

Monitoring Discrete Event Systems Using Petri Net Embeddings

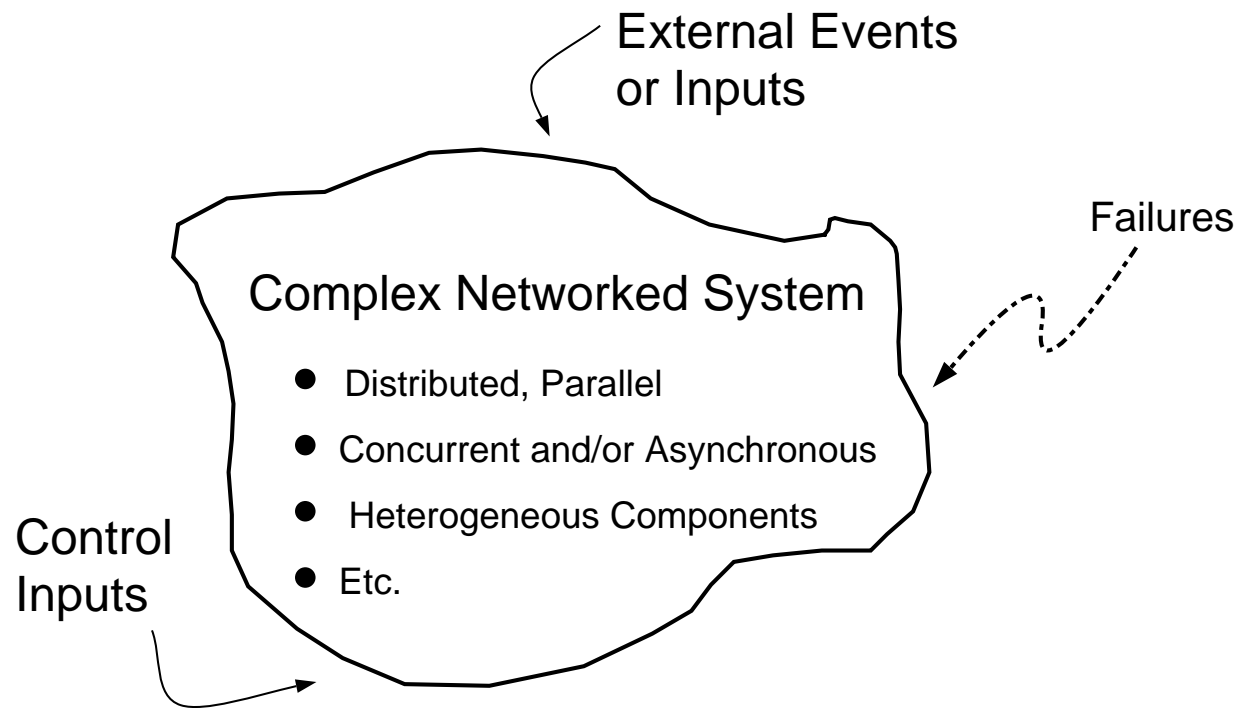
Christoforos Hadjicostis

Supervisor: Prof. George Verghese

Massachusetts Institute of Technology

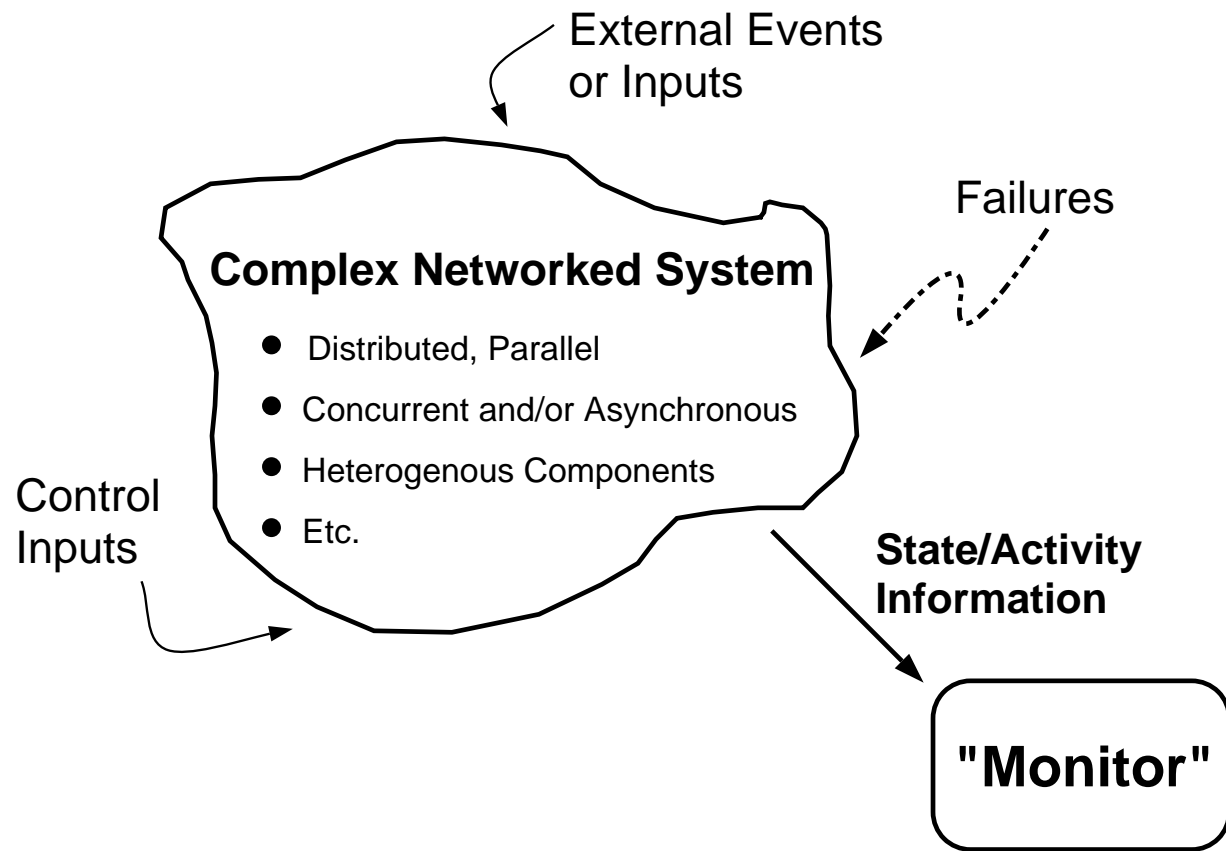
June 1999

FAILURES IN DISCRETE EVENT SYSTEMS



- **Examples:** Manufacturing systems, networked processors, protocols
- **Problem:** How do we model, detect, identify and correct failures?

MONITORING SCHEMES



- **Issues:** Communication cost, monitoring complexity, fault coverage, identification algorithm

