# Schedulability Analysis of Periodic Tasks Implementing Synchronous Finite State Machines

*Haibo Zeng,* Marco Di Natale
*McGill University – Montreal, Canada*
*Scuola Superiore S. Anna – Pisa, Italy*

ECRTS 2012
10-13 July
Pisa, Italy

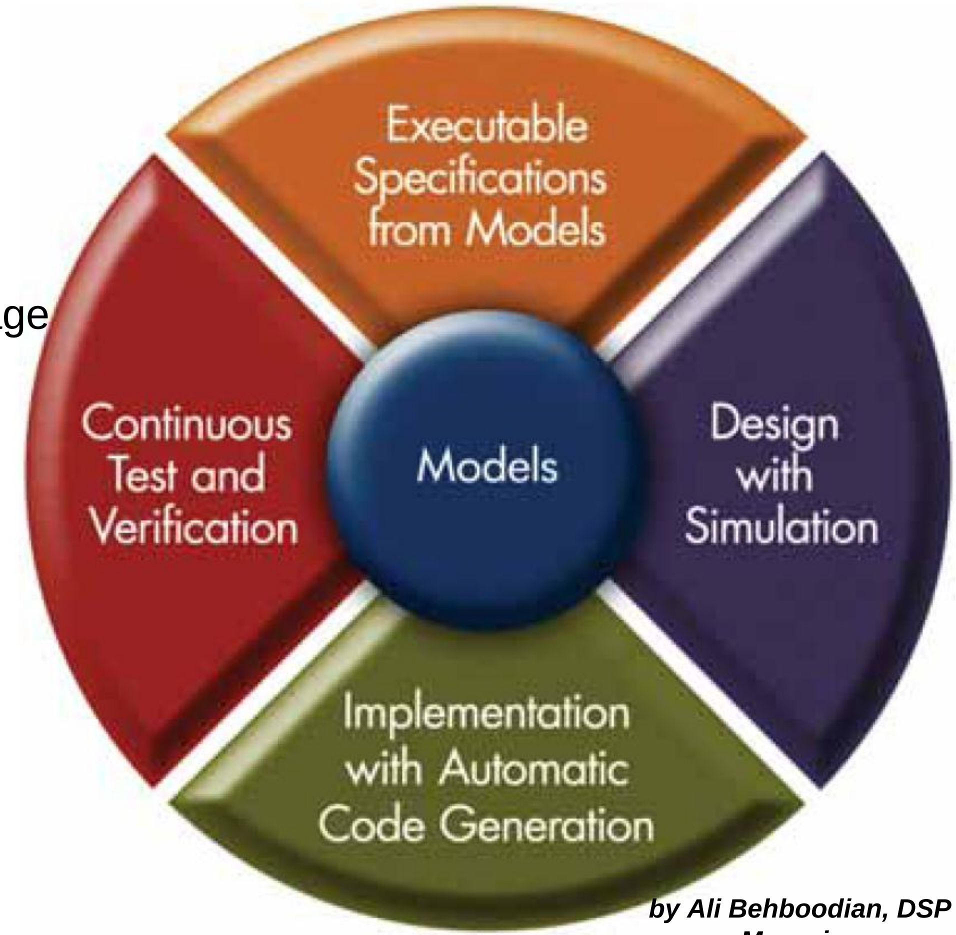
# Outlines

- Motivations
- Synchronous FSMs
  - actions, not tasks
    - Need to map reactions into tasks
  - Applicability of Existing Task Models
- Schedulability Analysis Overview
- Efficient Calculation of RBF and DBF
  - Execution Request Matrix
  - Periodicity of Execution Request Matrix
- Summary and Future Work

# Model-based Design

- Popular in many application domains of real-time systems
  - Automotive
  - Avionics
- To deal with complexity
  - Model everything for design (engineering) and analysis (science)
  - It is necessary to select a modeling language in the most natural and easy way
- The four tenets on the right are fundamental to model-based design

