Performance Models for a Hybrid Reactor System

Katinka Wolter[1], Andrea Zisowsky[2], and Günter Hommel[1]

[1] Technische Universität Berlin, Sekr. EN 10, Einsteinufer 17,
10587 Berlin, Germany. Email: {katinka,hommel}@cs.tu-berlin.de

[2] Fakultät für Mathematik und Informatik, Universität des Saarlandes, Postfach 15 11 50,
D-66115 Saarbrücken, Germany. Email: zisowsky@num.uni-sb.de

Abstract. In this paper we will present an improved numerical algorithm for the analysis of fluid stochastic Petri nets with two fluid places. This algorithm uses an alternating direction implicit (ADI) discretization, whereas the previously used algorithm was a θ-discretization scheme. Especially derived discrete reflecting boundary conditions ensure that both schemes are conservative in the probability mass. In the second part of this paper we will study a model of a hybrid reactor system. This system has been studied before with very simple models that are extended here. We will use this model not only to present performance and availability measures of the model, but also to compare the two solution algorithms using part of the full model. The new algorithm is almost equal to the θ-scheme in memory usage, but roughly twice as fast.

1 Introduction

Hybrid systems have gained much attention over the last years. Formal methods for their verification, performance and reliability analysis have been investigated intensively. For the latter a number of extensions to stochastic Petri nets have been proposed [5,4,12].

Stochastic models often suffer from not having closed-form solutions, so numerical methods or simulation techniques have to be applied [6,3,2]. We use a hybrid Petri net formalism, which we call *fluid stochastic Petri nets* (FSPNs). We solve these models with discretization methods. In this paper we will present a new numerical solution technique for solving FSPN models with two fluid places. The new algorithm uses an alternating direction implicit (ADI) discretization scheme, which will be described and compared with the formerly used θ-scheme.

In the second part of this paper we use an earlier developed tool for the analysis of a real-world hybrid application. We study the behaviour of a hybrid automated surveyance system that exists at the University of Hamburg-Harburg [7]. In this system work pieces are heated up on a heater and cooled down in a water tank. A robot is used for transportation of the metal pieces. One question that arises in designing these systems is whether there is sufficient cold water so that no dangerous heat-up of the pieces can happen because there is not sufficient cold water available. Also important is whether there are enough transportation facilities to remove and replace the hot metal sticks as quickly as necessary. Other aspects rather concern the performance of the system: is the transportation utilized as wanted or should the system be dimensioned differently.

At first the simple model from [11] will be presented, in which only one metal piece is used. The model has two continuous variables: one for the temperature of the metal piece and the other one for the temperature of the water in the tank. Two continuous variables can be conveniently handled by our solution algorithm. Increasing the number of metal pieces immediately increases the model complexity. An arbitrary number of continuous variables to represent the temperatures of an arbitrary number of work pieces can be handled by no existing software tool. In consequence the models have to be formulated differently.

We will show a more general formulation of the model, which will allow us to flexibly choose the number of work pieces used. The simple and the general model should match in the water temperature if the complex model uses only one work piece. All known formalisms for modelling hybrid systems have to go over to a discrete system or a discrete description of the hybrid system. They do so either on the level of the model, on the level of its mathematical description, or upon solving the equations describing the behaviour of the model. We decided to discretize as late as possible, that means to use the latter aproach. This has several advantages. The user of our formalism can operate in a purely hybrid world and does not see the discretization that has to take place. Using an intuitive hybrid modelling paradigm is considered much more convenient by many engineers. Furthermore the error of discretization is only due to numerical errors and is much smaller than the error of discribing a hybrid system with a discrete formalism.

The paper is organized as follows: in the next section the modelling formalism we use is defined. In section 3 the two numerical solution algorithms are described. Section 4 introduces the system we study, Section 5 presents the models we have developed and Section 6 concludes the paper.

2 The FSPN Formalism

For the definition of FSPNs the common notations [1,5] are taken over and extended where it is needed.

An FSPN is formally defined as a 8-tuple FSPN $= (\mathcal{P}, \mathcal{T}, \mathcal{A}, m_0, g, \lambda, r, w)$ where the set of places ($\mathcal{P} = \mathcal{P}_c \cup \mathcal{P}_d$) is divided into the *fluid (continuous)* and the *discrete* places. Discrete places (the elements of $\mathcal{P}_d$) are drawn as single lined circles and hold an integer number of *tokens*, whereas the continuous, or fluid, places (the elements of $\mathcal{P}_d$) are drawn as two concentric circles and they hold a real-valued amount of *fluid*. The set of transitions $\mathcal{T} = \mathcal{T}_E \cup \mathcal{T}_I$ is composed of the exponentially distributed, and the immediate transitions, respectively. Exponentially distributed transitions are drawn as empty rectangles and the immediate transitions as thin bars.

The set of arcs $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_c$ is divided into two subsets, the discrete arcs (the elements of $\mathcal{A}_d$), and the continuous, or fluid, arcs ($\mathcal{A}_c$). The discrete arcs are drawn as single-lined arrows, whereas the fluid double-lined arcs are drawn like pipes.

The *marking* $\mathbf{m} = (s, \mathbf{z}) \in \mathcal{M}$ consists of a discrete part $s = (\#p_i, i \in \mathcal{P}_d)$, where $\#p_i$ denotes the number of discrete tokens in the i-th discrete place p_i, and a continuous part, a vector representing the fluid level in each fluid place, $\mathbf{z} = (z_k, k \in \mathcal{P}_c) \in \mathcal{Z}$. The initial marking is $\mathbf{m}_0 = (s_0, \mathbf{z}_0)$. The total number of discrete states is S, the set of all discrete states is denoted $\mathcal{S}$.

$\mathrm{r}_{t,p}$, $\mathrm{r}_{p,t}$: $\mathcal{A}_c \times \mathcal{M} \to \mathbb{R}^2$ is the flow rate function along the fluid arc connecting the timed transition t and the fluid place p or vice versa. The flow rate $r = (\mu, \sigma^2)$ is a normally distributed random variable specified by expectation μ and variance σ^2. If it is specified by only one parameter, that one is the expectation and the variance is equal to zero.

Let Z_k be the fluid level in place $k \in \mathcal{P}_c$. The change rate of the fluid level is again a normally distributed random variable with

$$\frac{dZ_k}{dt} = \sum_{t \text{ enabled in } \mathbf{m}} r_{t,p}(\mathbf{m}) - \sum_{t \text{ enabled in } \mathbf{m}} r_{p,t}(\mathbf{m}).$$

Reflecting boundaries assure, that the fluid level in each place stays within its range $[z^{\min}, z^{\max}]$. For the sake of clearness and simplicity we write the first argument (the arc) as a subscript, defined by place and transition (source and destination).

$g_t : \mathcal{M} \to \mathbb{B}$ is the guard of transition t that can be a function of the discrete and the continuous state. $\lambda_t : \mathcal{M} \to \mathbb{R} \cup \{\infty\}$ is a function of both the continuous and the discrete marking. Immediate transitions have firing rate $\lambda_t = \infty$.

