Relaxing Event Densities by Lower Bounds on Event Streams

Steffen Kollmann, Karsten Albers and Frank Slomka
Embedded Systems / Real-Time Systems
University Ulm
{first name.second name}@uni-ulm.de

***Abstract*—The regular execution of tasks - e.g. sensor data acquisition - is common in embedded systems. So it is possible to determine not only a minimal distance in which events can occur, but also a maximum distance between events. In this paper we will show for fixed-priority systems how it is possible to use a lower bound of stimulation to improve the minimal distance between events. Taking this into account will relax the worst case response times of tasks in distributed real-time systems, leading to a more accurate schedulability analysis.**

I. Introduction

For some embedded systems it is necessary to interact with its environment in specific time intervals. Sensor data acquisition or control circuits are examples for such cases. So it is possible to determine not only a maximum density of stimulation in such systems, but also a minimal density.

In figure 1 it can be seen that the consideration of minimal distances can lead to an improvement of the density of outgoing events. The CPU has got the tasks τ_x and τ_y. We assume that τ_x has a strict periodic execution. In the right part of the figure the improvement of the distance of two events of τ_y can be observed. In the upper part the minimal stimulation is not considered, as usual in real-time analysis. In the lower part the minimal stimulation is considered. This leads to a relaxed outgoing event stream.

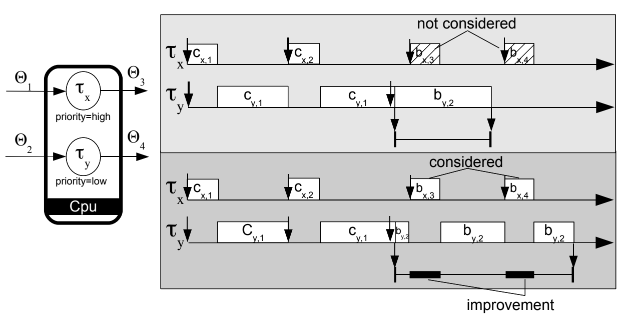

Fig. 1. Improvement of the event density by considering minimal event streams

The rest of the paper is organized as follows: In section II we give a short overview about related work in this area. The model is explained in section III. Section IV explains the calculation of event streams in distributed systems by considering minimal stimulation. An example is given in section V. Finally a conclusion follows.

II. Related Work

Some models consider lower bounds of event sequences that can occur in a system. Such models are for example the real-time calculus [7] or the periodic task model with jitter [5]. But only a very few contributions make use of these bounds. The transaction model [6], for example, does not use the lower bounds.

The real-time calculus defines no possibility to consider the best case execution time of the task under analysis in order to calculate the upper request curve. Only the lower service curve includes the minimal occurrence of the higher priority tasks. This lack is founded by the fact, that the real-time calculus can not distinct between best case and worst case execution times.

Redell, for example, shows in [4] how the calculation of a best case response time can be obtained when lower bounds of stimulations are considered. In this paper we also exploit the lower bound of the stimulations in order to improve the maximum density of events in a system. For this, we will adapt Redell's approach to the event stream model [2] and extend it in order to improve the calculation of event streams in distributed systems.

III. Model

In this section we introduce our models. We differentiate between the task model and the model for the stimulation.

A. Task Model

Γ is the set of tasks on one resource $\Gamma = \{\tau_1, ..., \tau_n\}$. A task $\tau = (c, b, d, \rho, \overline{\Theta}, \underline{\Theta})$. Where c is the worst case execution time, b is the best case execution time, d is the deadline, ρ is the priority for the scheduling (the lower the number the higher the priority), $\overline{\Theta}$ defines the maximum stimulation (maximum density of events) and $\underline{\Theta}$ the minimum stimulation (minimum density of events). Let τ_{ij} be the j-th job/execution of task τ_i.

In our model we assume that a task can only generate an event at the end of its execution to notify other tasks. Furthermore we assume a fixed-priority scheduling.

B. Maximum Event Streams

Event streams have been first defined in [2]. The purpose was to give a generalized description for every kind of stimuli. The basic idea is to define an event function $E(I, \Theta)$ which can calculate for every interval I the maximum amount of events occurring within I. In the following, when speaking of intervals we mean the length of the interval. The event function needs a properly described model behind it which makes it easy to extract the information. The idea is to notate for each number of events the minimum interval which can include this number of events. Therefore we get an interval for one event (which is infinitely small and therefore considered to be zero), two events and so on. The result is a sequence of intervals showing a non-decreasing behaviour. The reason for this behaviour is, that the minimum interval for n events cannot be smaller than the minimum interval for n-1 events since the first interval also includes n-1 events. This sequence of intervals shows a periodic behaviour and is called event stream. Each of the single intervals is called event stream element.