- ***No program by hand***
- ***Starting point is functional model***
- ***Automatic generation of implementation is key***
- ***Synthesis of tasks, priorities, allocation, communication mechanisms …***

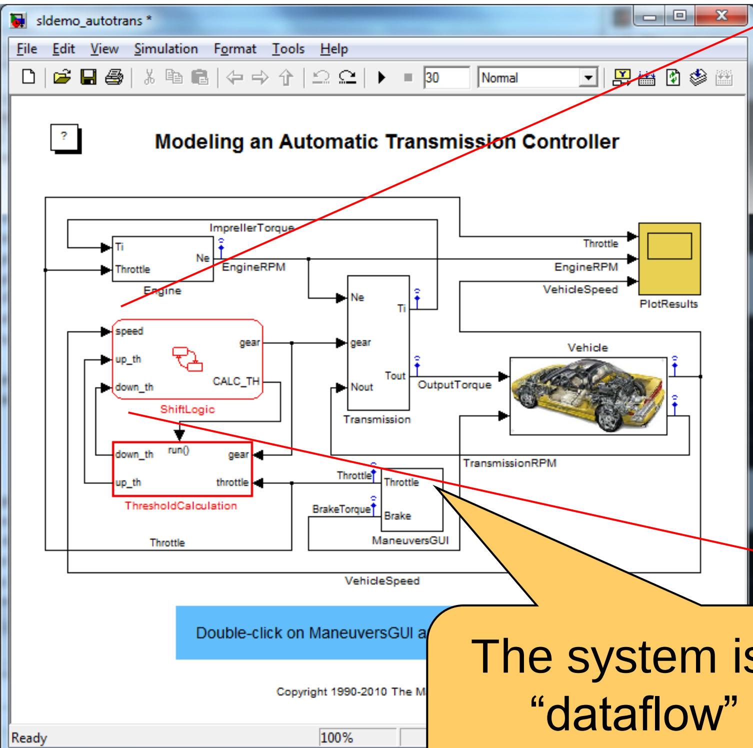


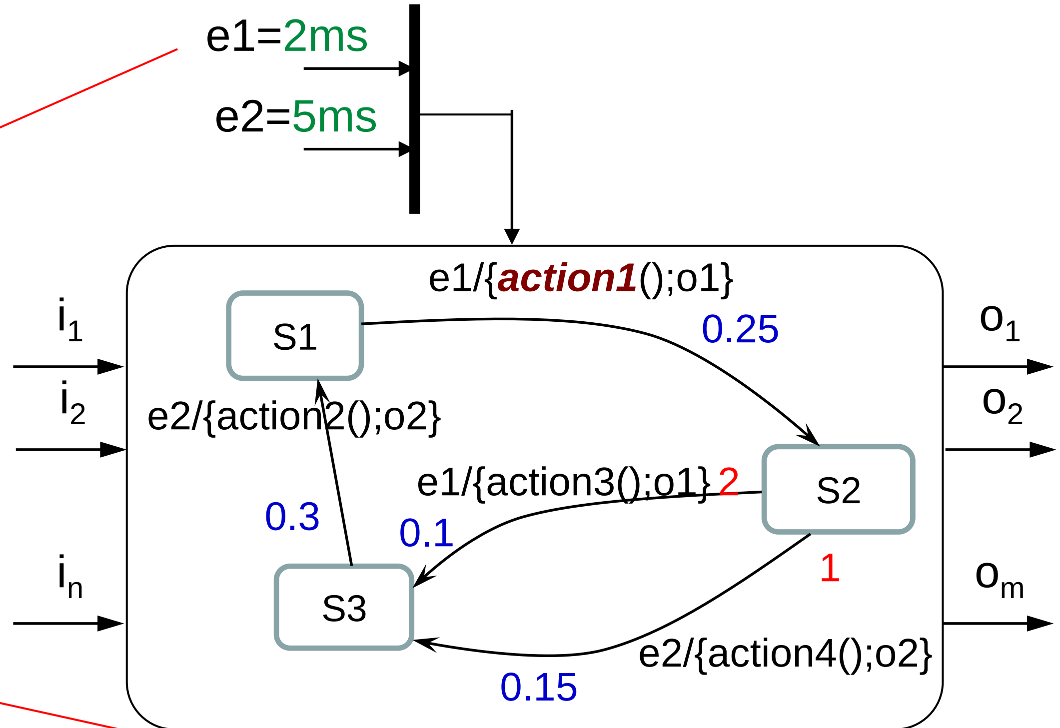

