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

Prozessorauslastung "U"

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

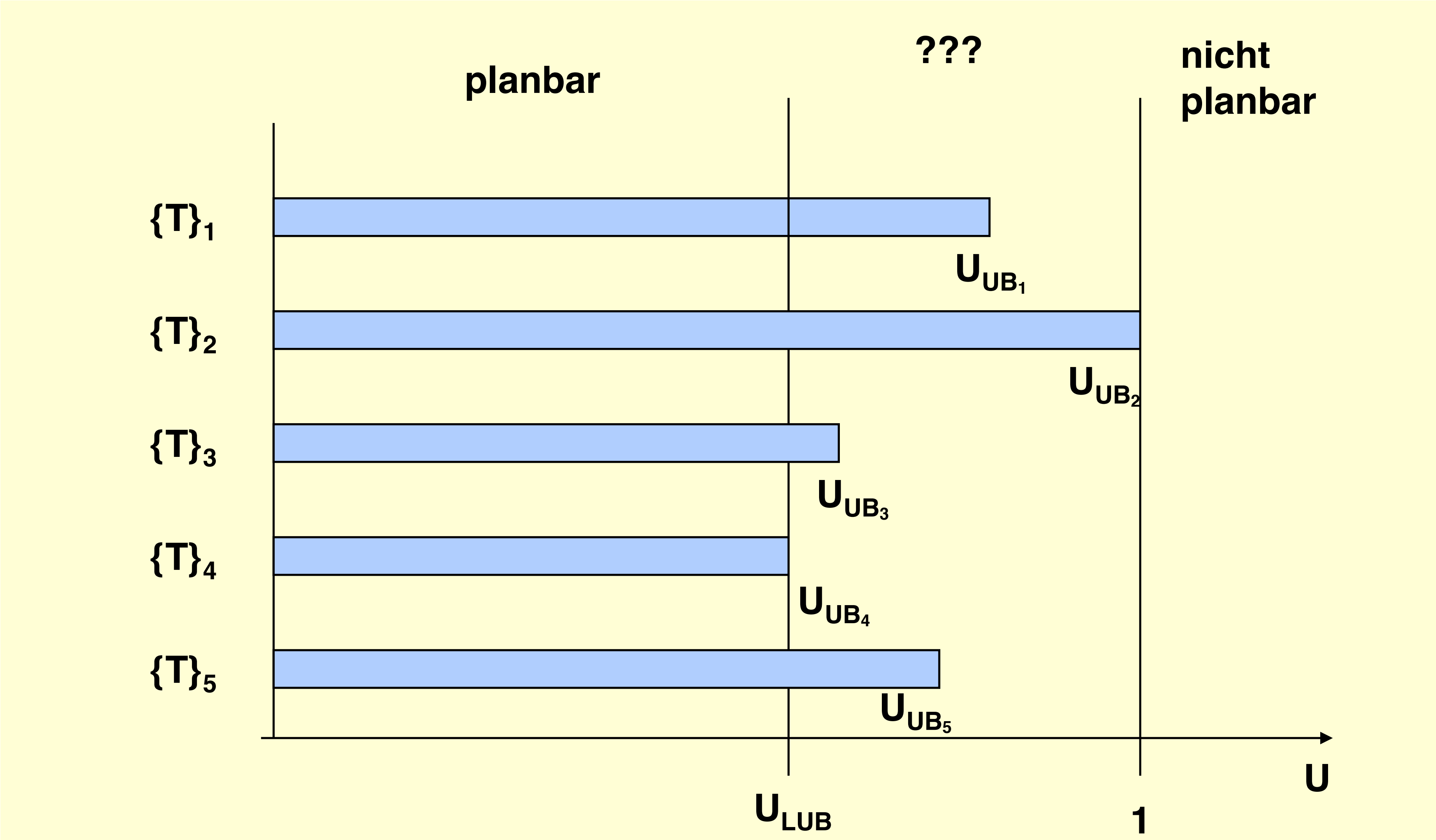
$\Delta e_i / \Delta p_i$ ist die Zeit, die Task T_i in der Periode Δp_i zur Ausführung nutzt.

Der Auslastungsfaktor für n Tasks ist dann gegeben durch:

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

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

$\sum_{(i=1,...,n)} ((\Delta P / \Delta p_i) \cdot \Delta e_i)$ ist die Gesamtrechenzeit aller Tasks, 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:

