Sponsored by IEEE Singapore SMC, R&A, and Control Chapters

Organized and invited by Professor Sam Ge, NUS

Wireless Sensor Networks for Monitoring Machinery, Human Biofunctions, and BCW Agents

F.L. Lewis, Assoc. Director for Research
Moncrief-O'Donnell Endowed Chair
Head, Controls, Sensors, MEMS Group

Automation & Robotics Research Institute (ARRI)
The University of Texas at Arlington

Singapore Institute of Manufacturing Technology
A*STAR

UTA

NTU National Technological University

Organized and invited by
Jing Bing Zhang

Sponsored by SIMTech & IEEE Singapore Control Chapter

IEEE Networking the World

Matrix Framework for Discrete Event Control

F.L. Lewis, Assoc. Director for Research
Moncrief-O'Donnell Endowed Chair
Head, Controls, Sensors, MEMS Group

Automation & Robotics Research Institute (ARRI)
The University of Texas at Arlington

F.L. Lewis
Moncrief-O'Donnell Endowed Chair
Head, Controls and Sensors Group

Automation & Robotics Research Institute (ARRI)
The University of Texas at Arlington

Discrete Event Control & Decision-Making

http://ARRI.uta.edu/acs

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

UTA

Discrete Event Control

$75K in ARO Funding for Networked Robot Workcell Control
$80K in NSF Funding for research and USA/Mexico Network

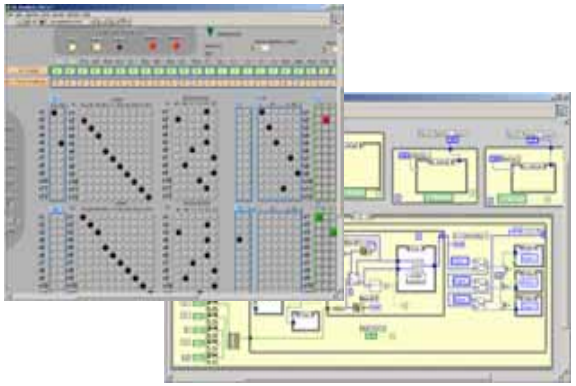
Dr. Jose Mireles- co-PI

Objective:

Develop new DE control algorithms for decision-making, supervision, & resource assignment WITH PROOFS

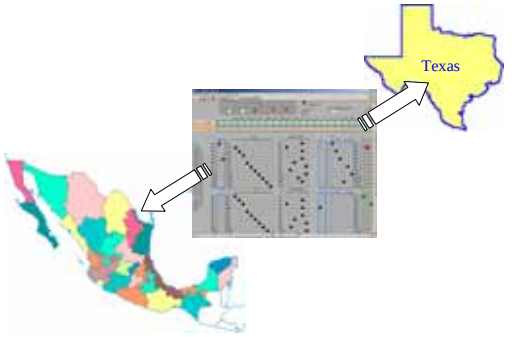
Apply to manufacturing workcell control, battlefield C&C systems, & internetworked systems

Intelligent Robot Workcell



Man/Machine User Interface

- **Patent on Discrete Event Supervisory Controller**
- **New DE Control Algorithms based on *Matrices***
- **Complete Dynamic Description for DE Systems**
- **Formal Deadlock Avoidance Techniques**
- **Implemented on Intelligent Robotic Workcell**
- **Internet- Remote Site Control and Monitoring**
- **USA/Mexico Collaboration**
- **Exploring Applications to Battlefield Systems**



USA/Mexico Internetworked Control

Matrix Formulation: Definition

Based on Manufacturing Bill of Materials

DE Model State Equation: Compare with $x_{k+1}=Ax_k+Bu_k$

$$\bar{x} = F_v \bar{v}_c + F_r \bar{r}_c + F_u \bar{u} + F_D \bar{u}_D$$

Where multiply = AND & addition = OR

where
- x is the task or state logic
- F_v is the job sequencing matrix (Steward)
- F_r is the resource requirements matrix (Kusiak)
- F_u is the input matrix
- F_D is the conflict resolution matrix

Job Start Equation: $V_s = S_v x$

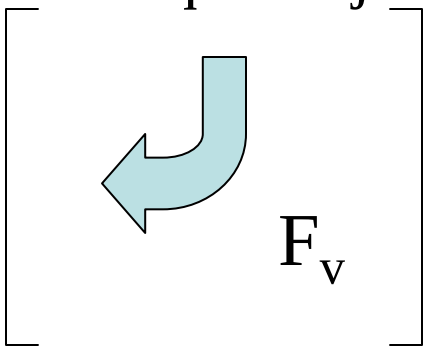
Resource Release Equation: $r_s = S_r x$

Product Output Equation: $y = S_y x$

Meaning of Matrices

Prerequisite jobs

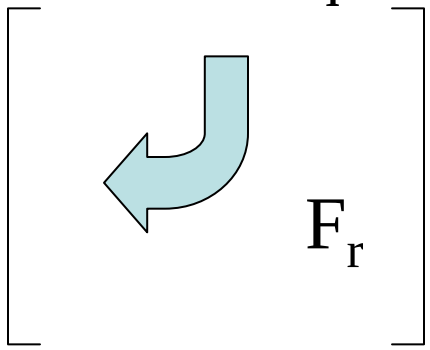
Next job



Steward's Task Sequencing Matrix

Resources required

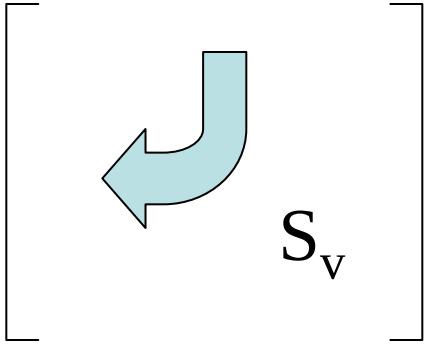
Next job



Kusiak's Resource Requirements Matrix
Bill of Materials (BOM)