DESIRABLE FEATURES FOR MONITORING SCHEMES

Detection and identification of failures should aim for:

- Simple design
- Systematic identification, fault coverage
- Robustness

Optional Features:

- Minimal communication cost, hardware overhead

Other Considerations:

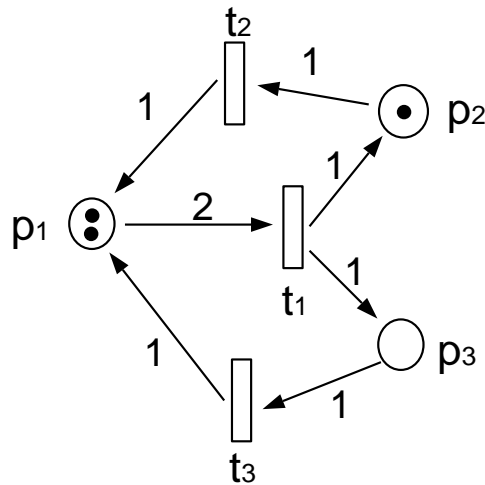
- Distributed and/or hierarchical
- Concurrent or non-concurrent

TALK OUTLINE

- Description of monitoring schemes
- **Petri nets, error model, Petri net embeddings**
- Examples of monitoring schemes
- Conclusions and future research