Definition 1: *A maximum event stream is a set of event stream elements* $\theta{:}\overline{\Theta}=\{\theta_1,\theta_2,...,\theta_n\}$ *and each event stream element* $\theta=(p,a)$ *consists of an offset-interval* a *and a period* p*. The maximum event stream complies the characteristic* $E(I_1+I_2,\overline{\Theta})\leq E(I_1,\overline{\Theta})+E(I_2,\overline{\Theta})$.

The characteristic of the maximum event stream is called sub-additivity. This means that the maximum number of events of an interval cannot exceed the cumulated maximum number of events of its subintervals.

Each event stream element describes a set of intervals of the sequence. For the event stream element θ the interval $a+k\cdot p$ is part of the sequence and all the intervals with $k\in\mathbb{N}$. An event stream models a given sequence if all the elements and only the elements of the sequence can be generated using the event stream elements. Therefore it is possible to calculate for each interval the maximum amount of events that can occur within this interval:

Event Stream Function:

$$E(I,\Theta)=\sum_{\theta\in\Theta}E(I,\theta) \quad ; \quad E(I,\theta)=\begin{cases}0 & I<a_\theta\\ \left\lfloor\frac{I-a_\theta}{p_\theta}+1\right\rfloor & I\geq a_\theta\wedge p_\theta<\infty\\ 1 & I\geq a_\theta\wedge p_\theta=\infty\end{cases} \tag{1}$$

As inverse function we define the following function which gives to a number of events the minimum interval in which these events can occur:

Request Time Function:

$$RT(n,\Theta)=min\{I|E(I,\Theta)=n\} \tag{2}$$

With an infinite (∞) period it is possible to model irregular behaviour. A detailed definition of the concept and the mathematical foundation can be found in [1].

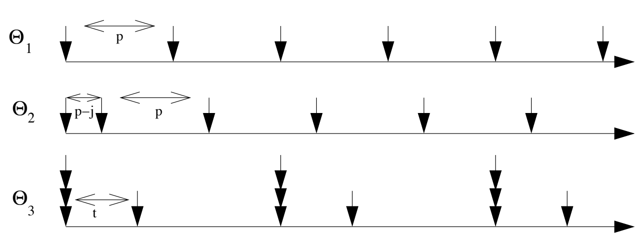

Fig. 2. This figure shows three different event sequences

In figure 2 some examples for event streams can be found. The first one $\overline{\Theta}_1$ = (p,0) has a strictly periodic stimulus with a period p. The second example $\overline{\Theta}_2$ = (∞,0), (p,p-j) shows a periodic stimulus in which the single events can jitter within a jitter interval of size j. In the third example $\overline{\Theta}_3$ = (p,0), (p,0) , (p,0), (p,t) three events occur at the same time and the fourth occurs after a time t. This pattern is repeated with a period of p. Event streams can describe all these examples in an easy and intuitive way.

C. Minimal Event Streams

Analog we define the minimal event streams which describe for every Interval I the minimum stimulation in such an interval.

Definition 2: *A minimum event stream is a set of event stream elements* $\theta{:}\underline{\Theta}=\{\theta_1,\theta_2,...,\theta_n\}$ *and each event stream element* $\theta=(p,a)$ *consists of an offset-interval* a *and a period* p*. The minimum event stream complies the characteristic* $E(I_1+I_2,\underline{\Theta})\geq E(I_1,\underline{\Theta})+E(I_2,\underline{\Theta})$.

The characteristic of the minimum event stream is called super-additivity. This means that the maximum number of events of an interval can exceed the cumulated maximum number of events of its subintervals.

Anymore, for the minimal event stream applies the following lemma.

Lemma 1: *For a minimum event stream aperiodic events occurring independently of the remaining event stream can be ignored.*

Proof: Let us assume that an aperiodic event (∞,a) exists in a minimal event stream $\underline{\Theta}$ and this event is the last aperiodic event:

$$\begin{aligned} & (\exists\underline{\Theta});(\exists\theta=(\infty,a))|(\theta\in\underline{\Theta}\wedge\neg\theta'\in\underline{\Theta}{:}a_\theta<a_{\theta'}\wedge p_{\theta'}=\infty)\\ \Rightarrow\quad & (\exists I_1\in\mathbb{R});(\exists I_2\in\mathbb{R})|(I_1>a_\theta\wedge I_2=I_1+I_1)\\ \Rightarrow\quad & E(I_1)>E(I_2)-E(I_1)\\ \Leftrightarrow\quad & E(I_1+I_1)<E(I_1)+E(I_1)\end{aligned}$$

Which is a contradiction to the assumption. Since there is no last aperiodic event in a minimal event stream, it follows that no aperiodic event exist in a minimal event stream. □

The examples in figure 2 can be described by the following minimal event streams: The first one $\underline{\Theta}_1$ = (p,p). The second example $\underline{\Theta}_2$ = (p,p+j). In the third example $\underline{\Theta}_3$ = (p,p-t), (p,p), (p,p), (p,p) .

