

Redes de Petri com Internet das coisas e outras tecnologias

Petri Nets with internet of things other technologies

RESUMO

Waldyr Turqueti Gonçalves
waldyr.1999@alunos.utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Ponta Grossa, Paraná,
Brasil

Rui Tadashi Yoshino
ruiyoshino@utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Ponta Grossa, Paraná,
Brasil

O objetivo desse trabalho é verificar a utilização o conceito de redes de Petri junto com outras tecnologias, como internet das coisas e RFID's para a indústria 4.0. As pesquisas foram feitas utilizando a metodologia PRISMA e com as plantas Festo MPS. Por questão de praticidade é usado redes de alto níveis e a modelagem de tal para os processos industriais com a integração com algoritmos de otimização como o algoritmo ACO e com linguagens reconhecidas por máquinas para realizar o processo industrial. Somando esses conhecimentos é obtidos resultados eficientes e otimizados que atingem os resultados que a indústria 4.0 procura.

PALAVRAS-CHAVE: Internet das Coisas. Redes de Petri. Radiofrequência-Identificação.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

The focus of this work is how the concept of Petri nets can be used together with other technologies, such as the internet of things and RFID for industry 4.0. The researches were made using the PRISMA methodology and with the Festo MPS plants. For the sake of practicality, high-level networks and the modeling of such are used for industrial processes with the integration with optimization algorithms such as the algorithm and with languages recognized by machines to carry out the industrial process. Adding this knowledge, efficient and optimized results are obtained that achieve the results that Industry 4.0 seeks.

KEYWORDS: Internet of things (IoT). Petri nets. RFID.



INTRODUÇÃO

A indústria 4.0 busca a inovação da tecnologia para aumentar a eficiência, qualidade e tempo em sua produção. Para isso são aplicados diversos conhecimentos para performance da automatização, citadas por Wu e Zhang (2019), como *IoT (Internet of things)* e algoritmos como ACO aplicadas nas redes de Petri. A linguagem de programação SCL (*Structured Control Language*) que o manual *Education* (2018) afirma ser uma linguagem estruturada respeitando as normas DIN EN-61131-3 e outras tecnologias aplicadas nas Redes de Petri. A rede de Petri foi criada em 1939 por Carl Adam Petri e possui potencial de modelagem e de representação de sistemas, sendo ainda muito utilizado em diversas áreas da automatização. Sua Representação em grafos, pode ser vista de outras formas (FRANCÊS, 2003). Linguagens como a XML podem representar as redes de Petri (WU, 2019). Sua representação gráfica permite obter uma interpretação muito boa de sistemas discretos, facilitando o entendimento de processos complexos.

Somando esses conhecimentos torna-se possível a implementação desse nível nas plantas da Festo no Laboratório Indústria 4.0 da Universidade Tecnológica Federal do Paraná (UTFPR) – Campus Ponta Grossa, porém nesse trabalho foram aplicados a modelagem em redes de Petri e o código SCL apenas da estação *Distribution* para a simulação da distribuição de copos, mesmo sendo algo simples visualmente é algo complexo de implementar.

MATERIAL E MÉTODOS

A metodologia do trabalho é dívida em parte teórica e prática, sendo que a teórica foi feita através da metodologia Prisma para a obtenção de material científico e a parte prática foi realizado na Planta Festo MPS *Distribution*.

Na parte prática da pesquisa foram utilizadas 3 plantas Festo da linha MPS, sendo elas: *Distribution* (Responsável por distribuir os copinhos na esteira), *Pick and Place* (Responsável por fechar os copinhos com as tampas) e a *Sorting* (Responsável por separar os copinhos). Com essas três plantas foi possível executar uma simulação industrial de fechamento de copinhos e a separação deles por cor e/ou códigos lidos nas etiquetas presentes nos copos. Essa leitura é realizada por RFID's.

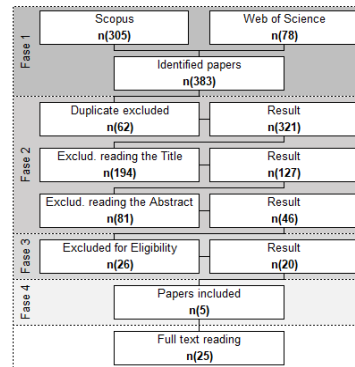
Para a revisão da literatura foi utilizada a metodologia Prisma nas bases de dados Scopus (305) e Web of Science (78). Essa metodologia consiste em um fluxograma com 4 etapas específicas, sendo elas: 1°- Busca com as palavras-chaves nas bases de dados e a exclusão por duplicatas; 2°- Exclusão por títulos e resumos; 3°- Exclusão por leitura dos arquivos; 4°- Inclusão de artigos que foram indicados pelo orientador e outros professores. Observado na Tabela 1 e na Figura 1.