PETRI NET MODELS FOR DES

If transition t_1 “fires”:



$$\mathbf{q}[1] = \underbrace{\begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}}_{\text{"state"}} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}}_{\text{"postconditions"}} - \underbrace{\begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}}_{\text{"preconditions"}}$$

$$\mathbf{B}^+ = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{B}^- = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{x}[k] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

“Evolution”:

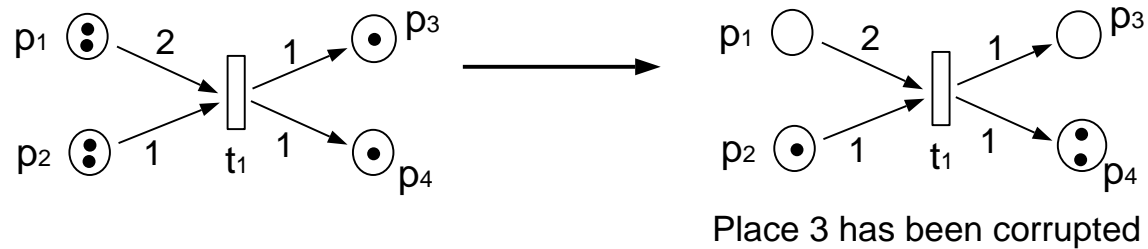
$$\mathbf{q}[k+1] = \mathbf{q}[k] + \underbrace{(\mathbf{B}^+ - \mathbf{B}^-)}_{\mathbf{B}} \mathbf{x}[k]$$

Interpretation depends on underlying DES:

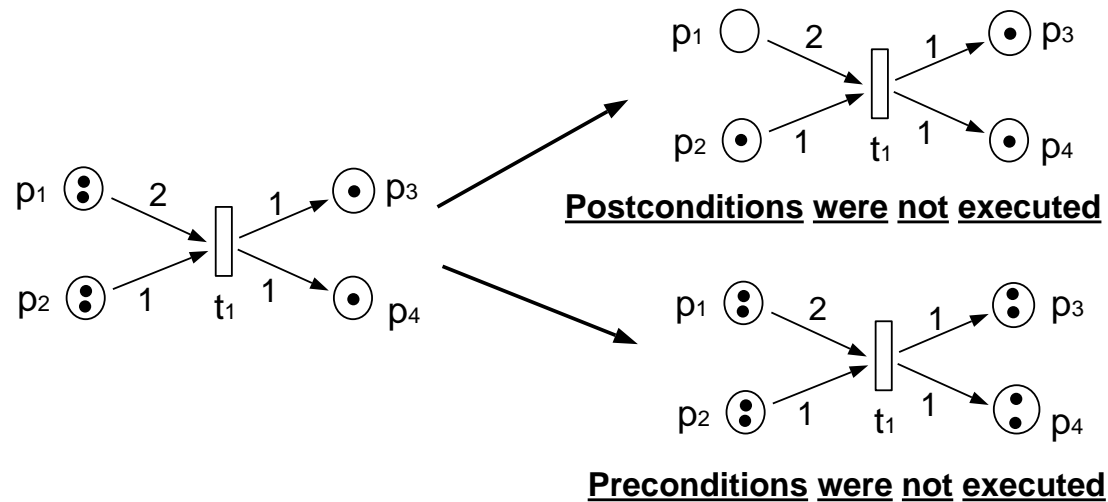
- *Tokens*: system resources, acknowledgments, packets
- *Places*: buffers, storage locations, preconditions, postconditions
- *Transitions*: events, actions, processors, servers, machinery

FAILURE MODELING IN PETRI NETS

Place Failure: Corrupts tokens in a single place



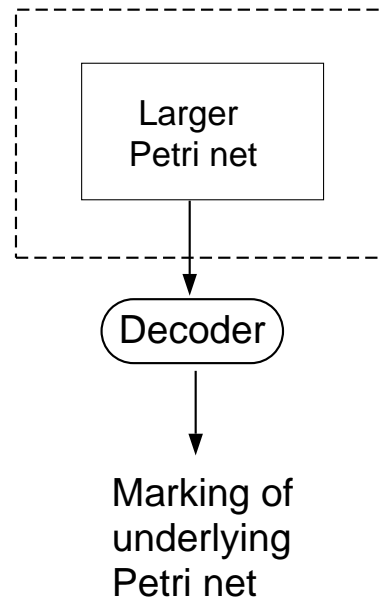
Transition Failure: Ignores *preconditions* OR *postconditions* of a transition



