field roughly divides the possible number of nodes for detecting an inconsistency in the pre-filtering phase by two. This is because the time difference between nodes on different sides of the trajectory is smaller than that on the same sides and therefore, the velocity is increased. A trajectory at the border of the sensor field causes a higher possibility to detect an inconsistent set of two nodes. Since the GA may find a wrong model with a too low $T$ value, the parameter should be selected higher. In pre-filtering phase, the case is judged by considering the TOA measurements as well as the angles of the minimum velocities. The fitness of a given model is obtained from

$$\text{Fitness} = 1 - \frac{\sum_{i \in S} (\text{TOA}_{\text{measured},i} - \text{TOA}_{\text{calculated},i})^2}{|S|}. \quad (6)$$

The Mating Process

In the mating process, two children are created from a set of two chromosomes. A random collection of genes participating in crossing is determined. Afterwards, for every gene, a random probability $p$ for the father and $(1-p)$ for the mother is determined. The new genes are created by

$$\text{child}_{\text{gene},i} = p \times \text{mother}_{\text{gene},i} + (1-p) \times \text{father}_{\text{gene},i}. \quad (7)$$

This method cannot create genes outside their boundaries. We have also observed scenarios where a gene of the model is located at its boundary. In this case, the model may not be determined accurately. Since this crossing strategy always drifts in the direction towards the average of the parents, the boundary of a gene is not reached by crossing in any case. Therefore, we define the parameter space of a gene of the chromosome to have a ring topology instead of line segment, and we determine the direction of the shortest distance between mother and father on the ring. The shortest distance is taken for the direction of the crossing, which results in a fair crossing to all directions in the feature space of a gene.

Selection Strategies of the Intermediate Population

We use Tournament Selection [8] for generating intermediate generation of chromosomes. It starts with a randomly created initial population and later on randomly selects an intermediate population with a given size called the tournament size. The best chromosomes of the intermediate population are taken into the mating pool. If the tournament size is too high, the population may get stuck in a local suboptimal solution. If the tournament size is too small, the algorithm will need unnecessary long time to reach the optimal solution.

Our realization of the tournament selection involves following steps:

1. Initial Population with a size of 5 000 chromosomes
2. Sort Population: sorts the chromosomes of the population by their fitness.
3. Delete Doubles: The procedure is executed every 10th generation.
4. Remove Chromosomes: removes worst chromosomes (Number: 20% of intermediate population)
5. Intermediate Population: randomly created from initial population with a size of 500 chromosomes
6. Mating: crosses best 20% of the chromosomes of the initial population
7. Next Generation: going back to step 2 and repeating the procedure until generation number of 150 is reached, or the number of the population is smaller than the number of the intermediate population.

Trajectory Estimation Simulator

We also developed a Trajectory Estimation Simulator with a GUI as shown in Fig. 4. The simulator allows flexible setting the position of the marksman, the velocity of the projectile and the trajectory of the projectile. The number and position of sensor nodes can also be selected. One can also specify positioning errors and delayed TOA values for individual nodes. In addition to the simulator, we also developed a Real Time Trajectory Estimator having the similar outlook that fuses the real TOA values gathered from the individual sensor nodes.
Figure 4: The simulation shows the \((x, y)\) plot for a sensor node field and the trajectory of a projectile going through it. The position of the marksman, the trajectory of the projectile and the individual positions of the sensor nodes can be set in a flexible way in the simulator. Furthermore, the simulator also allows to use different types of positioning errors and TOA errors models.

RESULTS

We use a set of 13 nodes in our simulator, so as to allow to discard several delayed TOA recordings in the pre-filtering and fitness selection phases. The dimensions of the sensor field used is 9 m by 22 m. Two sets of trajectories are analyzed: One trajectory goes through roughly the middle of the field of sensor nodes, whereas the other trajectory goes through the top part of the sensor node field and is very close to the border of the field. These two scenarios are referred to as Set1 and Set2, respectively.

Results for Set1

Figure 5 shows the azimuth and elevation angle errors for 100 iterations. We use a random positioning error of up to 10 cm in all dimensions and TOA errors of up to 10 \(\mu\)s. The mean azimuth angle error and the elevation angle error are found to be 0.12° and 0.04°, respectively. The mean velocity error for this case is found to be 4 m/s.

Figure 6 shows the results for Set1 with additional TOA error of 1200 \(\mu\)s for Node 11 and Node 12. The azimuth angle error is increased to a mean value of 0.3° and the elevation angle error to 0.23°. It may be noted that our pre-filtering approach finds Node 12 to be inconsistent to Node 7 and removes both of them from the subset of the nodes. Node 11 is automatically filtered out of the subset of nodes within the GA. Overall it can be seen that the system mitigates the erroneous measurements quite effectively.

Figure 7 shows Set1 for a scenario with four delayed TOA recordings of up to 4000 \(\mu\)s. Node 12 is found to be inconsistent with Node 3. After removing the two nodes, 11 nodes are left to find the best subset of 8 nodes, where the
other three nodes with increased TOA value have to be removed from the subset of nodes. We found an azimuth angle error of 1.4° and the elevation angle error of 0.4°. The mean velocity error of the projectile is found to be 19 m/s. However, in iteration number 40 a completely wrong model has been estimated.

