



SIMULATION VON HYBRIDEN PETRI NETZEN MIT DRUCKABHÄNGIGEN TRANSITIONEN

BACHELORARBEIT
zur Erlangung des akademischen Grades
BACHELOR OF SCIENCE

Westfälische Wilhelms-Universität Münster
Fachbereich Mathematik und Informatik
Institut für Informatik

Betreuung:

Prof. Dr. Anne Remke

Eingereicht von:

Alexander Bösel

Münster, Mai 2015

Plagiatserklärung

Hiermit versichere ich, dass die vorliegende Arbeit über „Simulation von Hybriden Petri Netzen mit druckabhängigen Transitionen“ selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken - auch elektronischen Medien - dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

Münster, 18. Mai 2015

Alexander Bösel

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

Münster, 18. Mai 2015

Alexander Bösel

Inhaltsverzeichnis

1	Einleitung	1
2	Physikalischer Hintergrund	3
2.1	Druckabhängiger Ausfluss	3
2.2	Zusätzliche konstante Ein- und Ausflüsse	6
2.3	Ergebnisse	8
3	Modellierung	12
3.1	Formale Definition	13
3.2	Zustand des Modells	15
3.3	Erlaubnis und Freigabe	15
3.4	Entwicklung der kontinuierlichen Variablen	16
3.5	Anpassung der effektiven Volumenströme	20
4	Simulation	24
4.1	Ereignissimulation	24
4.2	Anpassung an das Modell	25
4.3	Konstruktion der Ereignismenge	25
4.3.1	Ereignisse der kontinuierlichen Stellen	26
4.3.2	Ereignisse der Test- und Inhibitorkanten	28
4.4	Behandlung der Ereignisse	32
4.5	Betrachtete Größen	34
5	Implementierung	35
5.1	Anforderungen	35
5.2	Vorhandenes System	35
5.3	Systemarchitektur	36
5.3.1	Model	36
5.3.2	View	38
5.3.3	Controller	38
5.4	Ablauf einer Simulation	39

5.5	Persistenz	41
5.6	Generierung der Zufallszahlen	42
5.7	Numerische Schwierigkeiten	43
5.8	Erweiterbarkeit	44
6	Fallbeispiel	45
6.1	Fallbeschreibung	45
6.2	Auswertung	47
7	Zusammenfassung	49
	Abkürzungsverzeichnis	50
	Abbildungsverzeichnis	51
	Tabellenverzeichnis	52
	Auflistungsverzeichnis	53
	Algorithmenverzeichnis	54
	Literaturverzeichnis	55

1 Einleitung

In der heutigen Zeit werden kritische Infrastrukturen wie z. B. Gas-, Wasser- oder Elektrizitätsversorgung immer komplexer und beeinflussen sich gegenseitig. So kann z. B. ein Ausfall der Elektrizitätsversorgung zu einem Ausfall einer Pumpe und somit zum Ausfall der Wasserversorgung führen. Umso wichtiger erscheint es die Zuverlässigkeit kritischer Infrastrukturen gewährleisten können.

Zur Untersuchung dieser Strukturen, eignet sich der Modellformalismus der Hybriden Petri-Netze, wie in [DA05] eingeführt, der neben den klassischen diskreten Variablen noch kontinuierliche Variablen erlaubt. Um stochastische Vorgänge, wie das zufällige Eintreten von Fehlern, zu überprüfen, wurde das Modell zusätzlich um stochastische Variablen erweitert [GR13] und es wurde sowohl ein Simulator, als auch eine Analyse-Tool für die sogenannten Hybride Petri-Netze (HPnGs) entwickelt.

Hiermit können bereits viele Infrastrukturen auf ihre Zuverlässigkeit analysiert werden. Die Veränderungen der modellierten kontinuierlichen Variablen sind in dem Modellformalismus jedoch bisher auf lineare Übergänge beschränkt. Viele Übergänge, die in kritischen Infrastrukturen auftreten, wie z. B. das Laden einer Batterie oder die Bewegung von Gasen, sind aber nicht linear. Die aktuelle Forschung [ACG⁺15] beschäftigt sich mit diesen Problemen und sucht nach Lösungen, wie Anpassungen des Modells, um auch nicht-lineare Übergänge zu modellieren.

Eine allgemeine Lösung für nicht-lineare Übergänge erscheint vorerst sehr komplex. In dieser Arbeit wird daher zunächst ein konkreter nicht-linearer Übergang untersucht. Betrachtet wird das Auslaufen eines Tanks durch eine Öffnung in seinem unteren Bereich, bei dem die Strömungsgeschwindigkeit der auslaufenden Flüssigkeit abhängig vom Wasserdruck, bzw. abhängig vom Wasserstand ist. Ziel ist es das Modell der HPnGs so zu erweitern, dass Infrastrukturen, in denen druckabhängige Übergänge vorkommen, z. B. ein Wasserturm, der mehrere Verbraucher mit Wasser versorgt, modelliert werden können. Darauf aufbauend soll ein Simulator es ermöglichen die Zuverlässigkeit solcher Infrastrukturen zu testen.

Die Arbeit gliedert sich wie folgt: In Kapitel 2 untersuchen wir das Auslaufen eines

Tanks zunächst vor dem physikalischen Hintergrund und entwickeln Formeln, mit denen wir die zeitliche Veränderungen des Wasserstands berechnen können. Im darauffolgenden Kapitel 3 erweitern wir den Modellformalismus der HPnGs um druckabhängige Transitionen und passen in Kapitel 4 die Ereignissimulation an das erweiterte Modell an. Anschließend erläutern wir in Kapitel 5 die wichtigsten Aspekte und Entwurfsentscheidungen der Implementierung des Simulators. In Kapitel 6 modellieren wir mithilfe des neuen Formalismus die Wasserversorgung mit einem Wasserturm und testen sie unter Anwendung des Simulators auf ihre Zuverlässigkeit. Abschließend werden wir in Kapitel 7 die Ergebnisse dieser Arbeit zusammenfassen.

2 Physikalischer Hintergrund

In diesem Kapitel betrachten wir einen Tank mit verschiedenen Ein- und Ausflüssen und seinen physikalischen Größen. Zunächst werden wir einen Tank mit einem Ausfluss, der abhängig vom Füllstand des Tanks ist, betrachten und Formeln für die zeitliche Veränderung der Größen herleiten. Anschließend erweitern wir den Tank um einen zusätzlichen konstanten Ein- bzw. Ausfluss und leiten erneut die relevanten Formeln her. Wir vereinfachen hierbei an mehreren Stellen und bilden bei unserer Betrachtung nur eine Näherung zur Realität ab. Das Kapitel schließt mit einer Zusammenfassung der erzielten Ergebnisse ab und stellt die hergeleiteten Formeln auch graphisch dar. Schaltungen der Ein- und Ausflüsse werden hier noch nicht betrachtet.

2.1 Druckabhängiger Ausfluss

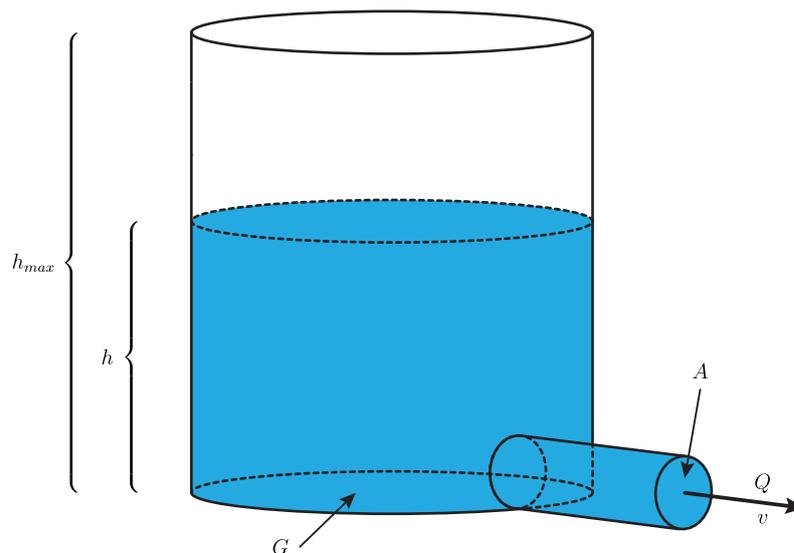


Abbildung 2.1: Tank mit druckabhängigem Ausfluss

Der in Abbildung 2.1 modellierte Tank zeichnet sich durch seine Grundfläche G , seine Höhe h_{max} und den Wasserstand h aus. Das Volumen der Wassermenge im Tank V , sowie das maximal Fassungsvermögen V_{max} des Tanks, sind dann proportional zu h und h_{max} .

$$V = G \cdot h \text{ und } V_{max} = G \cdot h_{max}. \quad (2.1)$$

Im unteren Bereich des Tanks geht eine Rohrleitung mit dem Querschnitt A ab, über die Wasser mit der Strömungsgeschwindigkeit v hinaus fließt. Die Strömungsgeschwindigkeit ist dabei abhängig vom Druck der Wassersäule im Tank. Wir gehen vereinfachend davon aus, dass der Ausfluss direkt am Boden des Tanks liegt und somit das komplette Wasser im Tank auf den Ausfluss drückt. Die Abhängigkeit zwischen v und h , bei fester Erdbeschleunigung g , ist durch das Gesetz von Torricelli [Gia10, S. 469] beschrieben:

$$v = \sqrt{2gh}. \quad (2.2)$$

Zusätzlich zur Strömungsgeschwindigkeit v betrachten wir den Volumenstrom Q , der proportional zu v ist und angibt wie viel Volumen pro Zeiteinheit fließt.

$$Q = A \cdot v. \quad (2.3)$$

Betrachtet man die zeitliche Veränderung des Wasserstands h im Tank, fassen wir die Größen h, V, v und Q als Funktionen der Zeit t auf.

Das Volumen der Flüssigkeit, die bis zum Zeitpunkt t ausgelaufen ist, bezeichnen wir mit $V_{out}(t)$. Es ist definiert als:

$$V_{out}(t) := \int_0^t Q(\tau) \, d\tau. \quad (2.4)$$

Der Volumenstrom $Q(t)$ ist dann die zeitliche Änderungsrate von $V_{out}(t)$:

$$Q(t) = \frac{dV_{out}(t)}{dt}. \quad (2.5)$$

Das Volumen zu einem Zeitpunkt $V(t)$ ist dann die Differenz aus Anfangsvolumen V_0 und der bis zu diesem Zeitpunkt ausgelaufenen Flüssigkeit $V_{out}(t)$:

$$V(t) = V_0 - V_{out}(t). \quad (2.6)$$

Mit diesen Beziehungen können wir eine lineare gewöhnliche Differentialgleichung 1. Ordnung bestimmen. Wir stellen 2.3 nach $v(t)$ um, ersetzen $Q(t)$ und anschließend $V_{out}(t)$ durch die Gleichungen 2.5 und 2.6.

$$\begin{aligned} v(t) &\stackrel{2.3}{=} \frac{Q(t)}{A} \\ &\stackrel{2.5}{=} \frac{1}{A} \cdot \frac{dV_{out}(t)}{dt} \\ &\stackrel{2.6}{=} \frac{1}{A} \cdot \frac{d(V_0 - V(t))}{dt}. \end{aligned} \quad (2.7)$$

Da V_0 konstant ist, ist $\frac{d(V_0)}{dt} = 0$. Wir ersetzen noch $V(t)$ durch Gleichung 2.1 und $h(t)$ durch $\frac{v^2(t)}{2g}$ (Gleichung 2.2 nach h umgestellt) und wenden die Kettenregel an.

$$\begin{aligned} v(t) &= -\frac{1}{A} \cdot \frac{d(V(t))}{dt} \\ &\stackrel{2.1}{=} -\frac{G}{A} \cdot \frac{d(h(t))}{dt} \\ &\stackrel{2.2}{=} -\frac{G}{2gA} \cdot \frac{d(v(t))^2}{dt} \\ &= -\frac{G}{gA} \cdot v(t) \cdot \frac{d(v(t))}{dt}. \end{aligned} \quad (2.8)$$

Eine Lösung finden wir, indem wir $v(t)$ ausklammern. Bei der nicht-trivialen Lösung

ist die Ableitung $\frac{d(v(t))}{dt}$ konstant. Eine Lösung für $v(t)$ ist dann eine lineare Funktion.

$$\begin{aligned}
 v(t) &= -\frac{G}{gA} \cdot v(t) \cdot \frac{d(v(t))}{dt} \\
 \Leftrightarrow v(t) + \frac{G}{gA} \cdot v(t) \cdot \frac{d(v(t))}{dt} &= 0 \\
 \Leftrightarrow v(t) \cdot \left(1 + \frac{G}{gA} \cdot \frac{d(v(t))}{dt}\right) &= 0 \\
 \Leftrightarrow v(t) = 0 \vee \frac{G}{gA} \cdot \frac{d(v(t))}{dt} &= -1 \\
 \Leftrightarrow v(t) = 0 \vee \frac{d(v(t))}{dt} &= -\frac{gA}{G} \\
 \Leftrightarrow v(t) = 0 \vee v(t) &= -\frac{gA}{G} \cdot t + K.
 \end{aligned} \tag{2.9}$$

Die Konstante K ist dabei die Strömungsgeschwindigkeit zum Zeitpunkt $t = 0$, die nur vom Wasserstand h_0 abhängt.

$$K = v_0 = \sqrt{2gh_0} = \sqrt{2g \frac{V_0}{G}}. \tag{2.10}$$

Daraus ergibt sich $v : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$,

$$v(t) = -\frac{gA}{G} \cdot t + \sqrt{2g \frac{V_0}{G}}. \tag{2.11}$$

Und mit Gleichung 2.2 $h : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$,

$$h(t) = \frac{1}{2} \cdot \left(\frac{A}{G}\right)^2 \cdot (t - t_b)^2. \tag{2.12}$$

Für den Fall, dass es n Ausflüsse aus dem Tank mit den Querschnitten A_1, \dots, A_n auf Bodenhöhe gibt, definieren wir A als die Summe der A_i und berechnen $v(t)$ und $h(t)$ wie bisher.

$$A := \sum_{i=1}^n A_i. \tag{2.13}$$

2.2 Zusätzliche konstante Ein- und Ausflüsse

Wir betrachten nun eine Modifikation des Tanks, in dem zusätzliche Ein- und Ausflüsse mit konstanter Strömungsgeschwindigkeit modelliert werden.