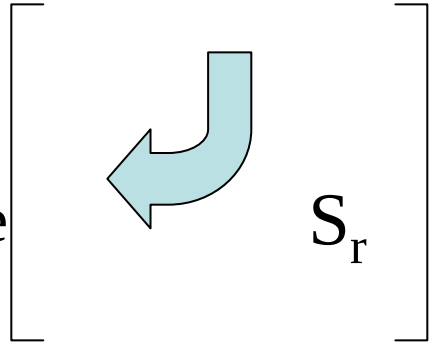
Conditions fulfilled

Next job



Conditions fulfilled

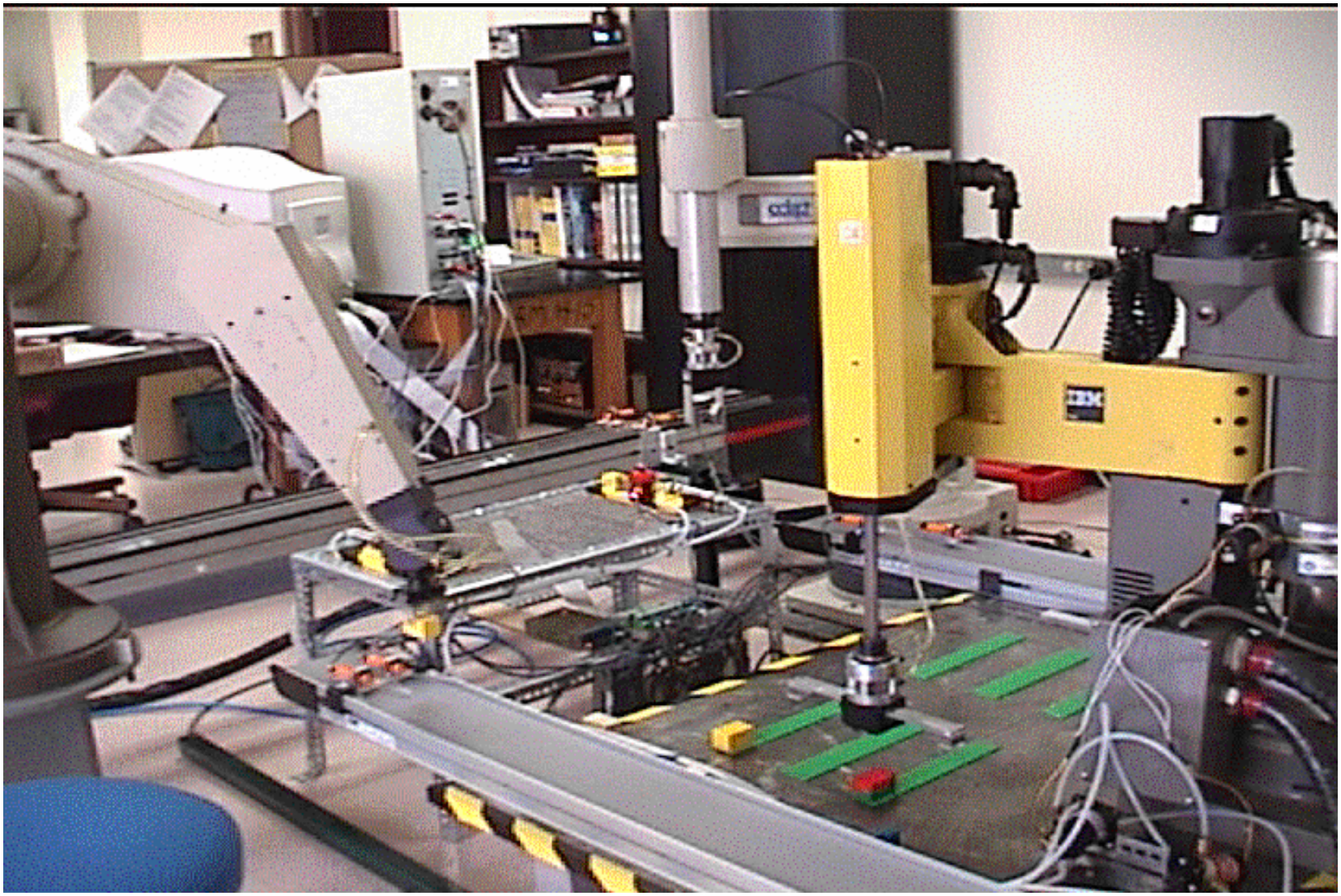
Release resource



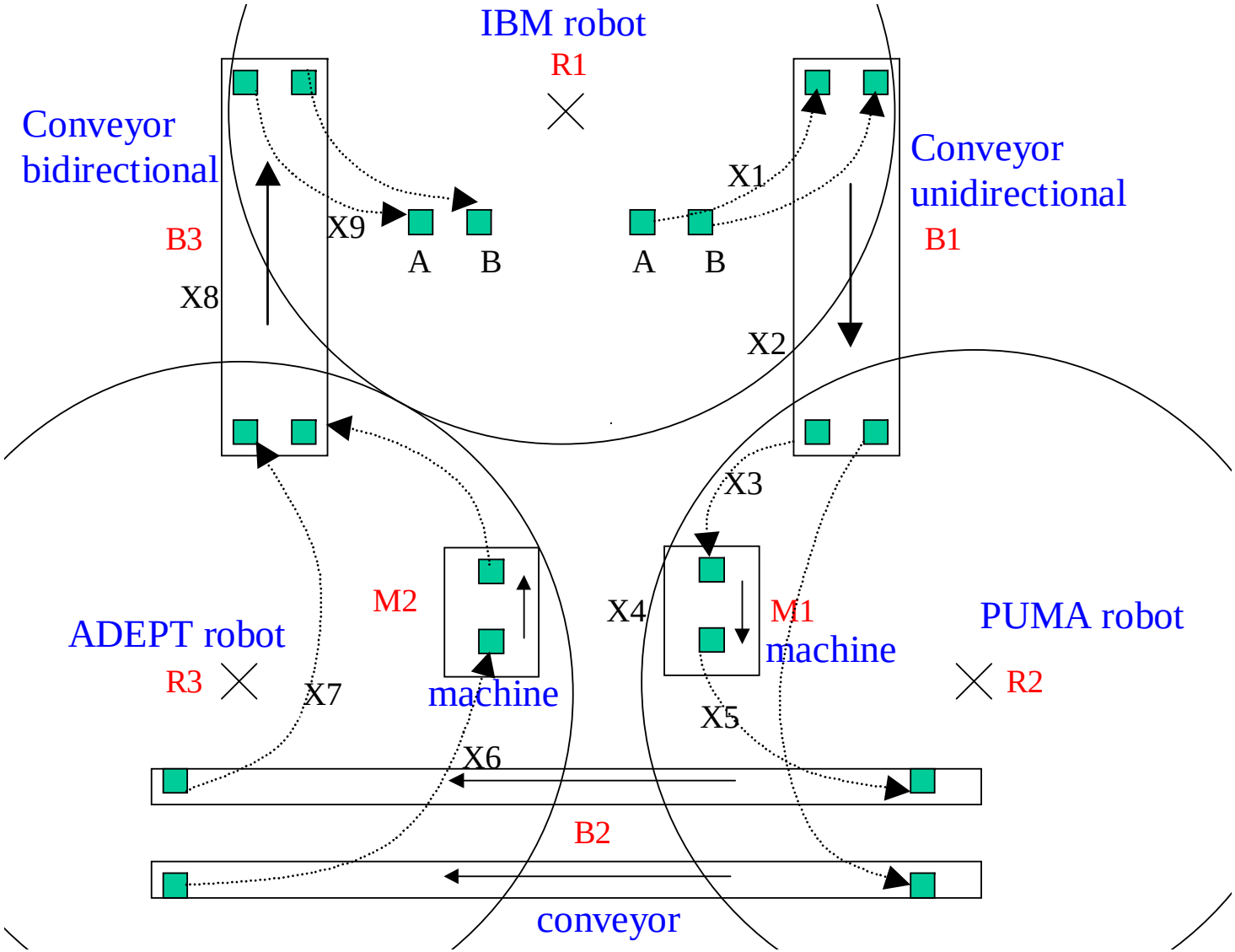
EXAMPLE

ARRI Intelligent Material Handling (IMH) Cell

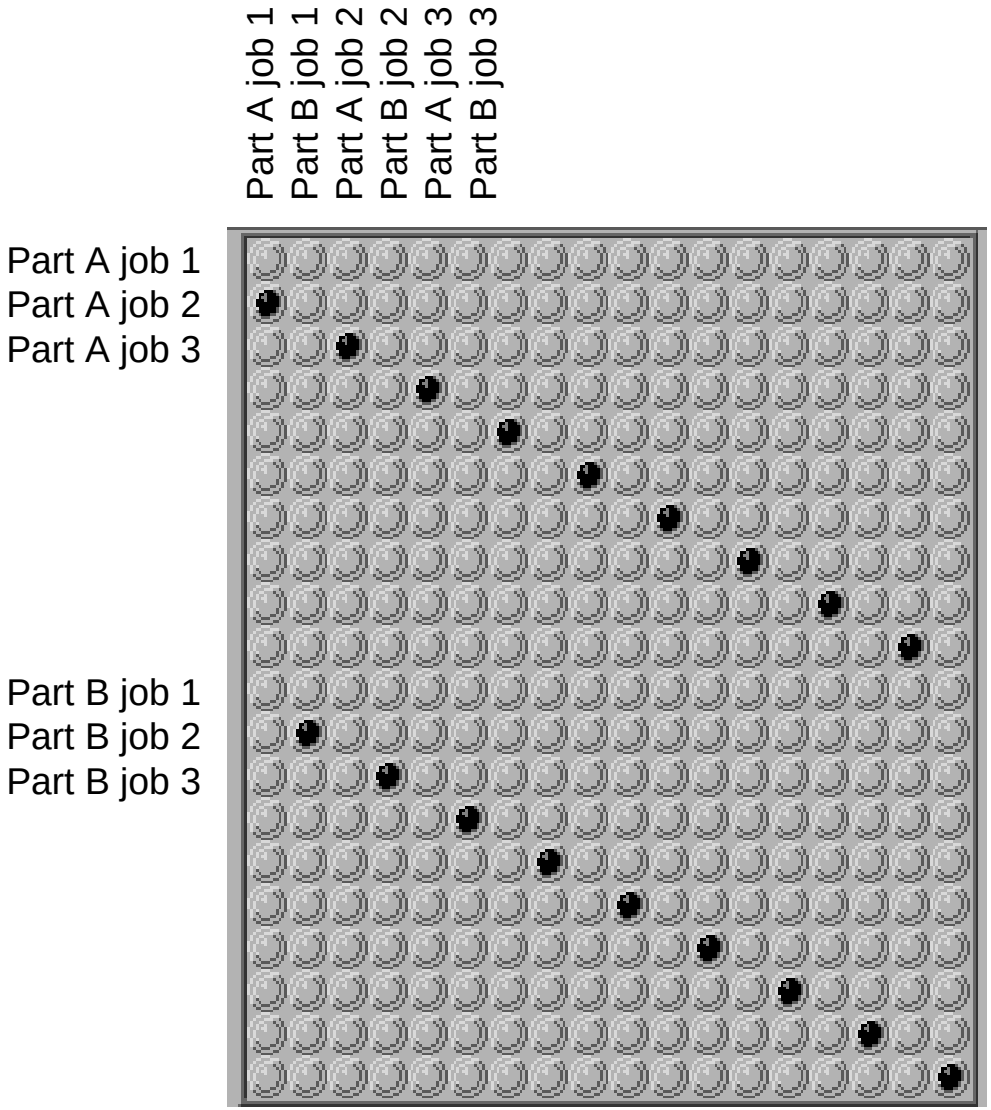
3 robots, 3 conveyors, two part paths



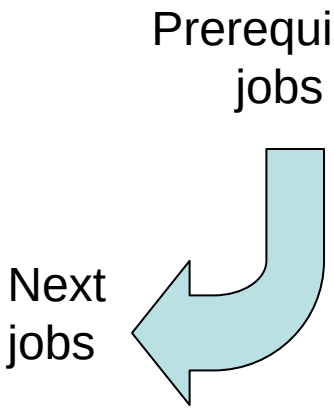
Layout of the IMH Cell



Construct Job Sequencing Matrix F_v

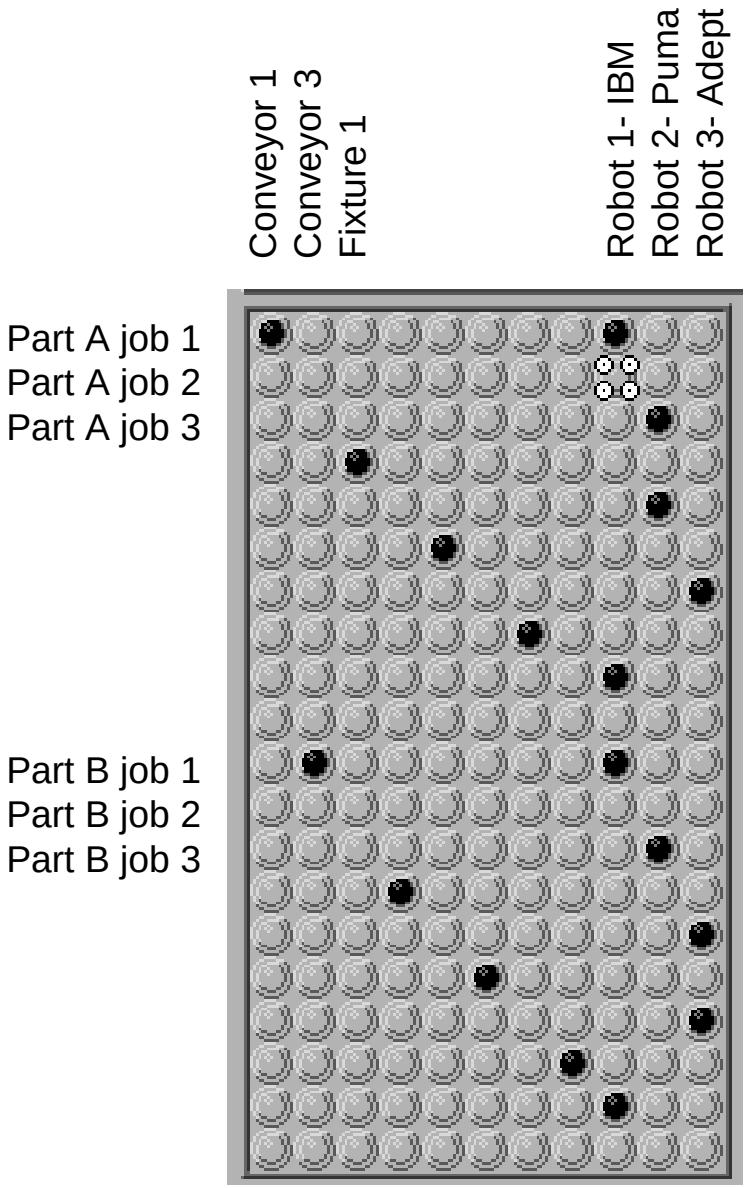


Used by Steward in Manufacturing Task Sequencing



Contains same information as the Bill of Materials (BOM)

Construct Resource Requirements Matrix F_r



Used by Kusiak in Manufacturing Resource Assignment

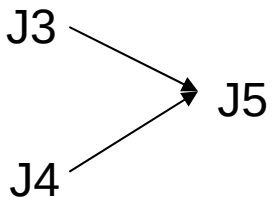
Contains information about factory resources

Prerequisite resources

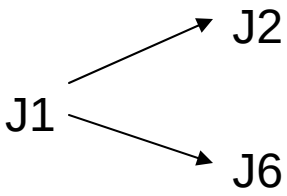
Next jobs

More About F_v

	J1	J3	J4
J2	1	0	0
J5	0	1	1
J6	1	0	0



Two 1's in same row = Assembly



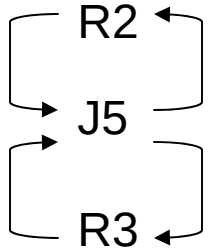
DECISION NEEDED!

Two 1's in same col. = Routing (Job Shop)

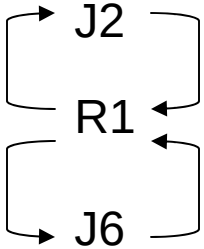
More About F_r

	R1	R2	R3
J2	1	0	0
J5	0	1	1
J6	1	0	0

Two 1's in same row = Job needs multiple res.