**Results for Set2**

Fig. 8 shows the azimuth and elevation angle errors for Set2 scenario. The mean azimuth angle error is found to be 2°.

![Figure 8: Azimuth and elevation angle errors for Set2 with random positioning error of up to 10 cm in all dimensions and a random additional TOA error of up to 10 µs.](image)

Figure 8 shows the azimuth and elevation angle errors for Set2 with random positioning error of up to 10 cm in all dimensions and a random additional TOA error of up to 10 µs.

larger than the same case of Set1. Since 11 estimations of the model are above this level, the mean accuracy is not good. Therefore approximately 11% of the iterations result in a wrong model. The elevation angle error is not as large as the azimuth angle error. In both cases the worst model was estimated in iteration number 56. In Fig. 9 Node 12 is delayed by 3000 µs. It is inconsistent with Node 7 and both of the nodes are removed from the set of measurements by the algorithm. The mean azimuth and the elevation angle errors are found to be 1.25° and 2°, respectively. The maximum azimuth angle error turns out to be 3.7°, which is quite smaller than that of the the previous estimation. Since two nodes had already been removed from the set of nodes, the algorithm is left with less opportunities to find a subset of nodes. Although the result looks more stable but the accuracy of correct models decreases when the number of nodes in the subset are decreased.

Therefore, we observed results in a scenario with two delayed TOA recordings and in a case when four nodes are removed from the subset of nodes, the algorithm is unable to find a wrong model by the small number of nodes left for the estimation. Fig. 10 shows the results for Set2 with a delayed TOA of Node 12 and Node 6. The Nodes 6, 12, 7 and 4 are found inconsistent and therefore, removed from the subset of nodes.

By a closed formation of sensor nodes scenario Set2 can be forced to be avoided. Therefore, the performance figures of the system designed for protection of an area of interest is represented by Set1 which has a good accuracy and error compensation.

In real measurements on a small bore shooting stand, we were limited by the spatial expanse of sensor nodes due to the lower velocity of a small bore projectile and the fixed direction of the shooter. Therefore, we observed only one scenario. The nodes were deployed very close to the shooter (within a distance of 6 m). Since the real direction and position of the shooter could not be exactly determined, we do not present quantitative performance evaluation of our real time trajectory estimator. However, it is worth mentioning that the projectile’s velocity was approximated to be 370 m/s, which is quite accurate. Qualitatively, based on visual estimates, the direction estimation was correct.
CONCLUSION AND FUTURE WORK

Our presented WSN based solution is able to accurately estimate the trajectory and velocity of a supersonic projectile. The gathering of the shockwaves’ TOA values at several positions, is realized using a shockwave detection circuit attached to each wireless node, deployed at different positions. The use of an inexpensive circuit board removes the need for an expensive DSP solution—capable of sampling broadband acoustic signal at a very high rate. We found from real shooting experiments that the system is able to gather highly reliable and synchronized TOA readings from sensor nodes. The TOA measurements from various nodes are fused together using a continuous genetic algorithm based data fusion framework. We show that our pre-filtering method and fitness function approach can effectively suppress the nodes with erroneous TOA values. A convex-hull based initial population generation scheme limits the spatial expanse of the search field, whereas the use of tournament selection scheme generates a stable population of chromosomes and avoids getting stuck into a local maxima. We have described the various parts of our data fusion framework in detail. We also developed a GUI-based trajectory estimator, which allows setting, testing and incorporating various types of noise and error models.

It takes approximately five seconds to find the direction of the marksman on a PC with Intel’s P4 processor clocked at 2.66 GHz and equipped with 512 MB of RAM. An additional 4–5 s are required to compute an accurate model. The computational bottleneck of our genetic algorithm is the sorting of chromosomes in every generation. We use the standard quick sort method for the whole population. Since the offspring of chromosomes are not sorted, only 100 chromosomes have to be sorted in every generation. There may be a faster algorithm to sort the added offspring in the population.

Using the simulator, we observed that by spatial expanse of the WSN, it is possible to receive a shockwave signal having LOS-component by using enough sensor nodes to determine the model. Furthermore, with greater spatial expanse of the nodes, the probability of estimating a wrong model is smaller than a deployment scheme with smaller spatial expanse. This is because model estimation has higher influence near the border of the sensor field as compared to the middle of the sensor field. We also observed that positioning errors of up to 30 cm do not influence the accuracy of the system very much. Even a few hall errors of sensor nodes are successfully detected and removed from the set of measurements.

However, erroneous measurements from several nodes cannot produce an accurate estimation of a model. This is because at least six TOA measurements (in a limited time-frame) from different nodes are needed to estimate the model. Using the analog circuit board, it is not possible to detect the type of the projectile generating the shockwave. For this purpose, we propose to use a few (approx. 15 % to 30 %) DSP based nodes, performing pattern matching analysis of the digital signature produced by a projectile at a certain distance from the shooter.

ACKNOWLEDGMENTS

We would like to thank European Union (projects IST-034963-WASP and IST-004536-RUNES) and RWTH Aachen University for the financial support.

REFERENCES