***by Ali Behboodian, DSP Magazine***

*Figure 1 – Elements of model-based design*

# Motivations

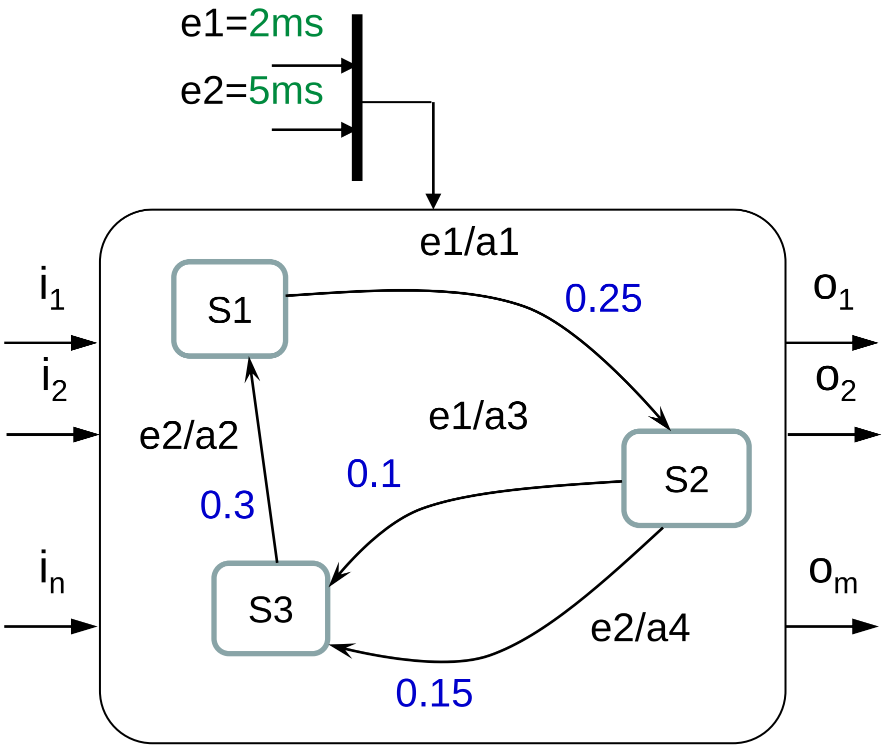
- Synchronous FSMs are used in the most popular model-based design tools
  - SCADE
  - Simulink/Stateflow