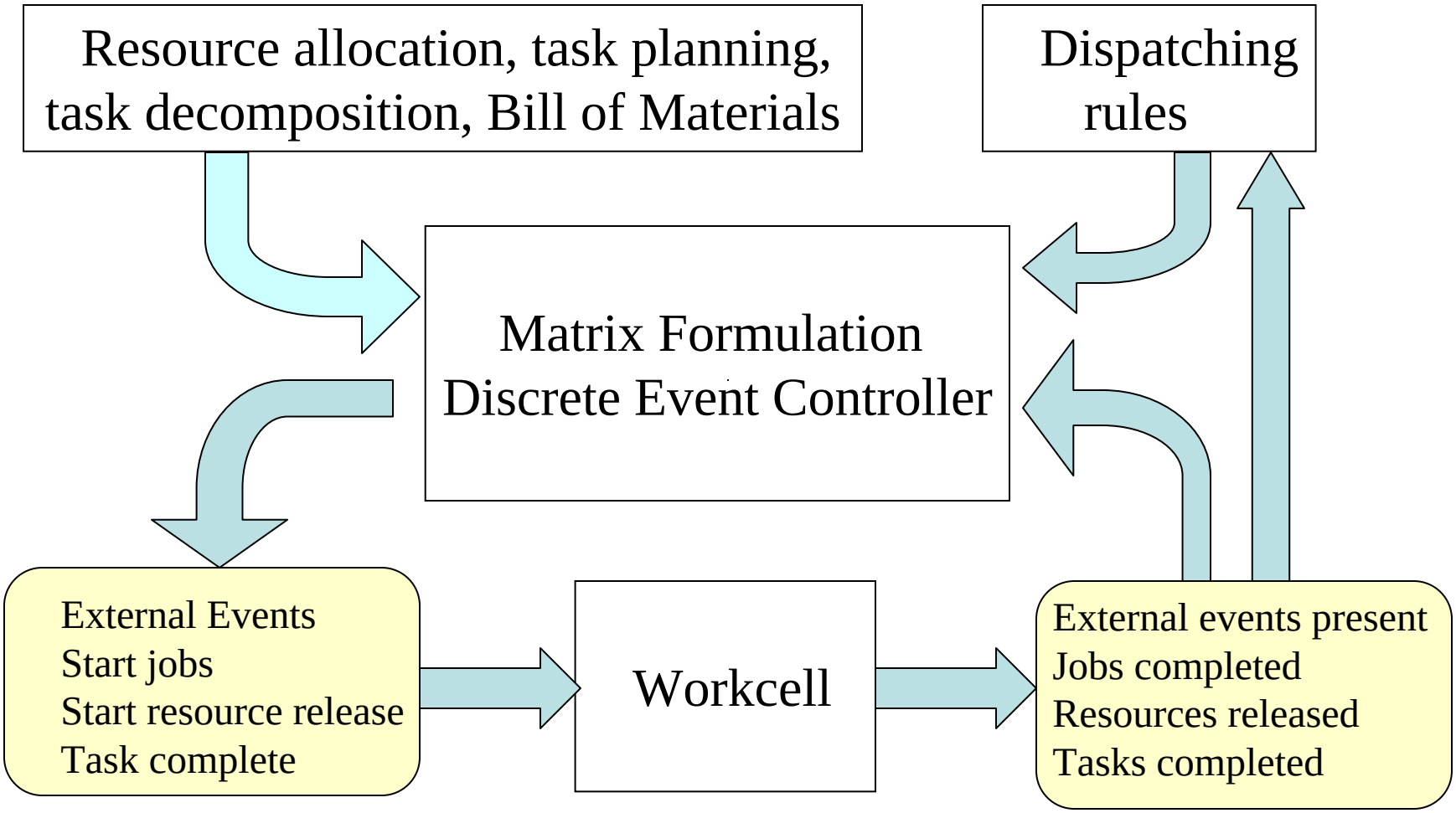


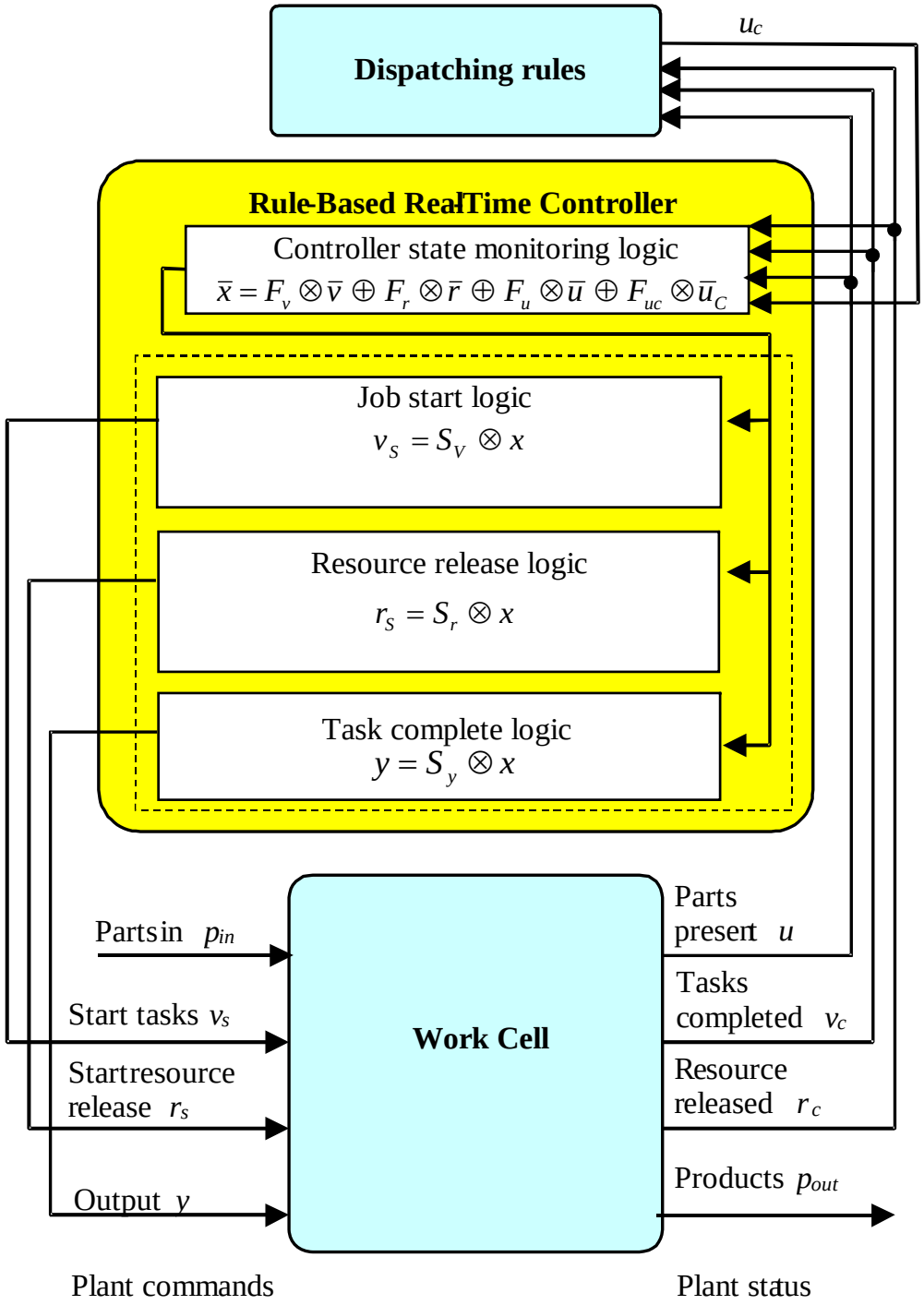
Two 1's in same col. = Shared Resource



DECISION NEEDED!

Controller based on Matrix Formulation

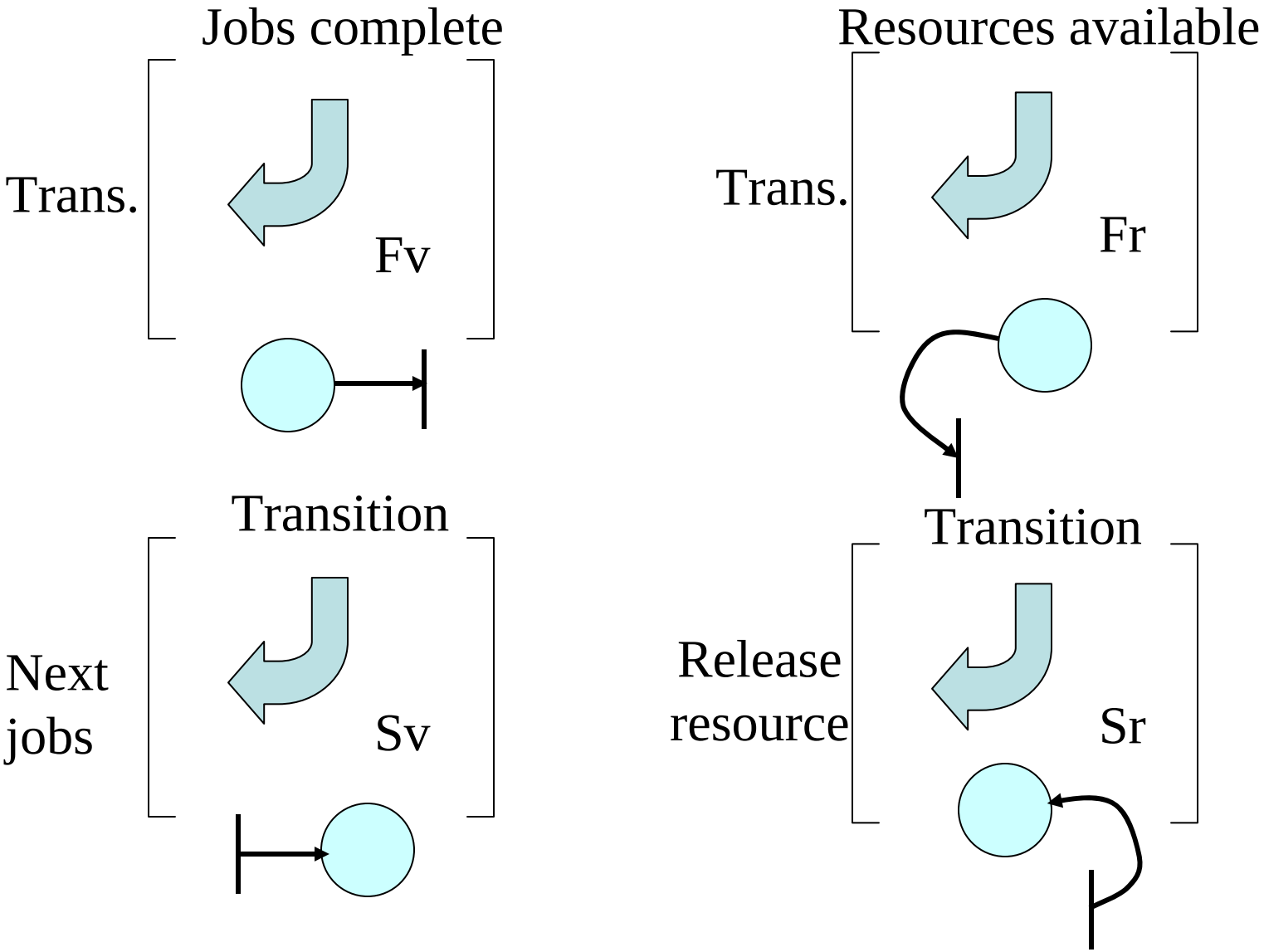



Dispatching rules
u_c
Rule-Based Real-Time Controller
Controller state monitoring logic
$\bar{x} = F_v \otimes \bar{v} \oplus F_r \otimes \bar{r} \oplus F_u \otimes \bar{u} \oplus F_{uc} \otimes \bar{u}_C$
Job start logic
$v_S = S_V \otimes x$
Resource release logic
$r_S = S_r \otimes x$
Task complete logic
$y = S_y \otimes x$
Parts in p_{in}
Start tasks v_s
Start resource release r_s
Output y
Work Cell
Parts present u
Tasks completed v_c
Resource released r_c
Products p_{out}
Plant commands
Plant status

