Grundlagen der Echtzeitplanung

1. Grundlegende Begriffe und Konzepte

2. Planungsverfahren (Scheduling)

   2.1 Planen aperiodischer Tasks
   - Planen durch Suchen
   - Planen nach Fristen
   - Planen nach Spielräumen

   2.2 Planen periodischer Tasks
   - **Planen nach monotonen Raten**
   - **Deadline Monotonic**
   - **EDF**

Folgende Annahmen gelten:

A1: Die Instanzen einer Task T_i werden mit einer konstanten Rate in regelmäßigen Abständen aktiviert. Das Intervall zwischen zwei Aktivierungen wird als *Periode* p_i bezeichnet.

A2: Alle Instanzen einer periodischen Task T_i haben dieselbe WCET Δe_1

A3: Alle Instanzen einer periodischen Task T_i haben dieselbe relative Deadline D_i, die der Periode von T_i entspricht.

A4: Alle Tasks sind unabhängig.

Reaktionszeit: Zeit von der Aktivierung bis zum Abschluß

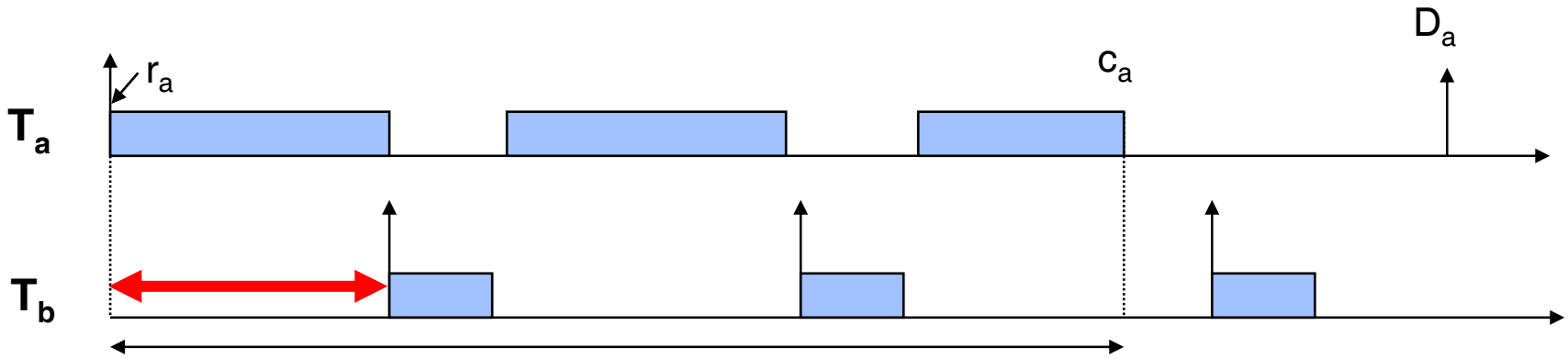
$$R_{i,k} = c_{i,k} - r_{i,k}$$

Kritische Instanz: Der Zeitpunkt an dem die Aktivierung einer Task die größte Reaktionzeit bedingt

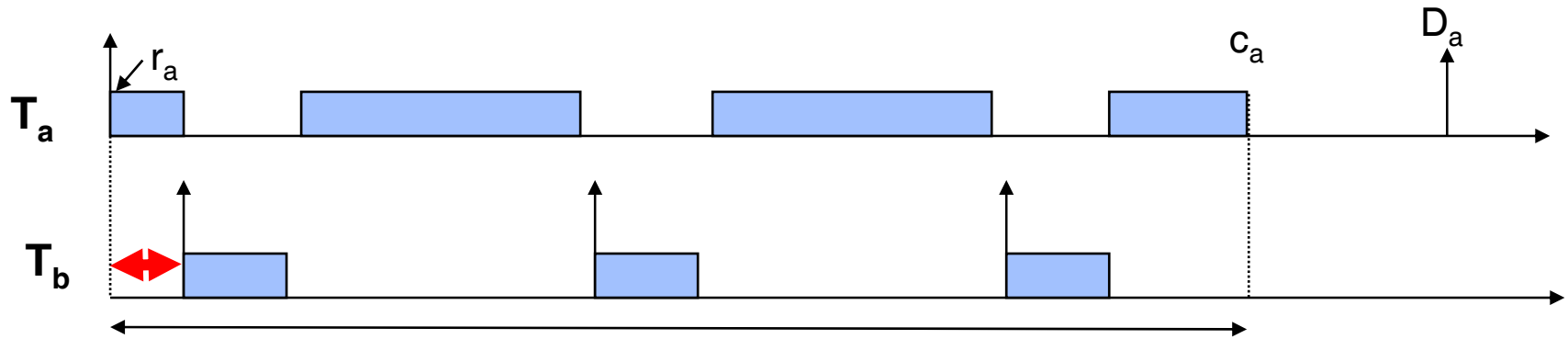
Kritisches Intervall: Das Intervall zwischen der kritische Instanz und der Abschlußzeit der entsprechenden Task

- Eine Task ist planbar, wenn alle ihre Instanzen vor der Deadline fertig werden.
- Eine Taskmenge heißt planbar, wenn alle Tasks planbar sind.

Kritische Instanz und kritisches Intervall



kritisches Intervall: $\Delta e_a + 2\ \Delta e_b$



kritisches Intervall: $\Delta e_a + 3\ \Delta e_b$

Die Antwortzeit von Ta wird durch die höher priorisierte Task Tb verlängert und zwar je mehr, je häufiger Tb während der Ausführungszeit von Ta aktiviert wird.

Das Maximum der Aktivierungen wird erreicht, wenn Ta und Tb zum selben Zeitpunkt aktiviert werden.

Prozessorauslastung "U"

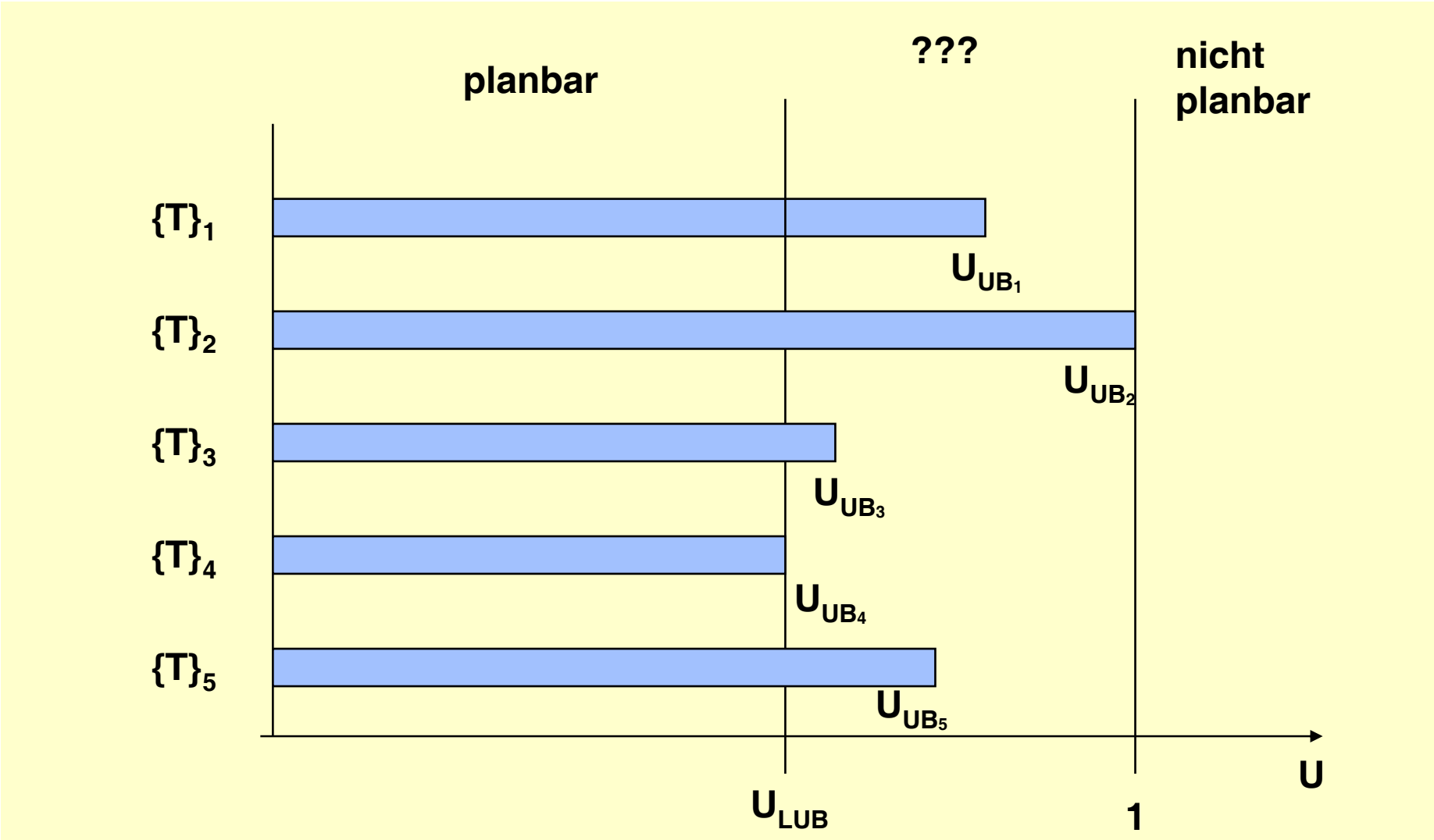
Gegeben sei die Menge periodischer Tasks T_1, T_2, .., T_n ,

$\Delta e_i / \Delta p_i$ ist die Zeit, die Task T_1 in der Periode Δp_i zur Ausführung nutzt.

Der Auslastungsfaktor für n Tasks ist dann gegeben durch:

$$U = \sum_{(i=1,...,n)} (\Delta e_i / \Delta p_i)$$

Sei U_{UB} ({T}, A) die obere Schranke des Auslastungsfaktors für die Menge der Tasks {T}, die von dem Schedulingalgorithmus A eingeplant wird.
Wenn $U = U_{UB}$ ({T},A), dann ist der Prozessor voll ausgelastet.



Die kleinste obere Schranke U_{LUB} (A) für die eine Taskmenge unter einem Schedulingverfahren planbar ist, ist gegeben durch:

$$U_{LUB}(A) = \min U_{UB}(\{T\}, A) \text{ über alle Taskmengen } \{T\}.$$

Satz: **Wenn der Auslastungsfaktor U >1 ist, kann eine Taskmenge von keinem Planungsalgorithmus eingeplant werden.**

Sei $\Delta P = \Delta p_1 \bullet \Delta p_2 \bullet \Delta p_3 \bullet ..... \bullet \Delta p_n$ (gemeinsames Vielfaches !)

Aus $U > 1$ folgt $U \cdot \Delta P > \Delta P$

$$\sum_{(i=1,...,n)} (\Delta P / \Delta p_i \cdot \Delta e_i) > \Delta P$$

mit $\Delta P / \Delta p_i$: Anzahl der Aktivierungen von T_i im Intervall ΔP
und $\Delta P / \Delta p_i \cdot \Delta e_i$:Ausführungszeit von T_i im Intervall ΔP

Daraus folgt, daß $\sum_{(i=1,...,n)} ((\Delta P / \Delta p_i) \cdot \Delta e_i)$ die Gesamtrechenzeit aller Tasks ist, die im Intervall ΔP benötigt wird. Diese kann nicht größer sein als die zur Verfügung stehende Gesamtzeit ΔP. Daher gilt, daß eine Menge von Tasks nur dann planbar ist, wenn $U \leq 1$ gilt.