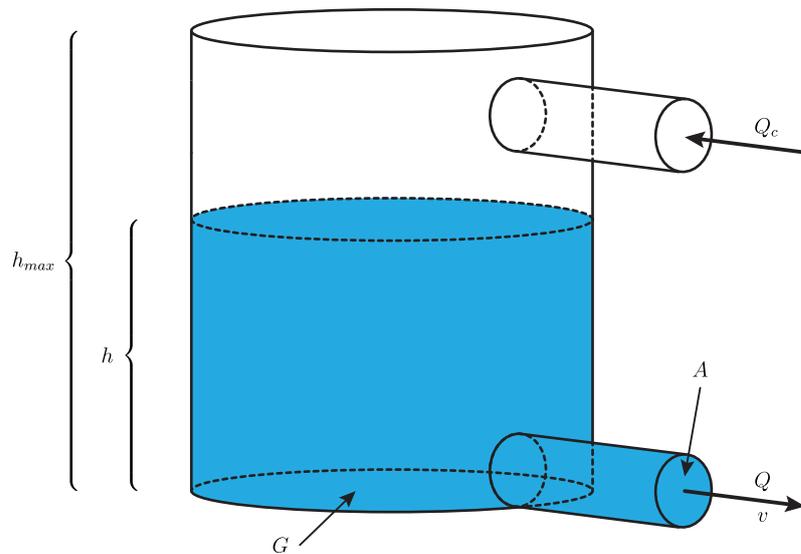


Abbildung 2.2: Tank mit konstantem Zufluss und druckabhängigem Ausfluss

In Abbildung 2.2 geht im oberen Bereich des Tanks eine zusätzliche Rohrleitung ab, über die der konstante Volumenstrom Q_c in den Tank fließt bzw. abgepumpt wird (negatives Vorzeichen). Gibt es mehrere konstante Ein- und Ausflüsse Q_1, \dots, Q_n definieren wir Q_c als ihre Summe.

$$Q_c := \sum_{i=1}^n Q_i. \quad (2.14)$$

Ist $Q_c = 0$ behandeln wir den Fall so wie im vorherigen Abschnitt ohne zusätzliche konstante Volumenströme.

Das Volumen $V(t)$ wird dann neben dem Ausfluss durch das untere Rohr $V_{out}(t)$ zusätzlich durch die konstanten Ströme beeinflusst. Die dadurch bedingte Volumenzunahme bzw. Volumenabnahme $V_c(t)$ verläuft linear mit der Zeit.

$$V_c(t) := \int_0^t Q_c \, d\tau = Q_c \cdot t. \quad (2.15)$$

Das Volumen zu einem Zeitpunkt $V(t)$ lässt sich dann ausdrücken, als:

$$V(t) = V_0 - V_{out}(t) + V_c(t). \quad (2.16)$$

Damit können wir wieder eine Differentialgleichung aufstellen. Wir beginnen wie zuvor indem wir 2.3 nach $v(t)$ umstellen und anschließend für $Q(t)$ die Gleichung 2.5 einsetzen.

Wir ersetzen dann $V_{out}(t)$ durch 2.16.

$$\begin{aligned}
 v(t) &\stackrel{2.3}{=} \frac{Q(t)}{A} \\
 &\stackrel{2.5}{=} \frac{1}{A} \cdot \frac{dV_{out}(t)}{dt} \\
 &\stackrel{2.16}{=} \frac{1}{A} \cdot \frac{d(V_0 - V(t) + V_c(t))}{dt}.
 \end{aligned} \tag{2.17}$$

Wieder ist $\frac{d(V_0)}{dt} = 0$. Mit Gleichung 2.15 erhalten wir $\frac{d(V_c)}{dt} = Q_c$ und können anschließend die Gleichungen 2.1 und 2.2 einsetzen.

$$\begin{aligned}
 v(t) &\stackrel{2.15}{=} -\frac{1}{A} \cdot \frac{d(V(t))}{dt} + \frac{Q_c}{A} \\
 &\stackrel{2.1}{=} -\frac{G}{A} \cdot \frac{d(h(t))}{dt} + \frac{Q_c}{A} \\
 &\stackrel{2.2}{=} -\frac{G}{2gA} \cdot \frac{d(v(t))^2}{dt} + \frac{Q_c}{A}.
 \end{aligned} \tag{2.18}$$

Die resultierende Gleichung ist eine nichtlineare gewöhnliche Differentialgleichung 1. Ordnung. Die Herleitung ihrer Lösung würde den Rahmen der Arbeit sprengen. Im Folgenden benutzen wir daher die Lösung eines Computer-Algebra-Systems [Map13]:

$$v(t) = \begin{cases} \frac{Q_c}{A} \left(W_{-1} \left(\frac{\left(A\sqrt{2g\frac{V_0}{G}} - Q_c \right) e^{\frac{A(\sqrt{2gV_0G} - 2Ag t)}{GQ_c}}}{Q_c} \right) + 1 \right) & \text{falls } Q_c < 0, \\ \frac{Q_c}{A} \left(W_0 \left(\frac{\left(A\sqrt{2g\frac{V_0}{G}} - Q_c \right) e^{\frac{A(\sqrt{2gV_0G} - 2Ag t)}{GQ_c}}}{Q_c} \right) + 1 \right) & \text{falls } Q_c > 0. \end{cases} \tag{2.19}$$

Wobei W_0 und W_{-1} die beiden Äste der Lambert-W-Funktion sind.

2.3 Ergebnisse

Wir fassen nun die die Ergebnisse aus Abschnitt 2.1 für $Q_c = 0$ und Abschnitt 2.2 für $Q_c \neq 0$ in den folgenden Sätzen zusammen:

Satz 1. Die Strömungsgeschwindigkeit für einen Tank wie in Abbildung 2.2 zum Zeitpunkt t wird durch die Funktion $v : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ bestimmt:

$$v(t) = \begin{cases} \frac{Q_c}{A} \left(W_{-1} \left(\frac{\left(A\sqrt{2g\frac{V_0}{G}} - Q_c \right) e^{\frac{A(\sqrt{2gV_0G} - 2Agt)}{GQ_c}}}{Q_c} \right) + 1 \right) & \text{falls } Q_c < 0, \\ \sqrt{2g\frac{V_0}{G}} - \frac{Agt}{G} & \text{falls } Q_c = 0, \\ \frac{Q_c}{A} \left(W_0 \left(\frac{\left(A\sqrt{2g\frac{V_0}{G}} - Q_c \right) e^{\frac{A(\sqrt{2gV_0G} - 2Agt)}{GQ_c}}}{Q_c} \right) + 1 \right) & \text{falls } Q_c > 0. \end{cases} \quad (2.20)$$

Wobei W_0 und W_{-1} die beiden Äste der Lambert-W-Funktion sind.

Mit den Gleichungen 2.1 und 2.2 können wir auch einen Satz für das Volumen formulieren:

Satz 2. Das Wasservolumen in einem Tank wie in Abbildung 2.2 zum Zeitpunkt t wird durch die Funktion $V : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ bestimmt:

$$V(t) = \begin{cases} \frac{Q_c^2 G}{2A^2 g} \left(W_{-1} \left(\frac{\left(A\sqrt{2g\frac{V_0}{G}} - Q_c \right) e^{\frac{A(\sqrt{2gV_0G} - 2Agt)}{GQ_c}}}{Q_c} \right) + 1 \right)^2 & \text{falls } Q_c < 0, \\ \frac{G \left(\sqrt{2g\frac{V_0}{G}} - \frac{Agt}{G} \right)^2}{2g} & \text{falls } Q_c = 0, \\ \frac{Q_c^2 G}{2A^2 g} \left(W_0 \left(\frac{\left(A\sqrt{2g\frac{V_0}{G}} - Q_c \right) e^{\frac{A(\sqrt{2gV_0G} - 2Agt)}{GQ_c}}}{Q_c} \right) + 1 \right)^2 & \text{falls } Q_c > 0. \end{cases} \quad (2.21)$$

Wobei W_0 und W_{-1} die beiden Äste der Lambert-W-Funktion sind.

In den folgenden Abbildungen, zur Veranschaulichung der komplizierten Formeln, ist jeweils auf der x-Achse die Zeit t und auf der y-Achse das Volumen V aufgetragen. Abbildung 2.3 zeigt einen beispielhaften Graphen von $V(t)$ für den Fall $Q_c < 0$ und Abbildung 2.4 einen beispielhaften Graphen für $Q_c = 0$.

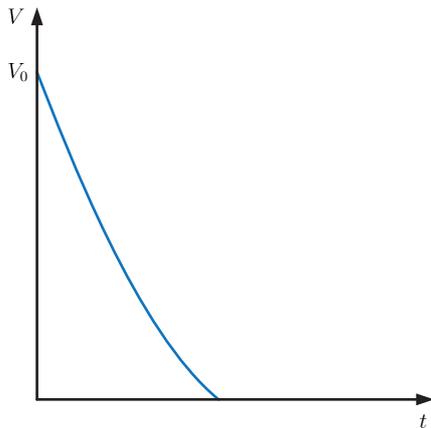


Abbildung 2.3: Zeitliche Entwicklung des Volumens (1. Fall)

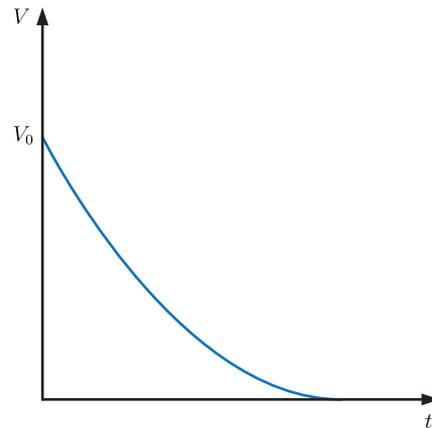


Abbildung 2.4: Zeitliche Entwicklung des Volumens (2. Fall)

Im Fall von $Q_c > 0$ nähert sich das Wasservolumen mit der Zeit der Asymptoten \hat{V} an, die dem Wasservolumen entspricht bei dem sich der konstante Einfluss Q_c und der druckabhängige Ausfluss Q ausgleichen. Abbildung 2.5 zeigt zwei beispielhafte Graphen von $V(t)$ für den Fall $Q_c > 0$.

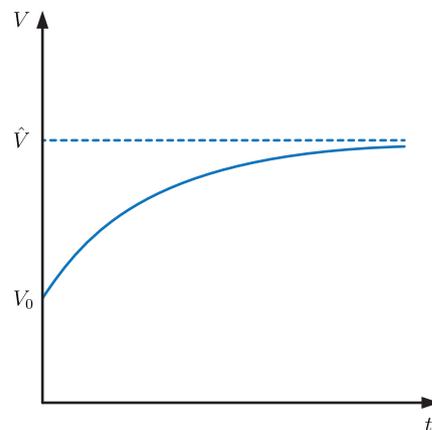
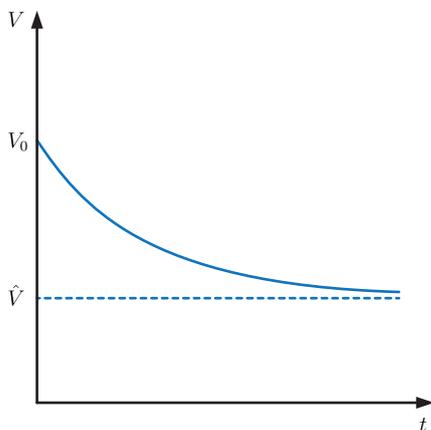


Abbildung 2.5: Zeitliche Entwicklung des Volumens (3. Fall)

Wir definieren die Asymptote \hat{V} durch:

$$\hat{V} := \lim_{t \rightarrow \infty} V(t). \quad (2.22)$$

Einen geschlossenen Ausdruck erhalten wir indem wir $Q = Q_c$ setzen, die Gleichungen

2.1, 2.2 und 2.3 benutzen und nach V auflösen.