IV. Improved Density by Minimal Event Streams

We have introduced a task model and a model for the stimulation. With these models we will show how it is possible to determine the stimulation density in the whole system. For this we have to determine when the worst case occurs.

Lemma 2: *A number of outgoing events occur in the maximum density when the first event is delayed as much as possible and all further events occur as early as possible.*

Proof: We assume that two outgoing events e_1 and e_2 exist having a higher density than the events fulfilling the assumption. If e_1 and e_2 are closer together than in the assumption, this would mean either e_1 arrives later than allowed by the assumption or e_2 arrives earlier than allowed by the assumption. This is a contradiction, because we assume already the maximum or minimum values for both arrival times. So there must be two other events later in the outgoing event stream having a shorter distance to each other. Assume that two events are occurring closer than in the assumption and the first event is delayed as much as possible and the second arrives as early as possible, this would mean that the corresponding incoming events also have a shorter distance to each other than the first two incoming events. But this is in contradiction to the event stream definition. The proof for another number of events is analog. □

For the calculation we need the worst case response time which determines the maximum delay of an event. Since we have minimal event streams it is also possible to determine a best case response time. So we first define the methodology in order to determine these two response times.

A. Worst Case Response Time

The most usual way to do a real-time analysis is to perform a response time analysis as introduced by Lehoczky et. al. [3]. The condition $\forall \tau \in \Gamma : WCRT_k(\tau) \leq d_\tau$ holds when the real-time analysis is successful. In order to calculate the worst case response time we have adapted the approach from [3].

$$WCRT_k(\tau)=min\{I|I=k\cdot c_\tau+\textstyle\sum_{\tau'\in HP} E(I,\overline{\Theta}_{\tau'})\cdot c_{\tau'}\} \tag{3}$$

The equation is similar to the common definition of the worst case response time. Only the calculation of the influence of higher priority events has been changed. The amount of execution produced by higher priority tasks can be calculated by the event function multiplied by the worst case execution time. By means of a fixed point iteration the worst case response time can be calculated for every k.

B. Best Case Response Time

Additionally to the worst case response time it is possible to determine a best case response time, since we have minimal event streams. For this we have adapted the best case response time from Redell [4].

$$BCRT(\tau)=min\{I|I=b_\tau+\textstyle\sum_{\tau'\in HP} E(I,\underline{\Theta}_{\tau'})\cdot b_{\tau'}\} \tag{4}$$

The equation adds to the best case execution time of task τ the best case execution time of the higher priority tasks. How many execution times are added depends on the minimal event streams of the higher priority tasks. As well as the worst case response time, it is possible to find the best case response time by a fix-point iteration. For a detailed description see [4].

C. Calculation of Outgoing Event Streams

For the calculation of the density of the outgoing events we define an interval function. This function gives for an amount of events the minimum interval in which they can occur. We call it interval function and define it as follows:

$$I(n,\tau)=\begin{cases} 0 & n=1 \\ RET(n,\tau)-RET(1,\tau) & n>1 \end{cases} \tag{5}$$

$$RET(n,\tau)=\begin{cases} WCRT_1(\tau) & n=1 \\ BC(\tau,RT(n,\tau),RET(n-1,\tau),WCRT_1(\tau),BCRT(\tau),\Gamma^{\tau}_{HP}) & n>1 \end{cases} \tag{6}$$

According to the event stream definition one event occurs always in the interval zero. Hence, we distinguish in the equation 5 between two cases. The first case describes the interval for one event which is always zero according to the event stream definition. All other events are covered by the second case via the **R**equest **E**nd **T**imes.

```
1  BC(τ, RT_n, RET_{n−1}, WCRT_τ, BCRT_τ, Γ_HP) {
2      CSTART = max(RT_n, RET_{n−1});
3      RET_new = CSTART + BCRT_τ;
4      while (true) {
5          B = b_τ;
6          for(∀τ' ∈ Γ_HP) {
7              ΔI = RET_new − (WCRT_τ − c_τ');
8              B = B + E(ΔI, Θ_τ') · b_τ';
9          }
10         ΔJ = RET_new − WCRT_τ;
11         If(B > ΔJ ) {
12             RET_new = WCRT_τ + B;
13         }
14         else {
15             return RET_new;
16 }   }   }
```

Fig. 3. Calculation of the improved Best Case Response Time

In order to explain the calculation of the events greater than one we use the figure 4.