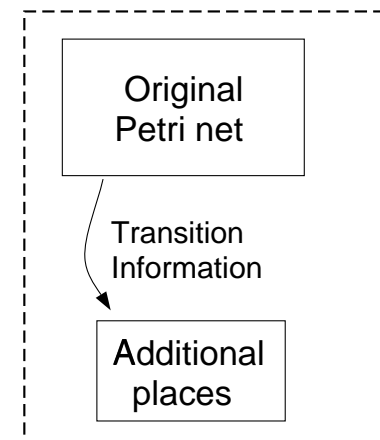
Additive Error Model: $\mathbf{q}_f = \mathbf{q} + \mathbf{e}$ (where \mathbf{q} is the fault-free state)

PETRI NET EMBEDDINGS

Non-Separate
Redundant Petri net
Embedding



Special Case:
Separate Redundant
Petri net Embedding



Embedding retains functionality:

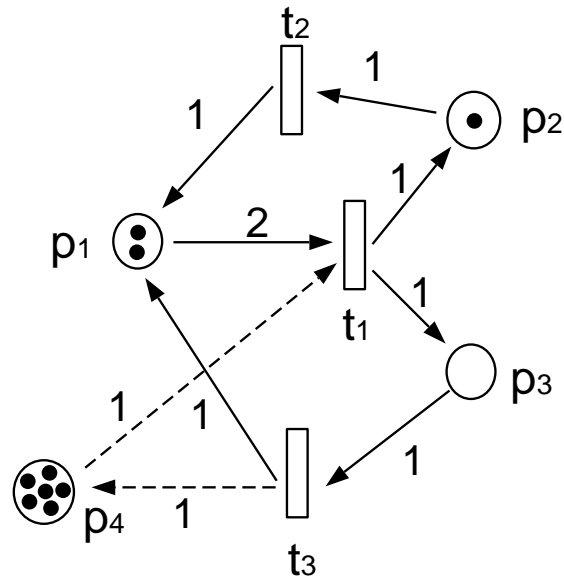
- Recovers marking of original Petri net, accepts same transition (event) sequences

**Goal: Structured and efficient introduction of redundancy
for failure identification**

TALK OUTLINE

- Description of monitoring schemes
- Petri nets, error model, Petri net embeddings
- **Examples of monitoring schemes**
- Conclusions and future research

EXAMPLE OF MONITORING SCHEME



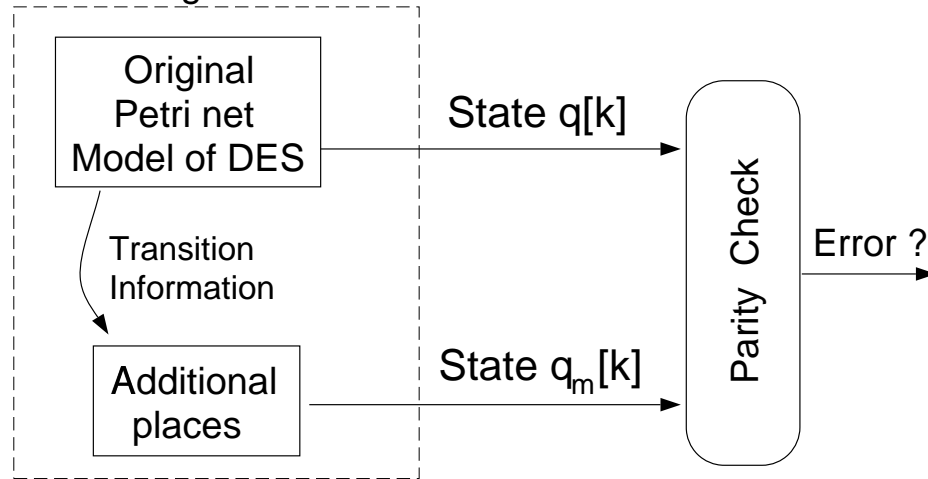
Can detect and identify single transition failures