The weight function $w_t : \mathcal{S} \to \mathbb{R}$ is defined for immediate transitions. The firing probability of each of the enabled immediate transitions in a vanishing state is $w_t(s) / \sum_{t_i \text{enabled in } \mathbf{m}} w_{t_i}(s)$.

The enabling rules for the transitions are the same as in discrete Petri net models. The fluid arcs act only on the fluid places and do not influence the enabling conditions of the transitions.

The reachability graph, for our system represented in Fig. 6, gives a graphical representation of the stochastic process underlying the model. Augmenting it with continuous variables, one for each fluid place, gives a description of the full stochastic process which is formally defined as

$$\{(S(t), Z_1(t), Z_2(t)), t \in \mathbb{R}_0^+\}, \tag{1}$$

if the model has two fluid places. $S(t)$ is a random variable for the discrete marking s at time t, and $Z_k(t)$ is a random variable, representing the fluid level in fluid place k at time t. Let $\pi_s(t, \mathbf{z}) = \partial/\partial\mathbf{z}\ P(S(t) = s, \mathbf{Z}(t) \le \mathbf{z})$ denote the transient probability of being in discrete state s with fluid levels in an infinitesimal environment around z_k at time t, for all fluid places $k \in \mathcal{P}_c$.

The stochastic process (1) can be mapped on a system of partial differential equations with appropriate initial and boundary conditions.

$$\frac{\partial}{\partial t}\boldsymbol{\pi} = \frac{1}{2}\frac{\partial^2}{\partial z_1^2}(\boldsymbol{\Sigma}_1^2\boldsymbol{\pi}) + \frac{1}{2}\frac{\partial^2}{\partial z_1^2}(\boldsymbol{\Sigma}_2^2\boldsymbol{\pi}) - \frac{\partial}{\partial z_1}(\boldsymbol{M}_1\boldsymbol{\pi}) - \frac{\partial}{\partial z_2}(\boldsymbol{M}_2\boldsymbol{\pi}) + \mathbf{Q}^T\boldsymbol{\pi}, \tag{2}$$

where the probability mass density $\boldsymbol{\pi}$ and the system parameter matrices $\boldsymbol{M}_k = \operatorname{diag}(\mu_{k,1}, \ldots \mu_{k,S})$ and $\boldsymbol{\Sigma}_k^2 = \operatorname{diag}(\sigma_{k,1}, \ldots \sigma_{k,S})$ depend on the two fluid (spatial) variables z_1 and z_2. Also $\boldsymbol{\pi}$ depends on the time variable t. $\boldsymbol{M}_k$ and $\boldsymbol{\Sigma}_k^2$ are the first two moments of the normally distributed change rate function $\frac{dZ_k}{dt}$ (cf. [10]).

Equation (2) is a system of S differential equations, one for each discrete state. These equations are coupled by the term $\mathbf{Q}^T\boldsymbol{\pi}$, that describes the transition from one state to another. The *generator matrix* $\mathbf{Q}$ contains the transition rates.

Two different discretization methods for this system of equations will be discussed in the next section.

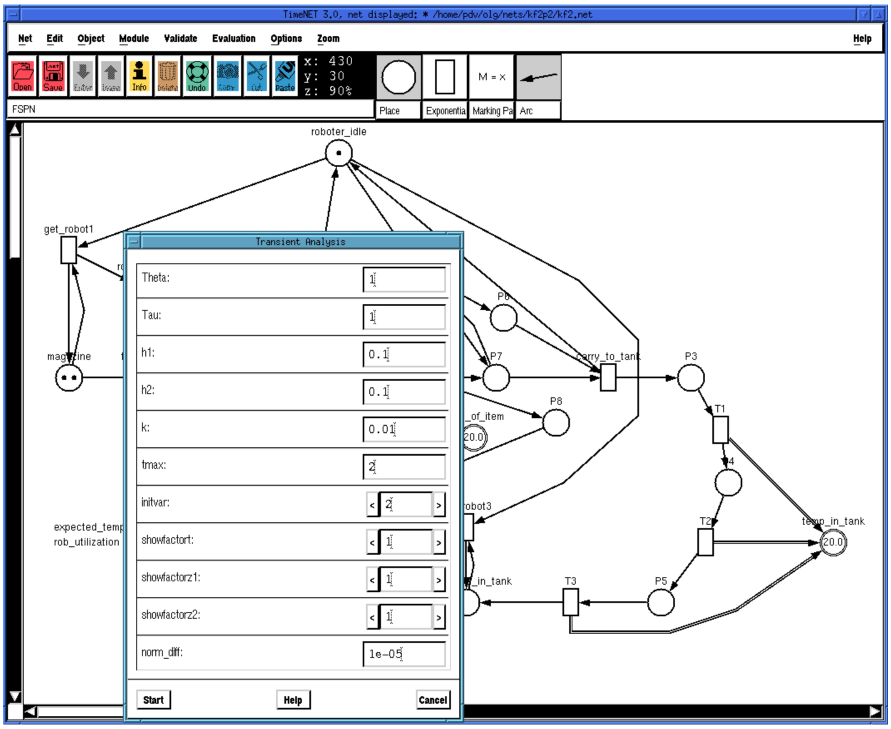

Fig. 1. Screen shot of the software tool

A screen shot of the graphical editor of the tool TimeNET [13] as it has been used for the analysis of the model in this paper can be seen in Fig. 1. In the graphical editor window the general model is displayed. All the textual description in Fig. 9 is hidden in attributes of the objects. The small window shows the parameters of the transient analysis.

3 The numerical solution algorithms

In this section we will describe two algorithms to solve the system of partial differential equations (2). First we introduce a discretization using a θ-*scheme* with *upwind strategy* together with boundary conditions (BCs) that results in a difference scheme which conserves the probability mass.

The discretization poses the problem to solve a system of linear difference equations given by a 5-banded matrix. In fact, no standard difference scheme for the chosen model yields a more simple matrix structure. But still the numerical solution is rather expensive: we use an iterative solution method, that works on sparse matrices, and thus has a computational effort of linear order for each iteration step (for full matrices it is of quadratic order). This means that it can be quite expensive for high accuracy; nevertheless it is less costly than direct methods.

To reduce expenses we consider a second solution algorithm: the *alternating direction implicit scheme* (ADI). This method is based on the discretization described above and splits the scheme into two successive steps with triangular matrices. These systems can be solved by a triangular Gaussian elimination, the so called *Thomas algorithm* (cf. [9]) with linear effort.