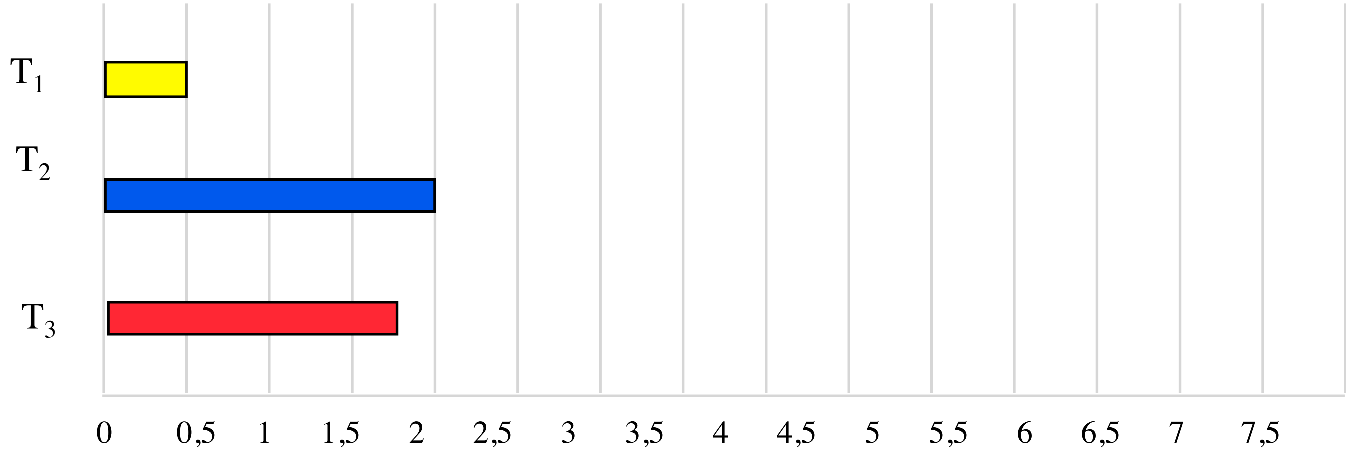
$$rms(i) < rms(j) \Leftrightarrow 1 / \Delta p_i < 1 / \Delta p_j$$

Nummerierung der Tasks gemäß ihrer Priorität:

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

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

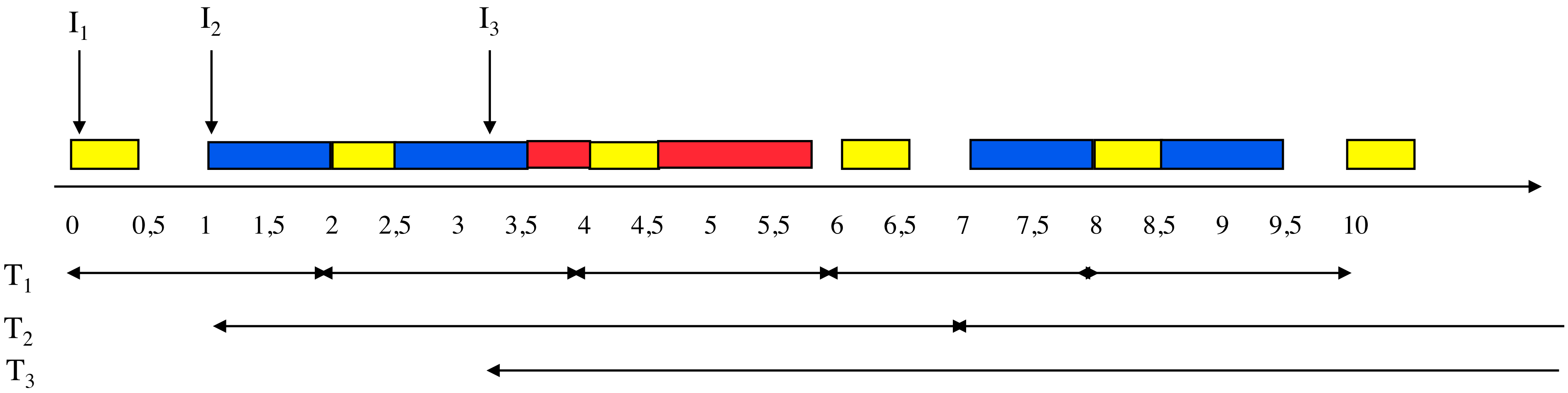
Beispiel:



$\Delta e_1 = 0{,}5, \Delta p=2$

$\Delta e_2 = 2 , \Delta p=6$

$\Delta e_3 = 1{,}75 , \Delta p=10$



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 → ∞ : $\lim U_{lub}(n) = \ln(2) = 0{,}693$

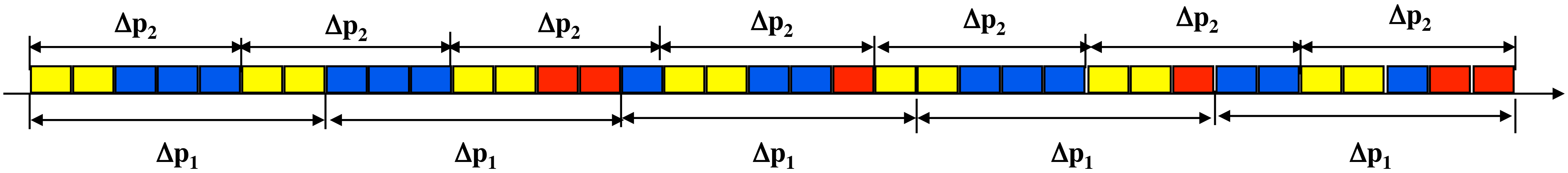
Eigenschaften von RMS :