Tabela 1 – Palavras chaves utilizadas nas bases de dados

Keywords	Filters	DataBases		
("Petri Nets" AND ("IoT" OR "Artificial Intelligence" OR "Internet of Things" OR "AI"))	Article	WoS	78	20,4 %
	Conference or Review	Scopus	305	79,6 %
	2015 →2020	Total	383	

Fonte: Autoria Própria (2020).

Figura 1 – Fluxograma Prisma



Fonte: Adaptado (MOHER et al., 2009).

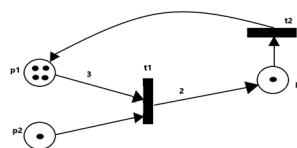
RESULTADOS E DISCUSSÕES

As redes de Petri, com seu poder de modelagem possibilitam serem representadas por grafos dirigidos, facilitando a rede ser traduzida por qualquer sistema autômato através de vários tipos de linguagens de programação. Na pesquisa em questão foram utilizadas as redes HLPN (*High Level Petri Net*) para representar o processo das plantas Festo MPS e a linguagem SCL (*Structured Control Language*) para representar as redes, o código gerado é compilado nos CLP's (Controladores Lógicos Programáveis) das plantas.

REDES DE PETRI

A ilustração de um grafo simples de uma rede de Petri com suas características pode ser vista na Figura 2. A Explicação aprofundada das redes de Petri e seus grafo são descritas detalhadamente em (FRANCÊS, 2003) e (CARDOSO; VALETTE, 1997).

Figura 2 – Exemplo Gráfico de uma rede de Petri



Fonte: (FRANCÊS, 2003).

A definição de uma rede de Petri em grafo pode ser representada pela n-upla N ilustrado na Eq. (1).

$$N = (S; T; F); \text{ tal que:} \quad (1)$$

$$i) S \cap T = \emptyset \text{ e } S \cup T \neq \emptyset;$$

$$ii) F \subseteq (S \times T) \cup (T \times S);$$

$$iii) \text{ dom}(F) \cup \text{ ran}(F) = S \cup T, \text{ onde}$$

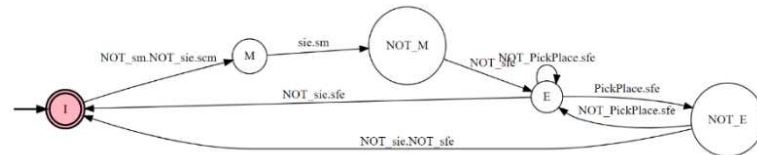
$$\text{ dom}(F) = \{x \in (S \cup T) \mid \exists y \in (S \cup T). (x, y) \in F\},$$

$$\text{ ran}(F) = \{y \in (S \cup T) \mid \exists x \in (S \cup T). (x, y) \in F\}.$$

REDES DE PETRI COM AS PLANTAS FESTO MPS E A LINGUAGEM SCL

Aplicando as redes de Petri na planta *Distribution* obtemos a rede vista na Figura 3. O Funcionamento de forma descritiva e visual de todas as 3 plantas é apresentado em (DIDATIC, 2013). Nesse trabalho é apresentado apenas a rede de Petri (Gerado usando a ferramenta criado por Zuzak (2017)) e o código SCL da *Distribution*.

Figura 3 – Rede de Petri da *Distribution*



Fonte: Autoria Própria (2020).

Onde os nomes dos estados e o que fazem são: I- Início; M- Ativação do Magazine; NOT_M- A não ativação do Magazine, ou seja, ele retorna; E-Ativação da esteira; NOT_E- A esteira não se move. Já as transições (condições de um estado pra outro) são: scm- Copinho presente no magazine; sm- Magazine avançado, sie- Sensor do inicio da esteira está identificando um copinho; PickPlace- A estação *Pick and Place* está ocupada; sfe- Sensor no final da esteira está identificando um copinho. E todos os com NOT_ na frente significam a negação desse acontecimento.

Já o código SCL é mostrado na Figura 4. Com a definição do que são cada variáveis e a sequências dos passos.