Prioritätsbasiertes statisches Scheduling

Es wird kein expliziter Plan aufgestellt, der (zeitbasiert) auf Fristen oder Spielräumen beruht, sondern es existiert ein impliziter Plan, der durch eine Prioritätszuordnung repräsentiert wird.

Praxixbezug: **Die meisten Echtzeitbetriebssystem(kern)e unterstützen prioritätsbasiertes Scheduling mit unterbrechbaren Prozessen.**

Zur Entscheidung, welchem Prozeß dem Prozessor zugeteilt wird, braucht man lediglich die Priorität des gerade laufenden Prozesses mit der Priorität des gerade bereitwerdenden Prozesses zu vergleichen.

Nach welchen Gesichtspunkten wird einem Prozeß seine Priorität zugeordnet ?

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

Annahmen:

1. Eine Task kann zu einem beliebigen Zeitpunkt unterbrochen werden.

2. Es wird nur die Ressource "Prozessorzeit" betrachtet.

3. Alle Tasks sind unabhängig und es gibt keine Vorrangrelation unter Tasks.

4. Es werden ausschließlich unterbrechbare periodische Tasks betrachtet.

5. Die relativen Deadlines der Tasks entsprechen ihren Perioden.

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

Def.:
Rate einer periodischen Task = Anzahl der Perioden im Betrachtungszeitraum
= Frequenz (über unbegrenzte Zeitraum)

Prioritätsordnung:

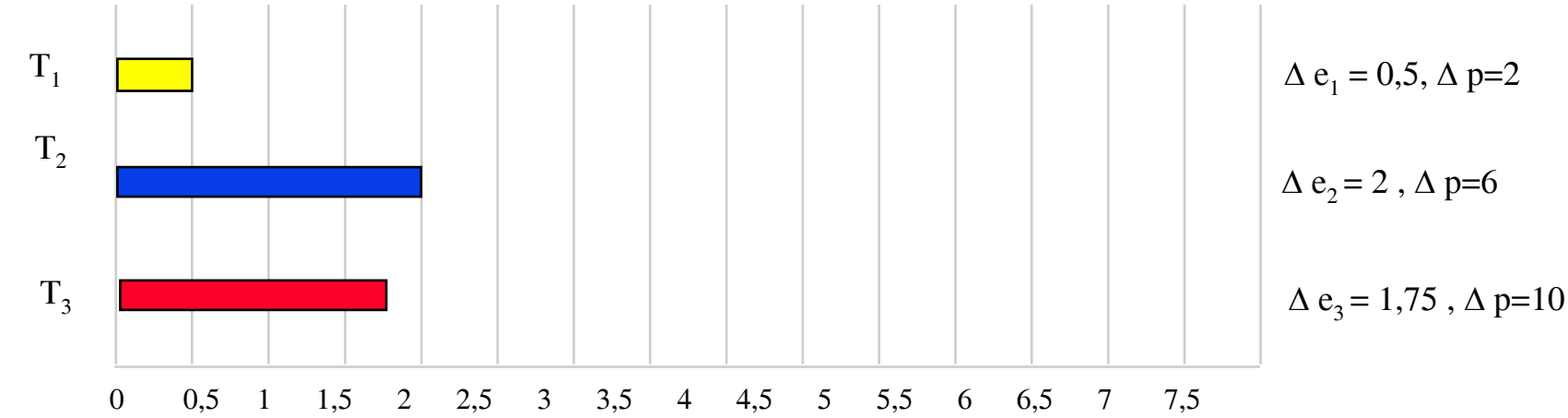
$$rms(i) < rms(j) \Leftrightarrow 1 / \Delta p_i < 1 / \Delta p_j$$

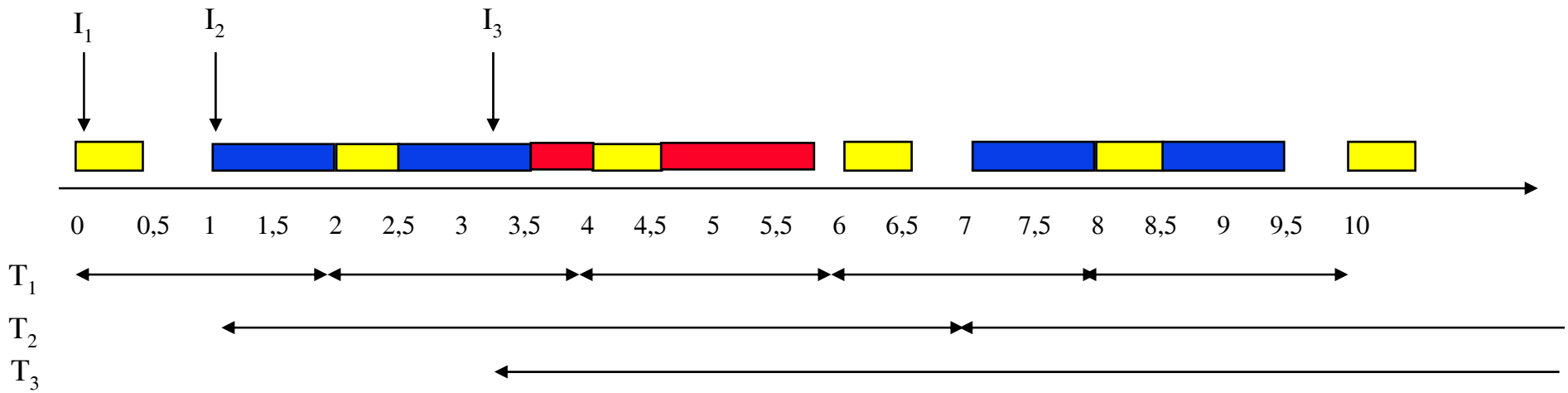
Nummerierung der Tasks gemäß ihrer Priorität:

$$i < j \Leftrightarrow rms(i) < rms(j)$$

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

Beispiel:





Eigenschaften von RMS :

Frage: Ist RMS optimal verglichen mit anderen statischen Planungsverfahren?

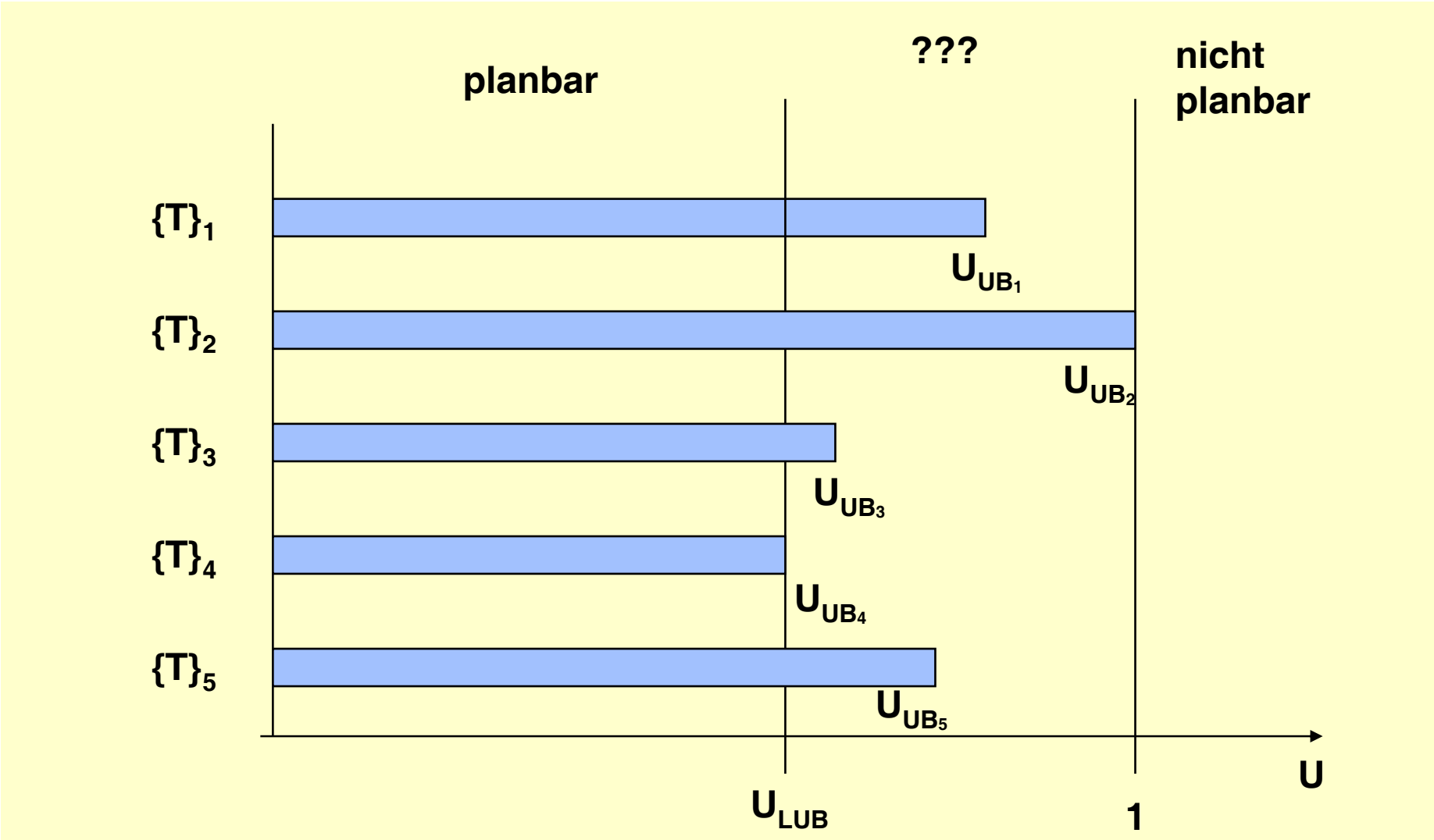
Intuitive Kritik: RMS berücksichtigt keine Deadlines, sondern zum aktuellen Zeitpunkt wird die Task mit der statisch höchsten Priorität bevorzugt.

Frage: Gibt es eine obere Schranke U_{lub} der Prozessorauslastung, für die immer ein Plan nach RMS garantiert werden kann (d.h. ein hinreichendes Kriterium für die Einplanbarkeit) ?

U_{lub} ist die Auslastung, für die RMS optimal ist, d.h. einen Plan findet, wenn überhaupt einer existiert. Es kann natürlich Verfahren geben, die eine bessere Auslastung realisieren.

Für n Tasks gilt: $U_{lub} = n\ (2^{1/n} - 1)$.

Für n = 1 : $U_{lub} = 1$
Für n = 2 : $U_{lub} = 0{,}828$
Für n $\rightarrow \infty$: $\lim U_{lub}(n) = \ln(2) = 0{,}693$



Die kleinste obere Schranke U_{LUB} (A) für die eine Taskmenge unter einem Schedulingverfahren planbar ist, ist gegeben durch:

$$U_{LUB}(A) = \min U_{UB}(\{T\}, A) \text{ über alle Taskmengen } \{T\}.$$

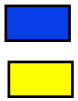
Eigenschaften von RMS :

Beispiel für Scheduling nach RMS, in dem die Auslastung über der RMS-Grenze liegt, d.h. RMS keinen Plan findet, obwohl einer existiert.