Check invariant condition: $s[k] = \begin{bmatrix} -2 & -2 & -1 & 1 \end{bmatrix} \xi[k]$

$\xi[k]$ is the marking of the redundant embedding

CONCURRENT MONITORING USING LINEAR CHECKS ON SEPARATE EMBEDDINGS

Separate Petri net
Embedding



Enforce Invariant condition:

$$\mathbf{q}_m[k] = \mathbf{C}\mathbf{q}[k]$$

$\mathbf{q}[\cdot]$ is n -dimensional
 $\mathbf{q}_m[\cdot]$ is d -dimensional

- State evolution of embedding: $\xi[k+1] = \xi[k] + \begin{bmatrix} \mathbf{B}^+ \\ \mathbf{X}^+ \end{bmatrix} \mathbf{x}[k] - \begin{bmatrix} \mathbf{B}^- \\ \mathbf{X}^- \end{bmatrix} \mathbf{x}[k]$

where $\xi[k] = \begin{bmatrix} \mathbf{q}[k] \\ \mathbf{q}_m[k] \end{bmatrix}$

- $\mathbf{X}^+ = \mathbf{C}\mathbf{B}^+ - \mathbf{D}$, $\mathbf{X}^- = \mathbf{C}\mathbf{B}^- - \mathbf{D}$, (\mathbf{C} , \mathbf{D} matrices with integer entries)
- Petri net embedding requires \mathbf{C} , \mathbf{D} , \mathbf{X}^+ , \mathbf{X}^- to be *nonnegative*

SYNDROME-BASED IDENTIFICATION OF TRANSITION AND/OR PLACE FAILURES

- **Parity check / Syndrome generation:** $\mathbf{s}[k] = \underbrace{\begin{bmatrix} -\mathbf{C} & \mathbf{I}_d \end{bmatrix}}_{\mathbf{C}'} \xi_f[k]$

- **Postcondition** failure for transition t_j :

$$\mathbf{s}[k] = \mathbf{D}(:, j)$$

- **Precondition** failure for transition t_j :

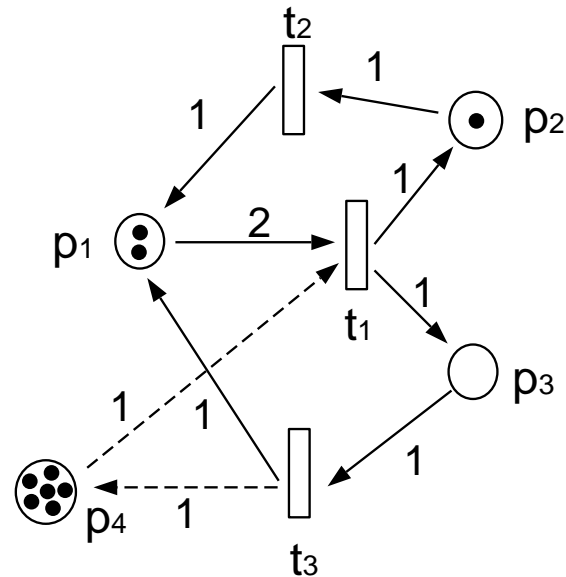
$$\mathbf{s}[k] = -\mathbf{D}(:, j)$$

- **Place** failure at p_i :

$$\mathbf{s}[k] = c \times \mathbf{C}'(:, i)$$

- **Conclusion:** Appropriate \mathbf{C} , \mathbf{D} allow error detection and identification;
applications of linear algebra and coding theory