3.1 The θ-scheme

The discretization of the system of differential equations is carried out on an equidistant grid with step-size $\triangle t$ in time and step-sizes $\triangle z_1, \triangle z_2$ in the two space directions z_1 and z_2. At the grid points the function $\pi_s(t, z_1, z_2)$ is approximated by the discrete function $u^n_{s,j}$ with

$$\begin{aligned} u^n_{s,j} \approx \pi_s(nk, z_1^{\min} + j_1 h_1, z_2^{\min} + j_2 h_2) \qquad & \text{for } n = 0, \dots, T \quad \text{and} \\ j_1 = 0, \dots, N_1, \quad j_2 = 0, \dots, N_2 \quad \text{with} \quad & j = j_1 N_2 + j_2. \end{aligned} \tag{3}$$

The system of equations has been discretized using a θ-scheme. The *implicitness parameter* θ weighs two neighbouring time levels. It can take on all values between zero and one. Depending on θ the discretization is explicit ($\theta = 0$), implicit ($\theta = 1$) or the well-known Crank-Nicolson scheme ($\theta = 1/2$). By $\boldsymbol{u}_j^{n+\theta}$ we will denote the weighted sum

$$\boldsymbol{u}_j^{n+\theta} := \theta \boldsymbol{u}_j^{n+1} + (1-\theta)\boldsymbol{u}_j^n. \tag{4}$$

For the first order spatial derivatives an *upwind scheme* is used (cf. [8]). 'Upwind' means that forward ($D^+_{z_k}$ in space direction z_k) and backward difference quotients ($D^-_{z_k}$) are used, weighed with the *upwind parameter* $\varrho_{k,s}$ and $(1-\varrho_{k,s})$, respectively. The parameter $\varrho_{k,s}$ depends on the sign of the convection parameter $\mu_{k,s}$. This is motivated by the fact that if e.g. $\mu_{k,s}(z_1, z_2) > 0$, the mass moves to the right and the backward difference is more appropriate in describing this motion and thus yields a stable scheme. Hence, a reasonable choice of $\varrho_{k,s}$ is the following:

$$\begin{array}{lll} \mu_{k,s}(z_1, z_2) > 0: & 0 \le \varrho_{k,s} < \frac{1}{2} & \text{backward difference weighs more,} \\ \mu_{k,s}(z_1, z_2) \le 0: & \frac{1}{2} \le \varrho_{k,s} \le 1 & \text{forward difference weighs more,} \end{array} \tag{5}$$

where s is the index of the state number. For the discretization of the second order spatial derivatives the second order difference quotient ($D^+_{z_k} D^-_{z_k}$) and for the time derivative the forward difference quotient is used. The coupling term $\boldsymbol{Q}^T \boldsymbol{\pi}$ is discretized explicitly, using the two previous time levels, not the new time level. Therefore the set up system of difference equations can be decoupled. A separate system of linear difference equations is solved for each discrete state, and the coupling term appears in each one as an inhomogeneity.

Thus the discrete form of (2) reads

$$\begin{aligned}\frac{\boldsymbol{u}_j^{n+1}-\boldsymbol{u}_j^n}{\Delta t} &= \frac{1}{2}\frac{1}{\Delta z_1{}^2}D_{z_1}^+D_{z_1}^-\left(\boldsymbol{\Sigma}_{1,j}^2\boldsymbol{u}_j^{n+\theta}\right)+\frac{1}{2}\frac{1}{\Delta z_2{}^2}D_{z_2}^+D_{z_2}^-\left(\boldsymbol{\Sigma}_{2,j}^2\boldsymbol{u}_j^{n+\theta}\right)\\ &-\frac{1}{\Delta z_1}\left(\boldsymbol{R}_1D_{z_1}^+ + (\boldsymbol{I}-\boldsymbol{R}_1)\,D_{z_1}^-\right)\left(\boldsymbol{M}_{1,j}\boldsymbol{u}_j^{n+\theta}\right)\\ &-\frac{1}{\Delta z_2}\left(\boldsymbol{R}_2D_{z_2}^+ + (\boldsymbol{I}-\boldsymbol{R}_2)\,D_{z_2}^-\right)\left(\boldsymbol{M}_{2,j}\boldsymbol{u}_j^{n+\theta}\right)\\ &+\boldsymbol{Q}^T\left((1-\theta)\,\boldsymbol{u}_j^n-\theta\boldsymbol{u}_j^{n-1}\right).\end{aligned}$$

When a reflecting boundary condition is implemented, usually the analytic BC is discretized in an intuitive way. But the discrete scheme will not necessarily adopt conservation properties of the analytic equation. Therefore we use a *discrete reflecting boundary condition*, which is called discrete, because it is completely derived on a discrete level. With these BCs the overall difference scheme conserves the probability mass and makes the commonly applied normalization step obsolete. An additional normalization has three disadvantages: first of course, it increases the computational effort, second the discrete model is no good approximation of the analytic model, since it has different properties and third it needlessly produces additional computational errors. To derive the discrete reflecting BC we sum up the difference equation at each interior grid point. This sum can be replaced using a discrete form of the mass conservation law, which involves all grid points and thus yields a condition for the grid points at the boundary. For more details on the θ-scheme and the reflecting boundary conditions we refer to [14].

3.2 The ADI method

The two-dimensional scheme is unconditionally stable (for the Crank-Nicolson scheme or any scheme with $\theta \geq 0.5$) but the discretization of both spatial derivatives yields a 5-band matrix, that is much more difficult to invert at each time step than the tridiagonal matrices encountered in one-dimensional problems. The ADI method is a strategy to reduce the two-dimensional problem to two successive one-dimensional problems.

For the discretization we again use the θ-scheme with upwind strategy as described above. We will show how to add a 'zero' to the discrete equation, that allows the equation to be factored. By introducing an artificial intermediate time step, the scheme can be split into two steps. At each of these steps a tridiagonal system of equations has to be solved.

Let $\boldsymbol{A}_1, \boldsymbol{A}_2$ denote the operators describing the discretization of the first and second order spatial derivatives in z_1 and z_2 direction respectively, i.e.

$$\boldsymbol{A}_{k,s}\boldsymbol{u}_s = -\frac{1}{2}D_{z_k}^+D_{z_k}^-\left(\boldsymbol{\Sigma}_{k,s}^2\boldsymbol{u}_s\right)+\left(\boldsymbol{R}_{k,s}\,D_{z_k}^+ + (\boldsymbol{I}-\boldsymbol{R}_{k,s})\,D_{z_k}^-\right)\left(\boldsymbol{M}_{k,s}\boldsymbol{u}_s\right)\,, \tag{6}$$

for $k = 1, 2$. The vector $\boldsymbol{u}_s$ denotes the approximation of the probability density function π_s at the grid points. The dimension of $\boldsymbol{u}_s$ is the number of spatial grid points. $\boldsymbol{R}_{k,s}$ contains the upwind parameter $\varrho_{k,s}$ at each grid point. For simplicity of notation, we will omit the index s for the operators $\boldsymbol{A}_{k,s}$.