Satz 3. Sei $Q_c > 0$ dann nähert sich $V(t)$ der Asymptoten \hat{V} an:

$$\hat{V} := \lim_{t \rightarrow \infty} V(t) = \frac{G}{2g} \left(\frac{Q_c}{A} \right)^2. \quad (2.23)$$

Um zu berechnen, wann das Wasservolumen einen bestimmten Wert erreicht, bilden wir die Umkehrfunktion $t(V)$.

Satz 4. Der Zeitpunkt, an dem ein Tank wie in Abbildung 2.2 das Wasservolumen V enthält, wird durch die Funktion $t : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ bestimmt:

$$t(V) = \begin{cases} \frac{1}{A} \sqrt{\frac{2G}{g}} \left(\sqrt{V_0} - \sqrt{V} \right) & \text{falls } Q_c = 0, \\ \frac{G}{2Ag} \left(\frac{Q_c}{A} \log \left(\frac{1 - \frac{A}{Q_c} \sqrt{2g \frac{V_0}{G}}}{1 - \frac{A}{Q_c} \sqrt{2g \frac{V}{G}}} \right) + \sqrt{2g \frac{V_0}{G}} - \sqrt{2g \frac{V}{G}} \right) & \text{falls } Q_c \neq 0. \end{cases} \quad (2.24)$$

Mehrere druckabhängige Ausflüsse können wie im vorherigen Abschnitt durch die Summierung ihrer Querschnitte wie in 2.13 behandelt werden.

3 Modellierung

In diesem Kapitel erweitern wir den Modellformalismus der HPnGs, wie in [GR13, S.4 ff.] beschrieben, zu einem Modellformalismus der Hybriden Petri-Netze (HPpTs), sodass das Auslaufen eines Tanks durch den Druck der Wassersäule, wie im vorherigen Kapitel beschrieben, abgebildet werden kann.

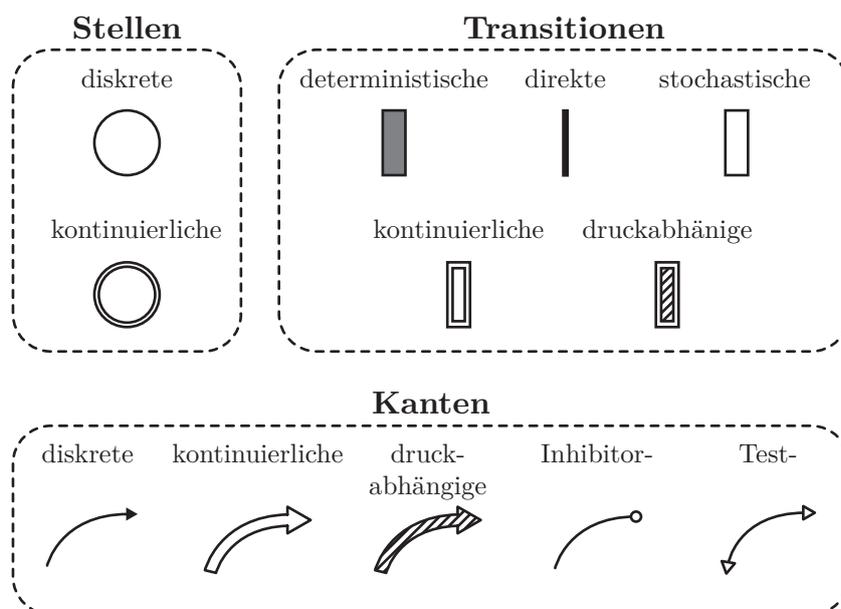


Abbildung 3.1: Elemente der erweiterten HPnGs

Dabei erweitern wir die HPnGs um die Elemente druckabhängige Transitionen und druckabhängige Kanten, sowie um die Eigenschaft der Grundfläche für kontinuierliche Stellen und die Eigenschaft der Querschnittfläche für druckabhängige Kanten. Abbildung 3.1 bildet alle Elemente der hier beschriebenen HPpTs ab. Dazu gehören diskrete und kontinuierliche Stellen, Transitionen und Kanten, die Transitionen und Stellen verbinden. Wir erlauben dabei nur druckabhängige Kanten von kontinuierlichen Stellen zu druckabhängigen Transitionen und nicht umgekehrt - modellieren also nur druck-

abhängige Ausflüsse (keine Zuflüsse).

Zunächst werden wir die bereits erwähnten Elemente der HPpTs und ihre Eigenschaften formal definieren. Danach werden wir Schritt für Schritt die Konzepte der bisherigen HPnGs um die Berücksichtigung der druckabhängigen Transitionen erweitern. Wir definieren auch für diese die Begriffe Erlaubnis und Freigabe bevor wir die Ergebnisse der physikalischen Betrachtung auf das Modell anwenden, um die Entwicklung der kontinuierlichen Variablen und die Anpassung der effektiven Volumenströme anzupassen.

Die Konzepte des Modells auf die in diesem Kapitel nicht eingegangen wird, z. B. die Auflösung von Schalt-Konflikten oder die Auswirkung des Schaltens einer deterministischen, direkten oder stochastischen Transition an sich, wurden ohne Änderung übernommen und werden hier nicht detailliert beschrieben. Sie können bei Bedarf in [GR13] nachvollzogen werden.

3.1 Formale Definition

Ein HPpT ist ein 6-Tupel $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathbf{m}_0, \mathbf{x}_0, \Phi)$. Die Elemente werden im folgenden einzeln vorgestellt.

Wie bisher ist $\mathcal{P} = \mathcal{P}^d \dot{\cup} \mathcal{P}^c = \{P_1^d, \dots, P_{n_d}^d\} \dot{\cup} \{P_1^c, \dots, P_{n_c}^c\}$ die Menge der Stellen, wobei zwischen diskreten Stellen \mathcal{P}^d und kontinuierlichen Stellen \mathcal{P}^c unterschieden wird.

Die Menge der Transitionen erweitern wir um die Teilmenge $\mathcal{T}^P = \{T_1^P, \dots, T_{n_P}^P\}$ der druckabhängigen Transitionen, sodass $\mathcal{T} = \mathcal{T}^I \dot{\cup} \mathcal{T}^D \dot{\cup} \mathcal{T}^G \dot{\cup} \mathcal{T}^F \dot{\cup} \mathcal{T}^P$. Wir definieren auch die Menge $\mathcal{T}^S = \mathcal{T}^I \dot{\cup} \mathcal{T}^D \dot{\cup} \mathcal{T}^G$ der Standard-Transitionen, die den diskreten Teil des HPpT beeinflussen.

Da wir in dieser Arbeit die Simulation und nicht die Analyse der modellierten HPnGs betrachten, können wir ohne Weiteres auch mehrere stochastische Transitionen erlauben. In der Simulation werden stochastische Transitionen dann wie einmalig schaltende deterministische Transitionen behandelt. Für sie wird bei jedem Durchlauf eine zufällige Schaltzeit berechnet.

Wir erweitern auch die Menge der Kanten um die Teilmenge $\mathcal{A}^p \subseteq (\mathcal{P}^c \times \mathcal{T}^P)$ der druckabhängigen Kanten, die von einer kontinuierlichen Stelle zu einer druckabhängigen Transition führen, sodass $\mathcal{A} = \mathcal{A}^d \dot{\cup} \mathcal{A}^f \dot{\cup} \mathcal{A}^p \dot{\cup} \mathcal{A}^h \dot{\cup} \mathcal{A}^t$. Die bekannten Notationen für Kanten ergänzen wir um $\langle P_i^c, T_j^P \rangle_p \in \mathcal{A}^p$ für alle druckabhängigen Kanten. Im

Gegensatz zu kontinuierlichen Transitionen und Kanten, sind druckabhängige Kanten, die von einer druckabhängigen Transition ausgehen, nicht zugelassen.

Der Vektor $\mathbf{m}_0 \in (\mathbb{N})^{n_d}$ gibt die Anfangsbelegung der diskreten Stellen an, der Vektor $\mathbf{x}_0 \in (\mathbb{R}^+)^{n_c}$ die der kontinuierlichen Stellen.

$\Phi = (\phi_a^{\mathcal{P}}, \phi_b^{\mathcal{P}}, \phi_w^{\mathcal{T}}, \phi_p^{\mathcal{T}}, \phi_d^{\mathcal{T}}, \phi_f^{\mathcal{T}}, \phi_g^{\mathcal{T}}, \phi_w^{\mathcal{A}}, \phi_s^{\mathcal{A}}, \phi_p^{\mathcal{A}}, \phi_a^{\mathcal{A}})$ ist ein 11-Tupel von Funktionen, die den Stellen, Transitionen und Kanten Eigenschaften zuordnen. Als Erweiterung zum bekannten Modell definieren wir die Funktionen $\phi_a^{\mathcal{P}} : \mathcal{P}^c \rightarrow \mathbb{R}^+$, die jeder kontinuierlichen Stelle eine Grundfläche zuordnet, und $\phi_a^{\mathcal{A}} : \mathcal{A}^p \rightarrow \mathbb{R}^+$, die jeder druckabhängigen Kante eine Querschnittfläche zuordnet.

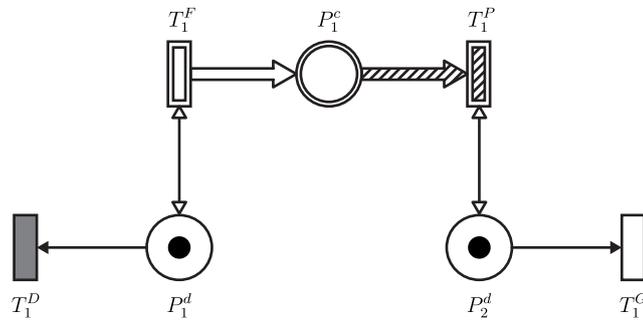


Abbildung 3.2: Beispiel für ein HPpT mit druckabhängiger Transition

Abbildung 3.2 zeigt in Anlehnung an das Reservoir-Beispiel aus [GR13, S. 23] ein Beispiel für ein einfaches HPpT mit einer druckabhängigen Transition. Im Mittelpunkt steht die kontinuierliche Stelle P_1^c , die einen Tank wie in Abschnitt 2.2 darstellt. Die Transition T_1^F entspricht einem kontinuierlichen Einfluss, die Transition T_1^P einem druckabhängigen Ausfluss. Darüber hinaus sind noch zwei diskrete Stellen P_1^d und P_2^d , sowie die beiden Transitionen T_1^D und T_1^G abgebildet. Auf ihre Funktion wird im Abschnitt 3.3 genauer eingegangen. Während die Anfangsbelegungen der diskreten Stellen $\mathbf{m}_0 = (1, 1)$ direkt in der Abbildung ablesbar sind, setzen wir explizit den Start-Füllstand der kontinuierlichen Stelle $\mathbf{x}_0 = (0)$. Zur vollständigen Beschreibung des HPpT gehört noch die Angabe der Funktionen Φ . Exemplarisch definieren wir die Grundfläche des Tanks mit $\phi_a^{\mathcal{P}}(P_1^c) = 50$, die Schaltzeit von $\phi_d^{\mathcal{T}}(T_1^D) = 120$ und die Gewichtung aller Kanten $A \in \mathcal{A}$ mit $\phi_w^{\mathcal{A}}(A) = 1$. Die übrigen Eigenschaften lassen wir für dieses Beispiel offen.

Zu Beachten ist, dass wir beim Übertragen eines Falls in ein Modell alle physikalischen Größen durch Angaben in ihren Basiseinheiten, also Meter bzw. Sekunden, beschreiben.

Der durch das Modell beschriebene Tank hätte also die Grundfläche von $50 m^2$ und die Transition T_1^D eine Schaltzeit von $120 s = 2 min$.

3.2 Zustand des Modells

Der Zustand des Modells ist gegeben durch das 4-Tupel $\Gamma = (\mathbf{m}, \mathbf{x}, \mathbf{c}, \mathbf{g})$ mit der Belegung der diskreten Stellen $\mathbf{m} \in (\mathbb{N})^{\mathbf{n}_d}$, den Füllständen der kontinuierlichen Stellen $\mathbf{x} \in (\mathbb{R}^+)^{\mathbf{n}_c}$, den Freigabezeiten für deterministische $\mathbf{c} \in (\mathbb{R}^+)^{\mathbf{n}_D}$ und den Freigabezeiten für stochastische Transitionen $\mathbf{g} \in (\mathbb{R}^+ \cup \{-1\})^{\mathbf{n}_G}$. Wobei unter Freigabezeit die Zeit verstanden wird, die vergangen ist während die Transition die Erlaubnis zum Schalten hatte.

