Jerzy MARTYNA

Jagiellonian University

Scheduling algorithm for delay and jitter reduction of periodic tasks in real-time systems

***Abstract**. Real-time systems, especially software control systems, are developed to meet the requirements of real-time automation systems. One such crucial requirement is reducing the delay and jitter of periodic tasks in such systems. In this paper, we present a new method for reducing delays and jitters of periodic tasks, which are enforced by the operating system, control tasks, kernel mechanisms, etc. Our algorithm is evaluated and compared with other scheduling algorithms in terms of jitter. The effectiveness of our algorithm is confirmed by the experimental results.*

***Streszczenie.** W pracy przedstawiono algorytm szeregowania dla redukcji fluktuacji i opóźnień zadań okresowych w systemach czasu rzeczywistego. Zastosowanie takiego algorytmu gwarantuje przewidywalność oraz poprawia efektywność działania systemów. Dodatkowo dzięki niemu jest możliwe zmniejszenie fluktuacji czasu odpowiedzi zadań okresowych. Możliwości działania prezentowanego algorytmu zostały potwierdzone w badaniach symulacyjnych. (**Algorytm szeregowania dla redukcji fluktuacji i opóźnień zadań okresowych w systemach czasu rzeczywistego**)*

Keywords: scheduling algorithms, real-time systems, periodic tasks, reducing the delay and jitter of periodic tasks.
Słowa kluczowe: algorytmy szeregowania, systemy czasu rzeczywistego, redukcja fluktuacji i opóźnień zadań okresowych.

Introduction

In real-time control applications many periodic tasks are executed which are associated with periodic activities such as periodic execution of control tasks, action planning, etc. These tasks are periodically invoked by the scheduler implemented in the kernel of the operating system.

The factors which degraduate the performance of real-time systems are delays and jitter appearing during the execution of periodic tasks. A jitter can be interpreted as disturbances acting upon the system which as a result of interrupts, calls of the operating system kernel, the activities of the scheduler, control programs, etc. In general, these disturbances have a negative effect on the system's performance. This was confirmed in the research studies [8], [16].

The jitter has also been studied in many papers. Among others, J. Nilsson et al. [18] have carried out a stochastic analysis and control of real-time systems with random time delays. A simulated annealing to find the optimal configuration task sets that minimizes jitter has been proposed by M. Di Natale and J. Stankovic in the paper [10]. In another paper, A. Cervin et al. [9] devised a method for finding the upper limit of the input-output jitter of each task by estimating the worst-case and best case response times under Earliest Deadline First (EDF) scheduling. This method introduce of the concept of the *jitter margin*, which guarantees system stabilization through the defined requirements of the jitter.

Independently of the given results, other authors of papers [1], [13], [12] proposed different algorithms for computing the minimum deadline of a newly arrived task, assuming that the existing task set is schedulable by EDF algorithm. In the first of these, P. Balbastre et al. [1] provided the deadline minimisation algorithm. In this solution the deadline reduction achievable in the first task is much higher than that achievable for the other tasks in the sequence. In the given solution the deadline reduction achievable in the first task is much high er than that achievable in the other tasks in the sequence. The authors also proposed a method for achieving uniform scaling all relative deadlines using the critical scaling factor for task deadlines. However, as was stated in the paper by H. Hoang and G.C. Buttazzo [13], using this method the jitter and delay carnot be reduced as expected and for some tasks they could even increase. The paper [12] presents a method for computating the minimum deadlines for periodic tasks in the real-time systems.

An optimal task rate selection method for fixed priority systems was prepared by E. Bini and M. Di Natale [4]. In this paper, the authors presented a method of periodic task selection. A schedulability analysis for a set of periodic tasks under an arbitrary fixed priority assignment was studied in depth by E. Bini and G.C. Buttazzo in paper [3]. A sensitivity analysis for a set of periodic tasks with fixe priorities was given In the paper [5]. The sensitivity analysis is of importance as the worst-case response time analysis.

The problem of jitter and delay reduction in real-time control applicationswas also dealt with in another way. By setting a suitable relative deadline it was possible to limit the execution interval of each periodic task. Based on this metod S. Baruah et al. [2] proposed two algorithms for guaranteeing the schedulability of the task set. G.C. Buttazzo and F. Sensini [6] presented an ”on-line” algorithmto compute the minimum deadline to be assigned to a new arriving task to guarantee schedulability under the EDF algorithm. The same function is performed by the algorithm given by A. Kwiecień [14], which includes some methods for reducing the cycle In the programmable controler.