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

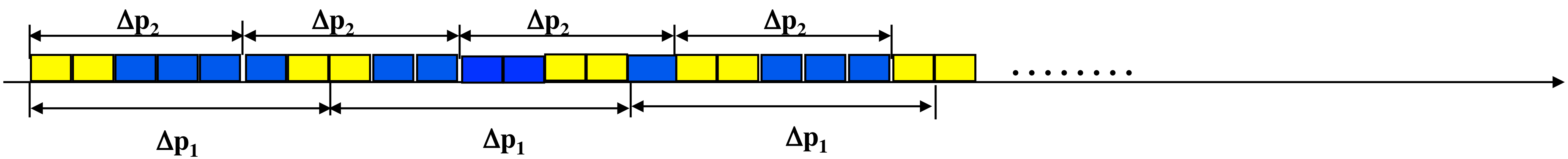
T_1 : 20 Zeiteinheiten

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

Beweis der Optimalität von RMS: 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. (kritische Instanz)

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

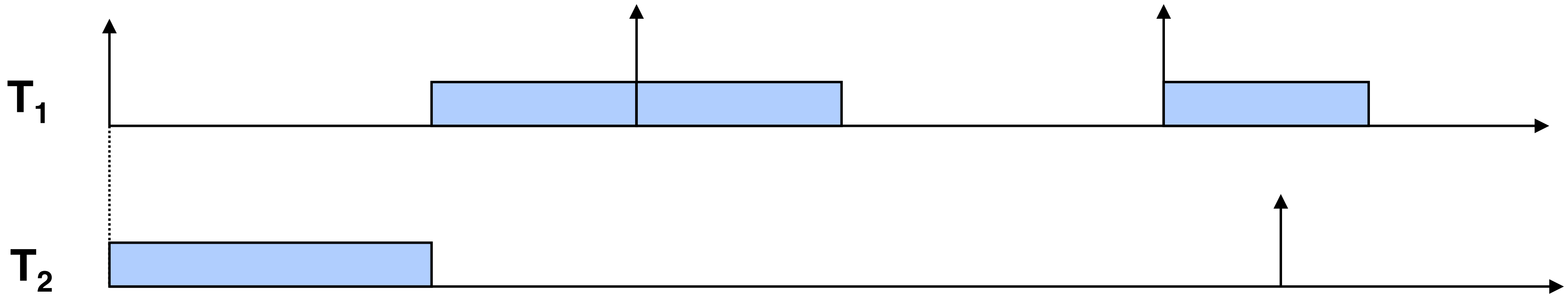
Die maximale (worst-case) Antwortzeit ist gegeben, wenn alle Tasks gleichzeitig aktiviert werden (kritisches Intervall).

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 .

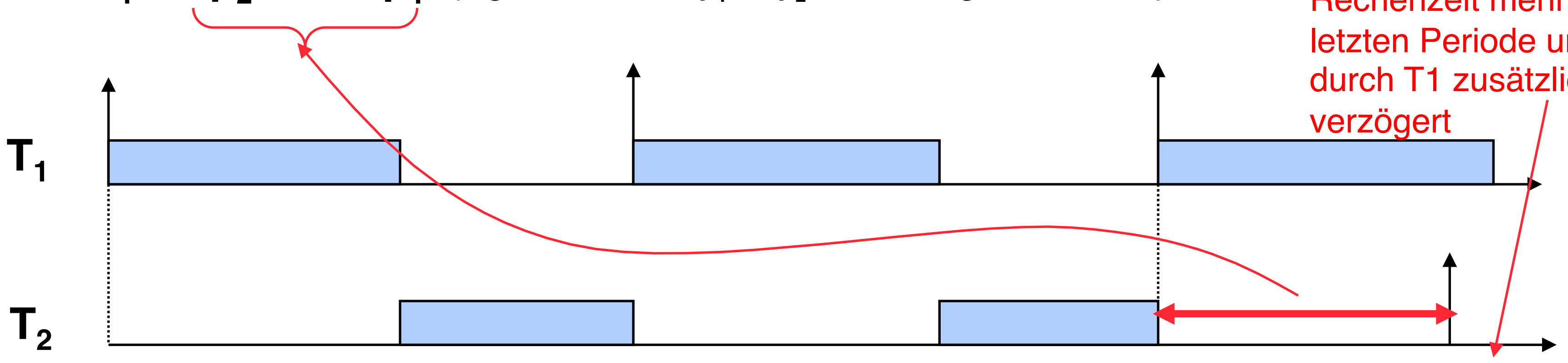
Bedingung, dass die Taskmenge sicher unter dieser Prioritätszuordnung geplant werden kann ist:

$$\boxed{\Delta e_1 + \Delta e_2 \leq \Delta p_1} \qquad \text{[G1]}$$

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

Fall 1: $\Delta e_1 \geq \Delta p_2 - F \cdot \Delta p_1$ (F gibt an, wie oft p_1 in p_2 vollständig enthalten ist)