EXAMPLE: MONITORING TRANSITION FAILURES



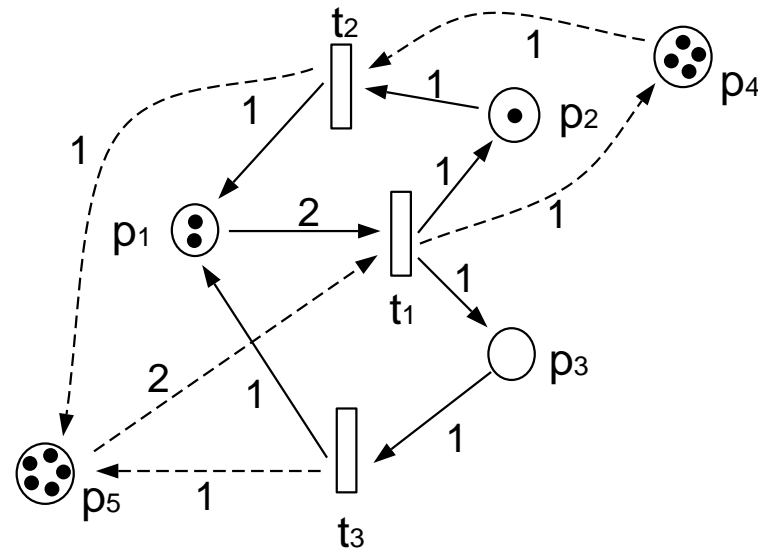
$$\mathbf{C} = \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$$

Detects and identifies single transition failures (columns of D differ)

Parity Check: $\mathbf{s}[k] = \begin{bmatrix} -2 & -2 & -1 & 1 \end{bmatrix} \xi[k]$

E.g., if $\mathbf{s}[k] = 3$, then t_1 has failed to execute postconditions at time epoch k ,
 if $\mathbf{s}[k] = -3$, then t_1 has failed to execute preconditions

EXAMPLE: MONITORING PLACE FAILURES



$$\mathbf{C} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix}$$

Detects and identifies single place failures (due to choice of C)

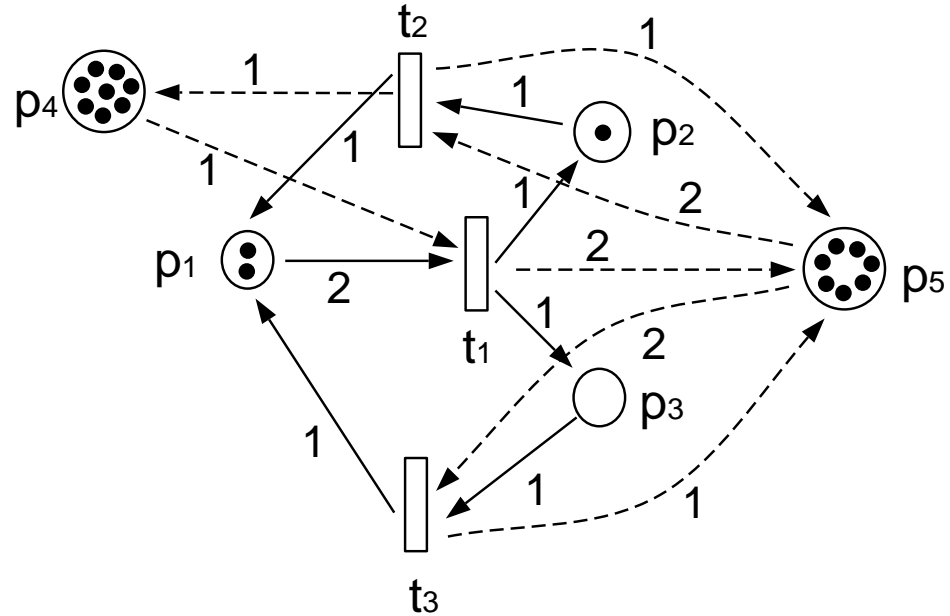
Parity Check: $\mathbf{s}[k] = \begin{bmatrix} -1 & -2 & -1 & 1 & 0 \\ -2 & -1 & -1 & 0 & 1 \end{bmatrix} \xi[k]$

E.g., if $\mathbf{s}[k] = c \times \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, then place p_1 has been corrupted

EXAMPLE: MONITORING TRANSITION AND PLACE FAILURES

$$\mathbf{C} = \begin{bmatrix} 3 & 2 & 3 \\ 2 & 3 & 3 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 5 & 2 & 3 \\ 4 & 1 & 1 \end{bmatrix}$$



Detects and identifies single transition or single place failures

Parity Check: $\mathbf{s}[k] = \begin{bmatrix} -3 & -2 & -3 & 1 & 0 \\ -2 & -3 & -3 & 0 & 1 \end{bmatrix} \xi[k]$

