Fixed Priority Scheduling of xy-tasks

Maksim Kavalerov

Perm National Research Polytechnic University
Perm, Russia

mkavalerov@gmail.com
Fixed Priority Scheduling of xy-tasks

xy-tasks Model:
1. Special instants (x and y)
2. Vector of external information
3. Quality function and states
4. Release algorithm

Figure from [Martin Stigge, Real-time workload models: Expressiveness vs. analysis efficiency, 2014] slightly adjusted
xy-tasks Model:

1. **Special instants (x and y)**
2. Vector of external information

Vector $\xi$ of information about:
- mode changes,
- changes in control law,
- human interference, etc.

Each vector $\xi$ of the job (task instance) is available no later than finish time ($f$) of the job.

Note: For hard real-time tasks possible values of $\xi$ (and the affect of them on the system) must be known (or estimated) before the system start.
xy-tasks Model:
3. **Quality function and states**
xy-tasks Model:

4. Release algorithm

Release algorithm \((R)\) can be seen as a generalization of the periodic release (activation) of a time-triggered task.
Fixed Priority Scheduling of xy-tasks

xy-tasks Model:
1. Special instants (x and y)
2. Vector of external information
3. Quality function and states
4. Release algorithm

Figure from [Martin Stigge, Real-time workload models: Expressiveness vs. analysis efficiency, 2014] slightly adjusted
Real-life examples of xy-tasks:
1. Event handling time-triggered task
Real-life examples of xy-tasks:
2. Control task with inter-job dependencies
Real-life examples of xy-tasks:

3. Control task with inter-job dependencies and averaged jitters
Real-life examples of xy-tasks:

4. Control task with nominal latency and response-time jitter figures are from [Amir Aminifar, Petru Eles, Zebo Peng, and Anton Cervin, “Stability-aware analysis and design of embedded control systems”, 2013]

\[
\begin{aligned}
&x_{i,j} = O_i + (j-1)h \\
&xy_{i,j} \geq xy_{i,j}^{\min} = L \\
&xy_{i,j} \leq xy_{i,j}^{\max} = (\alpha - 1) L / \alpha + \beta / \alpha \\
&L \geq 0 \\
&L \leq \beta
\end{aligned}
\]

Linear inequalities
Real-life examples of xy-tasks:

5. Control task that produces more linear inequalities

figure is from [Anton Cervin, “Stability and worst-case performance analysis of sampled-data control systems with input and output jitter”, 2012]

But take into account the convexity issue!
Real-life examples of xy-tasks:

5. Control task that produces **quadratic** inequalities

figures are from [Amir Aminifar, Petru Eles, Zebo Peng, and Anton Cervin, “Stability-aware analysis and design of embedded control systems”, 2013]

Linear dependency

\[ L + \alpha \cdot \Delta^w \leq \beta \]

but coefficients can be affected by the frequency \(1/h\) of the task, e.g., linearly:

\[ \alpha = \alpha_1 / h + \alpha_2 \]
\[ \beta = \beta_1 / h + \beta_2 \]

then we have quadratic inequalities in the timing constraint

\[
\begin{align*}
  x_{i,j} - (j-1) \cdot h &= O_i \\
  xy_{i,j} - L &\geq 0 \\
  xy_{i,j} - L - \Delta^w &\leq 0 \\
  L &\geq 0 \\
  L \cdot h - \beta_2 \cdot h &\leq \beta_1 \\
  \Delta^w &\geq 0 \\
  L \cdot h + \alpha_1 \cdot \Delta^w + \alpha_2 \cdot h \cdot \Delta^w - \beta_2 \cdot h &\leq \beta_1
\end{align*}
\]
How to schedule fixed priority xy-tasks: admissible release times
From general form of xy-task timing constraint to more specific, e.g. linear of quadratic
How to schedule fixed priority xy-tasks:
schedulability test for linear form of xy-task constraint

Definition 6. A set \( \{\sigma_{i,j}\}_{f_i,j-1} \) is called admissible and reachable if \( \{\sigma_{i,j}\}_{f_i,j-1} \) is produced by a sequence of admissible \( r_{i,v} \) for \( \forall v \in [1, j-1] \) starting from \( \sigma_{i,1} \).