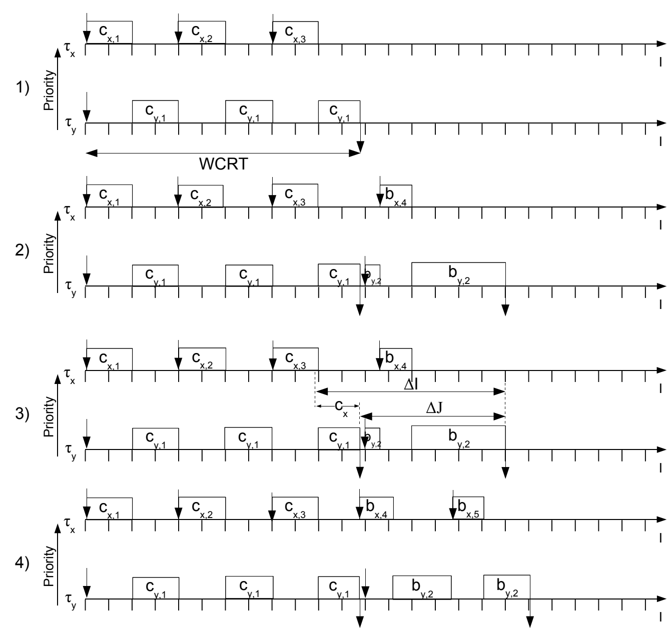

Fig. 4. Improvement of the event density considering minimal event streams

According to lemma 2 the first event is delayed as much as possible. This delay can be determined by the worst case response time of the first job. So the first calculation is the WCRT of instance one like in figure 4 (part 1). So the case for n=1 in equation 6 calculates the worst case response time.

The next events must occur as soon as possible. This happens when the task runs with its best case execution time and the job runs as soon as possible. For this calculation we use the algorithm depicted in figure 3. In line 2 we determine when the calculation can start. From this point in time we add the best case response time determined by Redell's approach (line 3). This can be seen in figure 4 (part 2).

The next step is to determine, whether more interrupts can occur from higher priority tasks or not. So we determine for every higher priority task an interval (ΔI) from the last possible stimulation of the task in the worst case response time up to the end of the best case response time (see figure 4 part 3). This is done in line 7.

Line 8 determines the absolute demand of execution of one task within the interval. If the execution demand of all tasks is greater than the interval ΔJ, the best case response time will be more relaxed (line 12). See figure 4 (part 4). Otherwise the best case response time will be not changed (line 15). This step must be repeated until the best case response time is unchanged. This is done by the while-loop which is equal to a fix-point iteration.

Is the request end time of the n-th event determined, the minimal interval for n events can be determined by the request end time of the n-th event minus the request end time of the first event (see equation 5).

V. Example

In order to show the significance of our approach we show by a short example the improvement of the density of events in a distributed system. Figure 5 shows this example. We calculate the density of events for $\underline{\Theta}_F$ and show the impact on the response time via task τ_4.

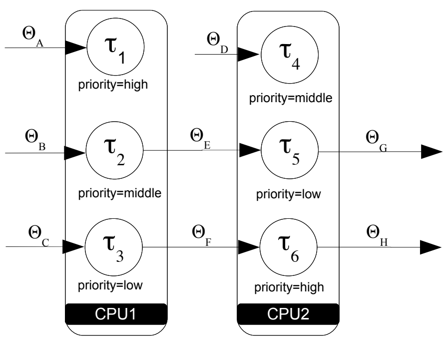

Fig. 5. Example of a distributed system

The next table describes the properties of the distributed system.