A number of recent papers have described the conditions of periodic task scheduling using the EDF and RM (Rate Monotonic) algorithms [19]. Ph. Michelon et al. [17] offer a mathematical model of periodic task scheduling and Lower limits of time scheduling. A comparative assessment and evaluation of jitter control methods was presented by G.C. Buttazzo and A. Cervin [7].

The main goal of this paper is to present the scheduling algorithmfor reducing the deadlines of the periodic tasks that minimalize jitters and delays in real-time systems. This algorithm leads to better tuning and also keeps the stability of the real-time system.

The paper is organized as follows. Section 2 presents the system model and the terminology used throughout the paper. In section 3, we give the scheduling algorithm for deadline reduction. Section 4 presents some simulation results and compares the proposed method with other algorithms. Finally, in section 5 we give our conclusions and plans for future work.

The System model

In this section, we present the system model which will be used throughout the paper.

Suppose set $T = \{\tau_1, \tau_2, \dots, \tau_n\}$ of n periodic tasks that must be executable in the uniprocessor system. In the simplest case we assume that the tasks are schedulable under the Earliest Deadline First (EDF) algorithm [15]. Task τ_i, $1 \le i \le n$, is defined by an infinite sequence of task instances,or jobs, which have the same worst-case execution time, C_i, a relative deadline and the same interarrival period. All tasks are fully preemptive. The following notion is used throughout the paper.

$\tau_{i,k}$ - denotes the k-th subtask of the task i, (where $k = 1,2, ...n$), that is the k-th instance of the task execution.

C_i - denotes the worst-case execution time of task τ_i.

T_i - denotes the period of task τ_i or the minimum interarrival time between successive subtasks.

D_i - denotes the relative deadline of task τ_i, that is, the maximum finishing time.

$r_{i,k}$ - denotes the release time of subtask $\tau_{i,k}$

$f_{i,k}$ - denotes the finishing time of subtask $\tau_{i,k}$, that is the time it takes the subtask to complete its execution.

$d_{i,k}$ - denotes the absolute deadline of subtask $\tau_{i,k}$, that is the maximum absolute time before wchich subtask $\tau_{i,k}$ must be completed ($d_{i,k} = r_{i,k} + D_i$).

U_i - denotes the utilization of task τ_i that is the fraction of CPU time used by τ_i , $U_i = C_i / \tau_i$

U - denotes the total utilization of the task set, that is the sum of all task utilizations $U = \sum_{i=1}^{n} U_i$

$R_{i,k}$ - denotes the response time of subtask $\tau_{i,k}$, that is the interwal between its release time at wchich the first instruction of $\tau_{i,k}$ is executed.

$IOD_{i,k}$ - denotes the input-output delay of subtask $\tau_{i,k}$, that is the interval between its start time and its finishing time: $IOD_{i,k} = f_{i,k} - s_{i,k}$.

RTJ_i - denotes the response time jitter of a task, that is the maximum variation In the response time of its subtasks:

$RTJ_i = \max_k \{f_{i,k}\} - \min_k \{R_{i,k}\}$

The dependencies between some of the above defined parameters are illustrated in Fig. 1.

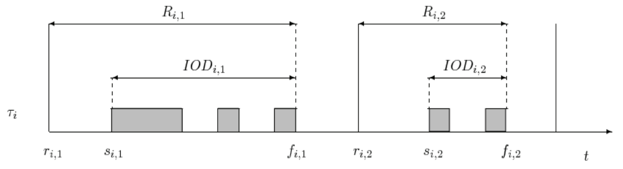

Fig. 1. An ex ample of periodic task

To better understand the given dependencies, we present an example.

Example 1

Suppose three periodic tasks τ_1, τ_2, τ_3, whose periods are equal to $T_1 = 6, T_2 = 9, T_3 = 12$ respectively. We assume that the worst-case execution times are equal to $C_1 = 1, C_2 = 2, C_3 = 3$. We suppose that τ_1 and τ_2 are tasks sensitive to delay and jitter. Task τ_3 is not sensitive to delay and jitter and can be executed within its period. We assume that for each periodic task $D_i = D_i^{max} = T_i$ and we can pro duce the schedule through the algorithm EDF (see Fig. 2).