1. Ausgangssituation: für diese Werte findet RMS einen Plan. Die Auslastung liegt mit 0,828 an der theoretischen Grenze

T_1 : $\Delta e_1 = 3$, $\Delta p_1 = 7$

T_2 : $\Delta e_2 = 2$, $\Delta p_2 = 5$

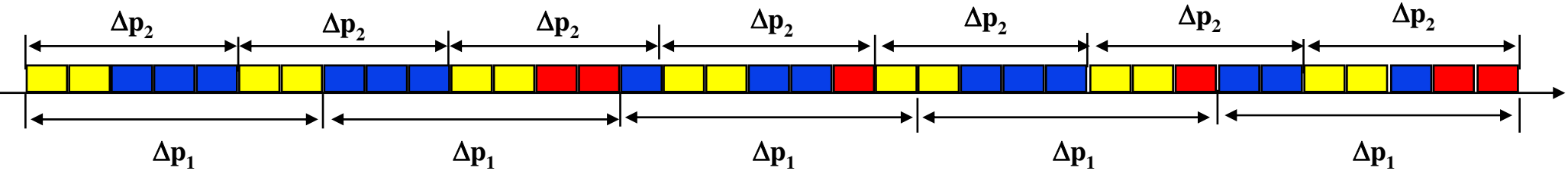


kgV = 35

T_1 : 15 Zeiteinheiten

T_2 : 14 Zeiteinheiten

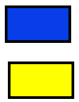
$29/35 = 0{,}828 \approx n\,(\sqrt[n]{2} - 1)$



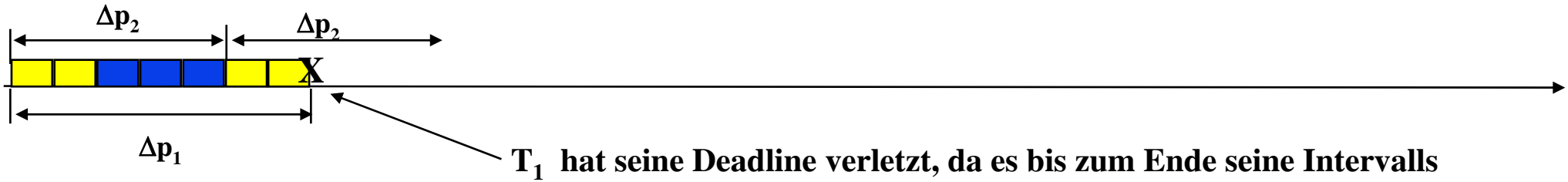
Eigenschaften von RMS :

Situation 2: Erhöhung des Rechenbedarfs von T_1 um 1 Einheit ($\Delta e_1 = 4$). RMS kann nicht mehr angewandt werden. Die theoretische Grenze der Auslastung wurde überschritten.

T_1 : $\Delta e_1 = 4$, $\Delta p_1 = 7$
T_2 : $\Delta e_2 = 2$, $\Delta p_2 = 5$

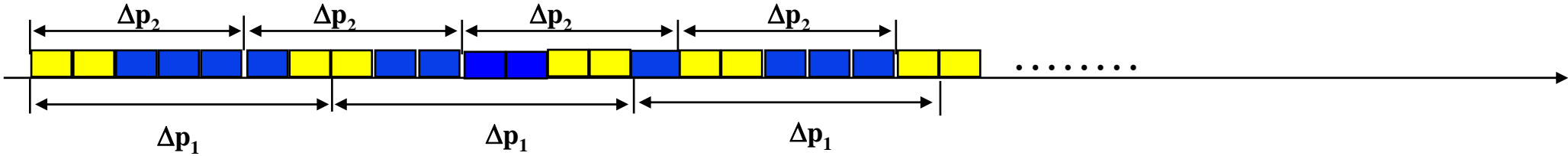


kgV = 35
T_1 : 20 Zeiteinheiten
T_2 : 14 Zeiteinheiten
$34/35 = 0{,}971 > n(\sqrt[n]{2} - 1) = 0{,}828$



T_1 **hat seine Deadline verletzt, da es bis zum Ende seine Intervalls noch keine 4 Zeiteinheiten zur Verfügung hatte.**

Für EDF ist das kein Problem, da das Verfahren den Prozessor bis 1 auslasten kann.



Optimalität von RMS

Satz:

Sei T eine Taskmenge, für die eine gefundene (statische) Prioritätszuordnung bereits einen brauchbaren Plan liefert. Dann wird auch die Prioritätszuordnung nach monotonen Raten einen brauchbaren Plan liefern.

C.L. Liu, J.W. Layland
Scheduling algorithms for multiprogramming in a hard- real-time environment.
Journal of the ACM 20(1), January 73, pp.46-61

Beobachtung:

Die Antwortzeit von T_a wird durch die höher priorisierte Task T_b verlängert und zwar je mehr, je häufiger T_b während der Ausführungszeit von T_a aktiviert wird.

Das Maximum der Aktivierungen wird erreicht, wenn T_a und T_b zum selben Zeitpunkt aktiviert werden.

Dieses Argument kann auf eine Taskmenge mit mehr als 2 Tasks erweitert werden, so dass gilt:

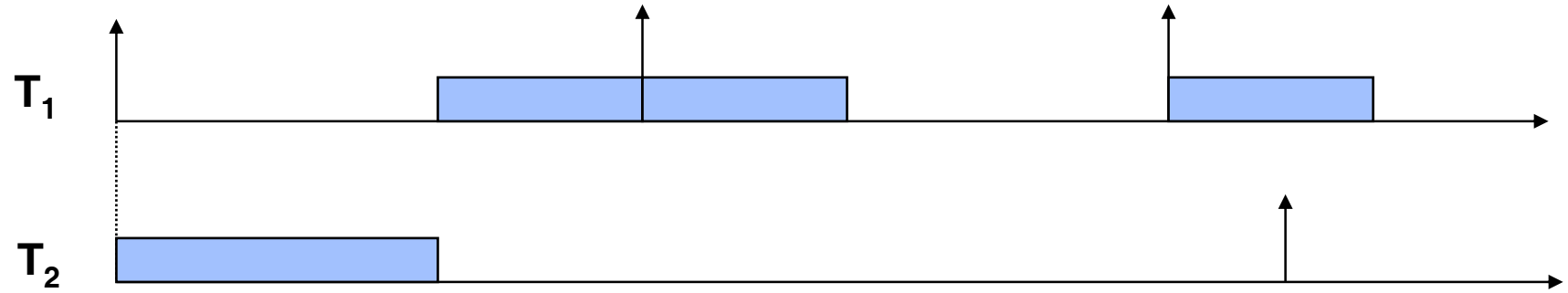
Die maximale (worst-case) Antwortzeit ist gegeben, wenn alle Tasks gleichzeitig aktiviert werden (s. kritische Intervall).

Schlussfolgerung: Wenn die Planbarkeit jeder Task an ihrer kritischen Instanz gezeigt werden kann, ist die Task auch unter allen anderen Bedingung planbar.

Der Beweis der Optimalität von RM nutzt diese Folgerung um zu zeigen, dass eine Taskmenge, die unter einer beliebigen Prioritätszuordnung planbar ist, auch unter RM geplant werden kann .

Beweis der Optimalität

Ausgangspunkt



T_1, T_2 periodische Tasks mit $\Delta p_1 < \Delta p_2$. Da die Prioritäten NICHT nach RM zugeordnet seien, gilt: Prio T_2 > Prio T_1 .

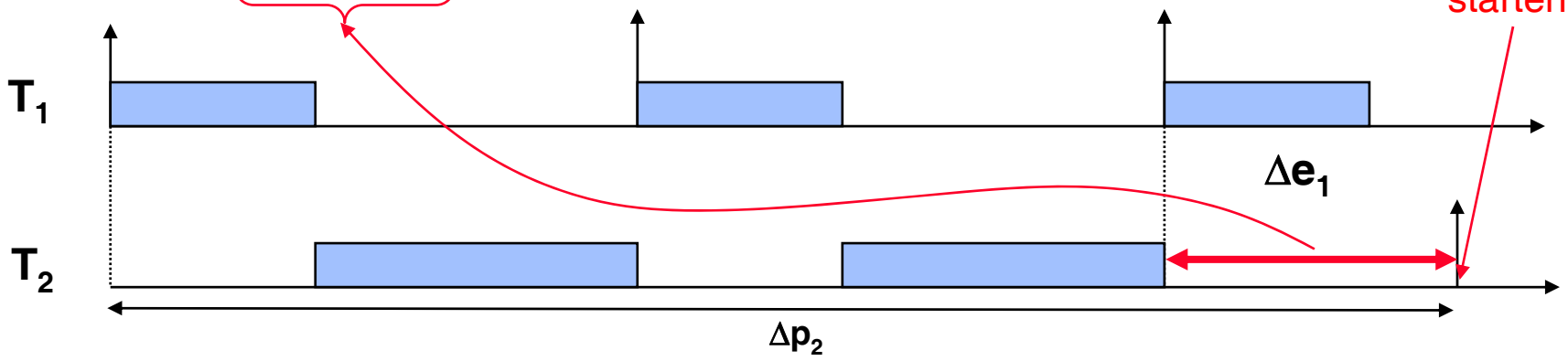
Bedingung, dass die Taskmenge sicher unter dieser Prioritätszuordnung geplant werden kann ist:

$$\Delta e_1 + \Delta e_2 \leq \Delta p_1 \qquad [G1]$$

Zu zeigen: G1 gilt auch in Plänen, die nach RMS erstellt wurden.

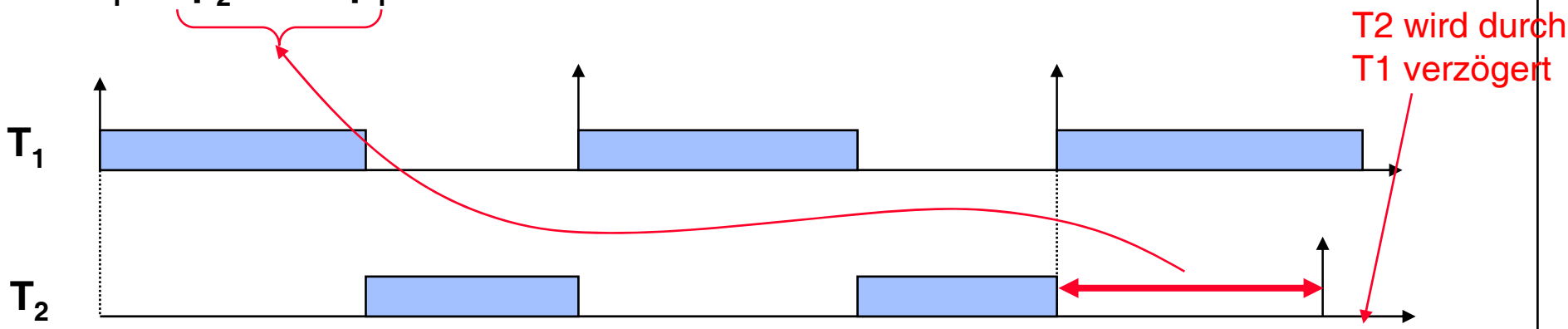
Fall 1: $\Delta e_1 < \Delta p_2 - F \bullet \Delta p_1$

(F gibt an, wie oft p_1 in p_2 vollständig enthalten ist)



Fall 1: Die Ausführungszeit Δe_1 ist kurz genug damit alle Anforderungen von T_1 innerhalb der kritischen Zone von T_2 beendet werden können, bevor die nächste Anforderung von T_2 gestartet wird.