The system is a network of “dataflow” blocks and Stateflow (EFSM) blocks

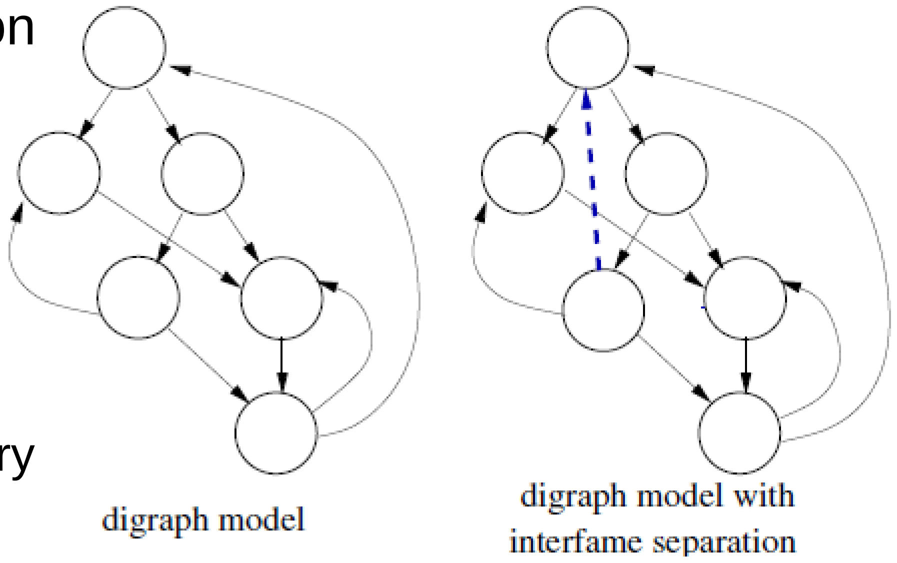
# Synchronous Finite State Machines




- Event $e_i$:
  - Period $T_i$
  - Offset = 0
- State $S_i$
- Transition $S_i$->$S_j$
  - Trigger event
  - Action $a_k$:
    - WCET $C_k$
  - guard, priority
- Hyperperiod $H$ = *lcm* of event periods
- Scheduled with static priority
  - As in commercial code generators (Simulink Coder, dSPACE TargetLink)

# Existing Task Models

- Actions and tasks:
  - Assumption: all actions are executed by a single task
  - *Other options are possible*
- Digraph task model [1] and its extension [2]
  - Accuracy issue
    - Arbitrary offsets
    - Dynamic priority scheduling (EDF)
  - Efficiency issue
    - Patterns of trigger events repeat every hyperperiod
    - Further periodicity by max-plus algebra




[1] M. Stigge et al. "The digraph real-time task model," in Proc. the 16th IEEE Real-Time and Embedded Technology and Applications Symposium, 2011.
[2] M. Stigge et al. "On the Tractability of Digraph-Based Task Models," in Proc. the 23rd Euromicro Conference on Real-Time Systems, 2011.