Now, for all periodic tasks we can give the response time jitters, which are as follows:$RTJ_1 = 4, RTJ_2 = 1, RTJ3 = 1$. For the data we can use the algorithm given by *P.* Balbastre et al. In the paper [1]. This algorithm reduces the response time jitter for each task by increasing it by 1/3. As was depicted In Fig. 3, the response time jitters for the tasks are as follows: $RTJ_1 = 1, RTJ_2 = 1, RTJ3 = 1$. Unfortunately, as was given by H. Hoang and G.C. Buttazzo In the paper [13], the scheduling algorithm cannot reduce the response time jitter for the some of the tasks.

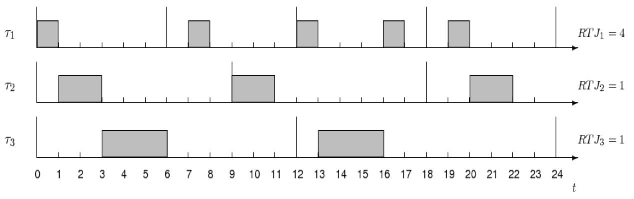

Fig. 2. Example showing the scheduling of three tasks using EDF algorithm

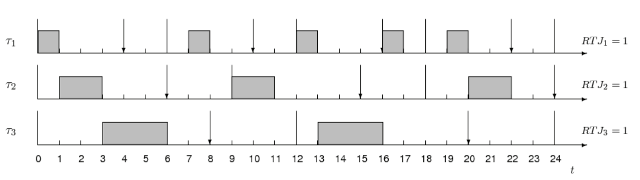

Fig. 3. Example showing the scheduling of three tasks using EDF algorithm with the increasing of deadlines. The arrows show the deadlines of periodic tasks

A scheduling algorithm for reducing the delay and jitter of periodic tasks in real-time systems

In this section, we outline our scheduling algorithm for reducing the delay and jitter of periodic tasks in real-time systems.

The scheduling algorithm is based on the *(m, k)* model [11] in which the multiple streams of tasks are scheduled in real-time. The basic idea of this algorithmis to assign higher priorities to tasks from streams that are closer to a dynamic failure so as to improve their chances of meeting their deadlines.

```
procedure scheduler_of_periodic_tasks;
  begin
   for $i$ = 0 to $n$ do
     begin
       if schedulable($\tau_i$)  then
        while schedulable($\tau_i$) do
        begin
          deadline_reduction($D_i$)
        end;
      else // task $\tau_i$ is not schedulable
       repeat
         deadline_increasing($D_i$);
    end;
end;

function deadline_reduction($D_i$);
  begin
     if $y_i > x_i$ then $x_i := x_i - 1$
else
if ($y_{i =} x_i$) and ($x_i > 0$) then
  $y_i := y_i - 1$;
if ($y_i = 0$) or ($y_i = y_i^{min}$ ) then reset($\beta_i$);
 $\beta_i = y_i / x_i$, $D_i := D_i - \beta_i$
end;

function deadline_increasing($D_i$);
  begin
    if $x_i > 0$ then  $y_i := y_i - 1$
    if ($x_i = y_i$) then $x_i := x_i - 1$;
    else if ($y_i = 0$) and ($x_i > 0$) then
    if ($y_i = 0$) or ($y_i = y_i^{min}$) then
    $x_i := x_{i+}1$
     $\beta_i = y_i / x_i$, $D_i := D_i + \beta_i$
end;
```

Fig. 4. Pseudo-code for reducing the delay and jitter of periodic tasks

The original algorithm guarantees a statistical number of real-time service constraints. The algorithm used here was modified and can guarantee more than its minimum required service. When it is impossible, it then produces a feasible schedule.

The activity of the modified algorithmis as follows: each periodic task τ_i, $(1 \leq i \leq n)$ is associated with window size $W_i = y_i / x_i$, where y_i is the original window-numerator and x_i is the original denominator. We suppose that the window-constraint is equal to the lowest constraint of the worst-case service time of periodic task τ_i with the lowest window-constraint being serviced first. Consequently, the current window-constraint is reduced, when this is possible.