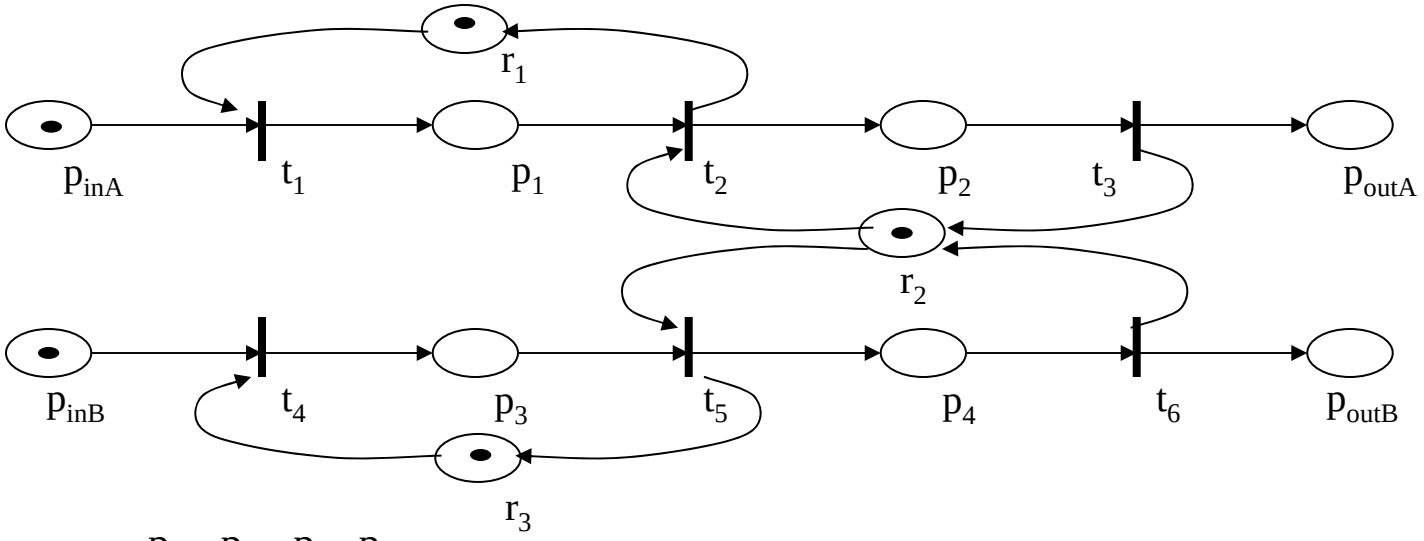
Advantages of the Matrix Formulation

- Formal rigorous framework
- Complete DE dynamical description
- Relation to known Manufacturing notions
- Formal relation to other tools- Petri Nets, MAX-Plus
- Easy to design, change, debug, and test
- Formal deadlock analysis technique
- Easy to apply any conflict resolution (dispatching) strategy
- Optimization of resources
- Easy to implement in any platform (MATLAB, LabVIEW, C, C++, visual basic, or any other)

Relation to Petri Nets



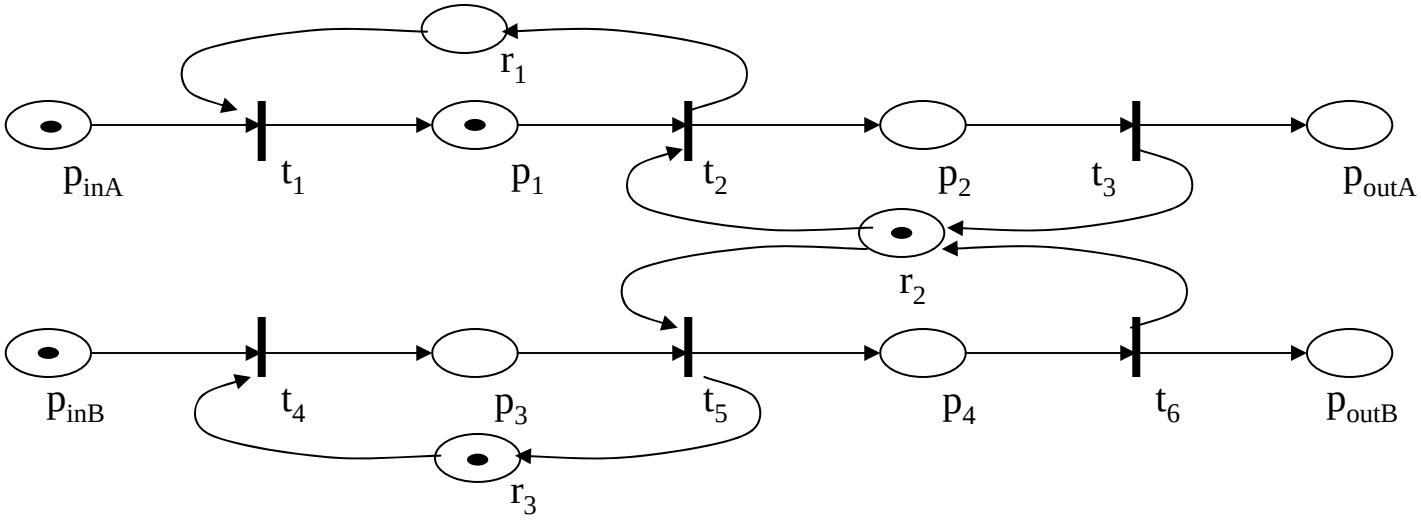
Example



$$F_v = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \qquad F_r = \begin{array}{c} \begin{array}{ccc} r_1 & r_2 & r_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \qquad F_u = \begin{array}{c} \begin{array}{cc} p_{inA} & p_{inB} \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array}$$

$$S_v^T = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad S_r^T = \begin{array}{c} \begin{array}{ccc} r_1 & r_2 & r_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \qquad S_y^T = \begin{array}{c} \begin{array}{cc} p_{outA} & p_{outB} \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

OR/AND Algebra- Locating transitions firing from current marking



$$\overline{x} = \underset{F_v}{\begin{bmatrix} 0&0&0&0\\ 1&0&0&0\\ 0&1&0&0\\ 0&0&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{bmatrix}} \otimes \underset{\overline{v}}{\begin{bmatrix} 0\\1\\1\\1 \end{bmatrix}} \oplus \underset{F_r}{\begin{bmatrix} 1&0&0\\ 0&1&0\\ 0&0&0\\ 0&0&1\\ 0&1&0\\ 0&0&0 \end{bmatrix}} \otimes \underset{\overline{r}}{\begin{bmatrix} 1\\0\\0 \end{bmatrix}} \oplus \underset{F_u}{\begin{bmatrix} 1&0\\ 0&0\\ 0&0\\ 0&1\\ 0&0\\ 0&0 \end{bmatrix}} \otimes \underset{\overline{u}}{\begin{bmatrix} 0\\0 \end{bmatrix}}$$

$$\overline{x} = \begin{bmatrix} 0\\0\\1\\0\\1\\1 \end{bmatrix} \oplus \begin{bmatrix} 1\\0\\0\\0\\0\\0 \end{bmatrix} \oplus \begin{bmatrix} 0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 1\\0\\1\\0\\1\\1 \end{bmatrix}, \quad \text{so} \quad x = \begin{bmatrix} 0\\1\\0\\1\\0\\0 \end{bmatrix}$$

i.e. fire t_2 and t_4

Complete DE Dynamic Formulation

Activity Completion Matrix F:

$$F = [F_u \quad F_v \quad F_r \quad F_y]$$

Activity Start Matrix S:

$$S = [S_u{}^T \quad S_v{}^T \quad S_r{}^T \quad S_y{}^T]$$

PN Incidence Matrix:

$$M = S^T - F = [S_u{}^T - F_u,\ S_v{}^T - F_v,\ S_r{}^T - F_r,\ S_y{}^T - F_y]$$

PN marking transition equation:

$$m(t+1) = m(t) + M^T x = m(t) + [S^T - F]x$$

Allowable marking vector:

$$x_k = \overline{F} \oplus \overline{\overline{m}_k} = \overline{[F_u \quad F_v \quad F_r \quad F_y]} \oplus \overline{\overline{[PI \quad v \quad r \quad PO]}}_k$$