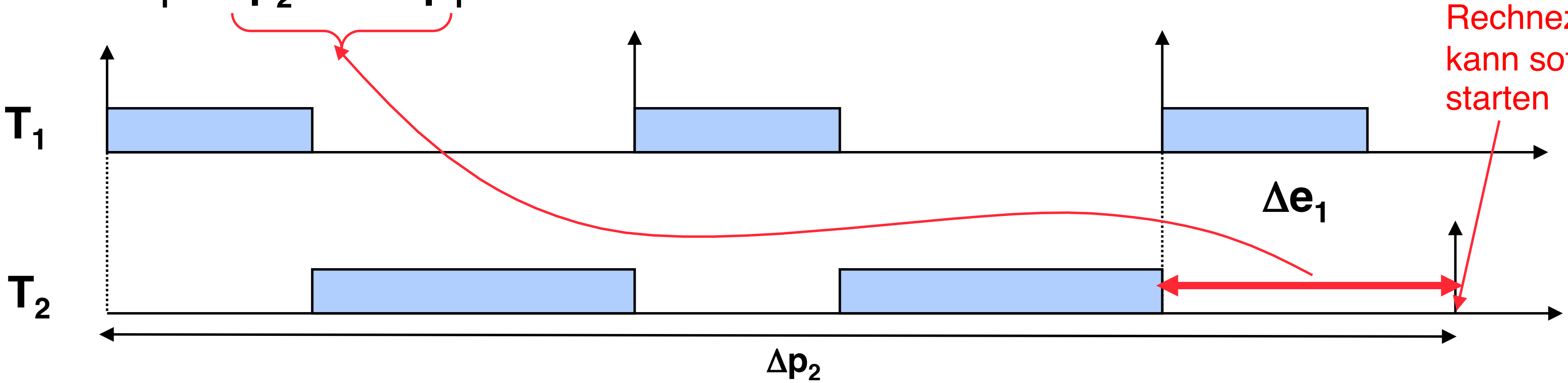
T2 erhält keine Rechenzeit mehr in der letzten Periode und wird durch T1 zusätzlich verzögert



Fall 1: 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.

Fall 2: $\Delta e_1 < \Delta p_2 - F \cdot \Delta p_1$

T2 erhält Rechnezeit und kann sofort starten

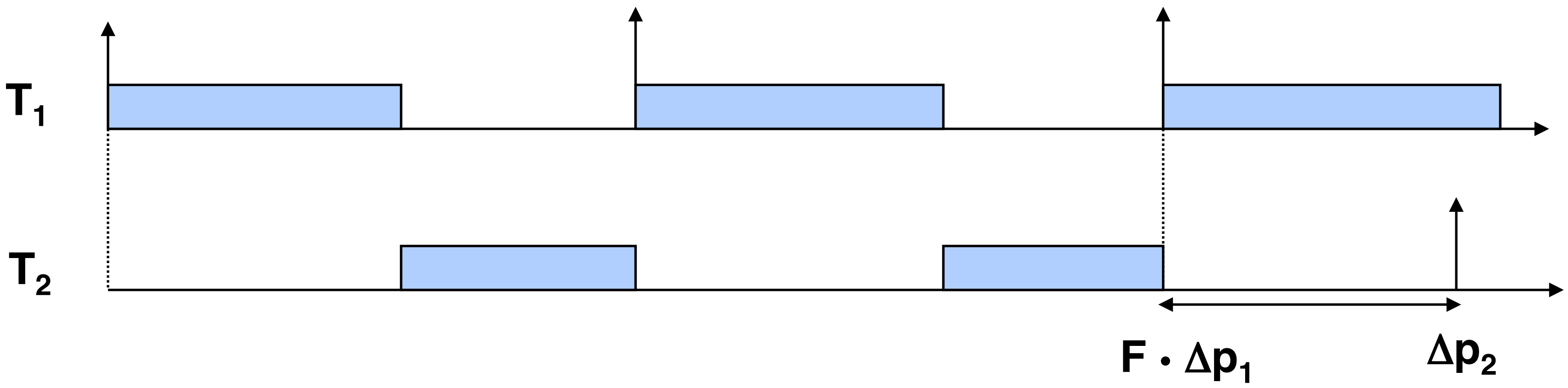


Fall 2: Die Ausführungszeit Δe_1 ist kurz genug damit alle Anforderungen von T_1 innerhalb der kritischen Zone von T_2 beendet werden können, bevor die nächste Anforderung von T_2 gestartet wird.