Figura 4 – Código SCL da Planta *Distribution*

```
//SEQUENCIA DE OPERAÇÃO DA MAQUINA MPS DISTRIBUTING
CASE "PASSO_D" OF
0:
  "IEC_Timer_0_DB_1".TON(IN := FALSE,
    PT := T#2s);
  "IEC_Timer_0_DB_2".TON(IN := FALSE,
    PT := T#1s);

  "G1MA1_D_F" := FALSE;
  "G1MA1_D_T" := FALSE;
  "G1MB1_D" := FALSE;
  "C1MB1" := FALSE;
  //RFID
  "COMMAND_CODE" := 16#0000; //comando de leitura
  "START_ADDR" := 16#0000; // W/R a partir do byte 0
  "LENGHT_ADDR" := 16#0001; //vai usar apenas 1 byte

  IF "C2BG1" THEN
    "PASSO_D" := 10;
  END_IF;

10:
  IF NOT "C2BG3" AND "M_START" THEN
    "PASSO_D" := 15;
  END_IF;
  IF "M_STOP" THEN
    "PASSO_D" := 200;
  END_IF;

15:
  "IEC_Timer_0_DB_2".TON(IN := TRUE,
    PT := T#1s);
  IF "IEC_Timer_0_DB_2".Q THEN
    "C1MB1" := TRUE;
    "PASSO_D" := 18;
  END_IF;
  IF "M_STOP" THEN
    "PASSO_D" := 200;
  END_IF;

18:
  IF "CH_AUTO_MAN" THEN
    "RFID_WRITE_BO" := 16#0002;
    "PASSO_D" := 20;
  ELSE
    "RFID_WRITE_BO" := 16#0001;
    "PASSO_D" := 20;
  END_IF;
  IF "M_STOP" THEN
    "PASSO_D" := 200;
  END_IF;

20:
  "IEC_Timer_0_DB_2".TON(IN := FALSE,
    PT := T#1s);
  IF "C2BG2" THEN
    "COMMAND_CODE" := 16#0001; //comando de escrita
    "G1MA1_D_F" := TRUE; // ESTEIRA
    "C1MB1" := FALSE; //MAGAZINE
    "PASSO_D" := 30;
  END_IF;
  IF "M_STOP" THEN
    "PASSO_D" := 200;
  END_IF;

30:
  IF NOT "G1GB3_D" THEN
    "PASSO_D" := 35;
  END_IF;
  IF "M_STOP" THEN
    "PASSO_D" := 200;
  END_IF;

35:
  IF NOT "PICK&PLACE_OCUPADA" THEN
    "G1MA1_D_F" := TRUE;
    "PASSO_D" := 40;
  END_IF;
  IF "PICK&PLACE_OCUPADA" THEN
    "G1MA1_D_F" := FALSE;
    "COMMAND_CODE" := 16#0000; //comando de leitura
  END_IF;

40:
  "IEC_Timer_0_DB_1".TON(IN := TRUE,
    PT := T#2s);
  IF "IEC_Timer_0_DB_1".Q THEN
    "PASSO_D" := 0;
  END_IF;
  IF "M_STOP" THEN
    "PASSO_D" := 200;
  END_IF;

200:
  "G1MA1_D_F" := FALSE;
  "G1MA1_D_T" := FALSE;

  IF "M_RESET" THEN
    "PASSO_D" := 0;
  END_IF;
END_CASE;
Onde:
"PASSO_D" = Valor do passo;
"G1MA1_D_F" = Esteira movendo para a direita;
"G1MA1_D_T" = Esteira movendo para a esquerda;
"G1MB1_D" = Ativa o Bloqueador;
"C1MB1" = Avança o Magazine;
"IEC_Timer_0_DB_1".TON(IN := FALSE, PT := T#2s) = Desativa o temporizador de 2 seg;
"IEC_Timer_0_DB_1".Q = Ativa o temporizador de 2 seg;
"COMMAND_CODE" := 16#0000 = Da o comando ao RFID de leitura;
"COMMAND_CODE" := 16#0001 = Da o comando ao RFID de escrita;
"START_ADDR" := 16#0000 = Informa ao RFID onde vai começar a ler/escrever;
"LENGHT_ADDR" := 16#0001 = Informa ao RFID o tamanho do que ser ler/escrever;
"RFID_WRITE_BO" := 16#0002 = Faz com que o RFID grave o valor 2 no copinho;
"C2BG1" = Magazine Recuado;
"C2BG2" = Magazine Avançado;
"C2BG3" = Magazine sem copinho;
"M_START" = Memória START;
"M_STOP" = Memória STOP;
"M_RESET" = Memória RESET;
"CH_AUTO_MAN" = Chave virada;
"PICK&PLACE_OCUPADA" = Sinal da Pick, informando que está ocupada;
"G1GB3_D" = Peça identificada no final da esteira.
```

Fonte: Autoria Própria (2019).

REDES COM IOT E RFID

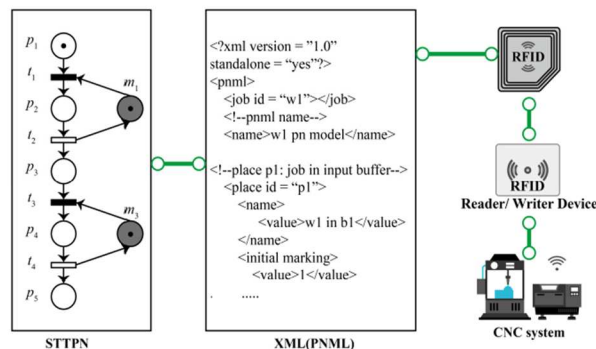
Para que a programação do processo industrial seja eficiente é necessária que ela seja dinâmica, pois há chances de imprevistos e acontecimentos externos e internos no chão de fábrica. Sendo tal planejamento complexo, fazendo-se necessária a utilização de *IoT* específicas para processos industriais, as *IoMT* (*Internet of Manufacturing Things*), onde são divididas em três camadas básicas: Camada de aplicativo, camada rede de transporte e camada perceptiva e de execução (WU; TIAN; ZHANG, 2019).