# Schedulability Analysis Overview

## Schedulability Analysis for Task *i*

FOR each priority level-*i* busy period *[s,f)*

IF $\exists t \in [s,f), \forall t' \in [s,t]$ such that

$$\tau_i.dbf[s,t] + \sum_{j \in hp(i)} \tau_j.rbf[s,t') > t' - s$$

THEN return **unschedulable**

ENDFOR

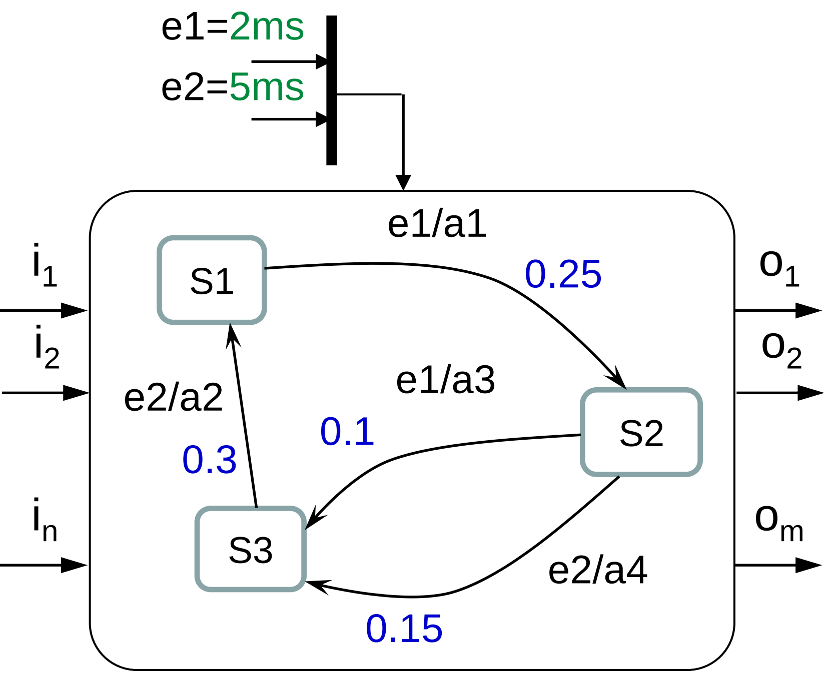
Return **schedulable**

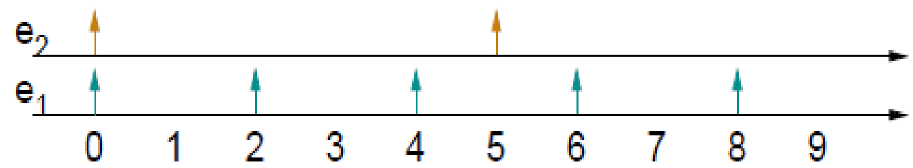
Remaining Question:

how to efficiently calculate *rbf(Δ)* and *dbf(Δ)* for a given time interval Δ?

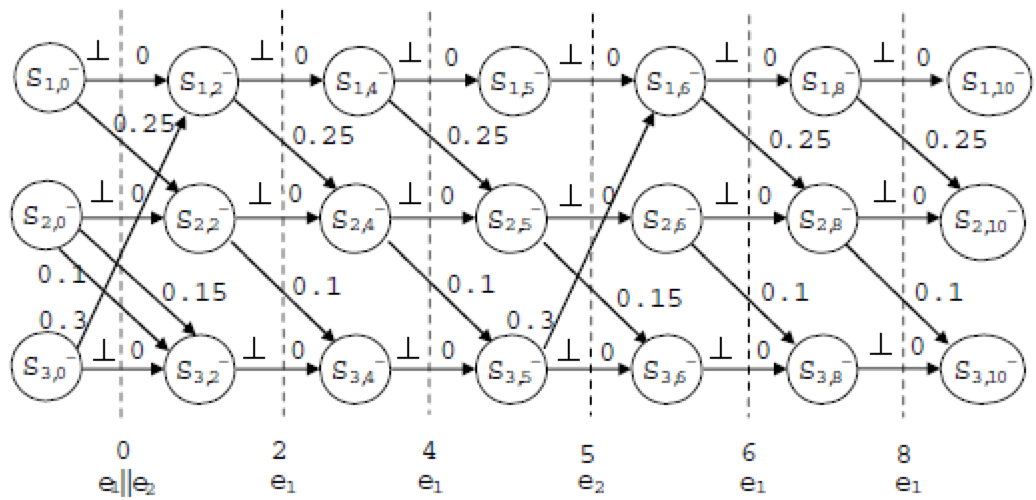
# Event Sequence Pattern and Reachability Graph

Need to compute the request bound function




**event sequence in one hyperperiod**




**reachability graph in one hyperperiod**

# Refinement of rbf

- $rbf_{i,j}(\Delta)$:
  - source state of the first transition is $S_i$
  - sink state of the last transition is $S_j$