Fall 2: $\Delta e_1 \geq \Delta p_2 - F \bullet \Delta p_1$

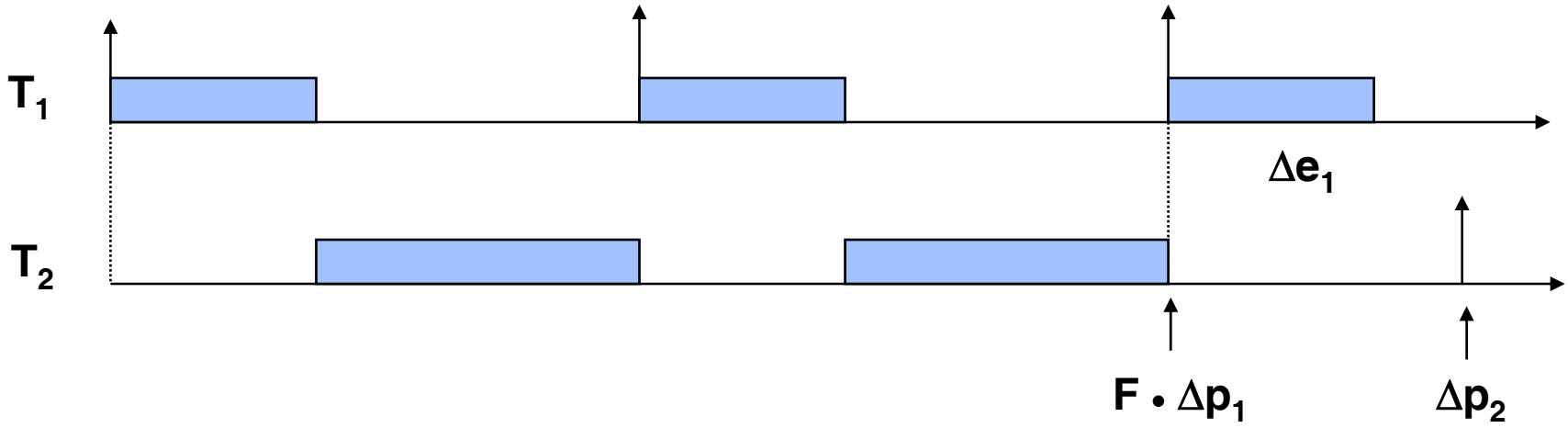


Fall 2: Die Ausführungszeit Δe_1 ist so lang, dass die letzte Anforderungen von T_1 innerhalb der kritischen Zone von T_2 mit der nächste Anforderung von T_2 überlappt.

Beweis der Optimalität

Betrachtung der möglichen Fälle unter RM und der Beweis, dass die Bedingung $\Delta e_1 + \Delta e_2 \leq \Delta p_1$ für einen beliebigen Plan die Bedingung für RM impliziert.

Fall 1: $\Delta e_1 < \Delta p_2 - F \bullet \Delta p_1$ **$F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$ = Anzahl der vollst. Akt. von T1**

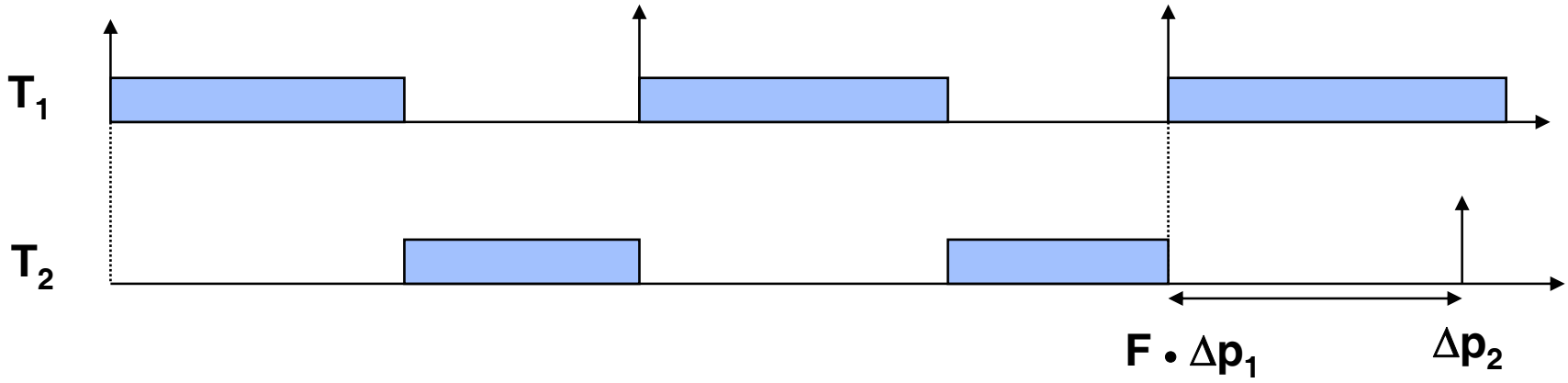


Die Bedingung für die Planbarkeit unter RM in diesem Fall ist:

$$(F+1)\,\Delta e_1 + \Delta e_2 \leq \Delta p_2 \qquad [G2]$$

Beweis der Optimalität

Fall 2: $\Delta e_1 \geq \Delta p_2 - F \bullet \Delta p_1$ $\qquad F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$



Die Bedingung für die Planbarkeit unter RM in diesem Fall ist:

$$F \bullet \Delta e_1 + \Delta e_2 \leq F \bullet \Delta p_1 \qquad \text{[G4]}$$

Herleitung der kleinsten oberen Schranke U_{LUB} für zwei Tasks für RM

Der Beweis erfolgt in zwei Schritten:

1. Es wird die minimale obere Schranke der Auslastung U_{UB} abhängig von den Ausführungszeiten der Tasks T_1 und T_2 hergeleitet. Es gelte $\Delta p_1 < \Delta p_2$.

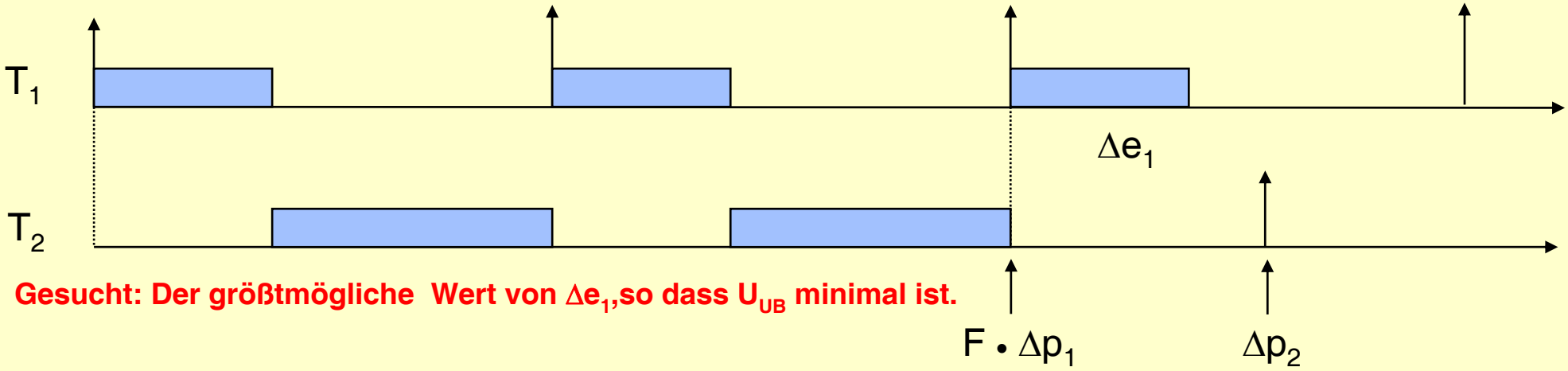
   Es wird angenommen, dass die Ausführungszeit der niedriger priorisierten Task T_2 so festgelegt wird, dass der Prozessor voll genutzt wird. Dann wird die Ausführungszeit der höher priorisierten Task so bestimmt, dass U_{UB} minimal wird.

2. Ist das Minimum von U_{UB} in Abhängigkeit von den Ausführungszeiten bestimmt, wird die obere Schranke im Hinblick auf die anderen Taskparameter, wie z.B. das Verhältnis der Periodelängen und der Phasen minimiert. U_{LUB} stellt dann eine im Hinblick auf alle Taskparameter minimierte obere Schranke dar.

Wie vorher werden die beiden Fälle für das kritische Intervall herangezogen, die auch zum Beweis der Optimalität benutzt wurden. Sei $F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$ die Anzahl der Perioden von T_1, die vollständig in T_2 enthalten sind.

1. Fall: Die Ausführungszeit von T_1 ist so kurz, dass alle Ausführungen von T_1 vor der zweiten Aktivierung von T2 vollständig beendet sind.

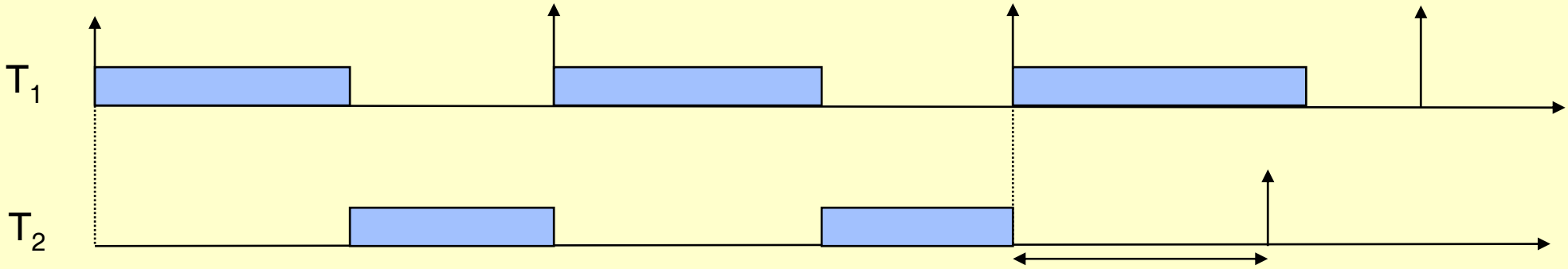
$\Delta e_1 < \Delta p_2 - F \bullet \Delta p_1$ $\quad F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$



Gesucht: Der größtmögliche Wert von Δe_1, so dass U_{UB} minimal ist.

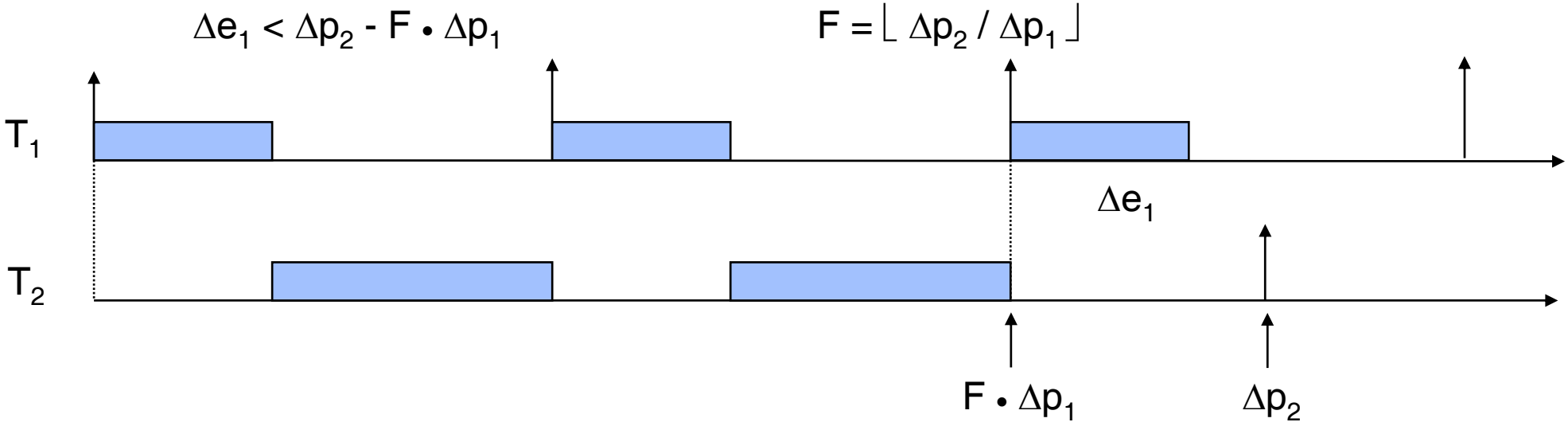
2. Fall: Die Ausführungszeit von T_1 im kritischen Intervall von T_2 überlappt die zweite Aktivierung von T_2 , so dass $\Delta e_1 \geq \Delta p_2 - F \bullet \Delta p_1$ gilt.