With the abbreviation $\boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q}$ for the discretization of the coupling term, the discrete form of (2) reads

$$\frac{\boldsymbol{u}_s^{n+1} - \boldsymbol{u}_s^n}{\triangle t} = -\theta \left(\boldsymbol{A}_1 + \boldsymbol{A}_2\right) \boldsymbol{u}_s^{n+1} - (1-\theta) \left(\boldsymbol{A}_1 + \boldsymbol{A}_2\right) \boldsymbol{u}_s^n + \boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q} \tag{7}$$

with an error in time of $O(\triangle t)$ and $O(\triangle t^2)$ for the Crank-Nicolson scheme; or sorted for $\boldsymbol{u}_s^{n+1}$ and $\boldsymbol{u}_s^n$

$$(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_1 + \theta \triangle t \boldsymbol{A}_2)\boldsymbol{u}_s^{n+1} = \left[\boldsymbol{I} - (1-\theta)\, \triangle t \boldsymbol{A}_1 - (1-\theta)\, \triangle t \boldsymbol{A}_2\right] u_s^n + \triangle t\; \boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q}\,. \tag{8}$$

The 'trick' now is to add the quadratic term $\triangle t^2 \theta^2 \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{u}_s^{n+1}$ on both sides of the equation and to add the terms $\pm(1-\theta)^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{u}_s^n$ on the r.h.s.

$$\begin{aligned}(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_1 &+ \theta \triangle t \boldsymbol{A}_2 + \theta^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2)\boldsymbol{u}_s^{n+1} \\ &= \left[\boldsymbol{I} - (1-\theta)\triangle t \boldsymbol{A}_1 - (1-\theta)\triangle t \boldsymbol{A}_2 + (1-\theta)^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2\right] \boldsymbol{u}_s^n \\ &\quad + \triangle t\; \boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q} + \theta^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2 u_s^{n+1} - (1-\theta)^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{u}_s^n\,. \end{aligned} \tag{9}$$

Now the equation can be factored as

$$\begin{aligned}(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_1)&(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_2)\boldsymbol{u}_s^{n+1} \\ &= (\boldsymbol{I} - (1-\theta)\triangle t \boldsymbol{A}_1)(\boldsymbol{I} - (1-\theta)\triangle t \boldsymbol{A}_2)\boldsymbol{u}_s^n + \triangle t\; \boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q}\,. \end{aligned} \tag{10}$$

The term $\theta^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{u}_s^{n+1} - (1-\theta)^2 \triangle t^2 \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{u}_s^n$ is of order $O(\triangle t^2)$ and thus of less (or equal) order than the ordinary discretization error.

Finally we split (10) by introducing an artificial intermediate value $\tilde{\boldsymbol{u}}_s$:

$$(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_1)\tilde{\boldsymbol{u}}_s = (\boldsymbol{I} - (1-\theta)\triangle t \boldsymbol{A}_2)\boldsymbol{u}_s^n + \theta \triangle t\; \boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q} \tag{11a}$$

$$(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_2)\boldsymbol{u}_s^{n+1} = (\boldsymbol{I} - (1-\theta)\triangle t \boldsymbol{A}_1)\tilde{\boldsymbol{u}}_s + (1-\theta)\triangle t\; \boldsymbol{u}_s \underset{n-1}{\overset{n}{\textcircled{T}}} \boldsymbol{Q}\,. \tag{11b}$$

A homogeneous two step scheme of this kind is called *Peaceman-Rachford scheme* and has the same stability behaviour as the underlying spatial discretization [8].

To verify this splitting, multiply (11a) with $(\boldsymbol{I} - (1-\theta)\triangle t \boldsymbol{A}_1)$ and (11b) with $(\boldsymbol{I} + \theta \triangle t \boldsymbol{A}_1)$ and use (a) to replace $\tilde{\boldsymbol{u}}_s$ in (b). This yields (10).

In each step of the ADI method just one z-direction is implicit, the other direction is explicit. This is alternated in the next step. For the efficient solution the exact structure of the matrices on the l.h.s. of (11) is important: (11a) is already tridiagonal, (11b) is 3-banded. To use the fast tridiagonal solver also for step (b), we reorder our data $\tilde{\boldsymbol{u}}_s$ after step (a). Reordering of data is equivalent to changing

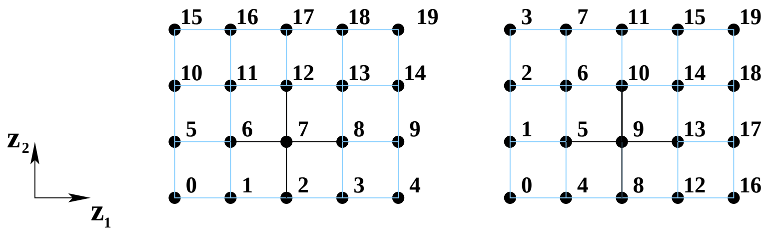

Fig. 2. Numeration of grid points for step (11a) row wise (left) and for step (11b) column wise (right)

the order of numeration from row wise to column wise as illustrated in Fig. 2. In the row wise numeration points 2,7 and 12 are involved in the spatial differences in z_2-direction in point 7. This yields a diagonal element and elements in the fifth off-diagonals. In the column wise numeration points 8,9 and 10 are involved in the spatial differences in z_2-direction in point 9. This results in entries in the diagonal and the first off-diagonals.

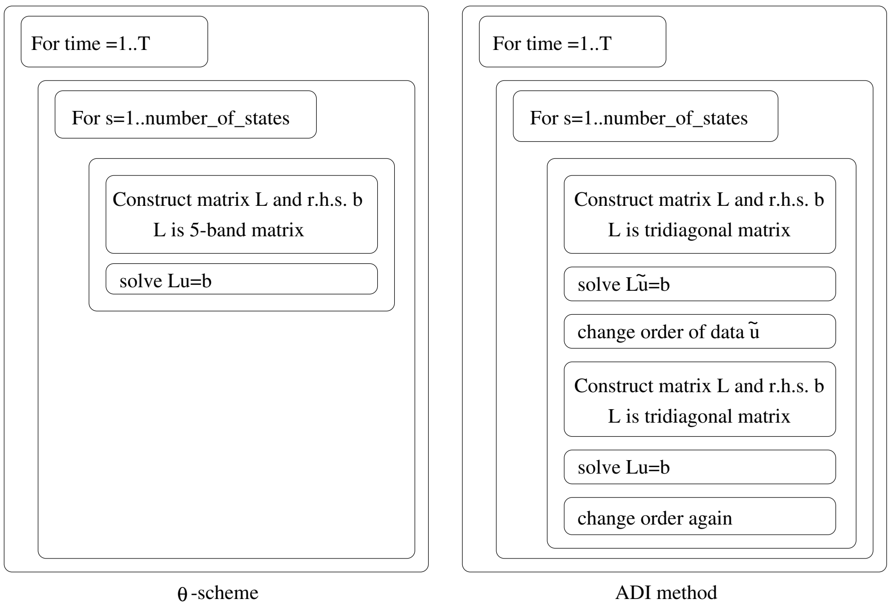

Fig. 3. Comparison of the structural progress of the θ-scheme and the ADI method

We used the overset tilde for the intermediate step to emphasize the fact, that it is no solution of our problem, but merely an artificial quantity.

Usually the development of boundary conditions for an ADI scheme is troublesome. For the intermediate quantity boundary values have to be prescribed, generally by extrapolation. Our aim was again to derive BCs that yield a probabil-

Table 1. Comparison of cpu-times of the two implemented algorithms for a test model

solution algorithm	cpu-time
ADI method	8 min 50
θ-scheme, $\varepsilon_{req}= 10^{-6}$	15 min 46
θ-scheme, $\varepsilon_{req}= 10^{-9}$	18 min 51
θ-scheme, $\varepsilon_{req}= 10^{-13}$	21 min 42

ity mass conserving discrete scheme. With the same technique as in the 'ordinary' θ-scheme it is possible to develop discrete reflecting boundary conditions for the ADI method. These BCs conserve the probability mass in each of the two steps (11) and therewith also the mass after a complete time step.