Der Zustand verändert sich mit der Zeit τ als $\Gamma(\tau)$. Der Startzustand ist $\Gamma(0) = (\mathbf{m}_0, \mathbf{x}_0, \mathbf{0}, \mathbf{0})$. In unserem Beispiel ist das $\Gamma(0) = ((1, 1), (0), (0), (0))$.

3.3 Erlaubnis und Freigabe

Das Verhalten der Transitionen ist je nach Art der Transition unterschiedlich. Um das Verhalten zu beschreiben verwenden wir die beiden Begriffe Erlaubnis und Freigabe.

Wir definieren zunächst wie im bisherigen Modell für alle Transitionen $T_i \in \mathcal{T}$ die Mengen $\mathcal{I}^h(T_i)$ und $\mathcal{I}^t(T_i)$, für alle Standard-Transitionen $T_i^S \in \mathcal{T}^S$ die Mengen $\mathcal{I}^d(T_i^S)$ und $\mathcal{O}^d(T_i^S)$, für alle kontinuierlichen Transitionen $T_i^F \in \mathcal{T}^F$ die Mengen $\mathcal{I}^f(T_i^F)$ und $\mathcal{O}^f(T_i^F)$ und für alle kontinuierlichen Stellen $P_i^c \in \mathcal{P}^c$ die Mengen $\mathcal{I}^F(P_i^c)$ und $\mathcal{O}^F(P_i^c)$. Zusätzlich definieren wir für alle druckabhängigen Transitionen $T_i^P \in \mathcal{T}^P$ die Mengen $\mathcal{I}^P(T_i^P) = \{P_j^c | \langle P_j^c, T_i^P \rangle_p \in \mathcal{A}^P\}$ und für alle kontinuierlichen Stellen $P_i^c \in \mathcal{P}^c$ die Mengen $\mathcal{O}^P(P_i^c) = \{T_j^P | \langle P_i^c, T_j^P \rangle_p \in \mathcal{A}^P\}$.

Eine Transition $T_i \in \mathcal{T}$ hat Erlaubnis, wenn alle 5 folgenden Bedingungen erfüllt sind:

$$\begin{aligned}
1. & T_i \in \mathcal{T}^S \Rightarrow (\forall P_j^d \in \mathcal{I}^d(T_i) : m_j \geq \phi_w^A(\langle P_j^d, T_i \rangle_d)), \\
2. & \forall P_j^d \in (\mathcal{I}^h(T_i) \cap P^d) : m_j < \phi_w^A(\langle P_j^d, T_i \rangle_h), \\
3. & \forall P_k^c \in (\mathcal{I}^h(T_i) \cap P^c) : x_k < \phi_w^A(\langle P_k^c, T_i \rangle_h), \\
4. & \forall P_j^d \in (\mathcal{I}^t(T_i) \cap P^d) : m_j \geq \phi_w^A(\langle P_j^d, T_i \rangle_t), \\
5. & \forall P_k^c \in (\mathcal{I}^t(T_i) \cap P^c) : x_k \geq \phi_w^A(\langle P_k^c, T_i \rangle_t).
\end{aligned} \tag{3.1}$$

Direkte Transitionen $T_i^I \in \mathcal{T}^I$ mit Erlaubnis haben immer auch Freigabe und schalten sofort.

Deterministische Transitionen $T_i^D \in \mathcal{T}^D$ mit Erlaubnis haben, wenn die zu wartende Schaltzeit $\phi_d^T(T_i^D)$ abgelaufen ist, d.h. $c_j = \phi_d^T(T_i^D)$ ist, Freigabe und schalten.

Stochastische Transitionen mit Erlaubnis haben, wenn sie nicht schon einmal geschaltet haben, Freigabe und können schalten.

Kontinuierliche Transitionen $T_i^F \in \mathcal{T}$ mit Erlaubnis beeinflussen durchgehend linear die Füllstände der kontinuierlichen Stellen. Ihnen wird ein effektiver Volumenstrom $\theta(T_i^F)$ (in [GR13]: actual flow-rate) zugeordnet. Hat eine Transition T_i^F keine Erlaubnis so ist $\theta(T_i^F) = 0$.

Druckabhängige Transitionen $T_i^P \in \mathcal{P}$ mit Erlaubnis beeinflussen durchgehend die Füllstände der kontinuierlichen Stellen. Die Stärke des Einflusses ist abhängig vom Füllstand.

In unserem Beispiel aus 3.2 haben die Transitionen T_1^F und T_1^P solange Erlaubnis, wie eine Marke auf den diskreten Stellen P_1^d bzw. P_2^d liegt. Die Transitionen T_1^D und T_1^G haben Erlaubnis und Freigabe. Wenn sie schalten, werden die Marken auf den diskreten Stellen P_1^d bzw. P_2^d entfernt. Dadurch verliert nicht nur die Transition die schaltet ihre Erlaubnis, sondern je nach Transition auch eine der Transitionen T_1^F oder T_1^P . T_1^D und T_1^G wirken quasi wie Ausschalter.

3.4 Entwicklung der kontinuierlichen Variablen

Die Füllstände der kontinuierlichen Stellen werden sowohl durch die kontinuierlichen Transitionen $T_i^F \in \mathcal{T}$ mit stückweise konstantem Volumenstrom $\theta(T_i^F)$, als auch von den druckabhängigen Transitionen beeinflusst $T_i^P \in \mathcal{P}$.

Als Maß für den Einfluss der kontinuierlichen Transitionen $T_j^F \in \mathcal{T}^F$ definieren wir für alle kontinuierlichen Stellen $P_i^c \in \mathcal{P}^c$:

$$q_i := \sum_{T_j^F \in \mathcal{I}^F(P_i^c)} \theta(T_j^F) \cdot \phi_w^A(\langle T_j^F, P_i^c \rangle_f) - \sum_{T_j^F \in \mathcal{O}^F(P_i^c)} \theta(T_j^F) \cdot \phi_w^A(\langle P_i^c, T_j^F \rangle_f). \quad (3.2)$$

Im Vergleich mit dem Tank aus 2.2 entspricht q_i dann der Summe Q_c der konstanten Volumenströme. q_i ist abhängig vom Zustand Γ .

Als Maß für den Einfluss der druckabhängigen Transitionen $T_j^P \in \mathcal{T}^P$ definieren wir für alle kontinuierlichen Stellen $P_i^c \in \mathcal{P}^c$:

$$a_i := \sum_{T_j^P \in \mathcal{O}^P(P_i^c) \wedge T_j^P \text{ hat Erlaubnis in } \Gamma} \phi_a^A(\langle P_i^c, T_j^P \rangle_p). \quad (3.3)$$

Im Vergleich mit dem Tank aus 2.2 entspricht a_i dann der Summe A der Querschnitte. a_i ist abhängig vom Zustand Γ .

Im Folgenden betrachten wir, wie die Füllstände \mathbf{x}' eines Folgezustands $\Gamma' = (\mathbf{m}', \mathbf{x}', \mathbf{c}', \mathbf{g}')$ konstruiert werden, wenn zwischen Γ und Γ' die Zeit $\Delta\tau$ vergeht. Wir unterscheiden für jede kontinuierliche Stelle $P_i^c \in \mathcal{P}^c$ vier Fälle und geben jeweils eine geschlossene Formel für x'_i an. Die Abbildungen 3.3, 3.4, 3.5 und 3.6 zeigen dabei für jeden Fall einen exemplarischen Graphen mit dem Verlauf des Füllstands über die Zeit τ . Eingezeichnet sind neben dem Füllstand des aktuellen Zustands x_i zudem die Zeit $\Delta\tau$ bis zum folgenden Zustand und der Füllstand x'_i des folgenden Zustands.

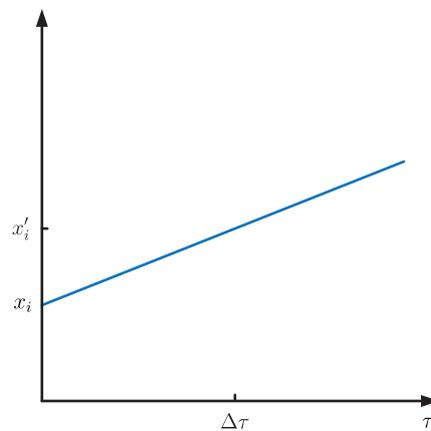


Abbildung 3.3: Entwicklung der Füllstände (1. Fall)

1. Fall: $a_i = 0$ (kein druckabhängiger Ausfluss) Wenn eine kontinuierliche Stelle keine Kanten zu druckabhängigen Transitionen mit Erlaubnis hat, verändert sich der Füllstand linear (Abb. 3.3). x'_i wird dann wie folgt berechnet:

$$x'_i = x_i + \Delta\tau \cdot q_i. \quad (3.4)$$

Die Fälle 2-4 treten ein, wenn eine oder mehrere Kanten zu druckabhängigen Transitionen mit Erlaubnis führen. x'_i wird dann mit den Formeln aus Satz 2 berechnet, wobei wir die Größen des Tanks durch die entsprechenden Größen des Modells ersetzen:

$$\begin{aligned} V_0 &\mapsto x_i, \\ \hat{V} &\mapsto \hat{x}_i, \\ G &\mapsto \phi_a^{\mathcal{P}}(P_i^c), \\ A &\mapsto a_i, \\ Q_c &\mapsto q_i, \\ t &\mapsto \tau. \end{aligned} \quad (3.5)$$

Das Startvolumen V_0 wird durch den Füllstand x_i des aktuellen Zustands, die Grundfläche des Tanks G durch $\phi_a^{\mathcal{P}}(P_i^c)$, die Querschnittfläche A des Ausflusses durch a_i und Q_c durch q_i ersetzt. Für die Asymptote \hat{V} schreiben wir \hat{x}_i und den Parameter t ersetzen wir durch τ .

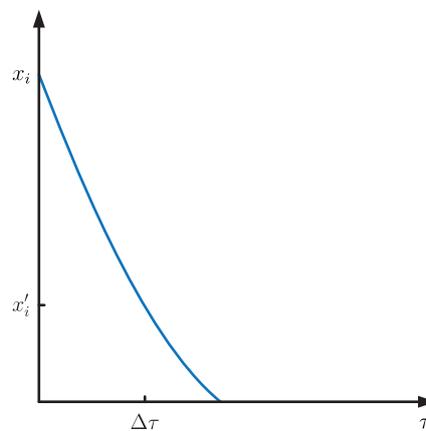


Abbildung 3.4: Entwicklung der Füllstände (2. Fall)

2. Fall: $a_i \neq 0 \wedge q_i < 0$ (**druckabhängiger Ausfluss und konstanter Ausfluss**) Der Füllstand x'_i nimmt schnell ab (Abb. 3.4) und wird wie folgt berechnet:

$$x'_i = \frac{q_i^2}{2a_i^2 g} \left(W_{-1} \left(\frac{\left(a_i \sqrt{\frac{2gx_i}{\phi_a^{\mathcal{P}}(P_i^c)}} - q_i \right) e^{\frac{a_i \left(\sqrt{2gx_i \cdot \phi_a^{\mathcal{P}}(P_i^c)} - 2a_i g \Delta\tau \right)}{\phi_a^{\mathcal{P}}(P_i^c) \cdot q_i}} - 1}{q_i}} \right) + 1 \right)^2 \cdot \phi_a^{\mathcal{P}}(P_i^c). \quad (3.6)$$

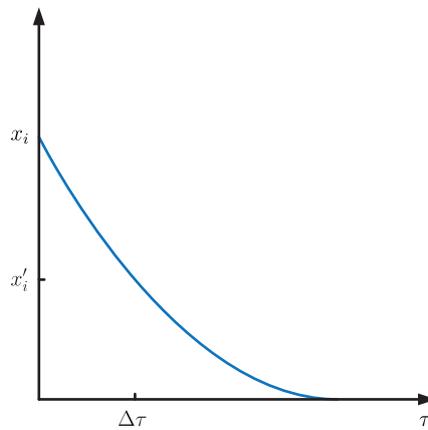


Abbildung 3.5: Entwicklung der Füllstände (3. Fall)

3. Fall: $a_i \neq 0 \wedge q_i = 0$ (**druckabhängiger Ausfluss und kein konstanter Aus- oder Einfluss**) Der Füllstand x'_i nimmt ab (Abb. 3.5) und wird wie folgt berechnet:

$$x'_i = \frac{1}{2g} \cdot \left(\sqrt{\frac{2gx_i}{\phi_a^{\mathcal{P}}(P_i^c)}} - \frac{a_i g \Delta\tau}{\phi_a^{\mathcal{P}}(P_i^c)} \right)^2 \cdot \phi_a^{\mathcal{P}}(P_i^c). \quad (3.7)$$