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: $\Delta e_1 \geq \Delta p_2 - F \cdot \Delta p_1$ **$F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$**



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

$$\boxed{F \cdot \Delta e_1 + \Delta e_2 \leq F \cdot \Delta p_1}$$ [G2]

Zu beweisen ist: $\Delta e_1 + \Delta e_2 \leq \Delta p_1 \Rightarrow F \cdot \Delta e_1 + \Delta e_2 \leq F \cdot \Delta p_1$

$\Delta e_1 + \Delta e_2 \leq \Delta p_1$

$F \cdot \Delta e_1 + F \Delta e_2 \leq F \cdot \Delta p_1$ da $F \geq 1$ gilt ($\Delta p_2 > \Delta p_1$!), können wir schreiben:

$F \cdot \Delta e_1 + \Delta e_2 \leq F \cdot \Delta e_1 + F \cdot \Delta e_2 \leq F \cdot \Delta p_1$ und daher:

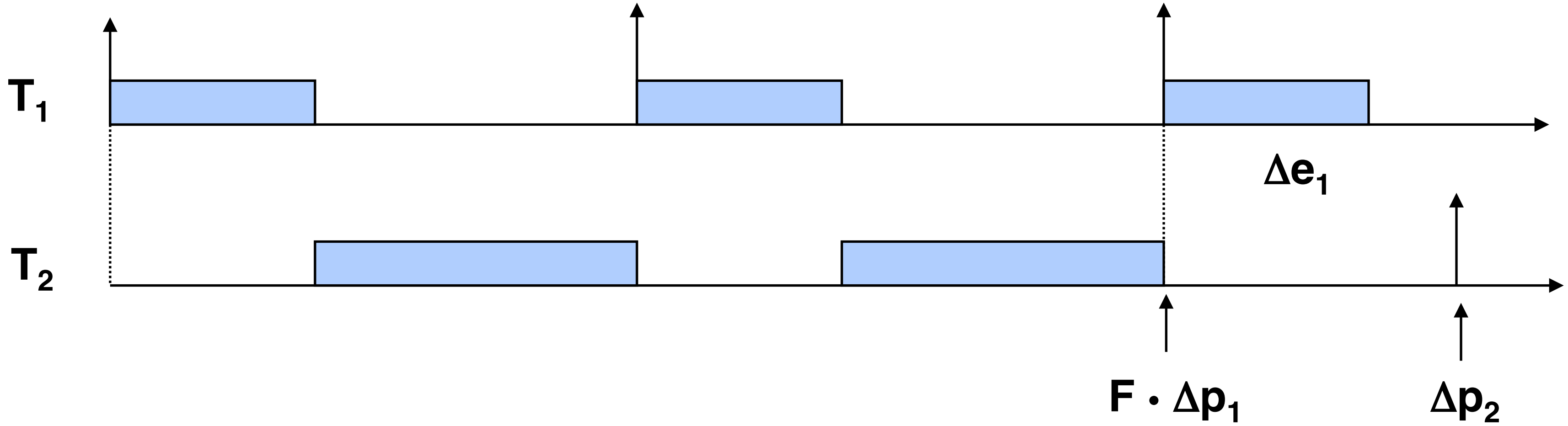
$\boxed{F \cdot \Delta e_1 + \Delta e_2 \leq F \cdot \Delta p_1}$ was die oben gezeigte Implikation beweist.

D.h.

Unter den Bedingungen eines beliebigen Planes kann auch ein Plan durch RM gefunden werden

Beweis der Optimalität

Fall 2: $\Delta e_1 < \Delta p_2 - F \cdot \Delta p_1$ **$F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$** **= Anzahl der vollst. Akt. von T1**



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

$$\boxed{(F+1)\,\Delta e_1 + \Delta e_2 \leq \Delta p_2} \qquad \text{[G3]}$$

Zu beweisen ist: $\overset{G1}{\Delta e_1 + \Delta e_2 \le \Delta p_1} \Rightarrow \overset{G3}{(F+1)\,\Delta e_1 + \Delta e_2 \le \Delta p_2}$

$\Delta e_1 + \Delta e_2 \le \Delta p_1$

$F \bullet \Delta e_1 + F \Delta e_2 \le F \bullet \Delta p_1$ da $F \ge 1$ gilt (wegen $\Delta p_2 > \Delta p_1$), können wir schreiben:

$F \bullet \Delta e_1 + \Delta e_2 \le F \bullet \Delta e_1 + F \bullet \Delta e_2 \le F \bullet \Delta p_1$ $\quad | +\Delta e_1$

$\boxed{F \bullet \Delta e_1 + \Delta e_1 + \Delta e_2 = (F+1)\,\Delta e_1 + \Delta e_2 \le F \bullet \Delta p_1 + \Delta e_1}$ [G4]

Da für den betrachteten Fall gilt: $\Delta e_1 < \Delta p_2 - F \cdot \Delta p_1$ erhalten wir nach Umformung

$\Delta e_1 < \Delta p_2 - F \cdot \Delta p_1$ $\quad | +F \cdot \Delta p_1$

$\Delta e_1 + F \cdot \Delta p_1 < \Delta p_2$ durch Einsetzen in [G4] ergibt sich dann:

$\boxed{(F+1)\,\Delta e_1 + \Delta e_2 \le \Delta p_2}$ was die oben gezeigte Implikation beweist.

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

Die Herleitung 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 $\Delta p_1 < \Delta p_2$.