The algorithm for reducing the delay and jitter of periodic tasks operates in detail as follows: for each task is check up the schedule of the task is checked by the EDF scheduling. If this is impossible, the window-constraint is increased by the *deadline_increasing* function. This function is performed so as long as the periodic task is schedulable. The extended deadline is then used for the next periodic task generated by the same source as the previous one. When the window-constraint is too large, the *deadline_reduction* function is performed. It will be used as long as the periodic task is schedulable. Note, the deadlines for periodic tasks are changed and the window-constraint is adjusted every time the periodic task misses a deadline. Each determined window-constraint guarantees the schedulability of the periodic task misses a deadline. Each determined window-constraint guarantees the schedulability of the periodic task. Consequently, the pseudo-code for reducing the delay and jitter of periodic tasks is shown in Fig. 4. We can give the following theorem.

Theorem 1
If a feasible schedule exists, the maximum delay for periodic task $\tau_i\,(1 \leq i \leq n)$ is equal at most to $(y_i + 1)\tau_i - C_i$, where C_i is the execution time for periodic task τ_i, T_i is the period of this task, y_i is the window parameter for each task τ_i.

Proof
When periodic task τ_i misses its deadline, the value of window parameter y_i is reduced by 1 until it reaches 0. If the execution time misses its deadline, it is delayed by T_i time units. Thus is satisfied the following dependency $y_i \leq y_i^{init}$, where y_i^{init} is the initial value of window parameter y_i. The execution of the periodic task can be delayed at most $y_i\ T_i$ time units. If the periodic task is delayed by more then T_i - C_i time units, the window-constraint is violated. The execution of the next subtask in the periodic task will be not complete by the end of its period T_i. In such situations, periodic task τ_i must be delayed by at most $(y_i + 1) \cdot T_i - C_i$ time units.

Example 2
Suppose three periodic tasks which have the following parameters: $C1 = 3$, $W_1 = ¾$, $C_2 = 5$, $W2 = 4/5$, $C_3 = 6$, $W_3 = 1/5$, $T_3 = 8$. Total utilization U is equal to $\sum_{i=1}^{3} \frac{1 - W_i C_i}{T_i} = 1$. This means that the schedulability cannot be realized. However, by using the given algorithm the schedule can be produced.

Suppose for periodic task τ_i the following dependencies:

$$c_i = K, t_i = \left\lfloor \frac{T_i}{C_i} \right\rfloor \cdot K, w_i = \frac{y_i}{x_i},$$

$$\frac{y_i}{x_i} = \frac{T_i \cdot c_i - C_i(1 - W_i)t_i}{T_i \cdot c_i}.$$

If the periodic tasks are divided into subtasks, we can obtain the following parameters:

$c_1 = 1, w_1 = \frac{17}{20}, t_1 = 1,$

$c_2 = 1, w_2 = 1, t_2 = 0, c_3 = 1, w_3 = \frac{2}{3}, t_3 = 1$. This means that for these parameters there is a feasible schedule. Thus, the total utilization equal to U = 0.68 is less than 1.

Simulation results
In order to evaluate the effectiveness of the proposed method for reducing delay and jitter of periodic tasks with respekt to the schedulability analysis, three experiments were performed.

Two periodic task sets were generated for this experiment. The first of these had 8 periodic tasks and the secondo 16 periodic tasks, respectively. Both sets had the the total utilization $U_d < 1$. It was assumed that $\sum_{i=1}^{n} U_i = U_d$.

The worst-case execution time C_i was generated by using the random number generator with uniform distri-bution from the interval [C_{min}, C_{max}], where C_{min} = 1, C_{max} = 20. The period of each periodic task τ_i was obtained from the dependency

$$T_i = \frac{C_i}{U_i}$$

It was assumed that the deadline $D_i^{\max}$ is equal to T_i and $D_i^{\max}$. For each periodic task was determined the maximum response time $R_i = \max_k\{R_{i,k}\}$ and the maximum response time jitter $RTJ_i = \max_k\{RTJ_{i,k}\}$.

The results of the simulation using of our algorithm were averaged for 1000 runs of simulation runs. In experimental ways was determined the graph of maximum response time jitter depending on the number of periodic tasks. By way of comparison, we performed a simulation for the same task sets using the EDF scheduling algorithm and computing the maximum response time jitter. The graphs for both simulations are given in Fig. 5 and 6.