Fall 2: $\Delta e_1 \geq \Delta p_2 - F \bullet \Delta p_1$ $\quad F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$



Gesucht: Der größtmögliche Wert von Δe_1, so dass U_{UB} minimal ist.

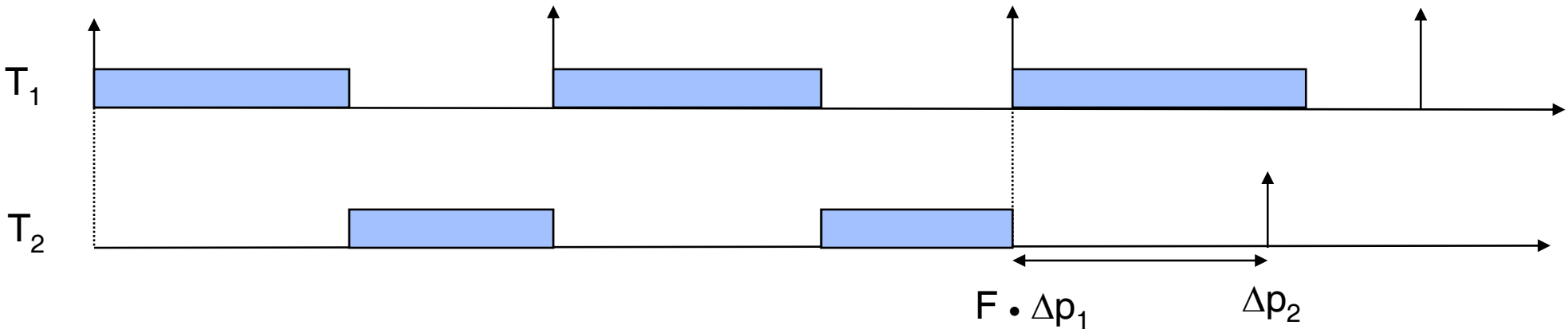
1. Fall: Die Ausführungszeit von T_1 ist so kurz, dass alle Ausführungen von T_1 im kritischen Intervall von T_2 vor der zweiten Aktivierung von T2 vollständig beendet sind.



Gesucht: Der größtmögliche Wert von $\Delta e1$,so dass U_{UB} minimal ist.

2. Fall: Die Ausführungszeit von T_1 im kritischen Intervall von T_2 überlappt die zweite Aktivierung von T_2 , so dass $\Delta e_1 \geq \Delta p_2 - F \bullet \Delta p_1$ gilt.

Fall 2: $\Delta e_1 \geq \Delta p_2 - F \bullet \Delta p_1$ $\quad F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$



Gesucht: Der größtmögliche Wert von $\Delta e1$,so dass U_{UB} minimal ist.

2. Schritt: Minimieren der übrigen Task Parameter (Verhältnis der Perioden)

Da U_{UB} in beiden Fällen bei $\Delta e_1 = \Delta p_2 - F \bullet \Delta p_1$ auftritt können wir diesen Wert in die Gleichung [G6] für die obere Schranke einsetzen und dann den Gesamtterm für das Verhältnis der Perioden minimieren. Die Phase der Perioden wurde bereits so gewählt, dass sie beide zum kritischen Zeitpunkt aktiviert werden.

$U = (\Delta p_1 / \Delta p_2) \bullet F + (\Delta e_1 / \Delta p_2) \bullet (\Delta p_2 / \Delta p_1 - F)$ **[G6]** Einsetzen von $\Delta e_1 = \Delta p_2 - F \bullet \Delta p_1$ ergibt:

$= (\Delta p_1 / \Delta p_2) \bullet F + ((\Delta p_2 - F \bullet \Delta p_1) / \Delta p_2) \bullet (\Delta p_2 / \Delta p_1 - F)$

$= (\Delta p_1 / \Delta p_2) \bullet [F + (\Delta p_2 / \Delta p_1 - F) \bullet (\Delta p_2 / \Delta p_1 - F)]$ **[G7]**

zur Vereinfachung der Schreibweise setzen wir $G = \Delta p_2 / \Delta p_1 - F$ und erhalten so:

$U = (\Delta p_1 / \Delta p_2) \bullet (F + G^2)$

$= (F + G^2) / (\Delta p_2 / \Delta p_1) = (F + G^2) / ((\Delta p_2 / \Delta p_1 - F) + F) = (F + G^2) / ((G) + F) = (F + G^2) / (F + G)$

$= 1 - (G(1-G)/(F+G)$

Da $0 \leq G < 1$ gilt, ist der Term (G(1-G) nicht negativ. Daher wird U mit wachsendem F größer. Der minimale Wert von U ist daher bei minimalem Wert von F gegeben, nämlich bei F=1 (kleiner kann F nach Definition von Δp_1 und Δp_2 nicht werden!). Dadurch erhalten wir:

$U = (1 + G^2) / (1 + G)$ **[G8]**

Da G das Verhältnis der Perioden ausdrückt, suchen wir den minimalen Wert von U indem wir dU/dG berechnen.

$dU/dG = (2G\,(1+G) - (1+ G^2)) / (1+G)^2$
$= (G^2 + 2G -1) / (1+G)^2$

dU/dG=0 erhalten wir für:

$(G^2 + 2G -1) = 0$ und damit die Lösungen:

$G_1 = -1 - \sqrt{2}$
$G_2 = -1 + \sqrt{2}$

Da $0 \leq G < 1$ gilt, kann die negative Lösung verworfen werden, so dass wir die kleinste Obere Schranke durch Einsetzen von G_2 in die Gleichung [G8]:

$U = (1 + G^2) / (1 + G)$ erhalten mit G= : $\sqrt{2} - 1$

$U_{LUB} = (1 + (\sqrt{2} -1)^2) / (1 + (\sqrt{2} -1)) = (4 - 2\sqrt{2}) / \sqrt{2} =$ **$2\,(\sqrt{2} -1\,)$** $\cong$ **0,83**

Auslastungsfaktor in Abhängigkeit des Verhältnisses der Periodendauern k

Wenn Δp_2 ein ganzzahliges Vielfaches von Δp_1 ist, wird $F = \Delta p_2 / \Delta p_1$ und $G = \Delta p_2 / \Delta p_1 - F = 0$. Da $U = (1 + G^2) / (1 + G)$ gilt, wird der Auslastungsfaktor U=1.

Frage: wie ändert sich allgemein der Auslastungsfaktor U in Abhängigkeit von dem Verhältnis zwischen **Δp_2 und Δp_1 ,** d.h. in Abhängigkeit von **$k = \Delta p_2 / \Delta p_1$?**

Ausgangspunkt: $U = (\Delta p_1 / \Delta p_2) \bullet (F + (\Delta p_2 / \Delta p_1 - F) \bullet (\Delta p_2 / \Delta p_1 - F))$ Formel [G6]

Einsetzen von k

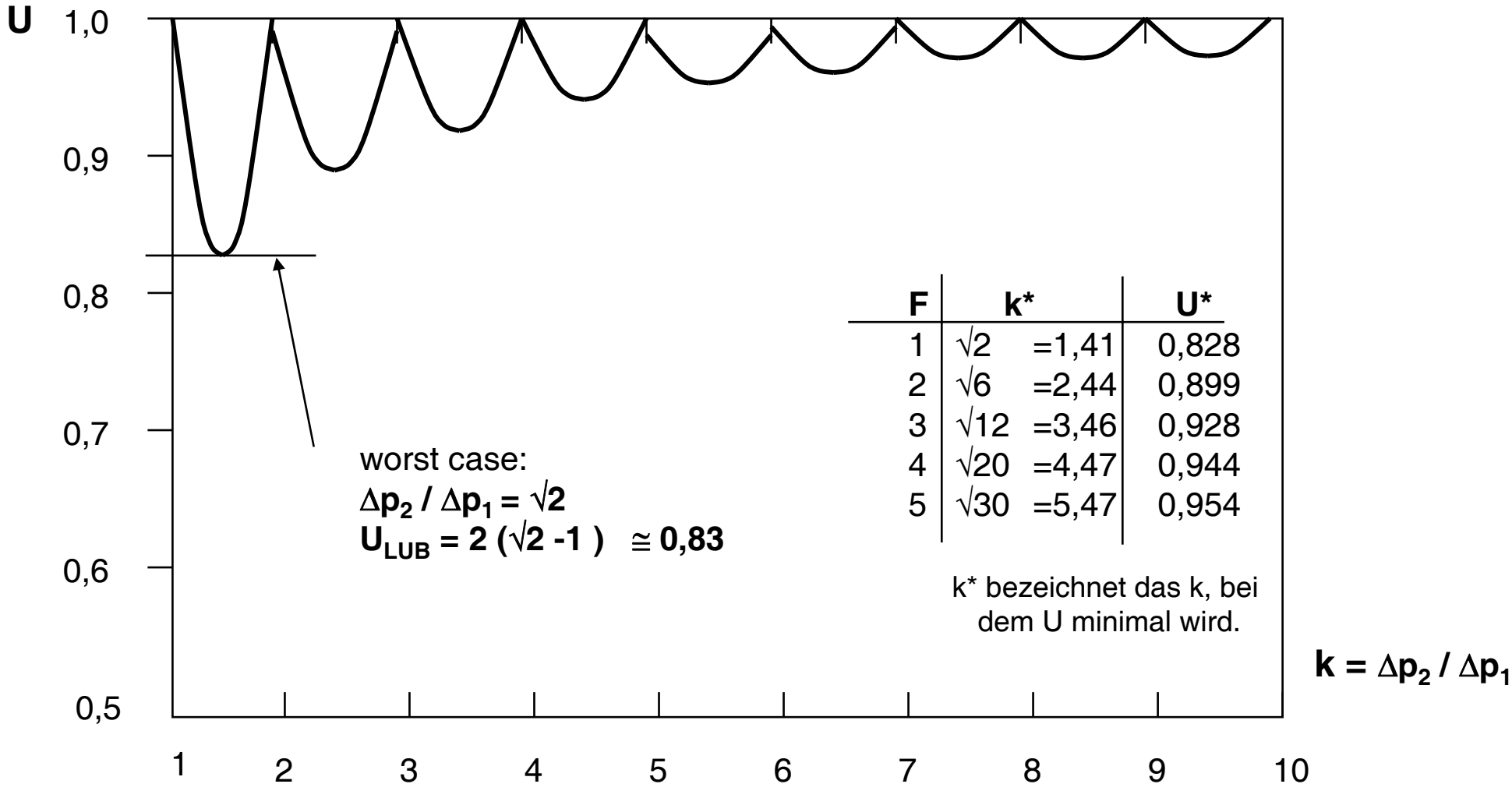
$= (1/ k) \bullet (F + (k-F) (k-F))$

$= F + (k-F)^2 / k$

$= k - 2F + (F(F+1) / k$

Minimieren nach dU/dk ergibt $U^* = 2 ((F (F+1)^{1/2} - F)$

Obere Schranke der Prozessorauslastung in Abhängigkeit von k (für 2 Tasks)