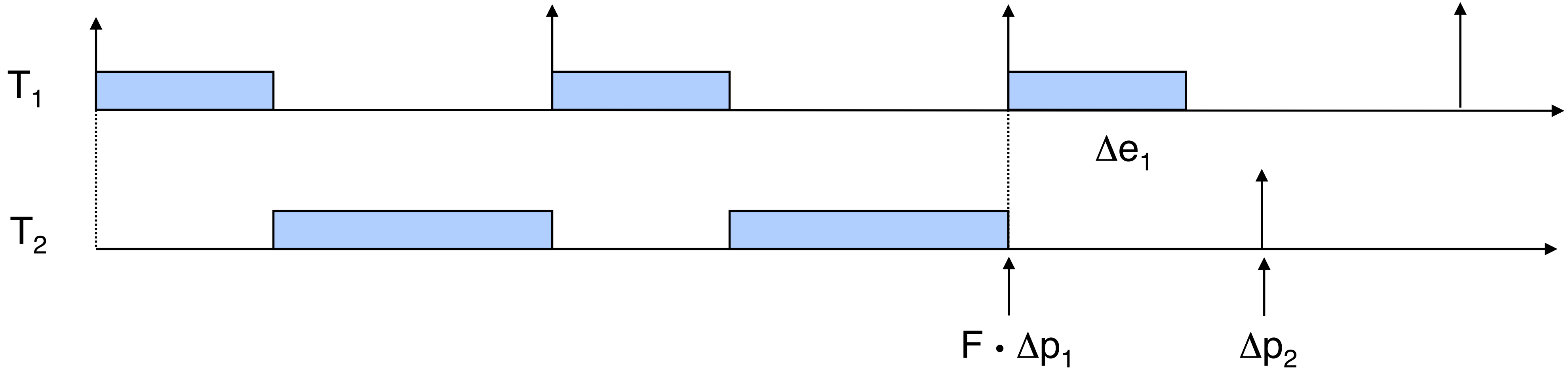
   Es wird angenommen, dass die Ausführungszeit der niedriger priorisierten Task T_2 so festgelegt wird, dass der Prozessor voll genutzt wird. Dann wird die Ausführungszeit der höher priorisierten Task so bestimmt, dass U_{UB} minimal wird.

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.

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.

$\Delta e_1 < \Delta p_2 - F \cdot \Delta p_1$ $\qquad$ $F = \lfloor \Delta p_2 / \Delta p_1 \rfloor$



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

Der größtmögliche Wert von Δe_2 ist gegeben durch: $\Delta e_2 = \Delta p_2 - \Delta e_1 (F+1)$

Die entsprechende obere Schranke ist:

$U_{UB} = \Delta e_1 / \Delta p_1 + \Delta e_2 / \Delta p_2$

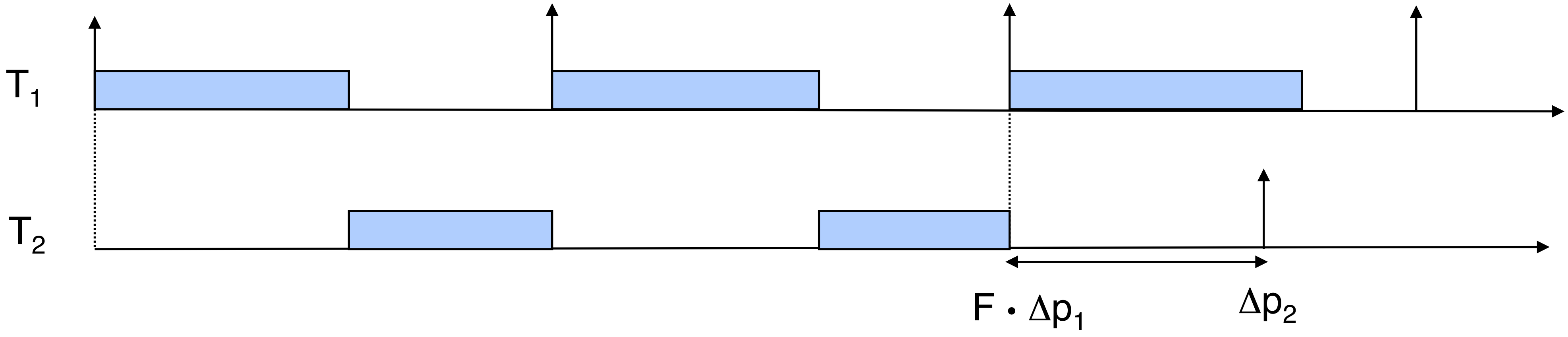
$\quad = \Delta e_1 / \Delta p_1 + \Delta p_2 - (\Delta e_1 (F+1)) / \Delta p_2$ $\qquad$ und nach weiterer Umformung

$\quad = 1 + (\Delta e_1 / \Delta p_2) \cdot [\Delta p_2 / \Delta p_1 - (F+1)]$ $\qquad$ **[G5]** $\qquad$ da der Ausdruck in [...] negativ ist, wird der Gesamtterm mit steigendem Wert für Δe_1 immer kleiner.