Petri Net Marking Transition Equation-- need to add Job Duration Times

$$m(t) = m_a(t) + m_p(t)$$

PN Marking Vector

Split transition equation in two steps

$$m_p(t+1) = m_p(t) + S^T\, x(t)$$

Add tokens

$$m_a(t+1) = m_a(t) - F\, x(t)$$

Subtract tokens when job complete

$$T = [O,\ vtimes^T, rtimes^T, O]^T$$

Add Time Duration Vector

$$T_{pend}(t+1) = diag\{m_p(t)\}[T_{pend}(t) - t_{sample}] + diag\{S^T x(t)\}T$$

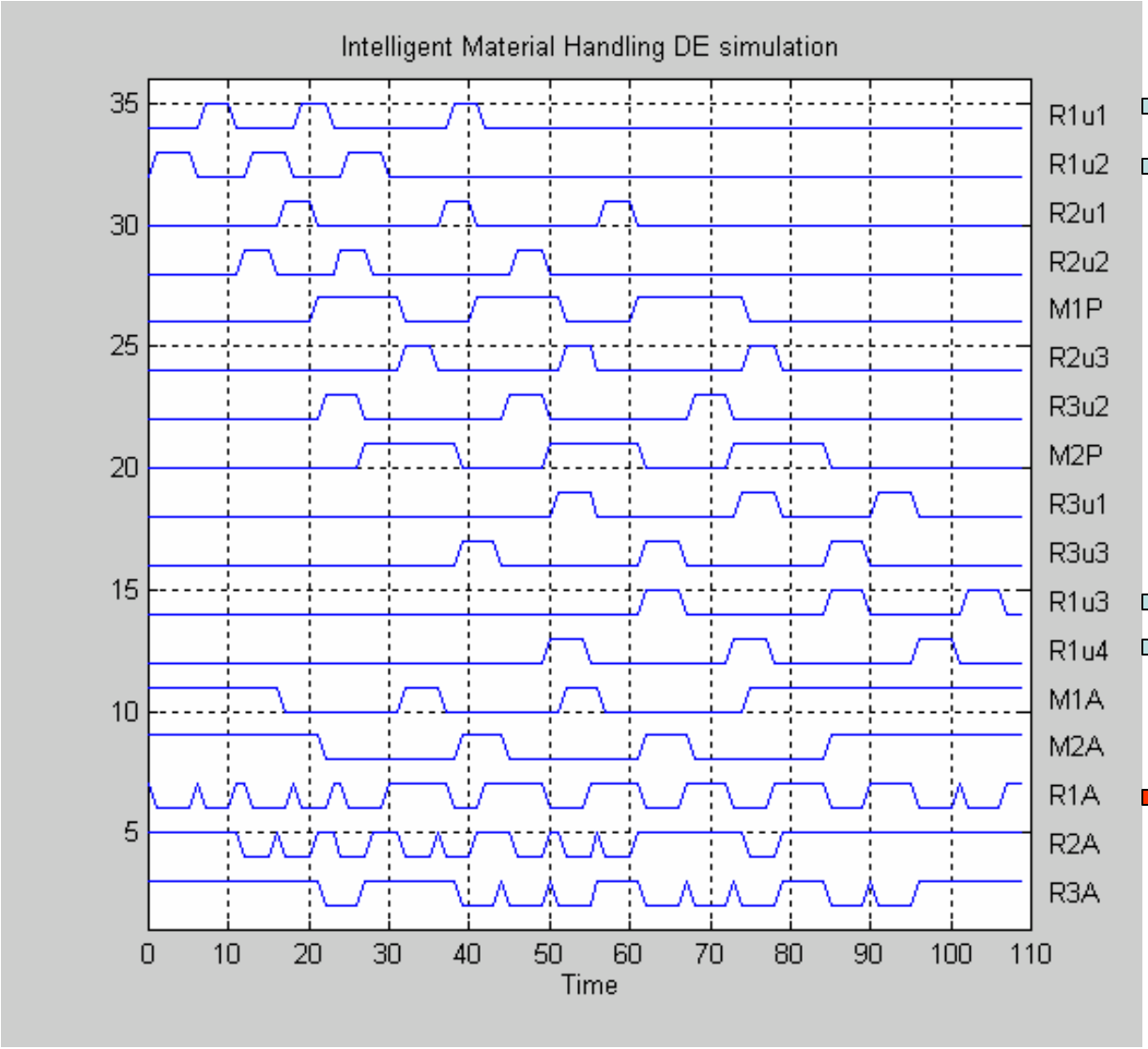
$$m_p(t) = m_p(t) - m_{finish}(t)$$

$$m_a(t) = m_a(t) + m_{finish}(t)$$

Corresponds to Timed Places

Allows Direct Simulations- e.g. MATLAB

c.f. DE version of ODE23



Jobs completed by Robot 1

Robot 1 busy or idle

Relation to Max-Plus Algebra

$$\bar{x} = F_v \bar{v}_c + F_r \bar{r}_c + F_u \bar{u} + F_D \bar{u}_D$$

State equation

$$V_s = S_v x$$
$$r_s = S_r x$$
$$y = S_y x$$

Output equations

OPERATIONS IN OR-AND ALGEBRA

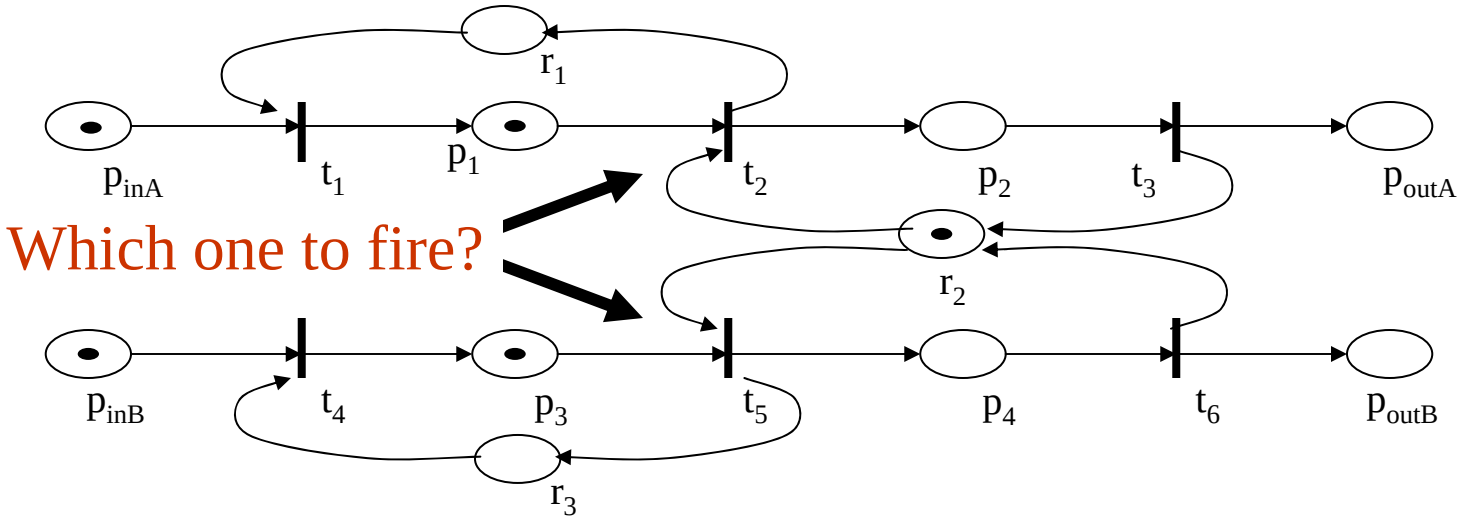
Define timing matrices. Then max plus is

$$x' = S_v T_v F_v x + S_r T_r F_r r$$

OPS. IN MAX-PLUS ALGEBRA

Can also include nonlinear terms- correspond to decisions

Conflict Resolution for Shared Resources



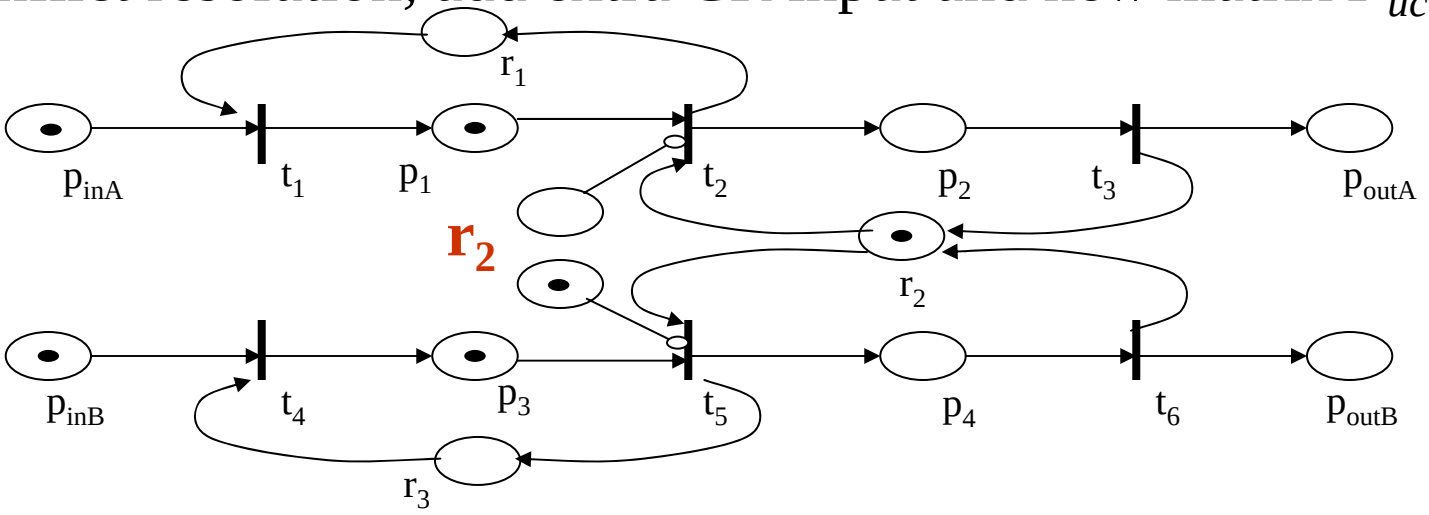
$$\overline{x} = \underset{F_v}{\begin{bmatrix} 0&0&0&0\\ 1&0&0&0\\ 0&1&0&0\\ 0&0&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{bmatrix}} \otimes \underset{\overline{v}}{\begin{bmatrix} 0\\1\\0\\1 \end{bmatrix}} \oplus \underset{F_r}{\begin{bmatrix} 1&0&0\\ 0&1&0\\ 0&0&0\\ 0&0&1\\ 0&1&0\\ 0&0&0 \end{bmatrix}} \otimes \underset{\overline{r}}{\begin{bmatrix} 1\\0\\1 \end{bmatrix}} \oplus \underset{F_u}{\begin{bmatrix} 1&0\\ 0&0\\ 0&0\\ 0&1\\ 0&0\\ 0&0 \end{bmatrix}} \otimes \underset{\overline{u}}{\begin{bmatrix} 0\\0 \end{bmatrix}}$$

Shared Resource- Two entries in same column

$$\begin{bmatrix} 0\\0\\1\\0\\0\\1 \end{bmatrix} \oplus \begin{bmatrix} 1\\0\\0\\1\\0\\0 \end{bmatrix} \oplus \begin{bmatrix} 0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 1\\0\\1\\1\\0\\1 \end{bmatrix}, \text{ so } x = \begin{bmatrix} 0\\1\\0\\0\\1\\0 \end{bmatrix}$$

But gives negative marking!
Cannot fire both.

Conflict resolution, add extra CR input and new matrix F_{uc}:

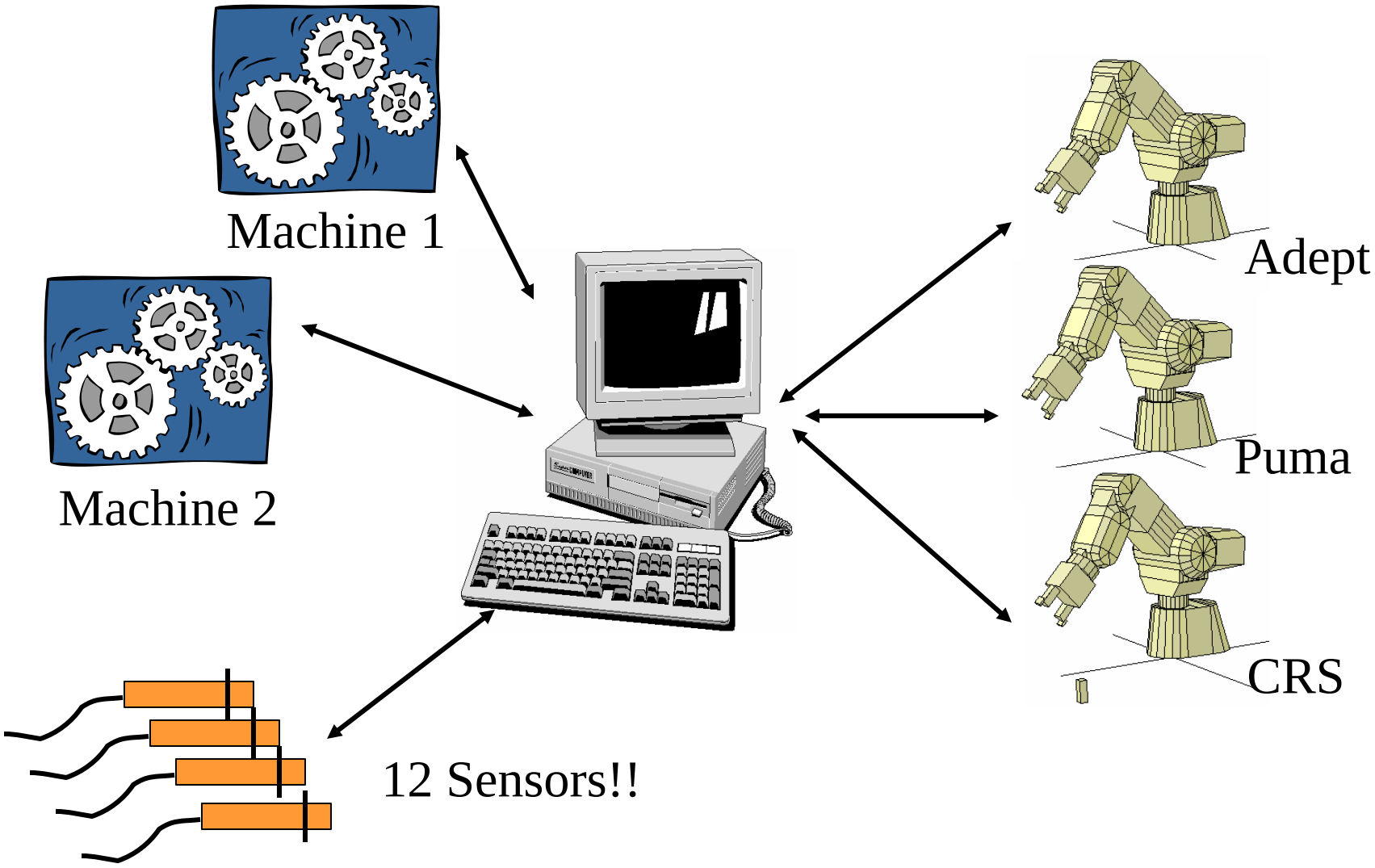


$$\overline{x} = \begin{bmatrix} 0&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&1 \end{bmatrix}_{F_v} \otimes \begin{bmatrix} 0\\1\\0\\1 \end{bmatrix}_{\overline{v}} \oplus \begin{bmatrix} 1&0&0\\0&1&0\\0&0&0\\0&0&1\\0&1&0\\0&0&0 \end{bmatrix}_{F_r} \otimes \begin{bmatrix} 1\\0\\1 \end{bmatrix}_{\overline{r}} \oplus \begin{bmatrix} 1&0\\0&0\\0&0\\0&1\\0&0\\0&0 \end{bmatrix}_{F_u} \otimes \begin{bmatrix} 0\\0 \end{bmatrix}_{\overline{u}} \oplus \begin{bmatrix} 0&0\\1&0\\0&0\\0&0\\0&1\\0&0 \end{bmatrix}_{F_{uc}} \otimes \begin{bmatrix} 1\\0 \end{bmatrix}_{\overline{r_2}}$$

$$\begin{bmatrix} 0\\0\\1\\0\\0\\1 \end{bmatrix} \oplus \begin{bmatrix} 1\\0\\0\\1\\0\\0 \end{bmatrix} \oplus \begin{bmatrix} 0\\0\\0\\0\\0\\0 \end{bmatrix} \oplus \begin{bmatrix} 0\\1\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 1\\1\\1\\1\\0\\1 \end{bmatrix}, \quad \text{so } x = \begin{bmatrix} 0\\0\\0\\0\\1\\0 \end{bmatrix}$$