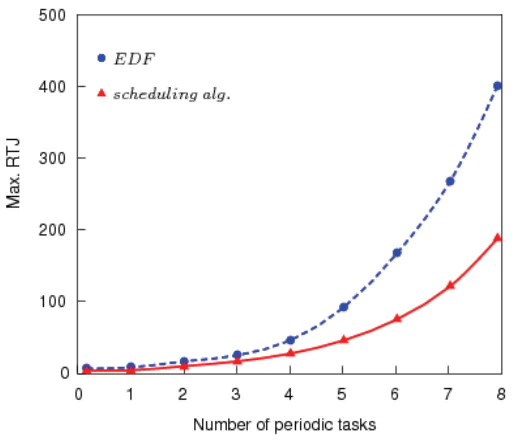

Fig. 5. Maximum Response Time Jitter (Max. RTJ) when applying the proposed algorithm to first set of periodic tasks (U_d = 0.8)

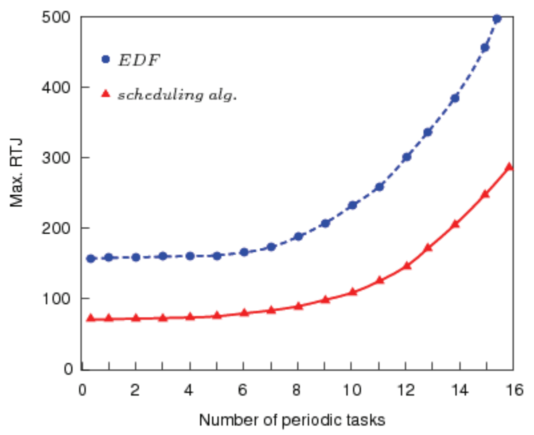

Fig. 6. Maximum Response Time Jitter (Max. RTJ) when applying the proposed algorithm to second set of periodic tasks (U_d = 0.6)

We observed that the proposed algorithm for reducing the delay and jitter of the periodic tasks in comparison with the EDF scheduling algorithm decreases the maximum response time jitter. This result is especially visible in the case of the greater task set of periodic tasks (see Fig. 5).

Thus, the maximum response time jitter is ca. 100% lower.

The difference in results between our algorithm and the EDF scheduling algorithm indicates that the proposed algorithm gives ca. 100% better performance than the EDF scheduling algorithm.

As shown in Figures 5 and 6 our algorithm is more effective when the task set has a high utilization. This confirms that our algorithm can guarantee the timing constraints for a larger number of periodic tasks.

Conclusion

In this paper, we have presented an algorithm for reducing the jitter and delay of all periodic tasks. The proposed algorithm can be implemented "on-line" in control software and thereby becomes efficient in use. By increasing the deadline of periodic tasks such as this it become possible to maintain the schedulability of all tasks, and we obtain the guarantee that the system will not lose its stability.

In our future work we will implement our algorithm in the real-time operating system. This will make it possible to estimate the effectiveness of the control software and provide the values for all parameters for the used algorithm. Additionally, we will test the sensitivity analysis of the real-time system with the proposed algorithm.

We also plan to apply the proposed algorithm in controlling the delay and jitter in software controlled systems for delay minimization in the control systems. It is also possibile to consider control systems that are able to do a tradeoff between the available computation time, i.e. how long time the system may spend calculating the new parameters of scheduling algorithm and the control loop performance.