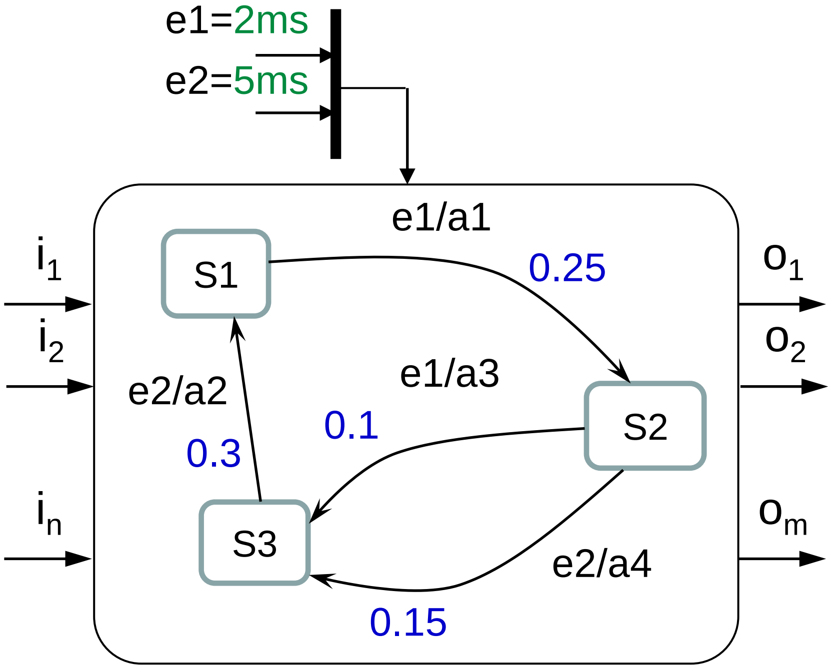
- $rbf_{i,j}(\Delta)$ is ***additive*** (but $rbf(\Delta)$ is not)
  - $rbf_{i,j}[s,f) = \max_m (rbf_{i,m}[s,t) + rbf_{m,j}[t,f))$
  - $rbf_{i,j}[s,f)$ *for a long interval [s, f) can be computed from its values for shorter intervals [s, t) and [t, f)*

Key property to enable dynamic programming techniques

# Execution Request Matrix

- The request bound function in one hyperperiod
  - $X=(x_{i,j})$, where $x_{i,j}=rbf_{i,j}[0,H)$

$$\Longrightarrow X=\begin{bmatrix}0.65 & 0.9 & 1.0\\ 0.45 & 0.7 & 0.8\\ 0.95 & 1.2 & 1.3\end{bmatrix}$$




**event sequence in one hyperperiod**

$$S1 \xrightarrow{e1/a1} S2 \xrightarrow{e1/a3} S3 \xrightarrow{e2/a2} S1 \xrightarrow{e1/a1} S2 \xrightarrow{e1/a3} S3$$

$$\Longrightarrow x_{1,3}=1.0$$


$$\Longrightarrow X=\begin{bmatrix}0.65 & 0.9 & 1.0\\ 0.45 & 0.7 & 0.8\\ 0.95 & 1.2 & 1.3\end{bmatrix}$$

# Execution Request Matrix

- The request bound function over several hyperperiods
  - $X^{(k)} = (x_{i,j}^{(k)})$, where $x_{i,j}^{(k)} = rbf_{i,j}[0,kH)$
  - $\forall i,j, \forall 1 \le l < k \quad x_{i,j}^{(k)} = \max_m (x_{i,m}^{(l)} + x_{m,j}^{(k-l)})$
  - $X^{(k+1)} = \begin{bmatrix} 0.65 & 0.9 & 1.0 \\ 0.45 & 0.7 & 0.8 \\ 0.95 & 1.2 & 1.3 \end{bmatrix} + k \times 1.3$