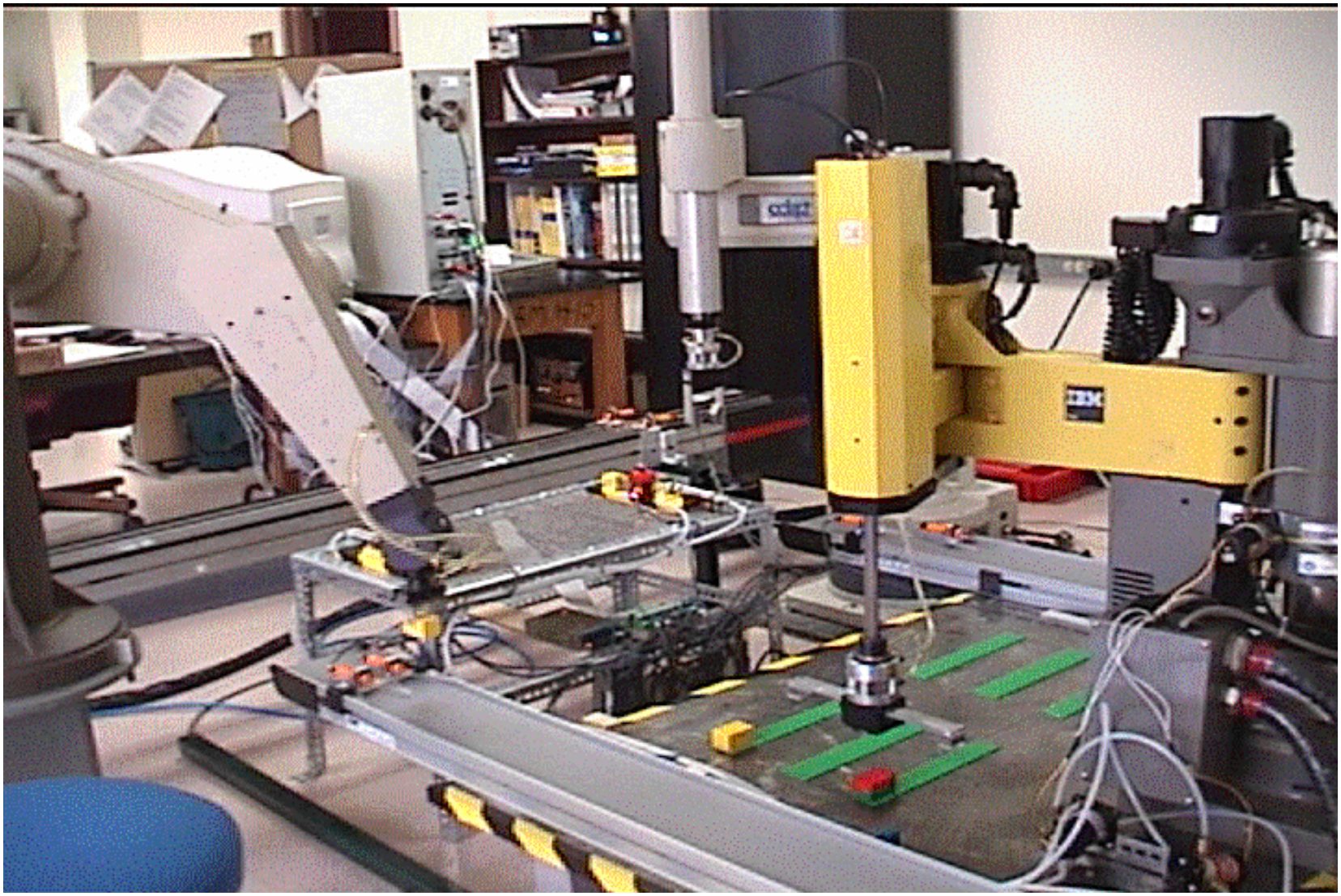
Now only t_5 fires

Application- Intelligent Material Handling

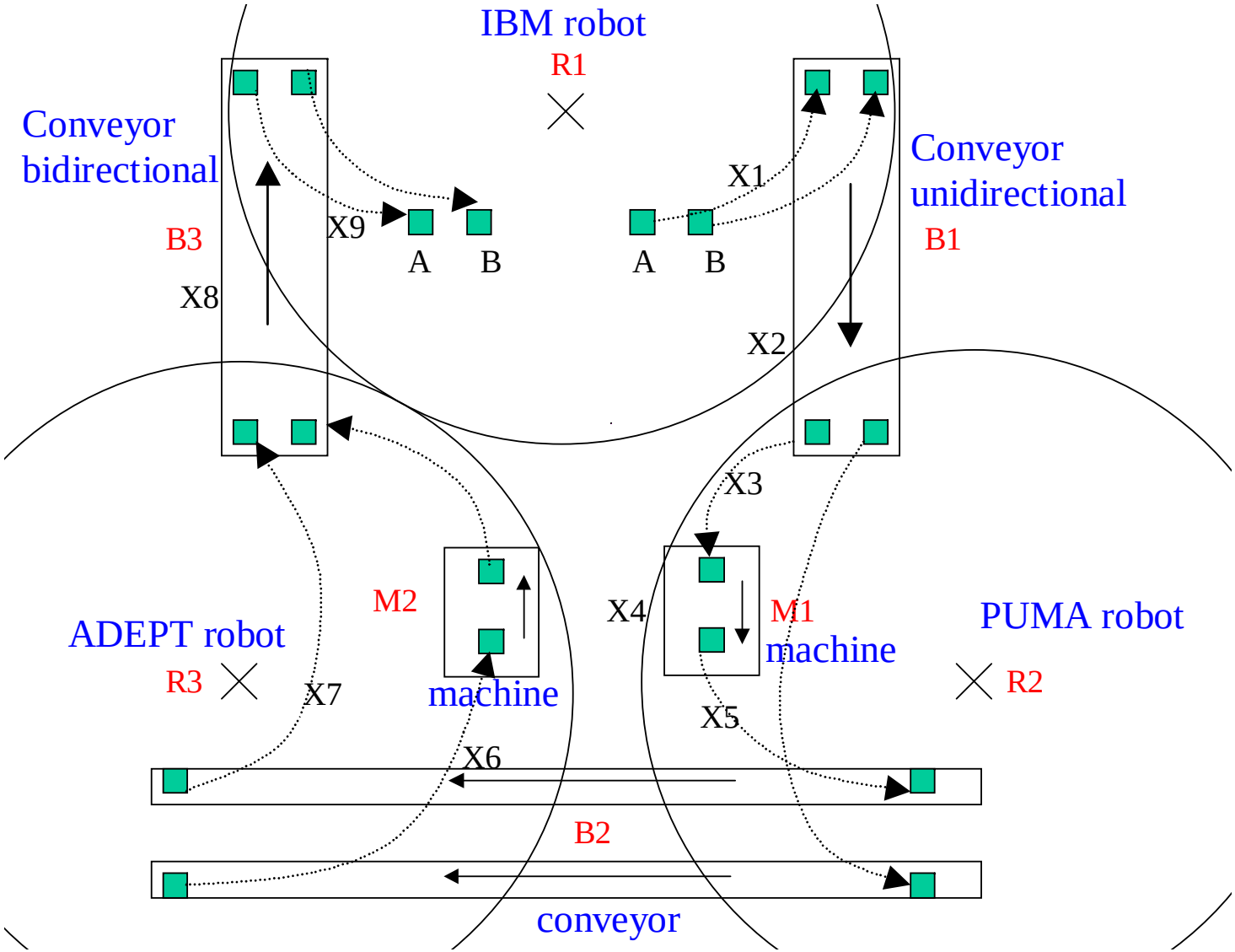


ARRI Intelligent Material Handling (IMH) Cell

3 robots, 3 conveyors, two part paths

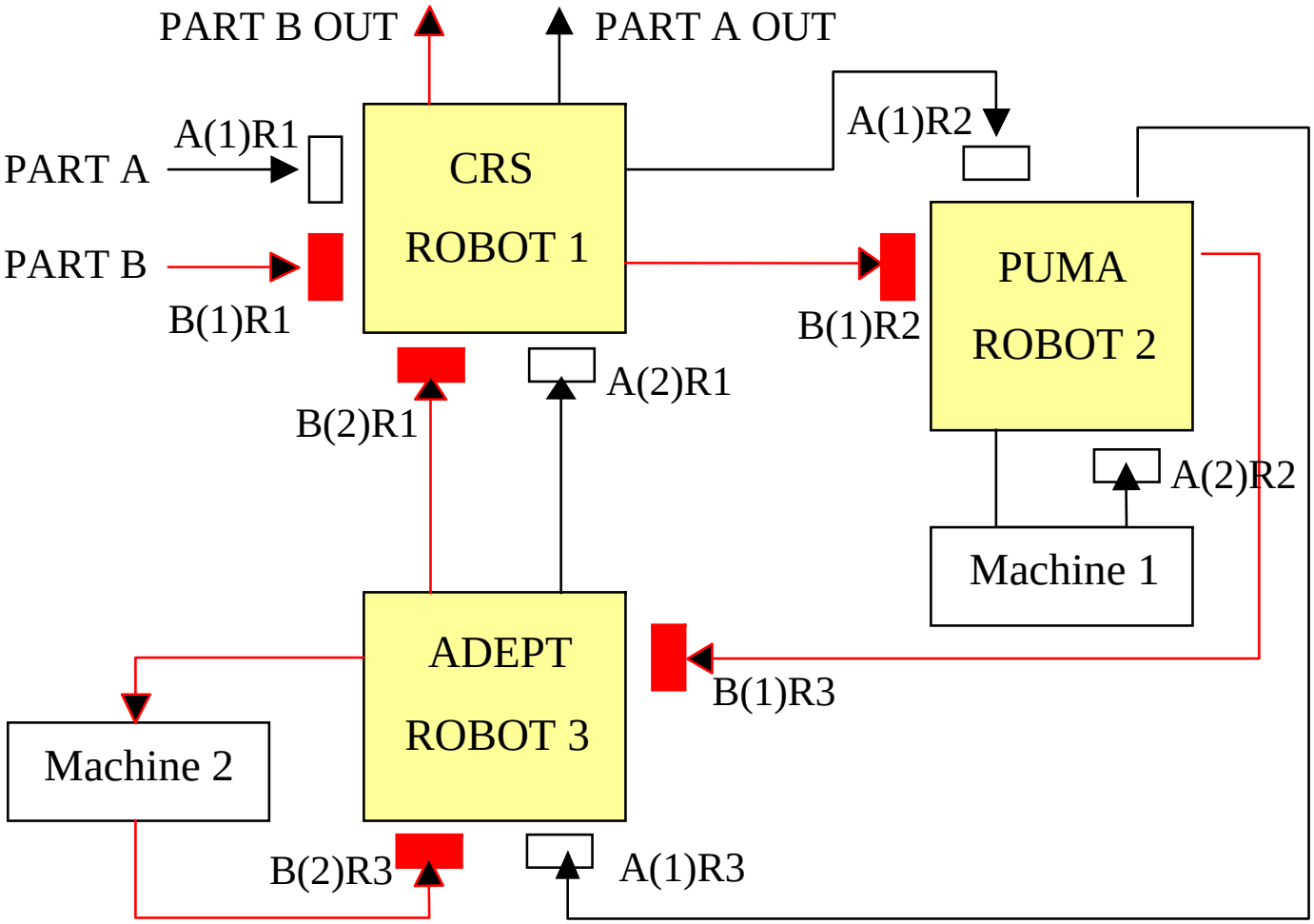


Layout of the IMH Cell

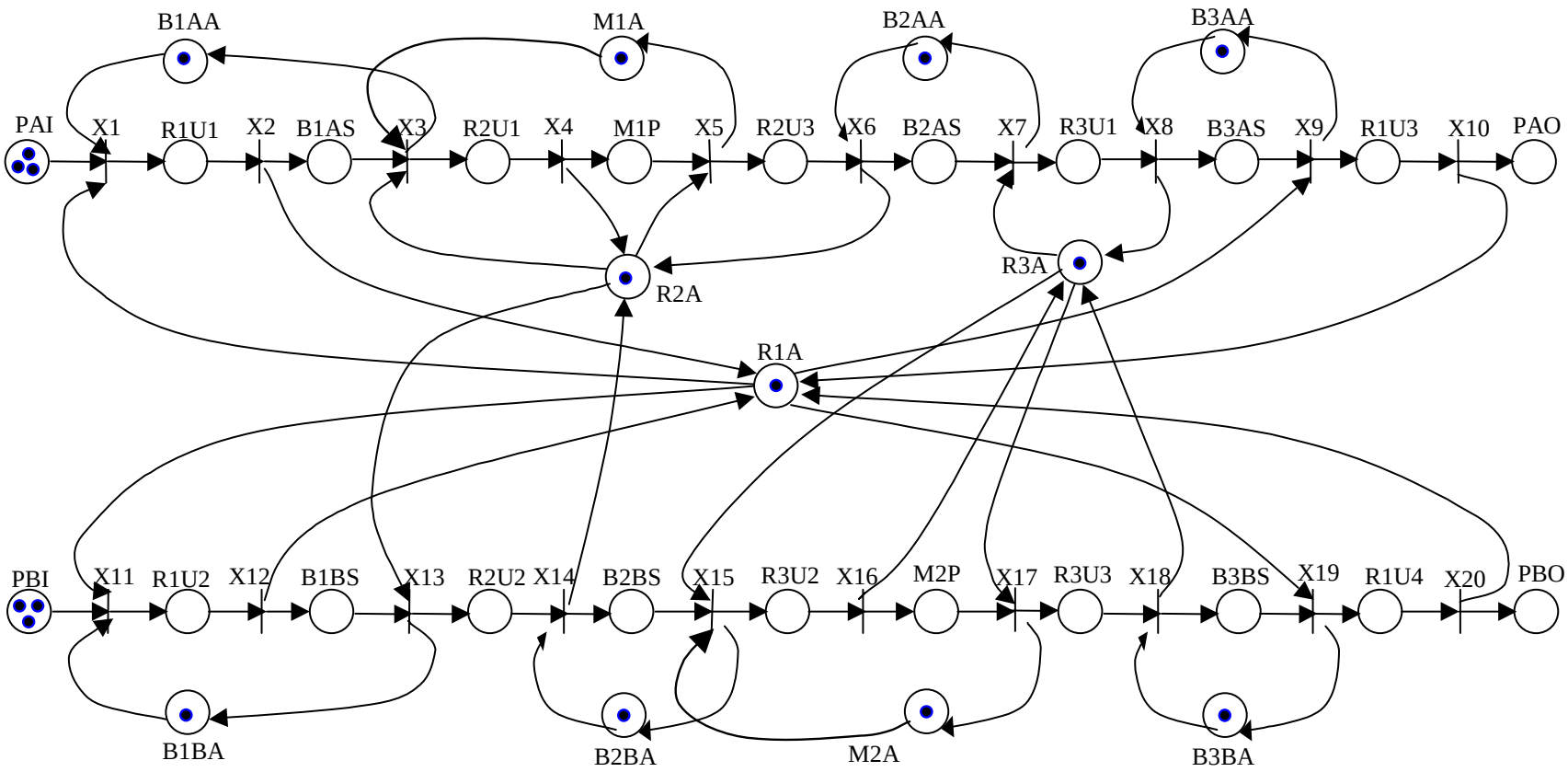