F	k*	U*
1	$\sqrt{2}$ =1,41	0,828
2	$\sqrt{6}$ =2,44	0,899
3	$\sqrt{12}$ =3,46	0,928
4	$\sqrt{20}$ =4,47	0,944
5	$\sqrt{30}$ =5,47	0,954

Bisher:

Hinreichende Bedingung für die Einplanbarkeit durch $U_{LUB} = n\,(2^{1/n} - 1)$

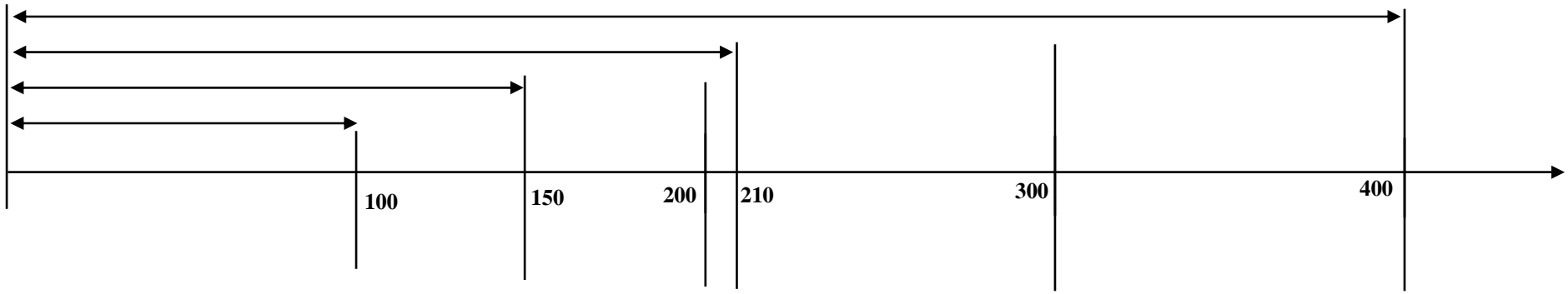
n	U_{LUB}
1	1.00
2	0,83
3	0,78
4	0,76
10	0,72

$\lim_{n\to\infty} U_{LUB}(n) = \ln 2 \cong 0{,}69$

Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS

Beispiel:

T_i	Δe_i	Δp_i
1	20	100
2	30	150
3	80	210
4	80	400



Wann kann T_1 eingeplant werden ?
Da es sich um die Task mit der höchsten Priorität handelt ist die notwendige und hinreichende Bedingung durch: $\Delta e_1 \leq \Delta p_1 = 100$ gegeben.

Wann können T_2 und T_3 eingeplant werden ?

T_2 kann erfolgreich eingeplant werden, wenn im Intervall $[0, \Delta p_2]$ genügend Prozessorzeit gefunden wird.

Angenommen T_2 wird zum Zeitpunkt t **abgeschlossen**.
Die Zeit, welche die höher priorisierte Task T_1 im Intervall [0,t] benötigt ist:

$$\lceil t/\Delta p_1 \rceil \bullet \Delta e_1 ,$$

so daß bis zum Zeitpunkt t insgesamt: $\lceil \mathbf{t/\Delta p_1} \rceil \bullet \Delta \mathbf{e_1} + \Delta \mathbf{e_2}$ benötigt wird.

Analog gilt für T_3 : $\lceil \mathbf{t/\Delta p_1} \rceil \bullet \Delta \mathbf{e}_1 + \lceil \mathbf{t/\Delta p_2} \rceil \bullet \Delta \mathbf{e_2} + \Delta \mathbf{e_3}$.

Kann man ein t im Intervall $[0, \Delta p_3]$ finden, so daß gilt : $t \geq \lceil t/\Delta p_1 \rceil \bullet \Delta e_1 + \lceil t/\Delta p_2 \rceil \bullet \Delta e_2 + \Delta e_3$
kann RM die Tasks einplanen.

Man muß nicht alle Zeitpunkte t überprüfen , sondern nur die, an denen neue Tasks aktiviert werden oder eine neue Periode der Task beginnt.

Formal ausgedrückt: Man muß die Gesamtausführungszeiten nur zu den Zeitpunkten

$$\tau_i = \{l \cdot \Delta p_j \mid j = 1, ..., i,\ l = 1, ..., \lfloor \Delta p_i / \Delta p_j \rfloor\}$$

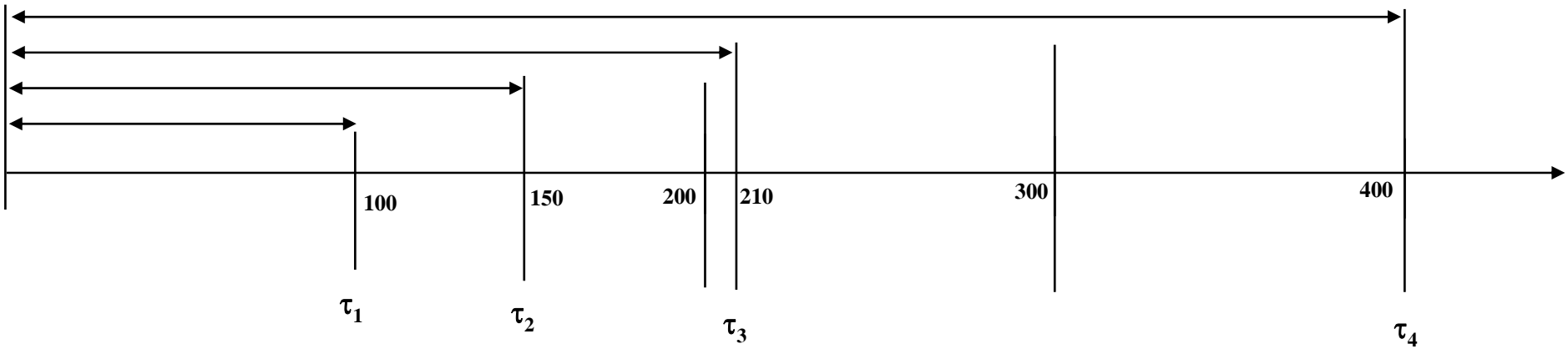
berechnen.

Für das Beispiel ergibt sich:

i	$\lfloor \Delta p_i / \Delta p_j \rfloor$	$l \cdot \Delta p_j$ j=1,..,i	τ_i
1	100/100=1	100	{100}
2	j=1: 150/100=1 j=2: 150/150=1	100 150	{100, 150}
3	j=1: 210/100=2 j=2: 210/150=1 j=3: 210/210=1	100, 200 150 210	{100, 150, 200, 210}
4	j=1: 400/100=4 j=2: 400/150=2 j=3: 400/210=1 j=4: 400/400=1	100, 200, 300, 400 150, 300 210 400	{100, 150, 200, 210, 300, 400}

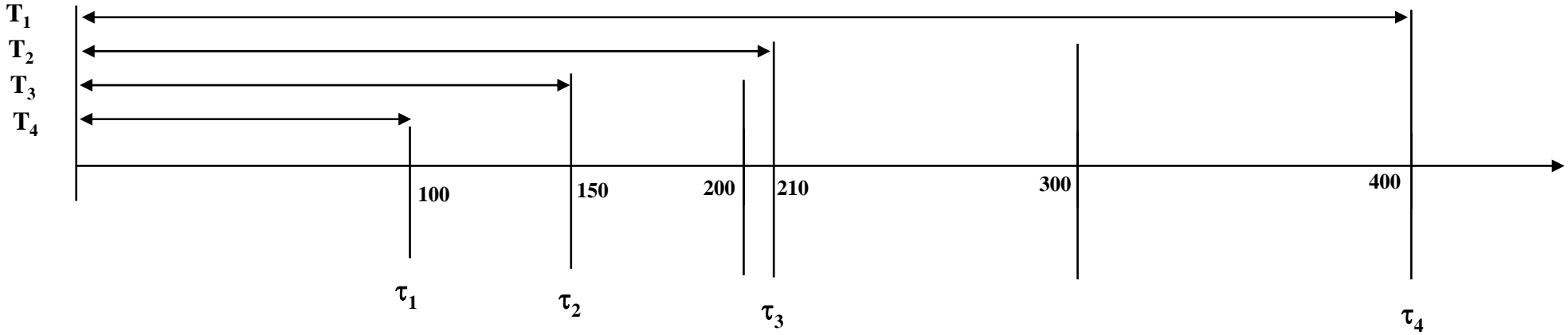
Beispiel:

T_i	Δe_i	p_i
1	20	100
2	30	150
3	80	210
4	80	400



$\tau_1 = \{100\}$
$\tau_2 = \{100, 150\}$
$\tau_3 = \{100, 150, 200, 210\}$
$\tau_4 = \{100, 150, 200, 210, 300, 400\}$

Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS (cont.)



Prozeß T_1 kann eingeplant werden, wenn:

$$\Delta e_1 \leq 100$$

Prozeß T_2 kann eingeplant werden, wenn:

$$\Delta e_1 + \Delta e_2 \leq 100$$

oder

$$2 \bullet \Delta e_1 + \Delta e_2 \leq 150$$

Prozeß T_3 kann eingeplant werden, wenn:

$$\Delta e_1 + \Delta e_2 + \Delta e_3 \leq 100$$

oder

$$2 \bullet \Delta e_1 + \Delta e_2 + \Delta e_3 \leq 150$$

oder

$$2 \bullet \Delta e_1 + 2 \bullet \Delta e_2 + \Delta e_3 \leq 200$$

oder

$$3 \bullet \Delta e_1 + 2 \bullet \Delta e_2 + \Delta e_3 \leq 210$$

Prozeß T_4 kann eingeplant werden, wenn:

$$\Delta e_1 + \Delta e_2 + \Delta e_3 + \Delta e_3 \leq 100$$

oder

$$2 \bullet \Delta e_1 + \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 150$$

oder

$$2 \bullet \Delta e_1 + 2 \bullet \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 200$$

oder

$$3 \bullet \Delta e_1 + 2 \bullet \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 210$$

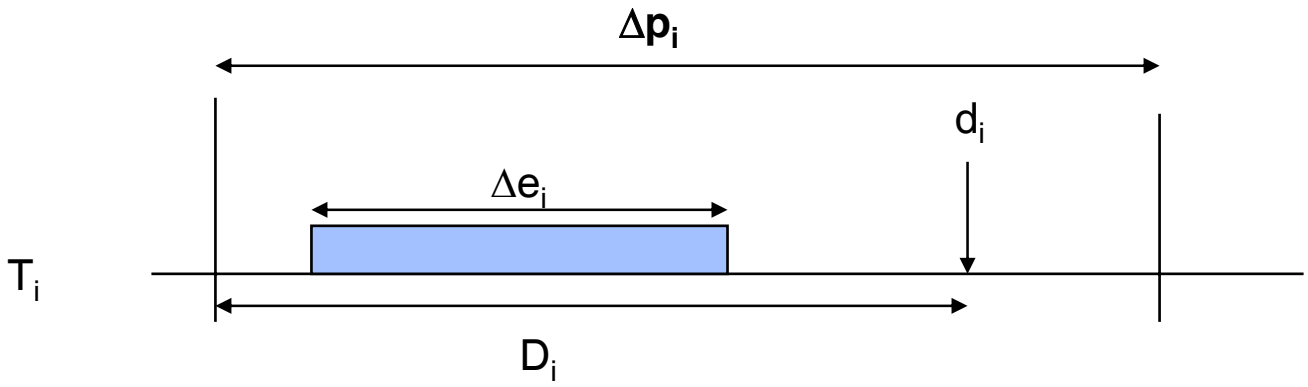
oder

$$3 \bullet \Delta e_1 + 2 \bullet \Delta e_2 + 2 \bullet \Delta e_3 + \Delta e_4 \leq 300$$

oder

$$4 \bullet \Delta e_1 + 3 \bullet \Delta e_2 + 2 \bullet \Delta e_3 + \Delta e_4 \leq 400$$

Deadline Monotonic



Eine Task T_i wird charakterisiert durch:

- **eine Phase Φ_i**
- **eine Ausführungszeit Δe_i**
- **eine relative Deadline D_i**
- **eine Periode Δp_i**

Es gelten folgende Beziehungen:

$$\Delta e_i \leq D_i \leq \Delta p_i$$

$$r_{i,k} = \Phi_i + (k-1)\,\Delta p_i$$

$$d_{i,k} = r_{i,k} + D_i$$

Das Verfahren ordnet einer Task die Prioritäten aufgrund der relative Deadline D_i zu. Da diese relative Deadline konstant ist, handelt es sich um ein statisches Verfahren mit festen Prioritäten. Es gilt:

Deadline Monotonic (DM) ist optimal, d.h. wenn irgendein statisches Verfahren eine Taskmenge mit Deadlines ungleich der Perioden planen kann, findet auch DM einen Plan.