The structural flow of the two algorithms is shown in Fig. 3. While the gain in memory for the ADI method is comparatively small, it is much faster than the θ-scheme. The computational complexity of both algorithms is linear in the number of discrete states S. But whereas the computational effort of the ADI method is independent of the model (for a fixed S), the iterative solution process of the θ-scheme terminates, when a required precision ε_{req} is reached. This depends strongly on the properties (e.g. the spectral norm) of the iteration matrix and thus on the model. We compared the algorithms on a Sun Ultra 5 with 333 MHz for a small test model with only three discrete states, 65×65 grid points and 200 time steps. In Table 1 we present the cpu-time of the θ-scheme and of the ADI method. The choice of $\varepsilon_{req} = 10^{-13}$ yields the same accuracy in the probability mass as the direct solution with the ADI method, but the ADI method is more than twice as fast. Even for a reduced accuracy of $\varepsilon_{req} = 10^{-6}$ the θ-scheme is comparatively slow.

In the following section we will regard a system that is modelled and solved with the ADI method.

4 The System

The system studied in this paper is a hybrid automated surveyance system in which metal pieces are heated and later cooled down again in a water tank. The metal pieces are the discrete items in the system, whereas the temperatures of both the water in the tank and the metal pieces are continuous variables. The control of the system switches between discrete states, depending on the discrete state and the values of the continuous variables.

The elements of the considered system are a magazine, holding small metal pieces, a heater on which those pieces are warmed up, a water tank in which the pieces cool down again - while increasing the temperature of the water - and a robot that is used for transportation.

The system performs the following control sequence: the robot carries a piece to the heater, where this item stays until some temperature is reached, then the robot picks the work piece up from the heater and carries it to the water tank in which the metal piece is placed. When the metal piece has cooled down to some

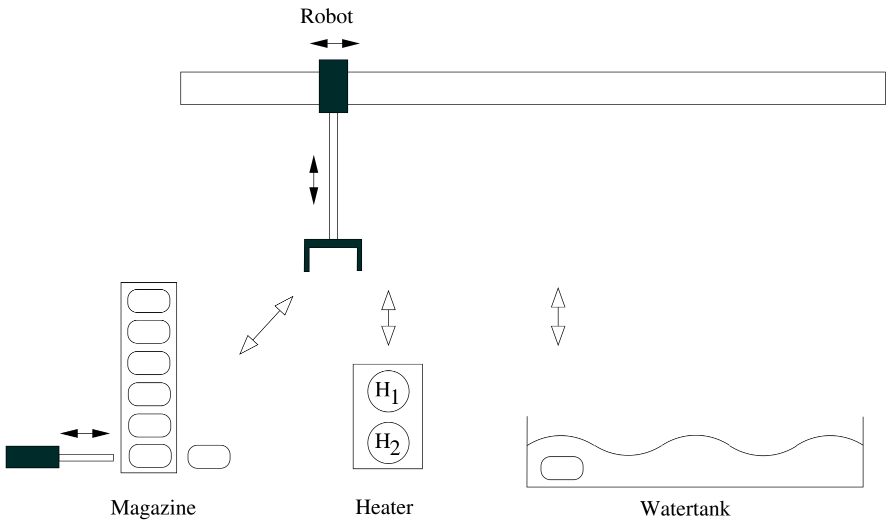

Fig. 4. Sketch of the hybrid manufacturing system

predefined temperature it is removed from the water tank and placed back on the heater. Figure 4 shows a sketch of the hybrid system.

Moving the robot to the position where it is needed (either the magazine, the heater or the tank) takes on the average 10 seconds, as does transportation of a work piece from one location to the next. The control sequence creates an action loop in the system, which will in a real nuclear power plant be ended after a known number of iterations. In our model system there is no limit to the number of iterations.

All parts have an initial temperature of 20° Celsius. Throughout this paper temperature is measured in degree Celsius. Transportation of items between locations and movements of the robot in general are assumed to be exponentially distributed with mean value of 10 seconds - that corresponds to a rate $\lambda = 0.1$, the time unit is one second, the metal pieces are assumed to be round disks with 25 mm height and a diameter of 40 mm. The water tank holds 0.5 $dm^3 = 0.5$ liters. In most cases the assumption of exponentially distributed times is a simplification. In availability or reliability models where we seek very precise results (e.g. 'five nines') a wrong distribution will severely change results. Here, however, we want to point out that there is variation and slight randomness in all execution times in technical systems, so the possibly slight uncorrectness of the exponential distribution will not significantly bias the results.

5 FSPN models of the hybrid system

In this section we will first present a simple model, that is specially designed to represent the situation with only one work piece in the system. The main metrics of interest are utilization of the robot - that is the probability that the robot is busy - and the temperature of the water in the tank.

A general model in which an arbitrary number of work pieces can be used will be presented in the subsequent subsection. The general model should give the same results as the simple one with respect to the two regarded measures if only one work piece is used.

5.1 A simple model of the hybrid system

Figure 5 shows an FSPN model of the hybrid system. The transitions *get_roboti* represent the time it takes to position the robot where it is needed. We study here a model with only one piece in the magazine. Initially this work piece is in the magazine and the robot is idle. After the robot has reached the magazine it picks up the metal piece and carries it to the heater. Here the work piece is put down and the robot is idle again.

While the piece lies on the heater transition *heat* is enabled and *temp_of_item*, the temperature of the metal piece increases as specified by rate *r1*. For this rate, as for both the other rates, only the expectation is given and the variance is assumed to be zero. Assuming a constant increase or decrease of the temperature might not be realistic, but to specify the variance of the rates is a problem we have not yet addressed. The expectation of the rates can be computed from the system description [7], but nothing is known about the nature of the variances. Transition *T8* has a fluid-dependent firing rate *q1* and is activated only if the temperature of the item on the heater exceeds $70°$. We model here only one heater position although the original system has two, for one item can only lie in one position. Once the heating process is finished the robot is called to the heater again. It picks up the item and carries it to the water tank, in which the item is placed. The temperature of the metal piece decreases (as represented by *r2* since *T2* is enabled) and the temperature of the water increases (through *r3*). Transition *T2* again has a fluid-depended firing rate and is enabled only if the item has cooled down to at most $30°$. It will then still cool down a bit while the robot moves to the tank. The item is picked up and carried back to the heater to be warmed up once more.