Multipart Reentrant Flow Line

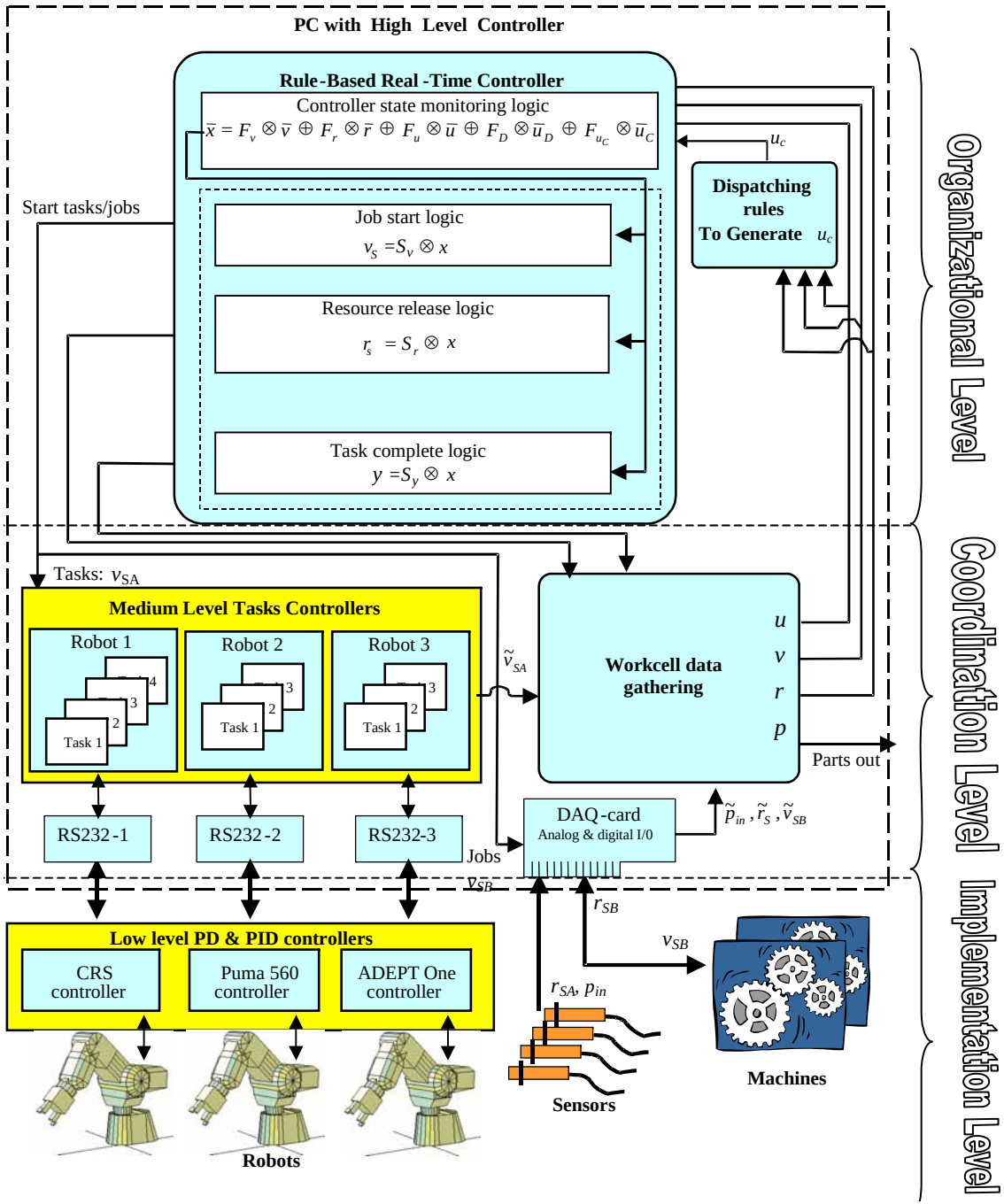
c.f. Kumar



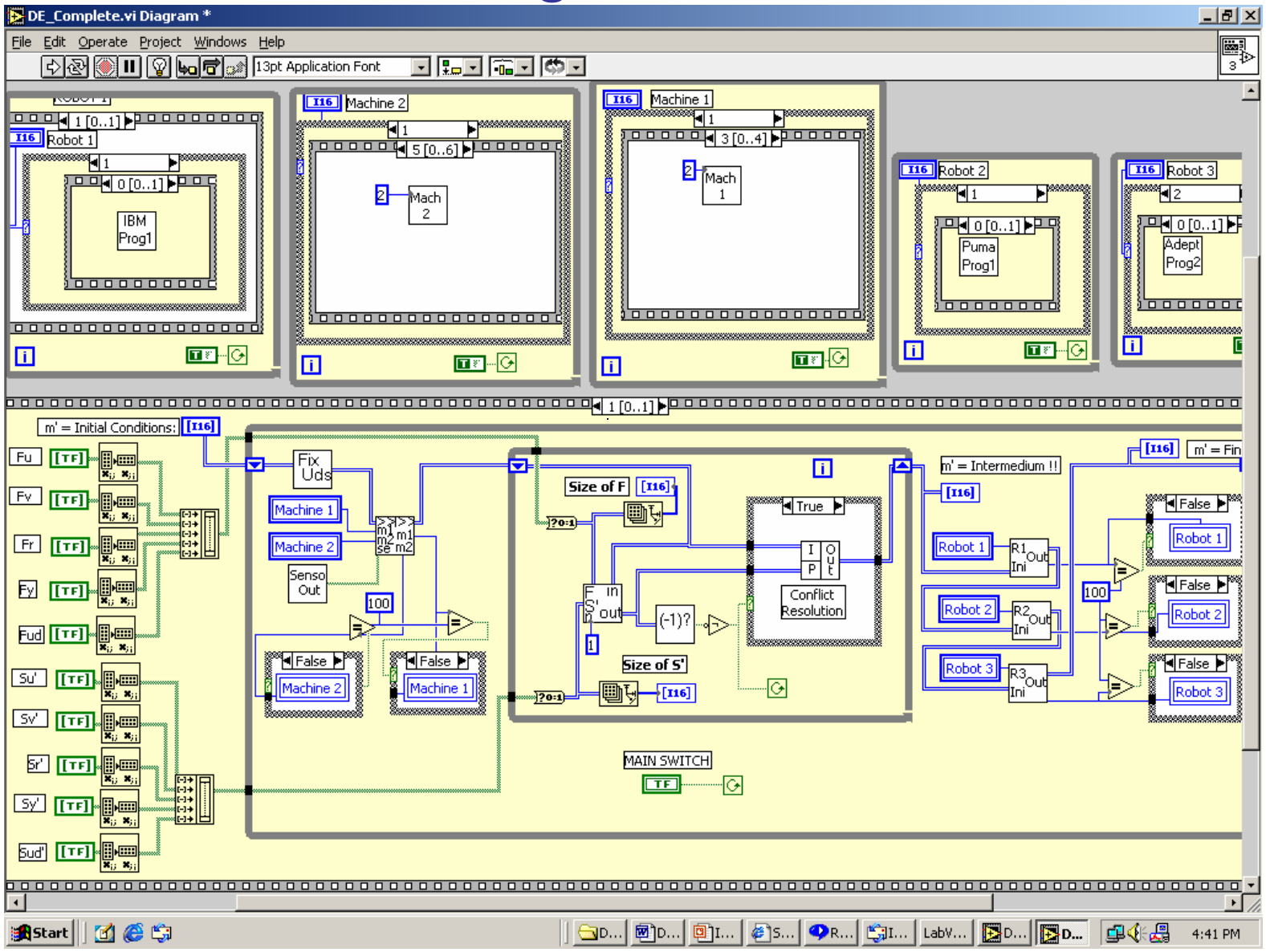
Petri Net flow chart



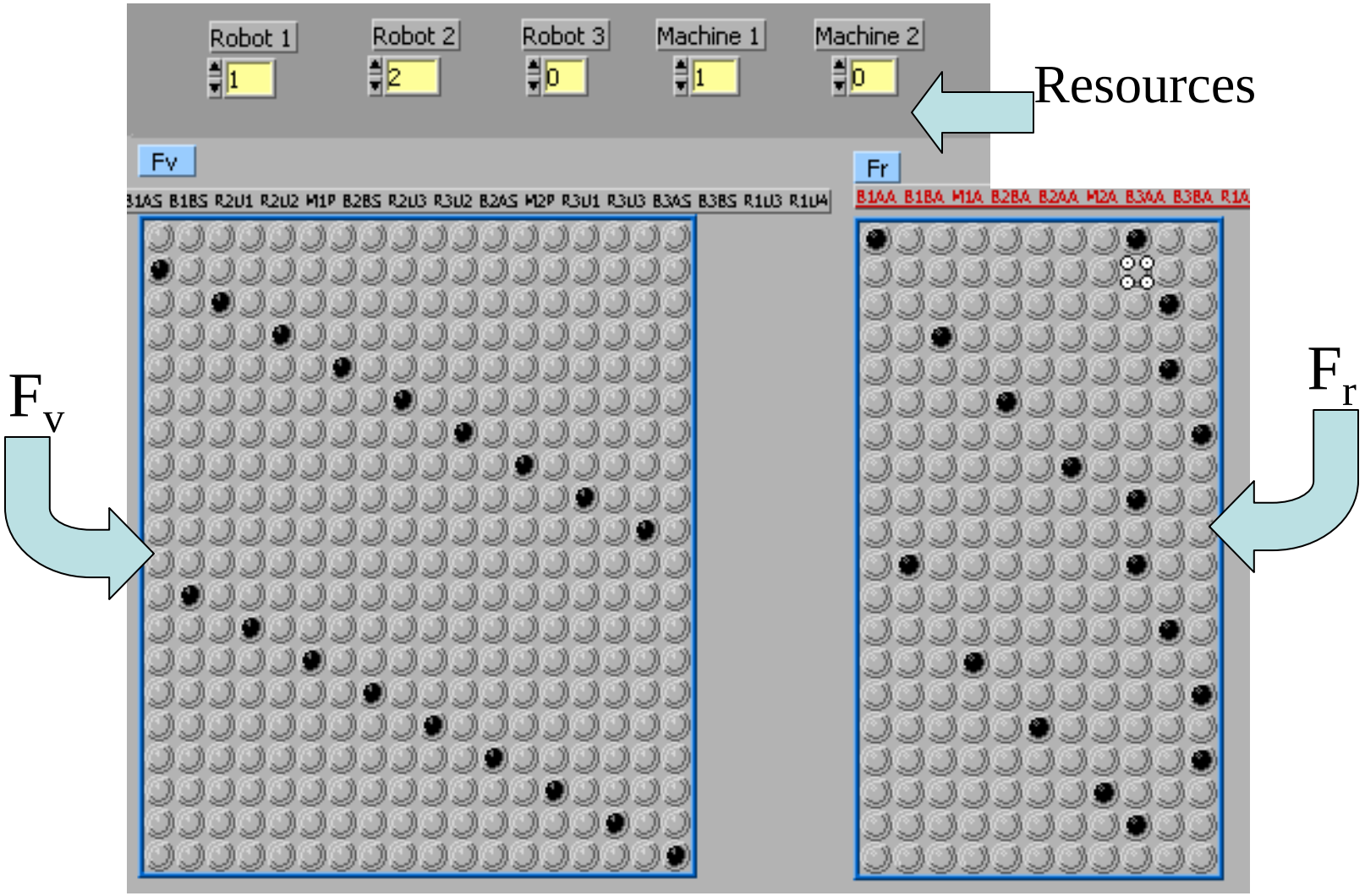
c.f. Saridis
Jim Albus



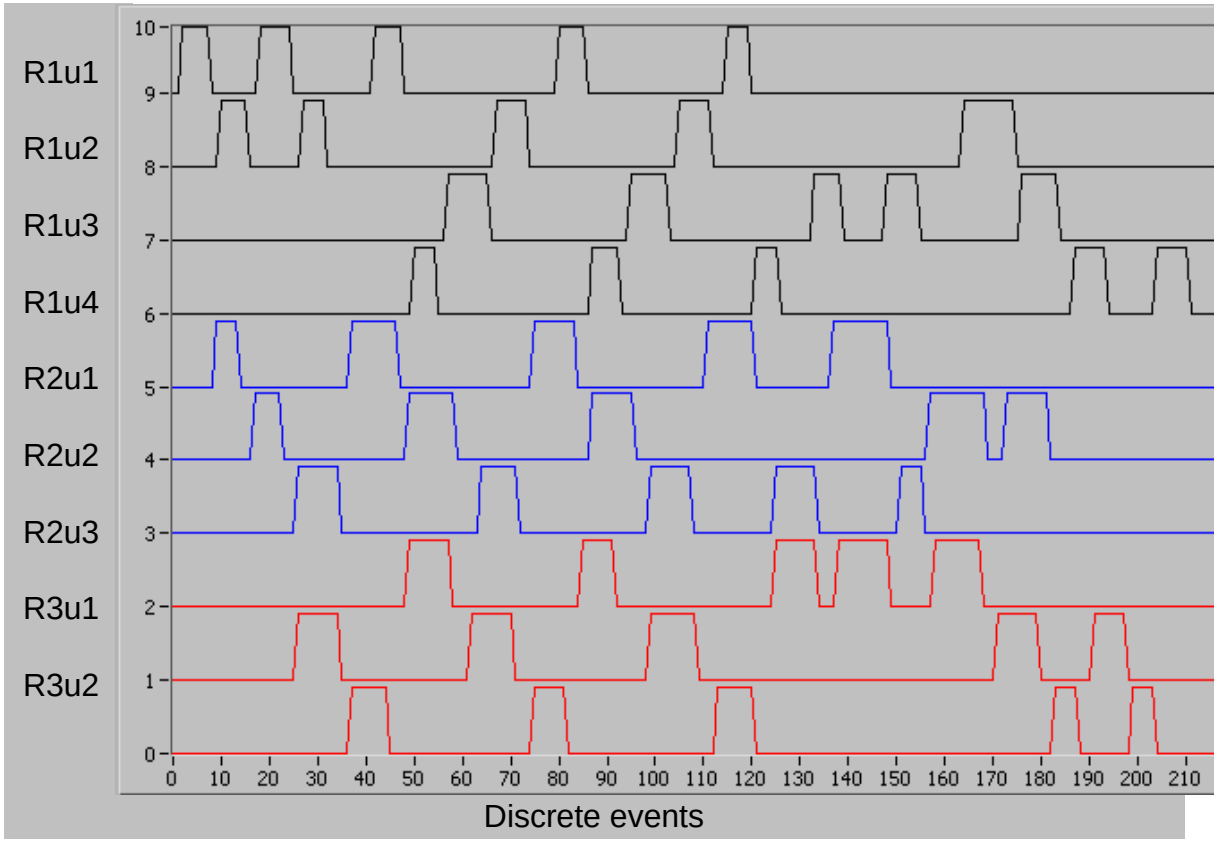
LabVIEW diagram of Controller



LabVIEW Controller's interface:



Results of LabVIEW Implementation on Actual Workcell

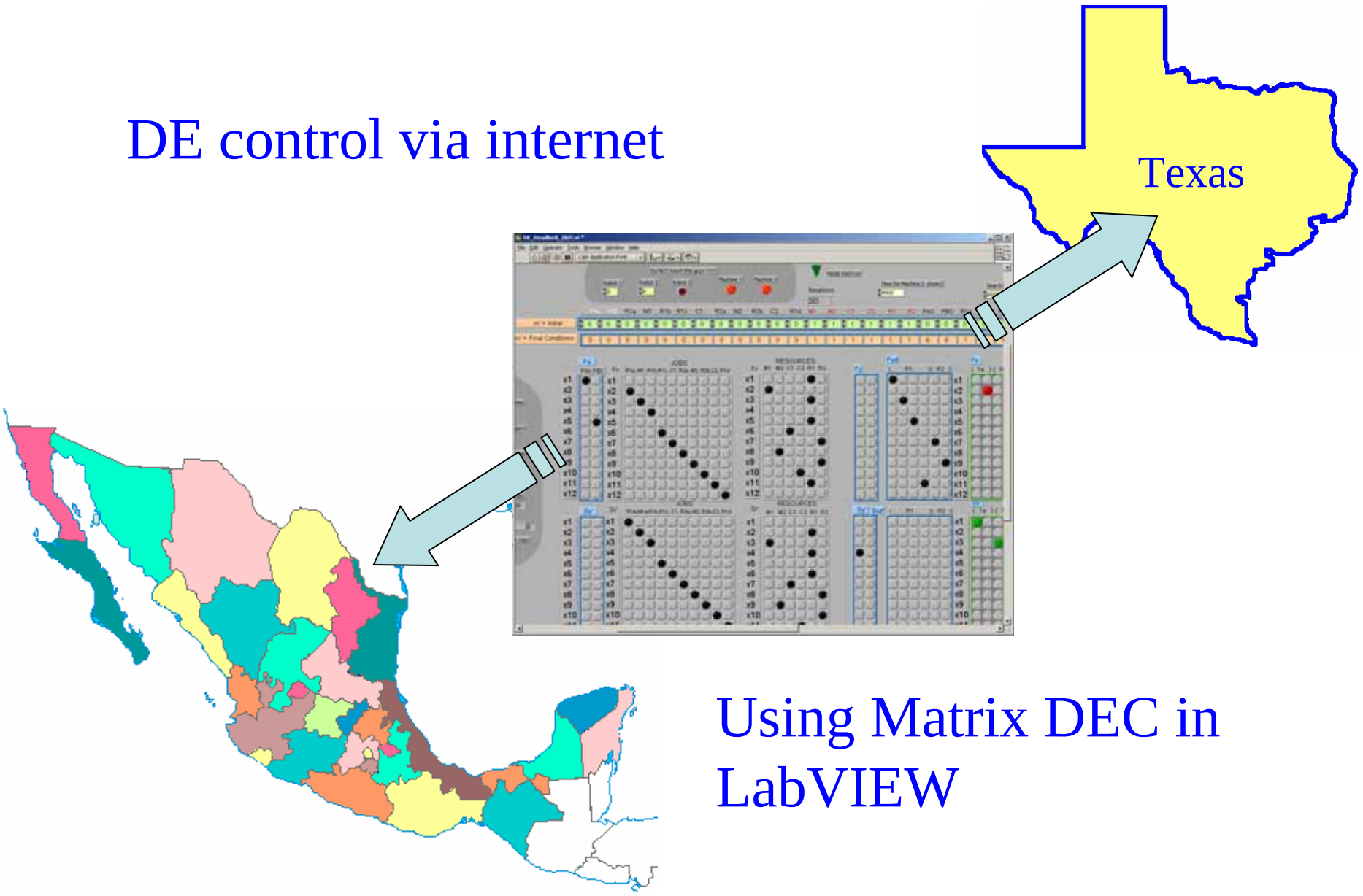


Compare with MATLAB simulation!

We can now simulate a DE controller and then implement it,
Exactly as for continuous state controllers!!

U.S.-Mexico shared research

DE control via internet



Using Matrix DEC in LabVIEW