Planbarkeitsanalyse

Hinreichende Bedingungen: Reduzierung der Taskperiode auf die relative Deadline.

1. Pessimistischer Test: $\sum_{i=1}^{n} \Delta e_i / D_i \leq n\,(2^{1/n} - 1)$

   folgt unmittelbar aus den Bedingungen von DM in Relation zu RM.
   Es ist allerdings keine gute Schranke, weil die Belastung des Prozessors überbewertet wird.

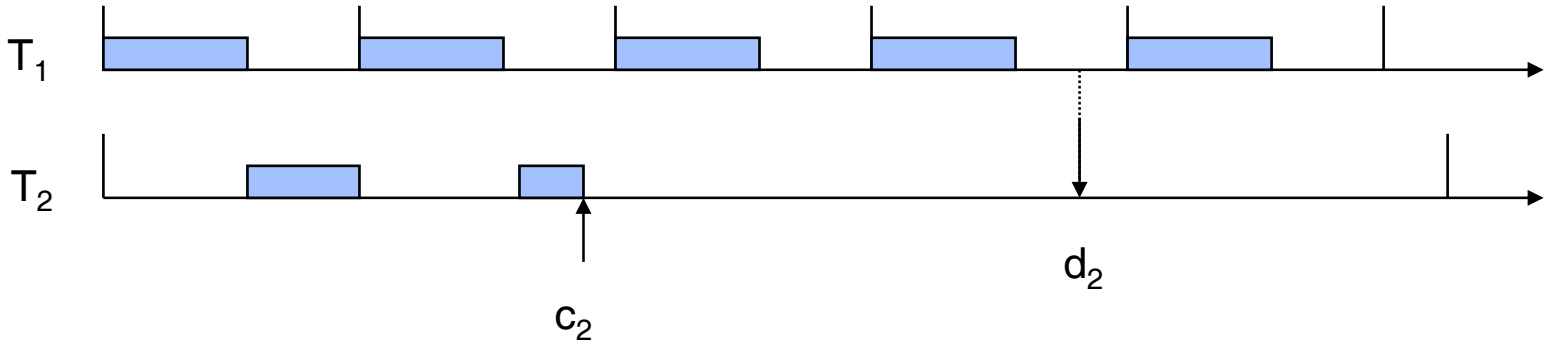
2. Weniger pessimistisch:
   - Der worst case Prozessorbedarf ensteht, wenn alle Tasks simultan bereit werden.
   - Für jede Task muß die Summe der Ausführungszeiten und der Interferenzen (I_i), die durch Unterbrechung (durch höher priorisierte Tasks entstehen) kleiner oder gleich der relativen Deadline D_i sein. Wenn alle Tasks nach Deadlines geordnet sind, d.h. $i<j \Leftrightarrow D_i < D_j$ gilt, ist ein solcher Test gegeben durch:

$$\forall\, i : 1 \leq i \leq n \qquad \Delta e_i + I_i \leq D_i$$

$$\text{mit} \quad I_i = \sum_{i=1}^{i-1} \lceil D_i / \Delta p_j \rceil \Delta e_j$$

Anzahl derUnterbrechnungen durch eine Task höherer Priorität.

Immer noch pessimistisch, wie das Beispiel zeigt:



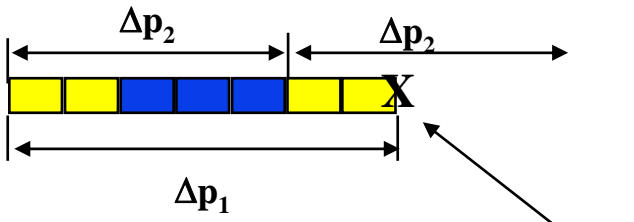
Bei dem Test wird angenommen, dass jede höher priorisierte Task eine mit niedrigerer Priorität genau $D_i / \Delta p_j$ -mal unterbricht, was nicht der Fall ist, wenn T_i bereits früher beendet ist.

Ein iterativer Algorithmus ähnlich der iterative Ermittlung des Prozessorbedarfs wird von Audsley, Burns, Richardson und Wellings (1991und 92) angegeben. Er ermittelt iterativ die Antwortzeit der Tasks unter allen Interferenzbedingungen.

N.C. Audsley, A. Burns, M.F. Richardson and A.J. Wellings: „Hard real-time scheduling: The deadline monotonic approach“ *IEEE Workshop on Real-Time Operating Systems, 1992.*
Auch diskutiert in Butazzo (siehe Literaturliste)

Das Argument für EDF

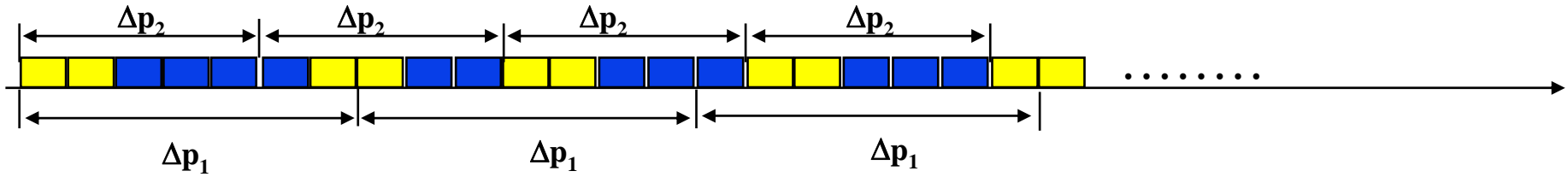
T_1 : $\Delta e_1 = 4$, $\Delta p_1 = 7$
T_2 : $\Delta e_2 = 2$, $\Delta p_2 = 5$



Planbarkeitsanalyse:
$\Delta e_1 / \Delta p_1 = 4/7$
$\Delta e_2 / \Delta p_2 = 2/5$
$4/7 + 2/5 = (20 + 14)/35 = 34/35$
$30/35 = 0{,}97 > 2\,(2^{1/2} - 1) = 0{,}828$

T_1 hat seine Deadline verletzt, da es bis zum Ende seine Intervalls noch keine 4 Zeiteinheiten zur Verfügung hatte.

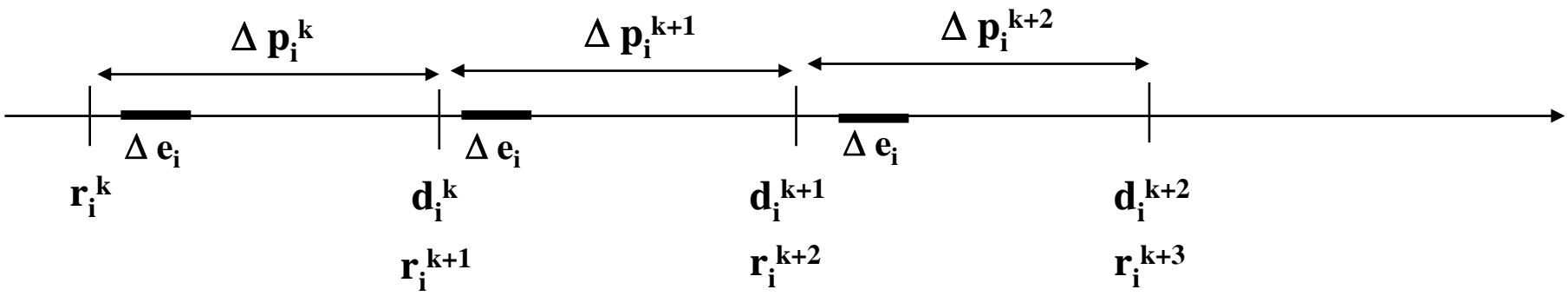
Für EDF ist das kein Problem, da das Verfahren den Prozessor bis 1 auslasten kann.



Wie kann man die Planbarkeitsanalyse für periodische Taskmengen unter EDF durchführen?

EDF bei periodischen Prozessen

Annahme: Perioden sind a priori bekannt. ⇨ **r_i und d_i sind für alle Prozesse bekannt.**



Bei periodischen Prozessen ist die Prozessorauslastung U durch die Summe der Anteile gegeben, die die jeweilige Ausführungszeit des Prozesses an seiner Periode hat.

Es gilt: $$U = \sum_{i \in P} (\Delta e_i / \Delta p_i)$$

Zu beachten ist, dass EDF keinerlei spezielle Annahmen über die Periodizität der Prozesse macht, d.h. zur Planung periodischer und aperiodischer Tasks genutzt werden kann.
Deshalb gilt die Optimalität, die wir bereits gezeigt haben auch für periodische Tasks.

Satz (Liu, Layland 1973) :

Eine Menge periodischer Tasks kann durch EDF eingeplant werden, dann und nur dann, wenn $U = \sum_{i=1,..,n} (\Delta e_i / \Delta p_i) \leq 1$

Beweis:

1. (nur dann) Für U>1 kann die Taskmenge nicht geplant werden.