This indicates some additional periodicity

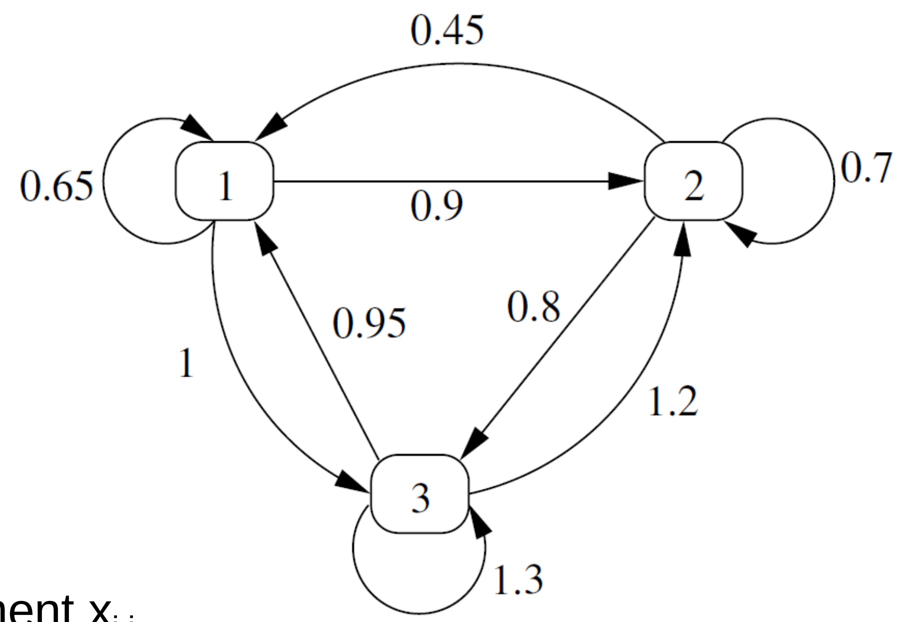
# Basics on Max-Plus Algebra

- Operations maximum (denoted by the max operator $\oplus$) and addition (denoted by the plus operator $\otimes$)
  - $a \oplus b = \max(a, b) \quad a \otimes b = a + b$
- Multiplication of two square matrices
  - $A \otimes B = C$, where $c_{i,j} = \oplus \left(a_{i,m} \otimes b_{m,j}\right) = \max_m(a_{i,m} + b_{m,j})$

# Periodicity of Matrix Power in Max-Plus Algebra

- Studied by its corresponding digraph $\mathcal{G}(X)$

$$X = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0.65 & 0.9 & 1.0 \\ 0.45 & 0.7 & 0.8 \\ 0.95 & 1.2 & 1.3 \end{bmatrix} \end{array}$$

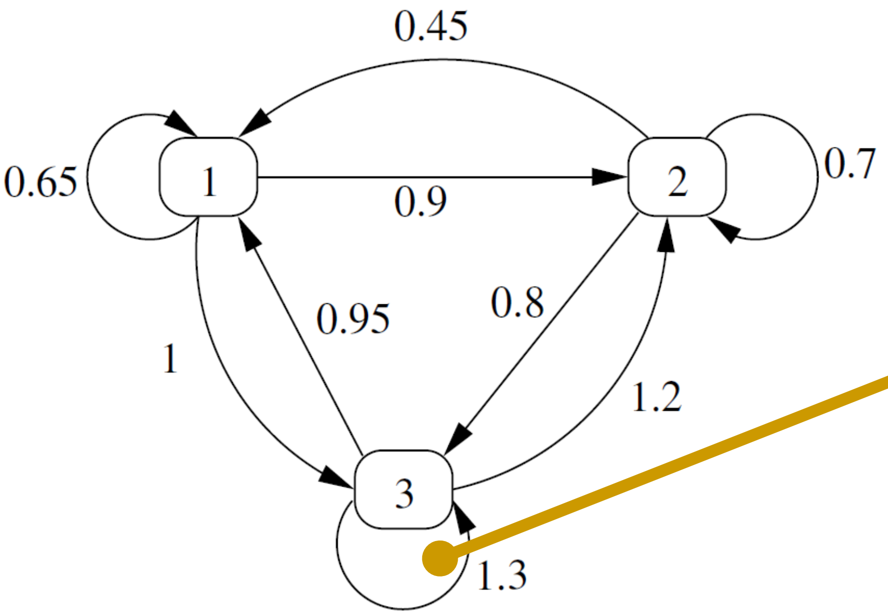



$\mathcal{G}(X)$

Edge (i,j) has weight equal to matrix element $x_{i,j}$

# Periodicity of Matrix Power in Max-Plus Algebra

- If the digraph $\mathcal{G}(X)$ is strongly connected
  - $X^{(k+p)} = X^{(k)} + p \times q$ for sufficiently large $k$
  - $p$ = maximum cycle mean of $\mathcal{G}(X)$
  - $q$ = *lcm* of all the cycles with mean equal to $p$