The rates *r1, r2* and *r3* are computed following the formula for temperature changes:

$$\frac{d\vartheta}{dt} = \frac{\dot{Q}_{in} - \dot{Q}_{out}}{mc_p} \tag{12}$$

where $\dot{Q}$ is the thermal input or output stream, m is the mass of the material whose temperature change shall be computed and c_p is the specific thermal capacity of the material. The change of temperature occurs where two substances of different temperature meet. It is proportional to the size of the contact area A and a coefficient of warmth transmission α. $\dot{Q}$ is defined as

$$\dot{Q} = \alpha A(\vartheta_1 - \vartheta_2). \tag{13}$$

The needed parameters for heating up the iron metal piece on the heater ($r1$), are $c_p = 0.46$, $A = 56.55cm^2$ and $\alpha = 13.85W/Km^2$. The heater has a heating power of 200 Watts (W), which is the warmth input to the metal piece. In the technical parameters temperature is always measured in Kelvin (K), whereas in all figures we use degree Celsius[1].

[1] $[°Celsius] + 273 = [Kelvin]$

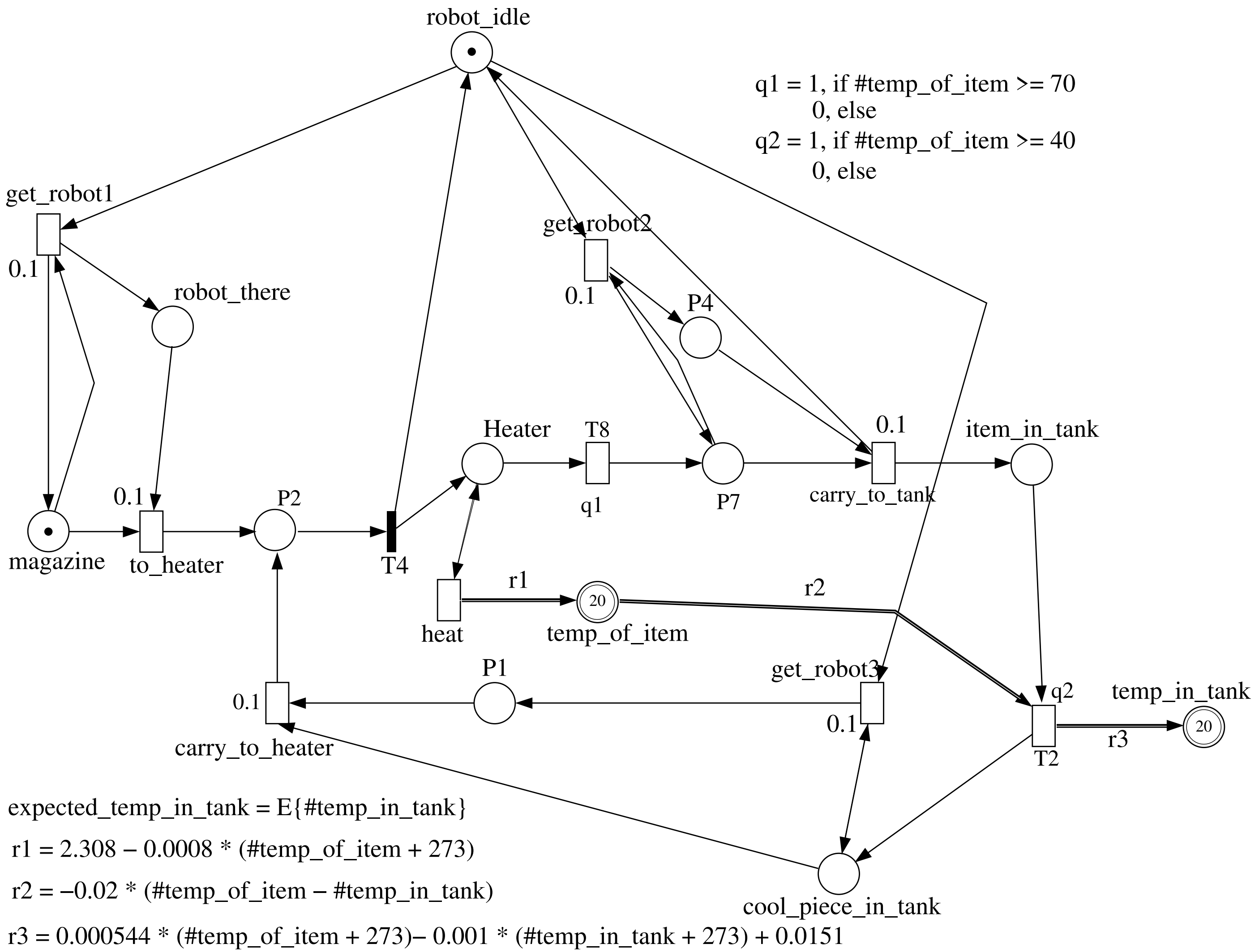

Fig. 5. FSPN model of the hybrid manufacturing system

Analogously, $r2$ is the temperature change of the work piece in contact with the water and $r3$ is the temperature change of the water, when being in touch with a metal piece. All rates are computed following the same principles.

Figure 6 shows the reachability graph of this simple model. The states are labelled with the discrete marking of the net and in addition they are enumerated, for simplicity in referencing.

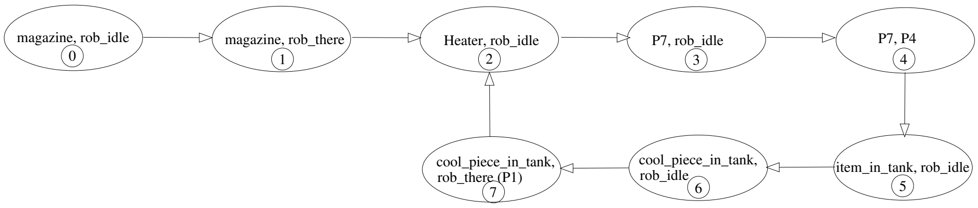

Fig. 6. Reachability graph to the FSPN model

In Fig. 7 and Fig. 8 some results are shown. The full solution of the model is a two-dimensional density at each point in time when summing over all discrete states. In each discrete state the curves are fractions of a two-dimensional density. Summing over the discrete states and then computing the mean values for each continuous variable yields a transient curve for the mean temperatures as shown in Fig. 7. Those means converge to the same and fairly low value. This shows that the amount of water is sufficient for cooling down the metal piece.

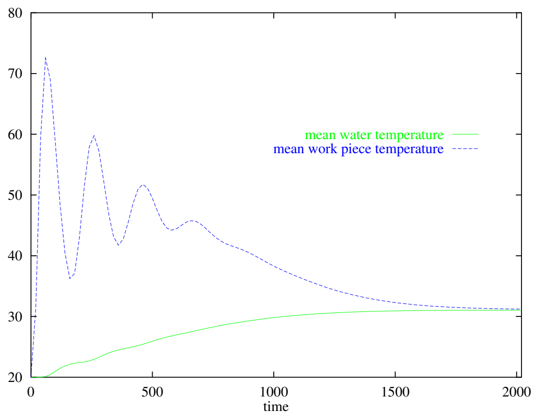

Fig. 7. Mean item temperature and mean water temperature

Fig. 8 shows the complement of the robot utilization (that is the probability that the robot is idle, Prob($\#rob\_idle = 1$)). The robot initially is used, but on the long-term average this usage is almost negligible. Most of the time the robot is idle. Consequently, the robot would be well capable of serving a system with more

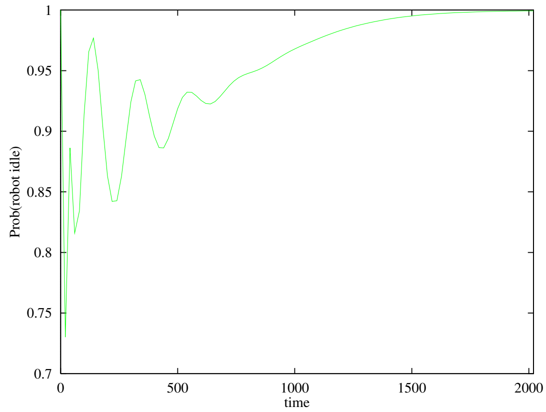

Fig. 8. Probability that the robot is idle

metal pieces. How many of them the system configuration can deal with well will be one of the questions posed to a more complex model.