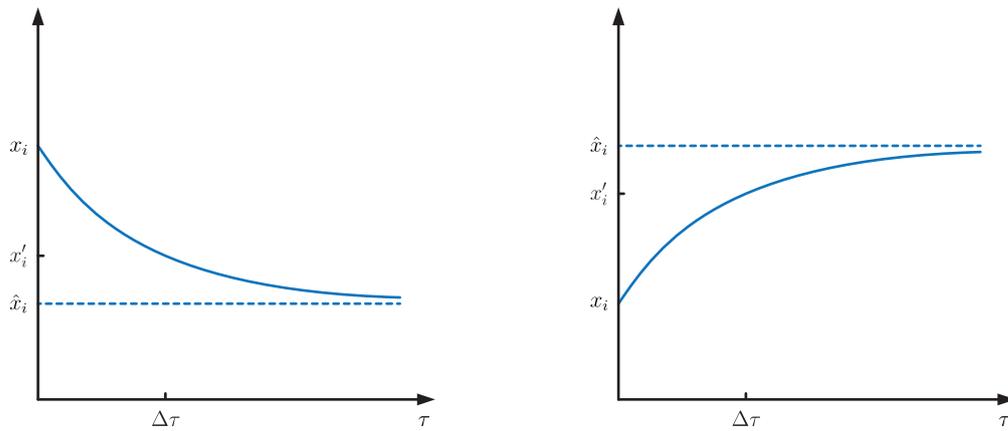


Abbildung 3.6: Entwicklung der Füllstände (4. Fall)

4. Fall: $a_i \neq 0 \wedge q_i > 0$ (**druckabhängiger Ausfluss und konstanter Einfluss**) Der Füllstand x_i' nähert sich mit der Zeit τ der Asymptote \hat{x}_i an (Abb. 3.6):

$$\hat{x}_i = \frac{\phi_a^P(P_i^c)}{2g} \left(\frac{q_i}{a_i} \right)^2. \quad (3.8)$$

Er wird wie folgt berechnet:

$$x_i' = \frac{q_i^2}{2a_i^2g} \left(W_0 \left(\frac{\left(a_i \sqrt{\frac{2gx_i}{\phi_a^P(P_i^c)}} - q_i \right) e^{\frac{a_i \left(\sqrt{2gx_i \cdot \phi_a^P(P_i^c)} - 2a_i g \Delta \tau \right) - 1}}{\phi_a^P(P_i^c) \cdot q_i}}}{q_i} \right) + 1 \right)^2 \cdot \phi_a^P(P_i^c). \quad (3.9)$$

In unserem Beispiel aus 3.2 würde der Füllstand sich zunächst wie in Fall 4 entwickeln. Schaltet die Transition T_1^D fällt der konstante Zufluss weg und der Füllstand entwickelt sich sich von da an weiter wie in Fall 3. Schaltet die Transition T_1^G , fällt der druckabhängige Ausfluss weg und der Füllstand entwickelt sich von da an linear wie in Fall 1.

3.5 Anpassung der effektiven Volumenströme

Kontinuierlichen Transitionen ist ein effektiver Volumenstrom $\theta(T_i^F)$ zugeordnet. Hat T_i^F Erlaubnis ist dieser im Normalfall gleich dem nominalen Volumenstrom $\phi_f^T(T_i^F)$.

In gewissen Situationen kann es aber vorkommen, dass der effektive Volumenstrom reduziert werden muss, damit die Begrenzungen der kontinuierlichen Stellen nicht überschritten werden. Dieses Verfahren der Regelung bezeichnen wir als Anpassung der effektiven Volumenströme (in [GR13]: rate adaption). Wir stellen hier die Unterschiede zur Anpassung, wie sie in [GR13, S. 10 ff.] beschrieben wird, heraus und verzichten auf eine vollständige Beschreibung.

Der Pseudo-Code in Algorithmus 1 zeigt die grundlegende Struktur des Algorithmus. Zunächst werden die effektiven Volumenströme $\theta(T_i^F)$ der kontinuierlichen Transitionen mit Erlaubnis durch ihren nominalen Wert $\phi_f^T(T_i^F)$ initialisiert (Zeile 1) und die q_i für alle Stellen $P_i^c \in \mathcal{P}^c$ berechnet.

Algorithmus 1 Anpassung der effektiven Volumenströme

```

1:  $\forall T_i^F \in \mathcal{T}^F$  mit Erlaubnis:  $\theta(T_i^F) = \phi_f^T(T_i^F)$ 
2:  $\forall P_i^c \in \mathcal{P}^c$ : berechne  $q_i$ 
3: setze Abbruchbedingung auf false
4: repeat
5:   setze Abbruchbedingung auf true
6:   for all  $P_i^c \in \mathcal{P}^c$  do
7:     if Fall 1 then
8:       if  $\hat{x}_i > \phi_b^P(P_i^c)$  then
9:         setze Abbruchbedingung auf false
10:         $\forall T_j^F \in \mathcal{I}^F(P_i^c)$ : passe  $\theta(T_j^F)$  an.
11:         $\forall P_j^c \in \mathcal{P}^c$ : berechne  $q_j$ 
12:       end if
13:     else if Fall 2 then
14:       setze Abbruchbedingung auf false
15:        $\forall T_j^F \in \mathcal{I}^F(P_i^c)$ : passe  $\theta(T_j^F)$  an.
16:        $\forall P_j^c \in \mathcal{P}^c$ : berechne  $q_j$ 
17:     else if Fall 3 then
18:       setze Abbruchbedingung auf false
19:        $\forall T_j^F \in \mathcal{O}^F(P_i^c)$ : passe  $\theta(T_j^F)$  an.
20:        $\forall P_j^c \in \mathcal{P}^c$ : berechne  $q_j$ 
21:     end if
22:   end for
23: until Abbruchbedingung ist true

```

Die darauf folgende Schleife (Zeile 4-23) wird solange durchlaufen, bis bei einem Durchlauf für keine der kontinuierlichen Stellen $P_i^c \in \mathcal{P}^c$, die in der inneren Schleife (Zeile 6-22) durchlaufen werden, eine Anpassungen der effektiven Volumenströme vorgenommen werden muss. Das ist im Grunde dann der Fall wenn keiner der 3 Fälle eintritt. Eine kleine Abweichung gibt es bei Fall 1. Hier ist eine Anpassung erst nötig wenn auch

die Bedingung $\hat{x}_i > \phi_b^{\mathcal{P}}(P_i^c)$ (Zeile 8) erfüllt ist. Ist eine Anpassung notwendig, müssen auch die q_i für alle Stellen $P_i^c \in \mathcal{P}^c$ neu berechnet werden. Die genaue Anpassung der effektiven Volumenströme wird für jeden der drei Fälle im Folgenden beschrieben.

1. Fall: $x_i = \phi_b^{\mathcal{P}}(P_i^c) \wedge q_i > 0 \wedge a_i > 0$ (**obere Begrenzung erreicht, druckabhängiger Ausfluss und konstanter Einfluss**) Wenn der Füllstand x_i einer kontinuierlichen Stelle P_i^c die obere Begrenzung $\phi_b^{\mathcal{P}}(P_i^c)$ erreicht, das Maß für den Einfluss der druckabhängigen Transitionen a_i positiv und das Maß für den Einfluss der kontinuierlichen Transitionen q_i positiv ist, muss das Verfahren aus [GR13, S. 10 ff.] angepasst werden.

Zunächst wird geprüft, ob die Asymptote \hat{x}_i größer ist als die obere Begrenzung $\phi_b^{\mathcal{P}}(P_i^c)$. Ist dies der Fall, so muss der konstante Einfluss so verringert werden, dass die Asymptote \hat{x}_i' gleich der Begrenzung ist. Dadurch bleibt der Füllstand anschließend konstant auf Höhe der oberen Begrenzung. Wir müssen dann einen kontinuierlichen Effekt q_i' finden, sodass genau die Asymptote \hat{x}_i' beibehalten wird. Dafür setzen wir $\hat{x}_i' = \phi_b^{\mathcal{P}}(P_i^c)$ in 3.8 ein und lösen die Gleichung nach q_i' auf.

$$q_i' = a_i \sqrt{2g \frac{\phi_b^{\mathcal{P}}(P_i^c)}{\phi_a^{\mathcal{P}}(P_i^c)}}. \quad (3.10)$$

Dieser Volumenstrom q_i' soll nach der Anpassung der effektiven Volumenströme das Ereignis aus Gleichung 3.2 sein. Um dies zu erreichen müssen die effektiven Volumenströme der Transitionen $T_j^F \in \mathcal{I}^F(P_i^c)$ verringert werden. Wir definieren den verfügbaren Volumenstrom (in [GR13, S. 11]: available flow-rate):

$$\alpha(P_i^c) := \sum_{T_j^F \in \mathcal{O}^F(P_i^c)} \theta(T_j^F) \cdot \phi_w^A(\langle P_i^c, T_j^F \rangle_f) + q_i'. \quad (3.11)$$

Sodass nach der Anpassung der effektiven Volumenströme, die Summe $\sum_{T_j^F \in \mathcal{I}^F(P_i^c)} \theta(T_j^F) \cdot \phi_w^A(\langle T_j^F, P_i^c \rangle_f)$ aus Gleichung 3.2 gleich $\alpha(P_i^c)$ ist.

Die genaue Aufteilung des verfügbaren Volumenstroms auf die Transitionen ist dann identisch wie in [GR13, S. 10 ff.].

2. Fall: $x_i = \phi_b^{\mathcal{P}}(P_i^c) \wedge q_i > 0 \wedge a_i = 0$ (**obere Begrenzung erreicht, kein druckabhängiger Ausfluss und konstanter Einfluss**) Wenn der Füllstand x_i einer kontinuierlichen Stelle P_i^c die obere Begrenzung $\phi_b^{\mathcal{P}}(P_i^c)$ erreicht, der Effekt der druck-

abhängigen Transitionen a_i gleich null und der Effekt der kontinuierlichen Transitionen q_i positiv ist, wird die Anpassung der effektiven Volumenströme genau wie in [GR13, S. 10 ff.] durchgeführt.

3. Fall: $x_i = 0 \wedge q_i < 0$ (untere Begrenzung erreicht und konstanter Ausfluss)

Wenn der Füllstand x_i einer kontinuierlichen Stelle P_i^c die untere Begrenzung 0 erreicht und der Effekt der kontinuierlichen Transitionen q_i negativ ist, bleibt der Füllstand unabhängig von a_i bei 0 und nähert sich keiner Asymptoten, wie in Fall 3.5. Die Anpassung der effektiven Volumenströme kann daher genau wie in [GR13, S. 10 ff.] durchgeführt werden.

4 Simulation

In diesem Kapitel wird zunächst der Begriff der Ereignissimulation erklärt, bevor wir konkret die Ereignissimulation für die im vorherigen Kapitel beschriebenen HPnGs beschreiben.

4.1 Ereignissimulation

Die Ereignissimulation ist eine Form der diskreten Simulation. Anstatt kontinuierlich den Zustand des simulierten Systems zu ändern, werden aus einem Zustand Ereignisse und die verbleibenden Zeiten bis sie eintreffen berechnet. Diese Ereignisse werden dann nach ihren verbleibenden Zeiten in eine Warteschlange eingereiht und nacheinander abgearbeitet bis irgendwann eine Abbruchbedingung erreicht wird.

Algorithmus 2 Ereignissimulation

```
1: setze Abbruchbedingung auf false
2: setze Simulationszeit auf 0
3: initialisiere das System
4: erzeuge Startereignis und setze es in die Warteschlange
5: while Abbruchbedingung nicht erfüllt do
6:   setze Simulationszeit auf die Zeit des nächsten Ereignisses
7:   führe Ereignis aus und entferne es aus der Warteschlange
8:   aktualisiere Statistiken
9: end while
10: werte Statistiken aus
```

Algorithmus 2 beschreibt den allgemeinen Fall der Ereignissimulation als Pseudo-Code. Nach der Initalisierung der Variablen (Zeile 1-4) wird die Schleife (Zeile 5-9) solange durchlaufen bis durch ein Ereignis die Abbruchbedingung erreicht wird. Bei der Ausführung eines Ereignisses wird die Simulationszeit aktualisiert (Zeile 6) und bei der Ausführung des Ereignisses (Zeile 7) ggfs. neue Ereignisse in die Warteschlange eingereiht. Je nach Anwendung werden außerdem Daten für Statistiken gesammelt (Zeile

8), die nach Durchlauf der Simulation ausgewertet werden können (Zeile 10).

4.2 Anpassung an das Modell

Um das Konzept der Ereignissimulation auf das hier vorgestellte Modell anzuwenden, müssen wir den allgemeinen Algorithmus anpassen.

Zum einen führen wir keine Warteschlange von Ereignissen, sondern berechnen für einen Zustand des Modells seine Ereignismenge (Kap. 4.3). Aus dieser wird ein Ereignis ausgewählt. Nach seiner Ausführung wird ein neuer Zustand erreicht, für den eine neue Ereignismenge berechnet wird. Nicht ausgewählte Ereignisse werden nicht weiter berücksichtigt.