Theorem 1. For a given \( \Xi_i \), xy-task \( \tau_i \) is guaranteed to be schedulable with any interval \( R_i \) if and only if the following condition holds:

\[
\begin{align*}
rx_i^{Up} - rx_i^{Lo} &\leq \min_{\alpha}(\min_{\beta}(x_{i,j}^{max} - x_{i,j}^{min})) \\
ry_i^{Up} - rx_i^{Lo} &\leq \min_{\alpha}(\min_{\beta}(y_{i,j}^{max} - x_{i,j}^{min})) \\
ry_i^{Lo} - rx_i^{Up} &\geq \max_{\alpha}(\max_{\beta}(y_{i,j}^{min} - x_{i,j}^{max})) \\
ry_i^{Up} - ry_i^{Lo} &\leq \min_{\alpha}(\min_{\beta}(y_{i,j}^{max} - y_{i,j}^{min})) \\
rx_i^{Up} &\leq \min_{\alpha}(\min_{\beta}(x_{i,j}^{max} - (r_{i,j-1} + T_i^{R}))) \\
ry_i^{Up} &\leq \min_{\alpha}(\min_{\beta}(y_{i,j}^{max} - (r_{i,j-1} + T_i^{R}))) \\
xy_i^{Lo} &\geq \max_{\alpha}(\max_{\beta}(xy_{i,j}^{min})) \\
xy_i^{Up} &\leq \min_{\alpha}(\min_{\beta}(xy_{i,j}^{max}))
\end{align*}
\]

where \( \alpha \) stands for “all admissible and reachable \( \{\sigma_{i,j}\}_{f_i,j-1} \) for \( \forall j \geq 1 \)”, and \( \beta \) stands for “\( \{\sigma_{i,j}\}_{f_i,j-1} \)”. 
How to schedule fixed priority xy-tasks: schedulability test for linear form of xy-task constraint (graphical interpretation)
How to schedule fixed priority xy-tasks: schedulability test for linear form of xy-task constraint (reduces to specific xy-tasks)

Response-time test for deadline task is a special case of the proposed schedulability test

Sch. test for control task with inter-job dependencies:

\[
\begin{align*}
T_i^R &\geq \max_{\{\xi_{i,j}\}}(T_{\text{xx}_i}^{\min}(\xi_{i,j})) - r_{x_{i,j}}^{L} + r_{x_{i,j}}^{U} \\
T_i^R &\leq \min_{\{\xi_{i,j}\}}(T_{\text{xx}_i}^{\max}(\xi_{i,j})) + r_{x_{i,j}}^{L} - r_{x_{i,j}}^{U} \\
xy_{i,j}^{U} &\leq \min_{\{\xi_{i,j}\}}(T_{\text{xy}_i}^{\max}(\xi_{i,j}))
\end{align*}
\]
How to schedule fixed priority xy-tasks:

**schedulability test** for linear form of xy-task constraint
(reduces to specific xy-tasks)

Sch. test for event handling time-triggered task:

\[
\begin{cases}
O_i + ry_i^{Up} \leq x_{i,0} + T_{i}^{xy} \max \\
T_i + ry_i^{Up} \leq r_{x_i}^{Lo} + T_{i}^{xy} \max 
\end{cases}
\]
How to schedule **fixed priority** xy-tasks:

**schedulability test** for linear form of xy-task constraint (reduces to specific xy-tasks)

Sch. test for with inter-job dependencies and **averaged jitters**:

\[
\begin{align*}
T_i + r x_i^{L0} & \geq r x_i^{Up} + T_i^{xx \text{ min}} \\
\lambda_i T_i + r x_i^{L0} & \geq r x_i^{Up} + \lambda_i T_i^{mxx \text{ min}} \\
T_i + r x_i^{Up} & \leq r x_i^{Lo} + T_i^{xx \text{ max}} \\
\lambda_i T_i + r x_i^{Up} & \leq r x_i^{Lo} + \lambda_i T_i^{mxx \text{ max}} \\
xy_i^{Up} & \leq T_i^{xy \text{ max}}
\end{align*}
\]

Compare this with response-time test for standard deadline task \( rf_i^{Up} \leq D_i \) or \( ry_i^{Up} \leq D_i \) when \( y_i = f_i \)

Maksim Kavalerov

mkavalerov@gmail.com
For periodic release pattern, period and priority assignment algorithms have been proposed

1. (Test Assignment) Modified optimal priority assignment algorithm [Audsley, 1991]. Instead of response-time feasibility test for derived period-deadline task, the proposed schedulability test is used for original xy-task.

2. (Maximal period assignment) First, priorities are assigned according to deadline-monotonic policy for derived period-deadline task (heuristic approach). Then period of each task is maximized being subject to the proposed schedulability test, while taking into account interference of high priority tasks with periods already assigned.
Experimental evaluation
Experimental evaluation

Fixed Priority Scheduling of xy-tasks

Maksim Kavalerov
mkavalerov@gmail.com
Experimental evaluation
Future research

1. How to find the optimal (or at least, suboptimal) on-line release algorithm $R_i$ or static release pattern for each xy-task under fixed priority policy? Notice that priority assignment is also the problem to solve in this case.

2. How to efficiently schedule the mixture of soft and hard xy-tasks? Notice that $R_i$ can be aware about soft xy-jobs awaiting in the queue.

3. Other approximations of generalized timing constraint of xy-task should be investigated to efficiently resolve the trade-off between the expressiveness of the constraint and the complexity of scheduling, taking into account the practical value of such approximation.