NON-SEPARATE EMBEDDINGS

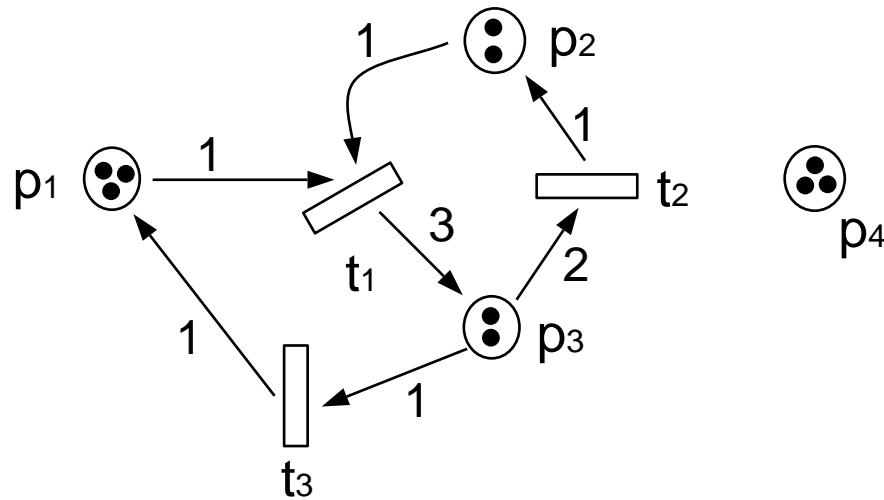
Key features:

- Retain functionality of original Petri net
- Admit same transition sequences
- “Encoded” marking

Advantages:

- Extended possibilities for monitoring schemes
- Flexibility for minimizing communication or hardware requirements

EXAMPLE: NON-SEPARATE EMBEDDING



Detects and identifies single transition failures

Original marking: $\mathbf{q}[k] = \underbrace{\begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & 1 & 1 & -2 \\ -3 & -4 & -2 & 7 \end{bmatrix}}_{\text{"decoding matrix"}} \xi[k]$

Parity check: $\mathbf{s}[k] = \underbrace{\begin{bmatrix} 1 & 2 & 1 & -3 \end{bmatrix}}_{\text{"check matrix"}} \xi[k]$

TALK OUTLINE

- Description of monitoring schemes
- Petri nets, error model, Petri net embeddings
- Examples of monitoring schemes
- **Conclusions and future research**

CONCLUSIONS

Monitoring schemes based on Petri net models:

- Systematic identification, simple design
- Automatic recognition of necessary acknowledgments, connections and weights
- Adjustable to changes in Petri net structure or initial state
- Decoupling of place and transition failures
- Connections with linear algebra and coding

RELATED FUTURE WORK

- **Optimizations:** connections, communication cost, monitor size, etc.
- Hierarchical and/or distributed schemes
- Robustness
- Non-separate vs. separate
- Applications to power systems, manufacturing systems, communication protocols
- Error recovery transitions

SOME RELATED LITERATURE

References

- [1] T. Murata, “Petri nets: properties, analysis and applications,” *Proceedings of the IEEE*, vol. 77, pp. 541–580, April 1989.
- [2] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat, *Synchronization and Linearity*. New York: Wiley, 1992.
- [3] C. G. Cassandras, *Discrete Event Systems*. Boston: Aksen Associates, 1993.
- [4] J. Sifakis, “Realization of fault-tolerant systems by coding Petri nets,” *Journal of Design Automation and Fault-Tolerant Computing*, vol. 3, pp. 93–107, April 1979.
- [5] M. Silva and S. Velilla, “Error detection and correction on Petri net models of discrete events control systems,” *Proceedings of the ISCAS*, pp. 921–924, 1985.
- [6] A. Aghasaryan, E. Fabre, A. Benveniste, and R. Boubour, “A Petri net approach to fault detection and diagnosis in distributed systems (Part 2),” *IEEE Conference on Decision and Control*, pp. 726–731, San Diego, CA, 1997.
- [7] K. L. Lo, H. S. Ng, and J. Trecat, “Distribution fault diagnostic using Petri net theory,” *Universities Power Engineering Conference*, vol. 2, pp. 575–578, London, 1995.