REFERENCES

[1] B a l b a s t r e P., R i p o l l I., C r e s p o A.,: Optimal Deadline Assignment for Periodic Real-Time Dynamic Priority Systems, *Proc. of of the 18th Euromicro Conference on Real-Time Systems*, Dresden, Germany, pp. 65 – 74, 2006.
[2] B a r u a h S., B u t t a z z o G.C., G o r i n s k y S., L i p a r i G., Scheduling Periodic Task Systems to Minimize Output Jitter, *Proc. of the 6th IEEE Int. Conference on Real-Time Computing Systems and Applications*, Hong Kong, pp. 62 – 69, Dec. 1999.
[3] B i n i E., B u t t a z z o G.C., Schedulability Analysis of Periodic Fixed Priority Systems, *IEEE Trans. on Computers*, 53(11), pp. 1462 – 1473, 2004.
[4] B i n i E., D i N a t a l e M., Optimal Task Rate Selection In Fixed Priority Systems, in: *Proc. of the 26th IEEE Real-Time Systems Symposium*, Miami, USA, 399 - 409, 2005.
[5] B i n i E., D i N a t a l e M., B u t t a z z o G., Sensitivity Analysis for Fixed-Priority Real-Time Systems, *Real-Time Systems, 39,* pp. 5 – 30, 2008.
[6] B u t t a z z o G.C., S e n s i n i F., Optimal Deadline Assignment for Scheduling Soft Aperiodic Task in Hard Real-Time Environments, *IEEE Trans. on Computers*, 48(10), pp. 1035 – 1052, 1999.
[7] B u t t a z z o G.C., C e r v i n A., Comparative Assessment and Evaluation of Jitter Control Method, in*: Proc. of the 15th Int. Conf. on Real-Time and Network Systems*, Nancy, France, March 29 - 30, 2007.
[8] C e r v i n A., Integrated Control and Real-Time Scheduling, Ph. D. Thesis, ISRN LUTFD2/TFRT-1065-SE, Dept. Of Automatic Control, Lund, Sweden, 2003.
[9] C e r v i n A., L i n c o l n B., E k e r J., A r z n K.-E., B u t t a z z o G.C., The Jitter Margin and Its Application in the Design of Real-Time Control Systems, *Proc. of the 10th Int. Conf. on Real-Time and Embedded Computing Systems and Applications* (RTCSA 2004), Gotheborg, Sweden, Aug. 25 – 27, pp. 1 – 9, 2004.
[10] D i N a t a l e M., S t a n k o v i c J., Scheduling Distributed Real-Time Tasks with Minimum Jitter, *IEEE Trans. on Computers*, 49(4), pp. 303 – 316, 2000.
[11] Hamdaoui M., Ramanathan P., A Dynamic Priority Assignment Technique for Streams with (m, k)-Firm Guarantees, *IEEE Trans. on Computers*, 44(12), pp. 1443 – 1451, 1995.
[12] H o a n g H., B u t t a z z o G.C., J o n s s o n M., arlsson S., Computing the Minimum EDF Feasible Deadline in Periodic Systems, *Proc. of the 12th IEEE Int. Conference on Embedded and Real-Time Computing Systems and Applications*, Sydney, Australia, pp. 125 – 134, Aug. 2006.
[13] H o a n g H., B u t t a z z o G.C., Reducing Delay and Jitter in Software Control Systems*, Proc. of the 15th Int. Conference on Real-Time and Network Systems*, Nancy, March 29 - 30, 2007.
[14] K w i e c i e ń A., S i d z i n a M., Some Methods for Reduction of Cycle in Free-programmable Controllers and Their Basic Studies, *Real-Time Systems*, 2, Warszawa: Wydawnictwa Komunikacji i Łączności, pp. 85 – 98, 2005 (In Polish).
[15] L i u C.L., L a y l a n d J.W., Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment, *Journal of the Association for Computing Machinery*, 20(1), pp. 46 – 61, 1973.
[16] M a r t i P., Analysis and Design of Real-Time Control Systems with Varying Control Timing Constraints, Ph. D. Thesis, Dept. of Automatic Control, Technical University, Barcelona, Spain, 2002.
[17] M i c h e l o n Ph., Q u a d r i D., N e g r e i r o s M., On a Class of Periodic Scheduling Problems: Models, Lower Bounds and Heuristics, *Proc. of the Int. Multiconference on Computer Science and Information Technology*, 3, pp. 899 – 906, 2008.
[18] N i l s s o n J., B e r n h a r d s s o n B., W i t t e n m a r k B., Stochastic Analysis and Control of Real-Time Systems with Random Time Delays, *Automatica*, 34(1), pp. 57 – 64, 1998.
[19] S h i n I., L e e I., Compositional Real-Time Scheduling Framework with Periodic Model, *ACM Trans. on Embedded Computing Systems*, 7(3), pp. 30:1 – 30:39, 2008.

***Author**: Ph. D. Jerzy Martyna, Institute of Computer Science, The Faculty of Mathematics and Computer Science, Jagiellonian University, ul. Prof. S. Łojasiewicza 6, 30-348 Kraków, Poland*
E-mail: martyna@ii.uj.edu.pl;