Zum anderen müssen wir den Algorithmus anpassen, da wir neben den diskreten Stellen auch kontinuierliche Stellen, also eine nicht-diskrete Komponente in der diskreten Ereignissimulation erlauben. Wir sind nicht nur an den Werten der Füllstände zu den konkreten Zuständen, sondern auch zwischen Zuständen interessiert. Zwischen zwei Zuständen können die Füllstände aber mit den Formeln aus Abschnitt 3.4 berechnet werden. Vor der Auswertung eines Ereignisses erheben wir daher die Daten für die Statistik zwischen dem aktuellen Zustand und dem nächsten Zustand.

Unsere Abbruchbedingung ist erreicht, wenn für den aktuellen Zustand kein Folgezustand existiert (leere Ereignismenge) oder wenn die Simulationszeit eine vorgegebene maximale Ausführungszeit erreicht hat.

4.3 Konstruktion der Ereignismenge

Die Entwicklung des Modells erfolgt ereignisorientiert, d.h. in jedem Zustand Γ wird eine Menge der absehbaren Ereignisse $\mathcal{E}^{\mathcal{P}}(\Gamma)$ berechnet, ein Ereignis ausgewählt, ausgewertet und ein Folgezustand Γ' bestimmt.

$\mathcal{E}(\Gamma) = \mathcal{E}^{\mathcal{T}}(\Gamma) \cup \mathcal{E}^{\mathcal{P}}(\Gamma) \cup \mathcal{E}^{\mathcal{A}}(\Gamma)$ ist die Menge der absehbaren Ereignisse und lässt sich in drei disjunkte Teilmengen unterteilen, nach den Elementen die Ereignisse auslösen können.

Die Teilmenge $\mathcal{E}^{\mathcal{T}}(\Gamma)$ umfasst die Ereignisse die eintreten, wenn eine direkte, deterministische oder stochastische Transition schaltet. Ihre Konstruktion bleibt unverändert

wie in [GR13] beschrieben.

4.3.1 Ereignisse der kontinuierlichen Stellen

$\mathcal{E}^{\mathcal{P}}(\Gamma)$ beinhaltet die Ereignisse die ausgelöst werden, wenn einer der Füllstände einer kontinuierlichen Stelle die obere Begrenzung $\phi_b^{\mathcal{P}}(P_i^c)$ oder die untere Begrenzung 0 erreicht.

Wir berechnen für jede Stelle $P_i^c \in \mathcal{P}^c$ die Zeit $\tau^{\mathcal{P}}(P_i^c, \Gamma)$ bis der Füllstand eine der Begrenzungen erreicht. Dabei treffen wir Fallunterscheidungen abhängig von den Größen a_i und q_i , x_i und $\phi_b^{\mathcal{P}}(P_i^c)$.

Wir unterscheiden nach dem Vorzeichen von q_i und ob a_i null ist. Im Fall von $q > 0$ und $a_i \neq 0$ vergleichen wir zusätzlich noch die Asymptote \hat{x}_i mit der oberen Begrenzung $\phi_b^{\mathcal{P}}(P_i^c)$. Abbildung 4.1 zeigt für jeden Fall einen Graphen mit der Entwicklung des Füllstands über die Zeit τ und die Begrenzung $\phi_w^{\mathcal{A}}(\langle P_i^c, T_j \rangle)$.

In fünf Fällen wird eine der Begrenzungen erreicht. Wir bezeichnen sie mit:

$$\begin{aligned}
 C_1^{\mathcal{P}} &: \Leftrightarrow a_i = 0 \wedge q_i < 0, \\
 C_2^{\mathcal{P}} &: \Leftrightarrow a_i = 0 \wedge q_i > 0, \\
 C_3^{\mathcal{P}} &: \Leftrightarrow a_i \neq 0 \wedge q_i < 0, \\
 C_4^{\mathcal{P}} &: \Leftrightarrow a_i \neq 0 \wedge q_i = 0, \\
 C_5^{\mathcal{P}} &: \Leftrightarrow a_i \neq 0 \wedge q_i > 0 \wedge \hat{x}_i > \phi_b^{\mathcal{P}}(P_i^c).
 \end{aligned} \tag{4.1}$$

Wobei die Fälle $C_1^{\mathcal{P}}$ und $C_2^{\mathcal{P}}$ (kein druckabhängiger Ausfluss) bereits aus [GR13] bekannt sind. Mit den Ergebnissen aus Kapitel 2 können wir in den übrigen Fällen die Zeit $\tau^{\mathcal{A}}(\langle P_i^c, T_j \rangle, \Gamma)$ mithilfe von Satz 4 berechnen. Wir ersetzen dafür die Größen des Tanks durch die entsprechenden Größen des Modells:

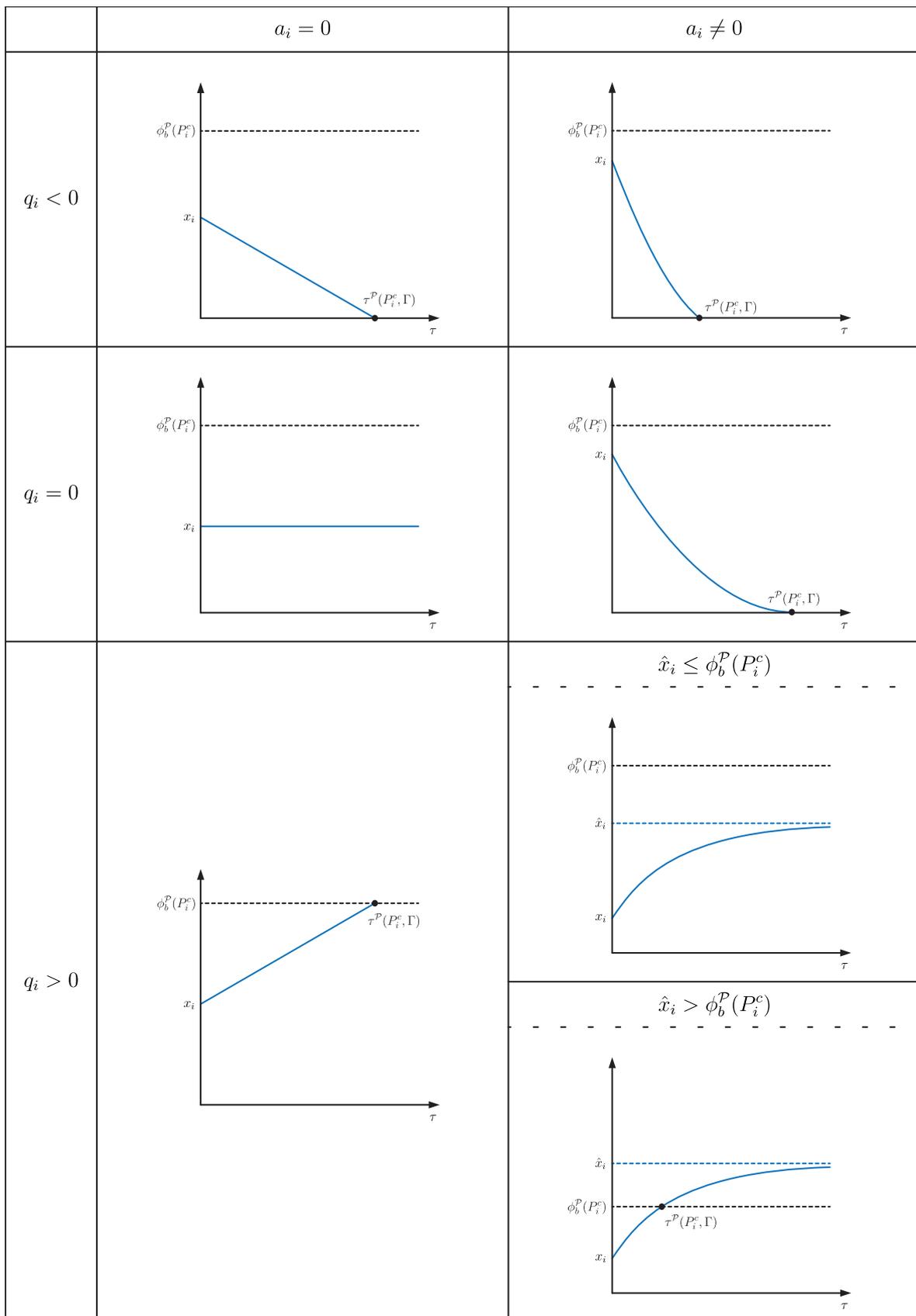


Abbildung 4.1: Fallunterscheidungen für die Ereignisse der kontinuierlichen Stellen

$$\begin{aligned}
V_0 &\mapsto x_i, \\
\hat{V} &\mapsto \hat{x}_i, \\
V &\mapsto \phi_b^{\mathcal{P}}(P_i^c) \text{ oder } V \mapsto 0, \\
G &\mapsto \phi_a^{\mathcal{P}}(P_i^c), \\
A &\mapsto a_i, \\
Q_c &\mapsto q_i.
\end{aligned} \tag{4.2}$$

Das Startvolumen V_0 wird durch den Füllstand x_i des aktuellen Zustands, die Grundfläche des Tanks G durch $\phi_a^{\mathcal{P}}$, die Querschnittfläche A des Ausflusses durch a_i und Q_c durch q_i ersetzt. Für die Asymptote \hat{V} schreiben wir \hat{x}_i und den Parameter V ersetzen wir je nach Fall durch die obere Begrenzung $\phi_b^{\mathcal{P}}(P_i^c)$ oder die untere Begrenzung 0 .

Für die Berechnung der Zeit $\tau^{\mathcal{P}}(P_i^c, \Gamma)$ gilt dann:

$$\tau^{\mathcal{P}}(P_i^c, \Gamma) = \begin{cases} -\frac{x_i}{q_i} & \text{falls } C_1^{\mathcal{P}}, \\ \frac{\phi_b^{\mathcal{P}}(P_i^c) - x_i}{q_i} & \text{falls } C_2^{\mathcal{P}}, \\ \frac{\phi_a^{\mathcal{P}}(P_i^c)}{2a_i g} \left(\frac{q_i}{a_i} \log \left(1 - \frac{a_i}{q_i} \sqrt{2g \frac{x_i}{\phi_a^{\mathcal{P}}(P_i^c)}} \right) + \sqrt{2g \frac{x_i}{\phi_a^{\mathcal{P}}(P_i^c)}} \right) & \text{falls } C_3^{\mathcal{P}}, \\ \frac{1}{a_i} \sqrt{\frac{2\phi_a^{\mathcal{P}}(P_i^c)}{g}} x_i & \text{falls } C_4^{\mathcal{P}}, \\ \frac{\phi_a^{\mathcal{P}}(P_i^c)}{2a_i g} \left(\frac{q_i}{a_i} \log \left(\frac{1 - \frac{a_i}{q_i} \sqrt{2g \frac{x_i}{\phi_a^{\mathcal{P}}(P_i^c)}}}{1 - \frac{a_i}{q_i} \sqrt{2g \frac{\phi_b^{\mathcal{P}}(P_i^c)}{\phi_a^{\mathcal{P}}(P_i^c)}}} \right) + \sqrt{2g \frac{x_i}{\phi_a^{\mathcal{P}}(P_i^c)}} - \sqrt{2g \frac{\phi_b^{\mathcal{P}}(P_i^c)}{\phi_a^{\mathcal{P}}(P_i^c)}} \right) & \text{falls } C_5^{\mathcal{P}}, \\ \infty & \text{sonst.} \end{cases} \tag{4.3}$$

Für alle kontinuierlichen Stellen $P_i^c \in \mathcal{P}^c$, für die $\tau^{\mathcal{P}}(P_i^c, \Gamma) < \infty$ ist, fügen wir ein Element $\Upsilon_i^{\mathcal{P}} = (\tau^{\mathcal{P}}(P_i^c, \Gamma), P_i^c)$ zur Menge $\mathcal{E}^{\mathcal{P}}(\Gamma)$ hinzu.

4.3.2 Ereignisse der Test- und Inhibitorkanten

$\mathcal{E}^{\mathcal{A}}(\Gamma)$ beinhaltet die Ereignisse die ausgelöst werden, wenn die Bedingung einer Test- oder Inhibitorkante, die zu einer kontinuierlichen Stelle führt, erfüllt wird bzw. nicht mehr erfüllt wird. $\mathcal{E}^{\mathcal{A}}(\Gamma) = \mathcal{A}^{\mathcal{T}^I}(\Gamma) \cup \mathcal{A}^{\mathcal{T}^D}(\Gamma) \cup \mathcal{A}^{\mathcal{T}^G}(\Gamma) \cup \mathcal{A}^{\mathcal{T}^F}(\Gamma) \cup \mathcal{A}^{\mathcal{T}^P}(\Gamma)$ lässt sich in fünf disjunkte Teilmengen partitionieren, entsprechend den Transitionen zu denen

die Kanten führen.

Wir berechnen für jede Kante $\langle P_i^c, T_j \rangle \in \mathcal{A}^t \cup \mathcal{A}^h$ die Zeit $\tau^A(\langle P_i^c, T_j \rangle, \Gamma)$ bis der Füllstand gleich der Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$ ist. Dabei treffen wir Fallunterscheidungen abhängig von den Größen a_i , q_i , x_i und $\phi_w^A(\langle P_i^c, T_j \rangle)$.