5.2 A general model of the hybrid system

In this section the model from the previous section is in one way simplified so we can make it more general in a different way. We want to allow for an arbitrary number of work pieces to be present in the system. This means that we can no longer trace the temperature of each work piece, since no more than two continuous dimensions can be used in a model. Therefore we have to partially discretize the warming process of the water. We do so by splitting the heating period into three phases, one where the piece is still very hot, a second where it is medium warm, and a third where it has almost cooled down completely. The rates at which the water is heated up (r_3, r_4 and r_5) are computed by assuming a fixed temperature of the work piece in each of the three phases and evaluating rate r_3 in the simple model for each of them. These temperatures are 70° in the first phase, 45° in the second phase and 22° in the third phase. In doing so, two additional discrete states are included in the model.

The discrete part of the model remains more or less the same as in the simple model, but the continuous part has to be changed significantly. Fluid place *temp_of_item* now denotes the temperature of the item that is present on the heater. As the item leaves the heater, the temperature has to be reset, so that the fluid place will represent the temperature of the next item to be put on the heater. The reset of the heater position temperature is modelled with the fluid rate r_2. An additional discrete state for the reset is inserted. The transition at the target of the fluid arc labelled with r_2, transition q_2, is enabled only when the item has left the heater and the heater position temperature is still beyond 30°. Before the temperature of the item on the heater is reset no new work piece can be positioned on the heater.

Figure 9 shows the generalized FSPN model where K is the number of work pieces in the magazine. We first chose $K = 1$ and then this model should be equivalent to the simple model shown in Fig. 5. Equivalent here means in first

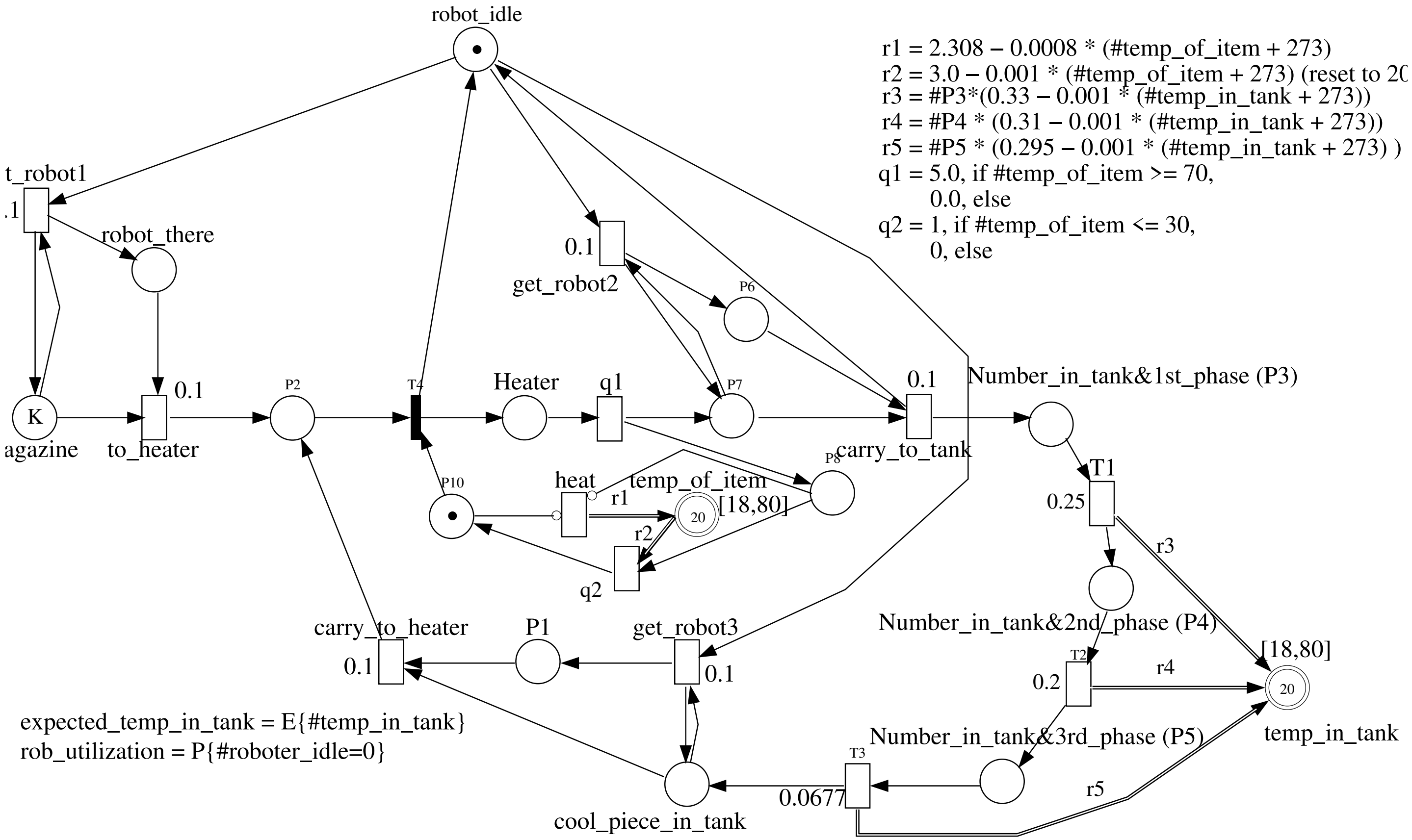

Fig. 9. General FSPN model of the hybrid manufacturing system

place, that the temperature of the water tank increases in a similar way in both models. We would also wish the models to be equivalent in the sense that the robot utilization should be similar. With $K = 1$ the model has 15 discrete states. The reachability graph is omitted here, since it is constructed following the same principles as the reachability graph for the simple model.

Figure 8 on the right hand side shows the mean water temperature for the simple model and the generalized model with only one work piece. These means do not increase in exactly the same way, but their behaviour is fairly similar and the stationary values matched sufficiently well.