Pelo fato do ambiente fabril ser dinâmico e flexível, deve ser dada atenção ao planejamento, segundo (WU; TIAN; ZHANG, 2019) a abordagem preditiva-reativa é a mais comum usada em escalonamentos dinâmicos por ser orientado a recuperação do sistema de interrupções. De acordo com esta abordagem os cronogramas iniciais são revisados e alterados para se adaptar ao novo ambiente dinâmico e flexível da fábrica.

Para obtenção dos resultados necessários, como integração com as redes de Petri com a *IoMT*, é necessário a modificação de redes de Petri para eventos dinâmicos que possam ocorrer no chão de fábrica. Para essas modificações foram usados como base algoritmos ACO (*Ant Colony Optimization*), ou mais conhecido como algoritmo de Otimização por Colônia de Formiga. Utilizando esse algoritmo para as redes de Petri torna-se PN-ACO (Para um aprofundamento mais profundo do algoritmo ler o (WU; TIAN; ZHANG, 2019).

Tais eventos dinâmicos, que ocorrem no meio do processo, por vezes só são identificados graças a periféricos como RFID, que podem escrever/ler de forma dinâmica no meio do processo através de etiquetas que ficam no produto criado no processo industrial. Através dessa etiqueta é possível gravar a rede de Petri como linguagem XML (linguagem de máquina) permitindo que a máquina faça o que a rede foi modelada a fazer, sendo o processo, valores de verificação, valores de comparação, etc. Como exemplificado na Figura 5.

Figura 5 – Exemplo de aplicação de redes de Petri com XML



Fonte: (WU; TIAN; ZHANG, 2019).

CONCLUSÃO

Nesse trabalho, foi possível verificar de forma prática a utilização do conceito das redes de Petri em conjunto com outras tecnologias. Em particular, foi possível aplicar a redes de Petri com a linguagem SCL e RFID's na planta *Distribution*.

Através do uso das redes de Petri no contexto de *IoT*, RFID's, ACO e XML aplicados em um cenário de indústria 4.0, foram observados fatores como a automatização e eficiência do processo industrial através da abordagem preditiva-reativa.

Com base nos testes feitos na planta MPS, utilizando-se da aplicação da rede de Petri e a implementação dessa modelagem com a linguagem SCL, se obteve um aumento na facilidade e execução de sistemas complexos.

Fundindo todos esses conceitos e tecnologias é completamente possível satisfazer as necessidades da indústria 4.0. Com aplicações que necessitam de uma resposta rápida, grande flexibilidade e garantia do produto final.

AGRADECIMENTOS

Meus sinceros agradecimentos pela participação minoritária do projeto Testbed e a ABDI e UTFPR pelos investimentos oriundos que foram feitos acreditando no potencial do projeto.

REFERÊNCIAS

CARDOSO, Janette; VALETTE, Robert. **REDES DE PETRI**. Florianópolis: Editora da UFSC, 1997. 157 p.

DIDACTIC, F. The current range of Festo Didactic products. [S.l.]: **Festo Didactic GmbH & Co. KG.**, 2013.

EDUCATION, **Siemens Automation Cooperates With**. Documentação de aprendizado/treinamento SCE | Módulo TIA Portal 051-201. São Paulo: Digital Factory, 2018. 39 p.

FRANCÊS, Carlos Renato Lisboa. **Introdução às Redes de Petri**. 2003. 19 f. Tese (Doutorado) - Curso de Computação, Laboratório de Computação Aplicada – Laca, Universidade Federal do Pará - Ufpa, Belém, 2003.

MOHER, D; LIBERATI, A; TETZLAFF, J; & ALTMAN, D. G. (2009). Preferred reporting items for systematic reviews and meta-analyses: the **PRISMA** statement. *Annals of internal medicine*, 151(4), 264-269.

WU, Xiaoqiang; TIAN, Songling; ZHANG, Lei. The Internet of Things Enabled Shop Floor Scheduling and Process Control Method Based on Petri Nets. **Ieee Access**, [S.L.], v. 7, p. 27432-27442, 2019. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/access.2019.2900117>.

ZUZAK, Ivan. **FSM simulator**. 2017. Disponível em: http://ivanzuzak.info/noam/webapps/fsm_simulator/. Acesso em: 4 ago. 2017.