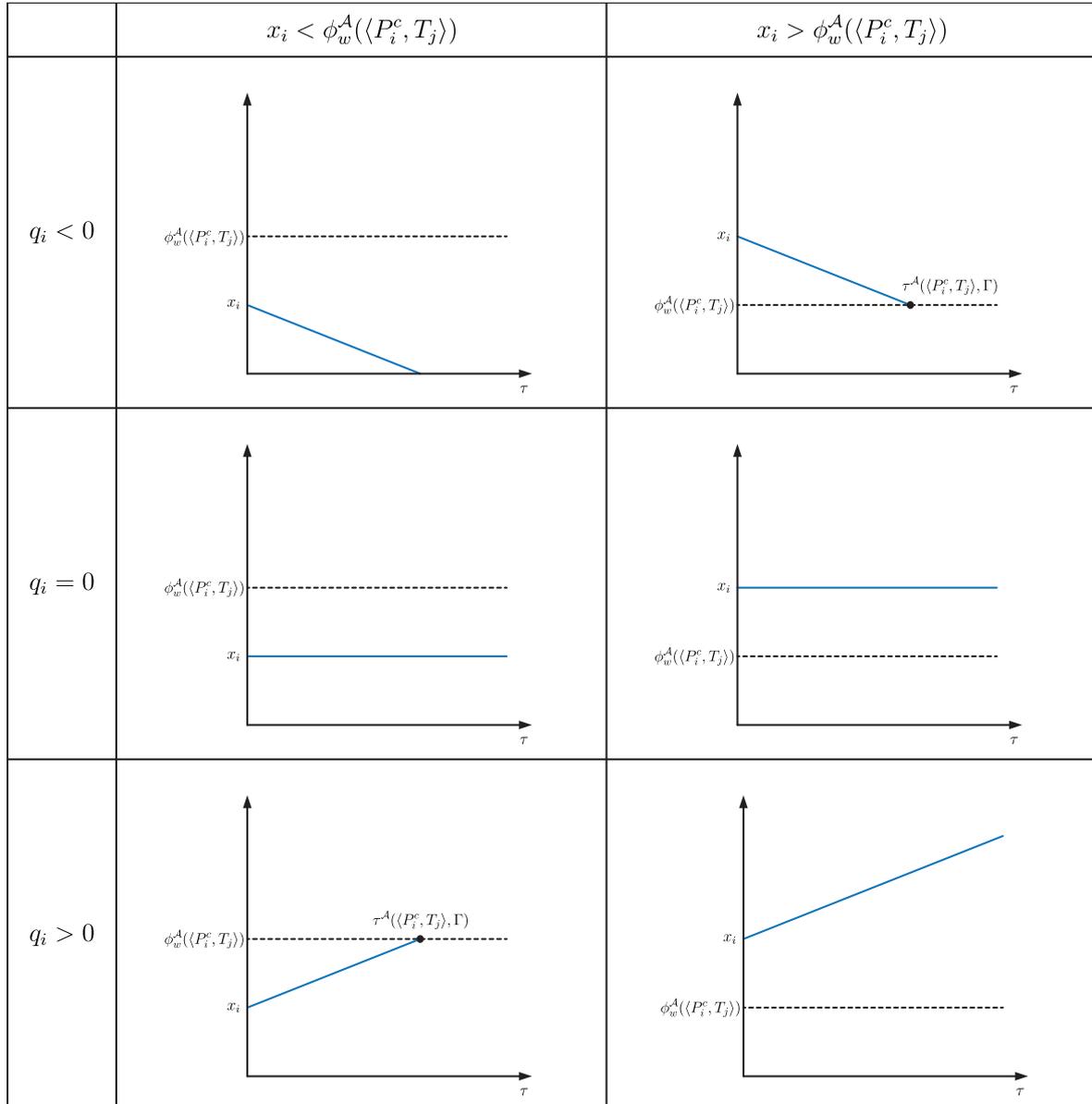


Abbildung 4.2: Fallunterscheidungen für die Ereignisse der Kanten (kein druckabhängiger Ausfluss)

Zunächst betrachten wir die aus [GR13] bekannten linearen Fälle für $a_i = 0$ (kein druckabhängiger Ausfluss). Dabei unterscheiden wir nach dem Vorzeichen von q_i und dem Vergleich von x_i mit der Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$ insgesamt sechs Fälle. Abbildung

4.2 zeigt für jeden Fall einen Graphen mit der Entwicklung des Füllstands über die Zeit τ und die Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$. In zwei Fällen wird die Bedingung erreicht. Wir bezeichnen sie mit:

$$\begin{aligned} C_1^A &:\Leftrightarrow a_i = 0 \wedge q_i < 0 \wedge x_i > \phi_w^A(\langle P_i^c, T_j \rangle), \\ C_2^A &:\Leftrightarrow a_i = 0 \wedge q_i > 0 \wedge x_i < \phi_w^A(\langle P_i^c, T_j \rangle). \end{aligned} \quad (4.4)$$

Als nächstes betrachten wir die nicht-linearen Fälle für $a_i \neq 0$ (druckabhängiger Ausfluss), die in Abbildung 4.3 dargestellt sind. Wir unterscheiden die Fälle nach dem Vorzeichen von q_i und dem Vergleich von x_i und die Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$. Im Fall von $q > 0$ vergleichen wir zusätzlich noch die Asymptote \hat{x}_i mit der Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$. Für jeden Fall ist ein Graph mit der Entwicklung des Füllstands über die Zeit τ und der Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$ abgebildet.

Wir finden so vier weitere Fälle, in denen die Bedingung erreicht wird:

$$\begin{aligned} C_3^A &:\Leftrightarrow a_i \neq 0 \wedge q_i < 0 \wedge x_i > \phi_w^A(\langle P_i^c, T_j \rangle), \\ C_4^A &:\Leftrightarrow a_i \neq 0 \wedge q_i = 0 \wedge x_i > \phi_w^A(\langle P_i^c, T_j \rangle), \\ C_5^A &:\Leftrightarrow a_i \neq 0 \wedge q_i > 0 \wedge x_i < \phi_w^A(\langle P_i^c, T_j \rangle) \wedge \hat{x}_i > \phi_w^A(\langle P_i^c, T_j \rangle), \\ C_6^A &:\Leftrightarrow a_i \neq 0 \wedge q_i > 0 \wedge x_i > \phi_w^A(\langle P_i^c, T_j \rangle) \wedge \hat{x}_i < \phi_w^A(\langle P_i^c, T_j \rangle). \end{aligned} \quad (4.5)$$

Mit den Ergebnissen aus Kapitel 2 können wir in diesen Fällen die Zeit $\tau^A(\langle P_i^c, T_j \rangle, \Gamma)$ mithilfe von Satz 4 berechnen. Wir ersetzen dafür die Größen des Tanks durch die entsprechenden Größen des Modells:

$$\begin{aligned} V_0 &\mapsto x_i, \\ \hat{V} &\mapsto \hat{x}_i, \\ V &\mapsto \phi_w^A(\langle P_i^c, T_j \rangle), \\ G &\mapsto \phi_a^P(P_i^c), \\ A &\mapsto a_i, \\ Q_c &\mapsto q_i. \end{aligned} \quad (4.6)$$

Im Unterschied zu den bereits beschriebenen Ersetzungen im vorherigen Abschnitt ersetzen wir den Parameter V dieses Mal durch die Bedingung $\phi_w^A(\langle P_i^c, T_j \rangle)$.

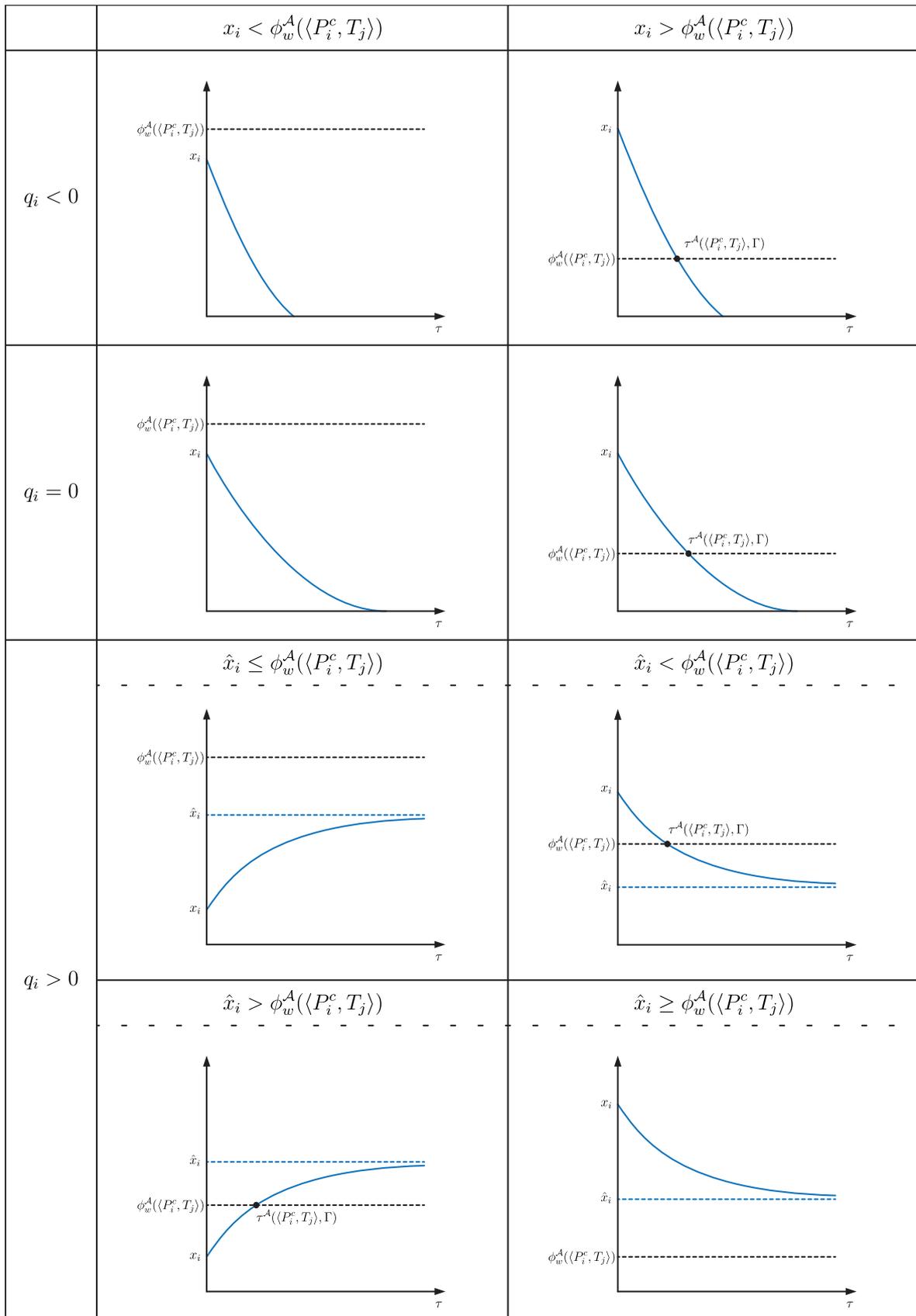


Abbildung 4.3: Fallunterscheidungen für die Ereignisse der Kanten (druckabhängiger Ausfluss)

Für die Berechnung der Zeit $\tau^{\mathcal{A}}(\langle P_i^c, T_j \rangle, \Gamma)$ gilt dann:

$$\tau^{\mathcal{A}}(\langle P_i^c, T_j \rangle, \Gamma) = \begin{cases} \frac{\phi_w^{\mathcal{A}}(\langle P_i^c, T_j \rangle) - x_i}{q_i} & \text{falls } C_1^{\mathcal{A}} \vee C_2^{\mathcal{A}}, \\ \frac{1}{a_i} \sqrt{\frac{2\phi_a^{\mathcal{P}}(P_i^c)}{g}} (\sqrt{x_i} - \sqrt{\phi_w^{\mathcal{A}}(\langle P_i^c, T_j \rangle)}) & \text{falls } C_4^{\mathcal{A}}, \\ \frac{\phi_a^{\mathcal{P}}(P_i^c)}{2a_i g} \left(\frac{q_i}{a_i} \log \left(\frac{1 - \frac{a_i}{q_i} \sqrt{2g \frac{x_i}{\phi_a^{\mathcal{P}}(P_i^c)}}}{1 - \frac{a_i}{q_i} \sqrt{2g \frac{\phi_w^{\mathcal{A}}(\langle P_i^c, T_j \rangle)}{\phi_a^{\mathcal{P}}(P_i^c)}}} \right) + \sqrt{2g \frac{x_i}{\phi_a^{\mathcal{P}}(P_i^c)}} - \sqrt{2g \frac{\phi_w^{\mathcal{A}}(\langle P_i^c, T_j \rangle)}{\phi_a^{\mathcal{P}}(P_i^c)}} \right) & \text{falls } C_3^{\mathcal{A}} \vee C_5^{\mathcal{A}} \vee C_6^{\mathcal{A}}, \\ \infty & \text{sonst.} \end{cases} \quad (4.7)$$

Für jede Kante $\langle P_i^c, T_j \rangle \in \mathcal{A}^t \cup \mathcal{A}^h$, für die $\tau^{\mathcal{A}}(\langle P_i^c, T_j \rangle, \Gamma) < \infty$ ist, fügen wir ein Element $\Upsilon_{i,j}^{\mathcal{A}} = (\tau^{\mathcal{A}}(\langle P_i^c, T_j \rangle, \Gamma), \langle P_i^c, T_j \rangle)$ zur Menge $\mathcal{E}^{\mathcal{A}}(\Gamma)$ hinzu.