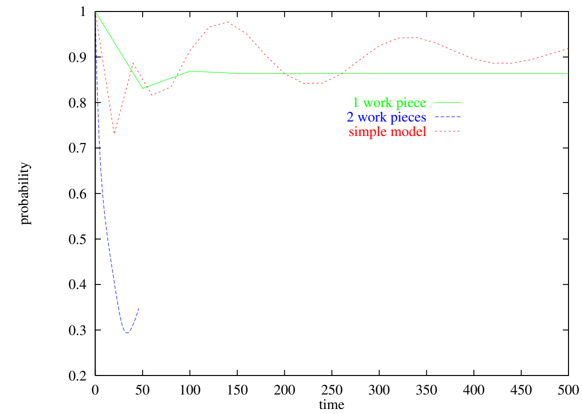

Fig. 10. Comparison of the probability that the robot is idle

With respect to the robot utilization, however, the models are not yet equivalent as can be seen in Fig. 10. Figure 10 shows the probability that the robot is idle in the different models. Robot utilization in the generalized model is much higher than in the simple model. This is most likely due to the fact that in the generalized model the robot has to wait holding a work piece in front of the heater until the heater position temperature is reset. We are still working on matching the robot utilization as well. A future refinement of the model should either increase the speed of reset of the heater, or model in such a way, that the robot is only called when the heater is immediately available.

Increasing K rapidly increases the complexity of the stochastic process. For $K = 2$ the model has already 74 discrete states and for $K = 3$ the model has 216 discrete states. Since in each discrete state the full continuous state space must be stored, complexity soon reaches a limit where none of our machines will be able to store the matrices anymore and runtime increases tremendously. In this paper we only solve for $K = 1, 2$. It can be seen already, that increasing the number of work pieces in the model with severely increase the usage of the robot. The water, however, is warmed equally slowly with two work pieces as it was earlier with only one. So the cooling capacity of the water seems to be sufficient for some more work pieces.

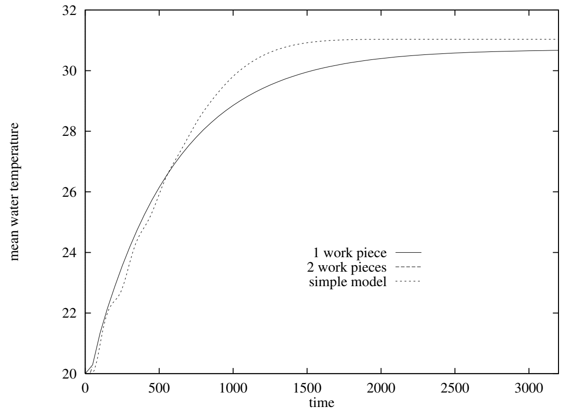

Fig. 11. Comparison of the mean water temperatures

6 Conclusions

Fluid stochastic Petri nets are designed to intuitively represent hybrid systems. The formalisms hides from the user the fact that a discrete solution is computed by the algorithm. Formulating the model on the discrete level would be highly unattractive. Not alone drawing a model that has some hundreds of places and transitions instead of one is not pleasant but also the solution of this model would not be any faster than our algorithm only more inaccurate. Analysis of hybrid systems using discrete models is not feasible, using FSPNs it becomes very simple.

In this paper we have introduced a new numerical solution algorithm for solving the system of equations underlying fluid stochastic Petri nets with two continuous dimensions. The algorithm implements an alternating direction implicit (ADI) discretization scheme. Formerly we used a θ-discretization. Due to especially constructed discrete reflecting boundary conditions, both schemes conserve the probability mass. It could be shown that the ADI method is about twice as fast as the θ-scheme, while not increasing the amount of memory needed.

We have used the new solution method to solve different models of a well-known manufacturing system, where work pieces are carried by a robot first to a heater and then to a water tank. On the heater the work piece is heated up and it is put in the water tank in order to cool down, while the water is warmed. These models allow us to obtain metrics such as the utilization of the robot and the mean temperature of the water tank. Appropriate dimensioning of the system as well as checking of security requirements can be carried out on the basis of the presented models.

Like many performability models and most hybrid models, our models suffer from numerical complexity, while the models as such are comparatively small. This problem can be addressed in different ways: by choosing even simpler models and by improving solution methods. The implementation of the ADI scheme is a step in the latter direction. If high accuracy is not a requirement one could improve the solution process still significantly by exchanging the data type *double* with the less accurate *float* in the implementation.

Acknowledgements

The first and third author's work is supported by the German Research Council (DFG) under Grant No. Ho 1257/11-3, and the second author's work is supported by DFG under Grant No. AR 277/3-1.

References

1. M. Ajmone Marsan and G. Balbo and G. Chiola and S. Donatelli and G. Franchеschinis (1995) Modelling with Generalized Stochastic Petri Nets. John Wiley & Sons
2. A. Bobbio and S. Garg and M. Gribaudo and A. Horváth and M. Sereno and M. Telek (1999) Modeling Software Systems with Rejuvenation, Restoration and Checkpointing through Fluid Stochastic Petri Nets. Proc. Eighth International Workshop on Petri Nets and Performance Models - PNPM'99, pages 82–91
3. G. Ciardo and D. Nicol and K.S. Trivedi (1999) Discrete-event Simulation of Fluid Stochastic Petri Nets. IEEE Trans. Softw. Eng. 25(2): 207–217
4. M. Gribaudo and M. Sereno and A. Bobbio (1999) Fluid Stochastic Petri Nets: An Extended Formalism to Include non-Markovian Models. In Proc. Eighth International Workshop on Petri Nets and Performance Models - PNPM'99, pages 74–81. Zaragoza, Spain
5. G. Horton and V. G. Kulkarni and D. M. Nicol and K. S. Trivedi (1998) Fluid Stochastic Petri Nets: Theory, Applications and Solution. European Journal of Operations Research, 105(1): 184–201
6. D. M. Nicol and A. S. Miner (1995) The Fluid Stochastic Petri Net Simulator. Proc. Sixth International Workshop on Petri Nets and Performance Models - PNPM'95, pages 214–215. Durham, North Carolina, USA. IEEE-CS Press
7. B. Nixdorf and J. Lunze (2000) Control of a manufacturing cell. Internal document
8. J. C. Strikwerda (1989) Finite Difference Schemes and Partial Differential Equations. Wadsworth & Brooks/Cole
9. J.W. Thomas (1995) Numerical Partial Differential Equations: Finite Difference Methods. Springer
10. K. Wolter (1999) Performance and Dependability Modelling with Second Order Fluid Stochastic Petri Nets. Shaker, Aachen
11. K. Wolter (2001) A Performability Model for a Hybrid Reactor System. K. Djemame and M. Kara, editors, Proc. 17th annual UK Performance Engineering Workshop, pages 13–22, Leeds, UK
12. K. Wolter and A. Zisowsky (2001) On Markov Reward Modelling with FSPNs. Performance Evaluation, 44:165–186
13. A. Zimmermann and R. German and J. Freiheit and G. Hommel (2000) Petri Net Modelling and Performability Evaluation with TimeNET 3.0. In Proc. 11th Int. Conf. on Computer Performance Evaluation; Modelling Techniques and Tools, LNCS 1786, pages 188–202. Schaumburg, IL, USA, March 2000
14. A. Zisowsky (1998) Entwurf und Implementierung eines Verfahrens für die transiente Analyse fluider stochastischer Petri-Netze. Diplomarbeit, TU Berlin