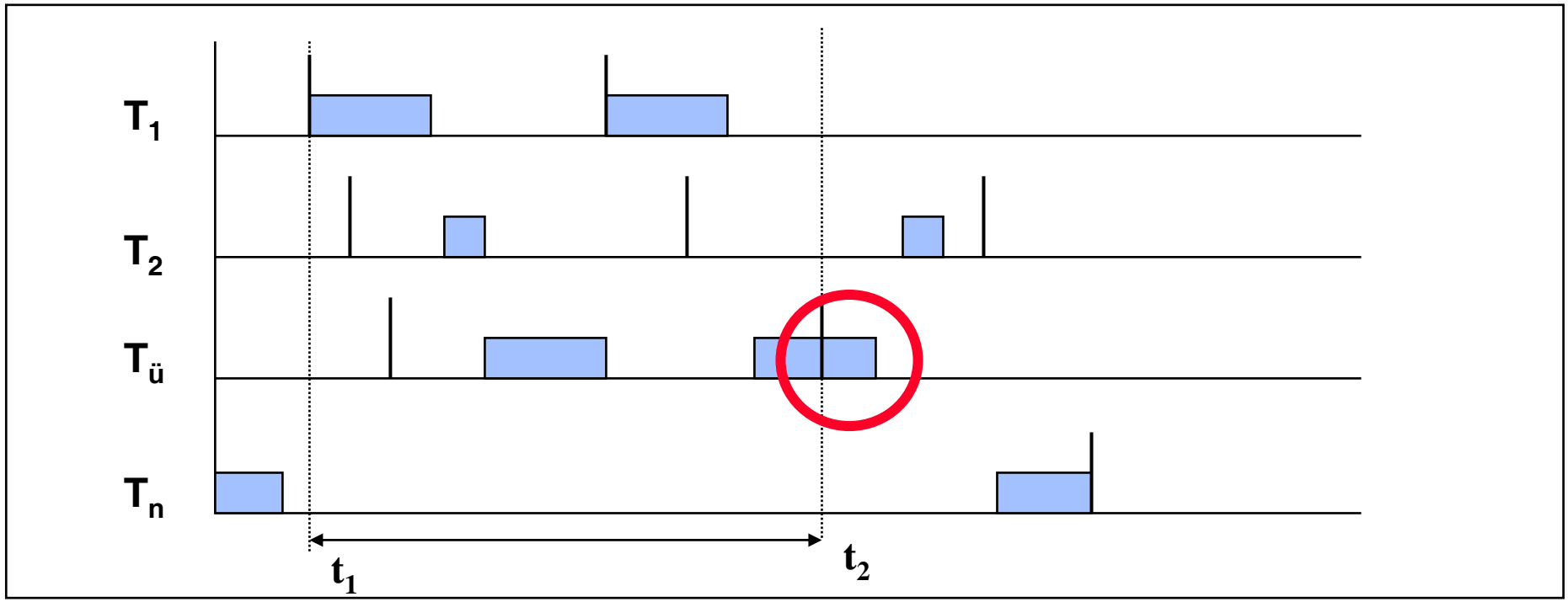
Man definiert $\Delta P = \Delta p_1 \bullet \Delta p_2 \bullet ..... \bullet \Delta p_n$, dann ist der Gesamtbedarf aller Tasks an Rechenzeit im Intervall ΔP gegeben durch:

$$\sum_{i=1,..,n} ((\Delta P / \Delta p_i) \bullet \Delta e_i) = \sum_{i=1,..,n} ((\Delta e_i / \Delta p_i) \bullet \Delta P) = U \bullet \Delta P$$

Wenn U>1 gilt, ist $U \bullet \Delta P$ größer als die zur Verfügung stehende Zeit ΔP , es kann also kein Plan gefunden werden.

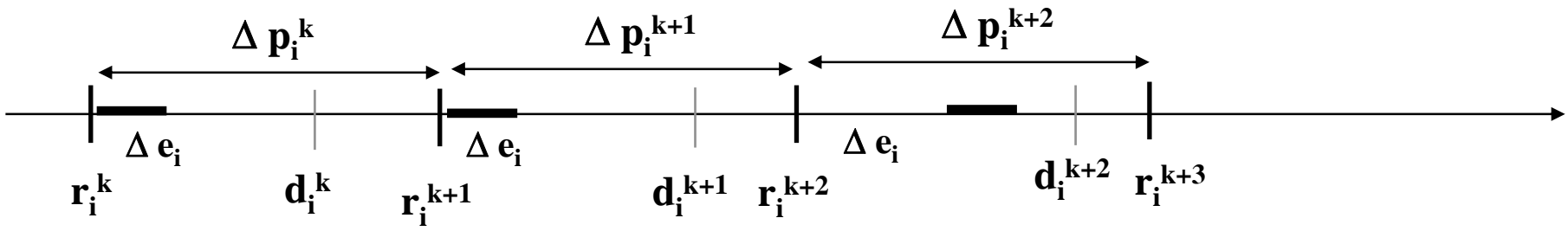
2. (dann, wenn) Für U≤1 kann EDF die Taskmenge planen.

Beweisidee: Beweis durch Widerspruch:
Annahme: U≤1 und die Menge der Tasks ist nicht planbar. Dann existiert ein Intervall (t_1 , t_2), das vollständig ausgelastet ist und in dem eine Zeitüberschreitung auftritt. Der Beweis zeigt, dass dann mit U>1 ein Widerspruch zur Annahme auftritt.



[t_1 , t_2] : längstes Intervall der kontinuierlichen Nutzung vor einer Zeitüberschreitung.
Im Intervall [t_1 , t_2] werden nur Tasks ausgeführt deren Fristen $\leq t_2$ sind.
t_1 ist die Bereitzeit einer periodischen Task.

EDF bei periodischen Prozessen, bei denen die Periode und die Deadline nicht zusammmenfallen.



Seien: $U = \Sigma\ (\Delta e_i / \Delta p_i)$, $d_{max} = \max_{1 \le i \le n} \{d_i\}$, und $P = kgV(p_1, \ldots, p_n)$.
Sei $h_T(t)$ die Summe der Ausführungszeiten aller Prozesse deren absoluten Deadlines vor t liegen.

Eine Prozeßmenge kann *NICHT* erfolgreich nach EDF eingeplant werden, genau dann wenn folgende Bedingungen gelten:

1.) $U > 1$ oder

2.) es gibt ein t_{min} mit $t_{min} < \min \left\{ P + d_{max}, (U/(1-U)) \max_{1 \le i \le n} \{(p_i - d_i)\} \right\}$
so daß $h_T(t) > t$

Beispiel 1:

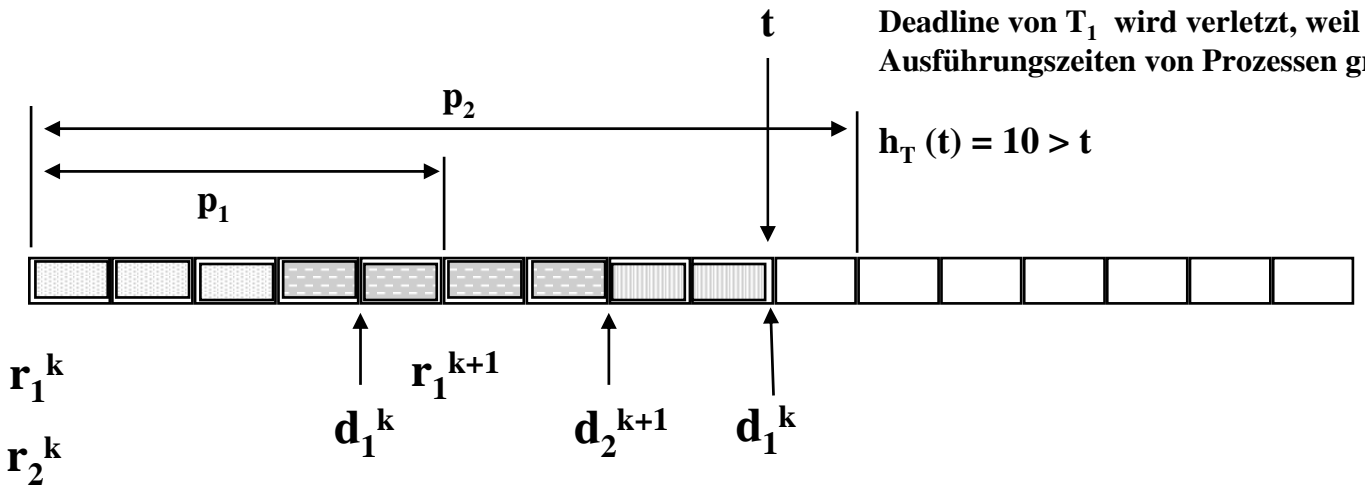
$\Delta p_1 = 5$
$\Delta p_2 = 10$

$\Delta e_1 = 3$
$\Delta e_2 = 4$

$d_1 = 4$
$d_2 = 7$

$U = \Sigma (\Delta e_i / \Delta p_i) = 3/5 + 4/10 = 1$
$P = kgV(5, 10) = 10$
$P + d_{max} = 10 + 7 = 17 < (U/(1-U)) \max_{1 \le i \le n} \{p_i - d_i\}$

d.h. t kann in dem Intervall von der Länge 17 gefunden werden.



Deadline von T_1 wird verletzt, weil die Summe der Ausführungszeiten von Prozessen größer ist als t.

$h_T(t) = 10 > t$

Beispiel 2:

$\Delta p_1 = 500$
$\Delta p_2 = 1000$

$\Delta e_1 = 3$
$\Delta e_2 = 4$

$d_1 = 4$
$d_2 = 7$

$U = \Sigma (\Delta e_i / \Delta p_i) = 3/500 + 4/1000 = 0{,}01$

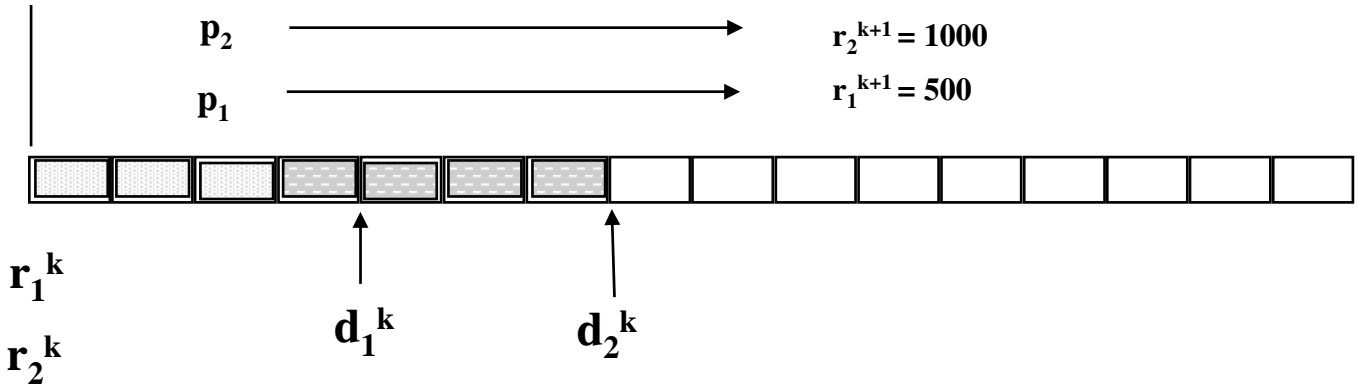
$P = kgV(500, 1000) = 1000$

$P + d_{max} = 1000 + 7 = 1007 > (U/(1-U)) \max_{1 \le i \le n} \{p_i - d_i\} = 0{,}01/0{,}99 * 993 \approx 10$

d.h. t kann in dem Intervall von der Länge 10 gefunden werden.

Deadline von P_1 wird nicht verletzt, weil kein t gefunden werden kann, für das gilt:

$$h_T(t) = 10 > t$$



Zusammenfassung:

Allgemeiner Fall: EDF ist optimal, es existieren notwendige und hinreichende Bedingungen für die Einplanbarkeit. Keine Garantien.

Periodische Prozesse

- **Deadline = Periode: notwendige und hinreichende Bedingung ist:** $U = \Sigma \ (\Delta e_i / \Delta p_i) = \Sigma \ \Delta e_i f_i \leq 1$

- **Deadline $\neq$ Periode:**

  Eine Prozeßmenge kann NICHT erfolgreich nach EDF eingeplant werden, wenn folgende (hinreichenden) Bedingungen gelten:

  1.) $u > 1$ **oder**

  2.) **es gibt ein** t_{min} **mit** $t_{min} < \min \{ P + d_{max} , (u/1-u) \max_{1 \leq i \leq n} \{(p_i - d_i\}\}$
  so daß $h_T (t) > t$