Da $\Delta e_1 \leq \Delta p_2 - F \bullet \Delta p_1$ gilt wird U_{UB} mit $\boxed{\Delta e_1 = \Delta p_2 - F \bullet \Delta p_1}$ minimal.

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

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



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

Der größtmögliche Wert von Δe_2 ist gegeben durch: $\Delta e_2 = F \cdot \Delta p_1 - F \cdot \Delta e_1$

Die entsprechende obere Schranke ist:

$$U_{UB} = \Delta e_1 / \Delta p_1 + \Delta e_2 / \Delta p_2$$
$$= \Delta e_1 / \Delta p_1 + (F \cdot \Delta p_1 - F \cdot \Delta e_1) / \Delta p_2$$
$$= (\Delta p_1 / \Delta p_2) \cdot F + \Delta e_1 / \Delta p_2 \cdot [\Delta p_2 / \Delta p_1 - F] \qquad \textbf{[G6]}$$

und nach weiterer Umformung da der Ausdruck in [...] positiv ist, wird der Gesamtterm mit steigendem Wert für Δe_1 größer.

Da $\Delta e_1 \geq \Delta p_2 - F \cdot \Delta p_1$ gilt, wird U_{UB} mit $\boxed{\Delta e_1 = \Delta p_2 - F \cdot \Delta p_1}$ minimal.

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

Da U_{UB} in beiden Fällen bei $\mathbf{\Delta e_1 = \Delta p_2 - F \cdot \Delta p_1}$ 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.

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

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

$$= (\Delta p_1 / \Delta p_2) \cdot [F + (\Delta p_2 / \Delta p_1 - F) \cdot (\Delta p_2 / \Delta p_1 - F)] \quad \textbf{[G7]}$$

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

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

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

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

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.

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

$dU/dG=0$ erhalten wir für:

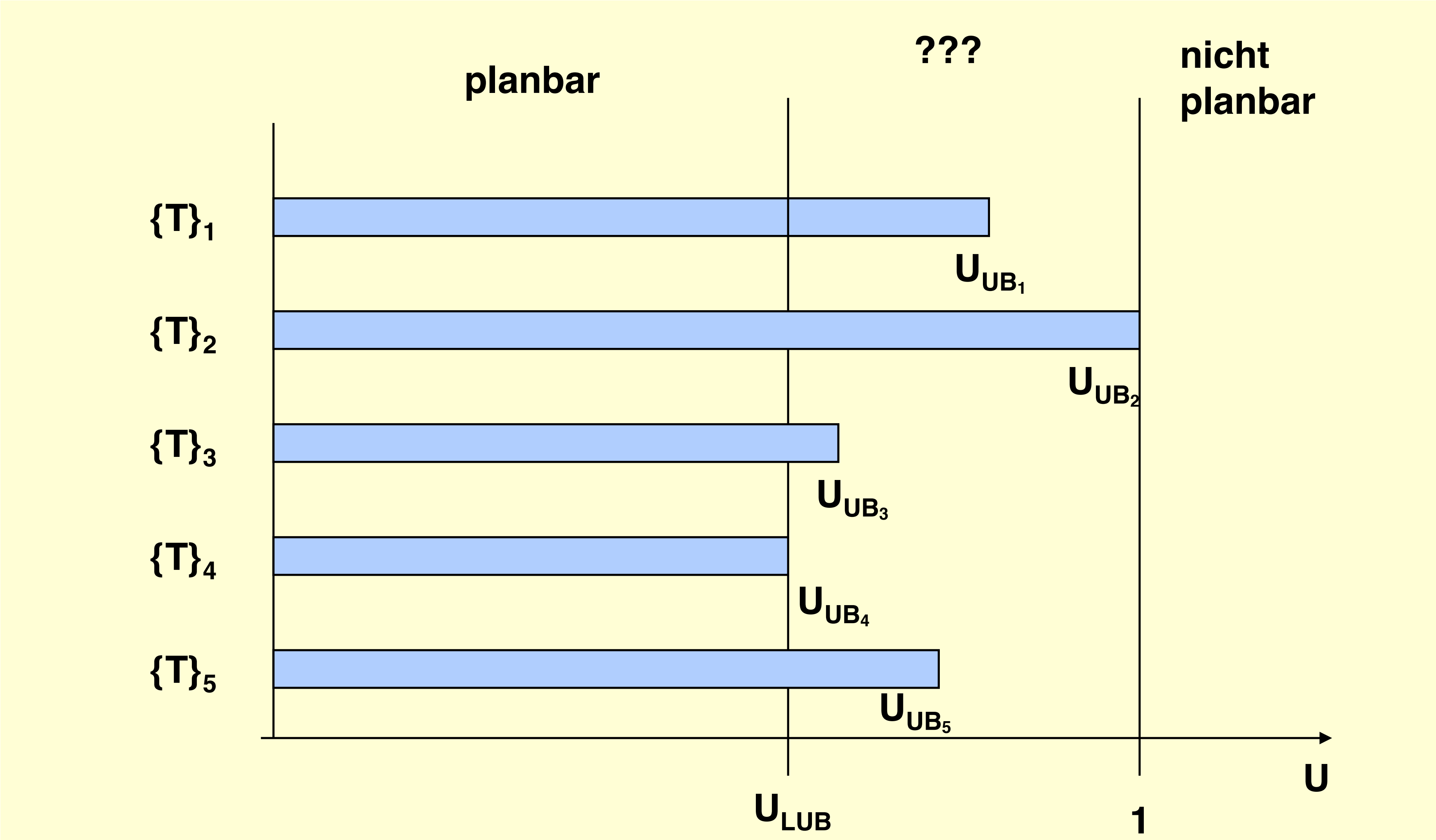
$(G^2 + 2G - 1) = 0$ und damit die Lösungen:

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

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**



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

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

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.

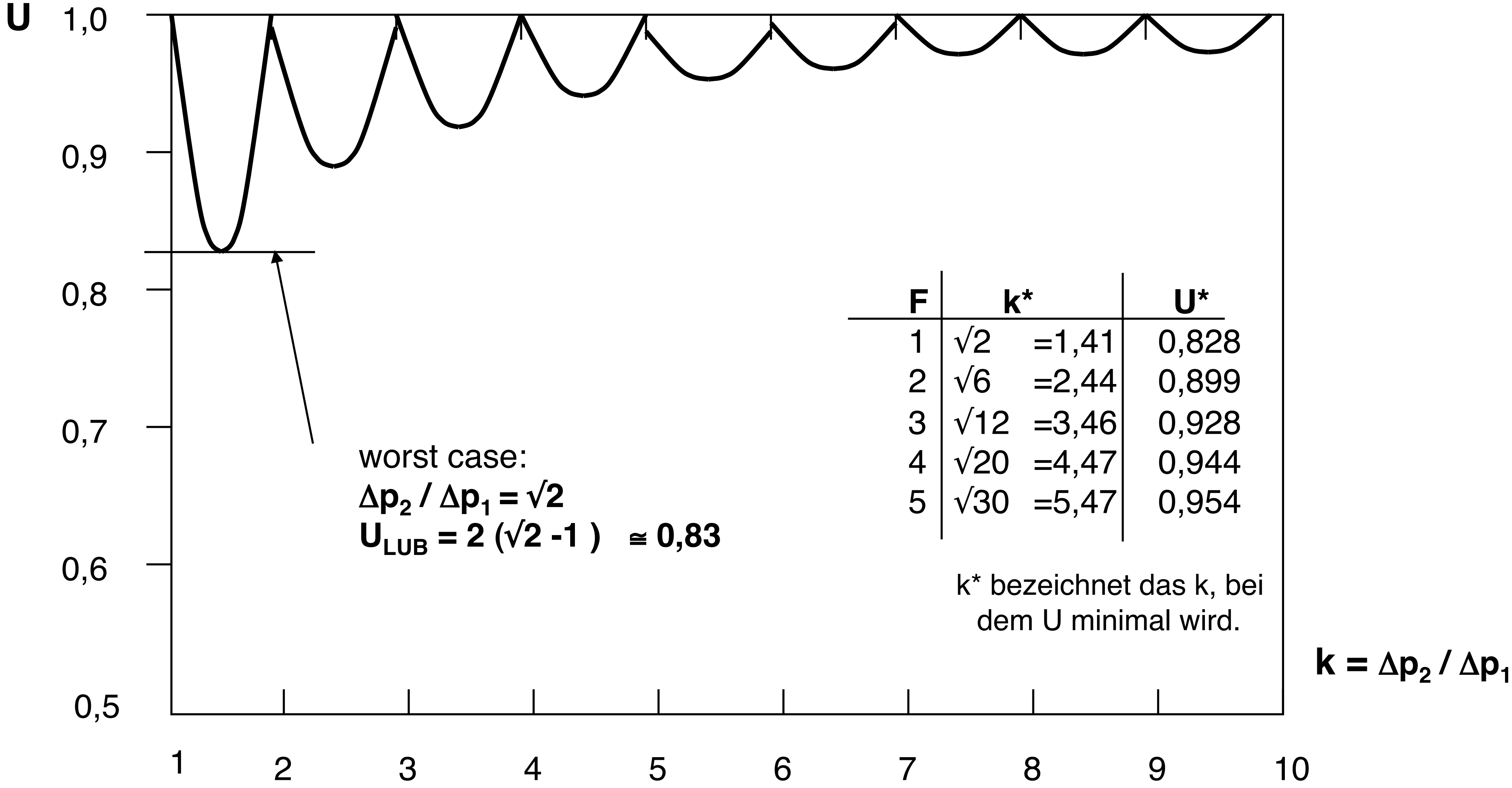
Frage: wie ändert sich allgemein der Auslastungsfaktor U in Abhängigkeit von dem Verhältnis zwischen **Δp_2 und Δp_1**, d.h. in Abhängigkeit von **$k = \Delta p_2 / \Delta p_1$?**

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