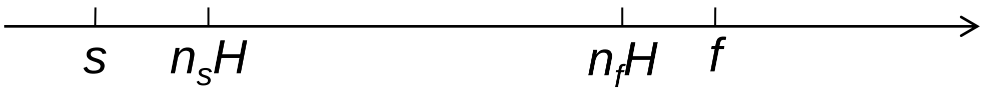



$p=1.3,\ q=1 \implies$

$$X^{(k+1)} = \begin{bmatrix} 0.65 & 0.9 & 1.0 \\ 0.45 & 0.7 & 0.8 \\ 0.95 & 1.2 & 1.3 \end{bmatrix} + k \times 1.3$$

# The Efficient Way of Calculating rbf

- For small intervals




$$
\begin{aligned}
rbf_{i,j}[s,f) = & \max_{k,l}(rbf_{i,k}[s,n_sH) + rbf_{k,l}[n_sH,n_fH) \\
& + rbf_{l,j}[n_fH,f)) \\
= & \max_{k,l}(rbf_{i,k}[s,n_sH) + x_{k,l}^{(n_f-n_s)} \\
& + rbf_{l,j}[0,f-n_fH))
\end{aligned}
$$

- For large intervals: the above equation applies, but

$$
x_{k,l}^{(n_f-n_s)} = x_{k,l}^{(n)} + (n_f - n_s - n) \times q_{k,l}(k)
$$

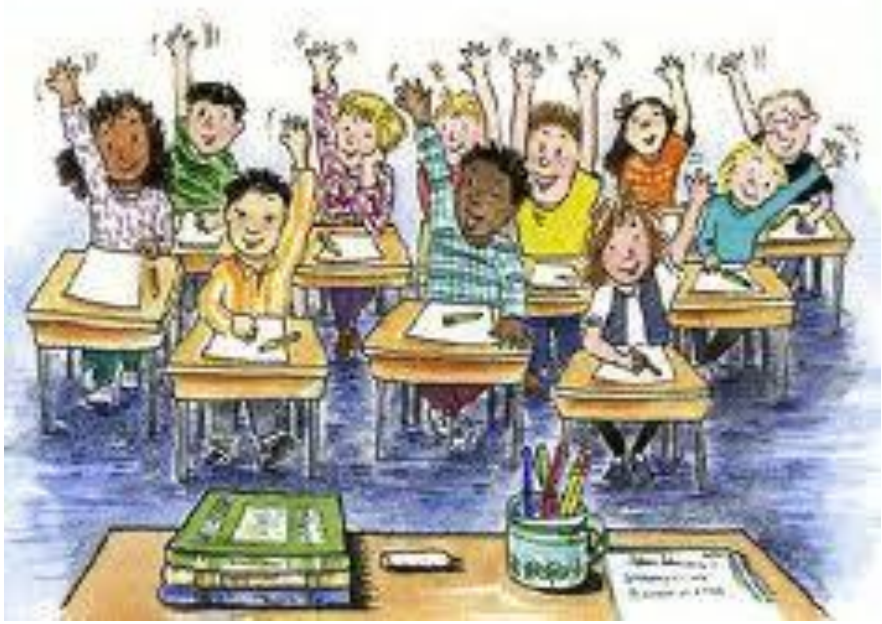
where $n \leq d$ and $n + p > d$, $\;n_f - n_s \equiv n \equiv k \mod p$.

→ Asymptotic complexity independent from length of interval

# Summary and Future Work

- Efficient and accurate schedulability analysis
  - Event sequence pattern within one hyperperiod
  - Max-plus algebra for evaluating the periodicity of the execution request matrix
- Multi-task implementation of an FSM
  - Issues with single task implementation
    - all actions executed at the same priority level
    - tight deadline (equal to the gcd of event periods)
    - inflexible for avoiding overhead from communication
- Extension of periodicity of *rbf* and *dbf* to generic digraph task models

# Thank you!



haibo.zeng@mcgill.ca