CPU1	τ_1	τ_2	τ_3	CPU2	τ_4	τ_5	τ_6
c	4	4	14	c	31	2	9
b	4	4	13	b	15	1	5
d	40	50	50	d	55	60	40
ρ	1	2	3	ρ	2	3	1
$\overline{\Theta}$	$\overline{\Theta}_A$	$\overline{\Theta}_B$	$\overline{\Theta}_C$	$\overline{\Theta}$	$\overline{\Theta}_D$	$\overline{\Theta}_E$	$\overline{\Theta}_F$
$\underline{\Theta}$	$\underline{\Theta}_A$	$\underline{\Theta}_B$	$\underline{\Theta}_C$	$\underline{\Theta}$	$\underline{\Theta}_D$	$\underline{\Theta}_E$	$\underline{\Theta}_F$

TABLE I

Parameters of the distributed System which is depicted in figure 5

The maximum event streams are: $\overline{\Theta}_A = \{(12, 0)\}$, $\overline{\Theta}_B = \{(12, 0)\}$, $\overline{\Theta}_C = \{(30, 0)\}$ and $\overline{\Theta}_D = \{(70, 0)\}$.

The minimum event streams are: $\underline{\Theta}_A = \{(12, 12)\}$, $\underline{\Theta}_B = \{(12, 12)\}$, $\underline{\Theta}_C = \{(30, 30)\}$ and $\underline{\Theta}_D = \{(70, 70)\}$.

We have calculated the minimal intervals of the first five events of $\overline{\Theta}_F$ to show the improvement of the approach. This can be seen in table II where we have calculated the densities with approach, without approach and with Redell's approach. The table III shows the three different event streams of $\overline{\Theta}_F$.

n	$\overline{\Theta}_{F1}$	$\overline{\Theta}_{F2}$	$Impr.$	$\overline{\Theta}_{F1}$	$\overline{\Theta}_{F3}$	$Impr.$
1	0	0	0%	0	0	0%
2	29	21	27,58%	29	13	55,17%
3	50	42	16%	50	27	46%
4	71	65	8,45%	71	57	19,71%
5	95	95	0%	95	87	8,42%

TABLE II

Shows the improvement of the approach on the event streams. $\overline{\Theta}_{F1}$ shows the intervals with the new approach, $\overline{\Theta}_{F2}$ shows the intervals with Redell's approach and $\overline{\Theta}_{F3}$ without any approach. The Improvement is given in %

$\overline{\Theta}_{F1} =$	{(∞,0),(∞,29),(∞,50),(∞,71),(30,95)}
$\overline{\Theta}_{F2} =$	{(∞,0),(∞,21),(,42),(30,65)}
$\overline{\Theta}_{F3} =$	{(∞,0),(∞,13),(30,27)}

TABLE III

Results of the event streams with the different approaches.

So we are able to calculate the worst case response time of tasks τ_4. The response time without any approach is 67 t.u., with Redell's approach 58 t.u. and with the new approach 49 t.u. This leads to an improvement of 15,51% against Redell's approach and to 26,86% against without any approach.

VI. Conclusion

In this paper we have shown how to use lower bounds of stimulation in order to improve the real-time analysis of distributed systems. We have shown how the approach of Redell [4] can be adapted and extended in order to improve the calculation of event sequences. Furthermore we have shown that this leads directly to more realistic response times in the system. In the future we would like to develop an efficient approach to calculate the maximum and minimum event streams in the systems. Additionally, we will extend the introduced approach further so we obtain tighter bounds in the analysis. Another approach is to determine the real occurrence of the last events of the higher priority tasks during the worst case execution time. This would lead to a greater interval ΔI. An extension to dynamic scheduling is also an aim.

References

[1] Karsten Albers and Frank Slomka. An event stream driven approximation for the analysis of real- time systems. In *ECRTS '04: Proceedings of the 16th Euromicro Conference on Real-Time Systems*, pages 187–195. IEEE, July 2004.

[2] Klaus Gresser. An event model for deadline verification of hard real-time systems. In *Proceedings of the 5th Euromicro Workshop on Real-Time Systems*, 1993.

[3] John P Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *Proceedings of the 11th IEEE Real-Time Systems Symposium*, pages 201–209, December 1990.

[4] Ola Redell and Martin Sanfridson. Exact best-case response time analysis of fixed priority scheduled tasks. In *ECRTS '02: Proceedings of the 14th Euromicro Conference on Real-Time Systems*, page 165, Washington, DC, USA, 2002. IEEE Computer Society.

[5] Kai Richter. *Compositional Scheduling Analysis Using Standard Event Models - The SymTA/S Approach.* PhD thesis, University of Braunschweig, 2005.

[6] Ken Tindell. Adding time-offsets to schedulability analysis. Technical report, University of York, Computer Science Dept, YCS-94-221, 1994.

[7] Ernesto Wandeler. *Modular Performance Analysis and Interface-Based Design for Embedded Real-Time Systems.* PhD thesis, ETH Zurich, September 2006.