Die konkrete Teilmenge, in die wir $\Upsilon_{i,j}^{\mathcal{A}}$ einfügen, ist abhängig von der Transition T_j zu der die Kante $\langle P_i^c, T_j \rangle$ führt. Führt z. B. eine Kante $\langle P_i^c, T_j \rangle$ zu einer deterministischen Transition $T_j^D \in \mathcal{T}^D$, so fügen wir ein Element $\Upsilon_{i,j}^{\mathcal{A}^D} = (\tau^{\mathcal{A}}(\langle P_i^c, T_j^D \rangle, \Gamma), \langle P_i^c, T_j^D \rangle)$ zur Menge $\mathcal{E}^{\mathcal{A}^D}(\Gamma)$ hinzu

4.4 Behandlung der Ereignisse

Wir sortieren die Ereignisse $\Upsilon_i = (\Delta\tau_i, e_i)$ der Menge $\mathcal{E}(\Gamma)$ nach der Zeit $\Delta\tau$ bis sie eintreffen würden und filtern nur die Elemente mit minimaler verbleibender Zeit heraus. Diese Ereignisse fassen wir als Menge $\mathcal{E}_{min}(\Gamma)$ zusammen:

$$\mathcal{E}_{min}(\Gamma) = \{\Upsilon_i = (\Delta\tau_i, e_i) \in \mathcal{E}(\Gamma) \mid \nexists \Upsilon_j = (\Delta\tau_j, e_j) \in \mathcal{E}(\Gamma) : \Delta\tau_j < \Delta\tau_i\} \quad (4.8)$$

Den unterschiedlichen Ereignistypen wird eine Ordnung zugeteilt. Von den Ereignissen mit minimaler verbleibender Zeit werden nur diejenigen mit minimaler Ordnung betrachtet. Tabelle 4.1 ist die Anpassung der Tabelle aus [GR13, S. 17]. Sie zeigt die Reihenfolge in der Ereignisse mit minimaler verbleibender Zeit berücksichtigt werden und welche Effekte diese auslösen.

Ordnung	Ereignis	Effekt
I	$\Upsilon_k^{\mathcal{T}^I} \in \mathcal{E}^{\mathcal{T}^I}(\Gamma)$	a. E2a Auflösung des Schaltkonflikts b. E3a Schalten der Transition c. E4 Berechnung der druckabhängigen Maße d. E5 Anpassung der effektiven Volumenströme
II	$\Upsilon_k^{\mathcal{A}^I} \in \mathcal{E}^{\mathcal{A}^I}(\Gamma)$	a. E1 Entwicklung der Füllstände
III	$\Upsilon_k^{\mathcal{T}^D} \in \mathcal{E}^{\mathcal{T}^D}(\Gamma)$	a. E1 Entwicklung der Füllstände b. E2b Auflösung des Schaltkonflikts c. E3b Schalten der Transition d. E4 Berechnung der druckabhängigen Maße e. E5 Anpassung der effektiven Volumenströme
IV	$\Upsilon_k^{\mathcal{P}} \in \mathcal{E}^{\mathcal{P}}(\Gamma)$ $\Upsilon_k^{\mathcal{A}^F} \in \mathcal{E}^{\mathcal{A}^F}(\Gamma)$	a. E1 Entwicklung der Füllstände b. E4 Anpassung der effektiven Volumenströme
V	$\Upsilon_k^{\mathcal{A}^D} \in \mathcal{E}^{\mathcal{A}^D}(\Gamma)$ $\Upsilon_k^{\mathcal{A}^G} \in \mathcal{E}^{\mathcal{A}^G}(\Gamma)$	a. E1 Entwicklung der Füllstände
VI	$\Upsilon_k^{\mathcal{T}^G} \in \mathcal{E}^{\mathcal{T}^G}(\Gamma)$	a. E1 Entwicklung der Füllstände b. E2c Auflösung des Schaltkonflikts c. E3c Schalten der Transition d. E4 Berechnung der druckabhängigen Maße e. E5 Anpassung der effektiven Volumenströme
VII	$\Upsilon_k^{\mathcal{A}^P} \in \mathcal{E}^{\mathcal{A}^P}(\Gamma)$	a. E4 Berechnung der druckabhängigen Maße b. E5 Anpassung der effektiven Volumenströme

Tabelle 4.1: Behandlung der Ereignisse

Bei den Ereignissen von Ordnung I und III muss ggfs. zunächst unter mehreren Ereignissen ein Ereignis, das eintritt, bzw. eine Transition, die schaltet, ausgewählt werden. Dies geschieht mit den Algorithmen zur Konfliktauflösung wie in [GR13] beschrieben.

Da wir im Gegensatz zu [GR13] mehrere stochastische Transitionen zulassen, könnte der äußerst unwahrscheinliche Fall eintreten, dass zwei (oder mehrere) stochastische Transitionen gleichzeitig schalten würden. In diesem Fall wählen wir zufällig unter gleichen Wahrscheinlichkeiten eine Transition aus.

4.5 Betrachtete Größen

Bei jeder Simulation will man bestimmte Größen betrachten um Daten zu erheben und später auszuwerten. Im Fall des Simulators für Hybride Petri-Netze mit druckabhängigen Transitionen unterscheiden wir zwischen zwei unterschiedliche Arten der Simulation.

Bei der ersten zeichnen wir den Füllstand einer kontinuierlichen Stelle über die Zeit auf. Es findet nur ein einziger Durchlauf statt. Diese Art der Simulation eignet sich insbesondere für Petri-Netze ohne stochastische Transition, um den Ablauf besser nachvollziehen zu können.

Mit der zweiten Simulationsart kann die Zuverlässigkeit des HPPt getestet werden. Dazu formulieren wir Bedingungen an Eigenschaften des HPPt, die dann bei einer festzulegenden Anzahl von Durchläufen überprüft werden. Eine Bedingung ist eine Aussage über Eigenschaften des HPPt, wie z. B. die folgenden Aussagen über einen Füllstand x_i oder eine Markierung m_i :

$$\begin{aligned}\psi_{x_i} &= x_i < a \mid x_i > a, \\ \psi_{m_i} &= m_i < a \mid m_i = a \mid m_i > a, \\ &\dots\end{aligned}\tag{4.9}$$

Neben diesen hier vorgeschlagenen Bedingungen, können beliebige andere Bedingungen formuliert werden. Alle für eine Simulation formulierten Bedingungen fassen wir in einer Menge Ψ zusammen. Bei der Simulation wird dann bei jedem Durchlauf und bei jedem Zeitschritt für den aktuellen Zustand überprüft, ob alle Aussagen $\psi \in \Psi$ wahr sind. Nach Ablauf einer festen Zahl von Durchläufen wird für jeden Zeitschritt die relative Häufigkeit bestimmt, mit der die Bedingungen zu diesem Zeitschritt erfüllt sind. Bei einem Petri-Netz ohne stochastische Transitionen wären alle relativen Häufigkeiten unabhängig von der Anzahl der Durchläufe entweder 0 oder 1. Diese Art der Simulation eignet sich daher insbesondere für Petri-Netze mit stochastischen Transitionen.

5 Implementierung

Dieses Kapitel beschreibt die Implementierung des Simulators. Zunächst werden die Anforderungen an den Simulator formuliert und Programme aus vorhergegangenen Arbeiten auf ihre Wiederverwendbarkeit untersucht. Im Hauptteil des Kapitels beschreiben wir dann die Systemarchitektur der Implementierung und erläutern das Zusammenspiel ihrer Komponenten beim Ablauf einer Simulation. Abschließend gehen wir noch auf das verwendete Dateiformat, die Generierung der Zufallszahlen, numerische Schwierigkeiten und mögliche Verbesserungen ein.

5.1 Anforderungen

Mit dem Simulator soll es möglich sein die beiden in 4.5 beschriebenen Simulationsarten, das Aufzeichnen des Füllstands über die Zeit und das Testen der Zuverlässigkeit durch Bedingungen, für HPPTs durchzuführen.

Das Programm fordert dabei als Eingabe ein korrektes HPPT und je nach Simulationsart zusätzliche Parameter, wie die maximale Ausführungszeit, Bedingungen, etc.

Das Programm soll als Ausgabe einen Graphen liefern, der je nach Simulationsart den Füllstand einer Stelle oder die Wahrscheinlichkeit, dass alle Bedingungen erfüllt sind, über die Ausführungszeit zeigt. Für weitere Untersuchungen soll es auch möglich sein eine Liste mit den einzelnen Wertepaaren auszugeben.

5.2 Vorhandenes System

Mit dem C-Programm Fluid Survival Tool (FST) [Pos13] existierte bereits ein Analyse-Tool für HPnGs nach [GR13]. Es wurde im Zuge der Arbeit [vdG14] um eine Diskrete Ereignis Simulation (DES) erweitert.

Ursprünglich war geplant das bestehende C-Programm so zu erweitern, dass es Hybride Petri-Netze mit druckabhängigen Transitionen simulieren kann. Neben der Simulation lag der Anwendungsschwerpunkt des FST auf der Analyse von HPnGs [GR13, S. 18 ff.]. Diese ist bisher nur für HPnGs mit einer stochastischen Transition möglich und wurde noch nicht auf das HPpT Modell übertragen. Da die unterschiedlichen Funktionen, Analyse von HPnGs und Simulation von HPpTs, nur eine kleine Schnittmenge haben, hätten große Teile des bestehenden Programms wie z. B. die Anpassung der effektiven Volumenströme oder die Erstellung der Ereignismenge auf Grund der Erweiterung des Modells stark umgeschrieben werden müssen. Dies hat letztendlich zu der Entscheidung geführt den Simulator von Grund auf neu zu schreiben.

Das FST benutzte zudem ein textbasiertes Format für die Eingabe von Petri-Netzen mit einem eigenen Parser. Das Format war verhältnismäßig unübersichtlich und leichte Abweichungen, wie z. B. zusätzliche Leerzeichen, konnten schnell zu Fehlern führen. In der Implementierung des neuen Simulators sollte daher auch das Eingabeformat durch ein XML-Format für HPpTs ersetzt werden.

Der neue Simulator ist nun in Java und nicht in C geschrieben. Es wurden keine der vorhandenen C-Bibliotheken verwendet. Die einzigen externen Bibliotheken, die verwendet wurden, sind JMathPlot [Ric15], zur Ausgabe der Graphen, und Commons Math [Fou15], für diverse mathematische Funktionen. Gewisse Aspekte des FST sind aber dennoch in den Simulator eingeflossen. So ist z. B. die Benutzeroberfläche des neuen Simulators stark an die des FST angelehnt.

5.3 Systemarchitektur

Der Simulator ist unter dem Model-View-Controller-Architekturmuster entworfen. Wir teilen daher Klassen für Datenmodell, Darstellung und Steuerung auf die Pakete `model`, `view` und `controller` auf. Ein zusätzliches Paket `lambertw` umfasst die Klasse zur Berechnung der beiden Äste W_0 und W_{-1} der Lambert-W-Funktion.

5.3.1 Model

Die Model-Klassen sind die Umsetzung des im Kapitel 3 formal als 6-Tupel definierten HPpT und seiner Elemente (Transitionen, Stellen und Kanten) als Klassen. Abbildung 5.1 zeigt das `model`-Paket mit den Klassen `HPpT` und `State` und seinen Unterpaketen.

Exemplarisch sind außerdem die Klassen der Unterpakete `arc` und `place` abgebildet. Die Pfeile kennzeichnen Vererbung zwischen den Klassen. Abstrakte Klassen bündeln dabei Eigenschaften, indem sie solche Eigenschaften implementieren, die mehreren Elementen zugeordnet werden. Beispielsweise zeichnen sich sowohl die Kanten von kontinuierlichen Stellen zu kontinuierlichen Transitionen $\langle P_j^c, T_i^F \rangle \in \mathcal{P}^c \times \mathcal{T}^F$ als auch die umgekehrten Kanten $\langle T_i^F, P_j^c \rangle \in \mathcal{T}^F \times \mathcal{P}^c$ durch die ihnen zugeordneten Eigenschaften ϕ_s^A und ϕ_p^A aus. Im Programm werden diese Eigenschaften durch die abstrakte Klasse `FluidArc` implementiert, von der die Klassen `FluidInputArc` und `FluidOutputArc`, die sich durch Start- und Endpunkt unterscheiden, erben.

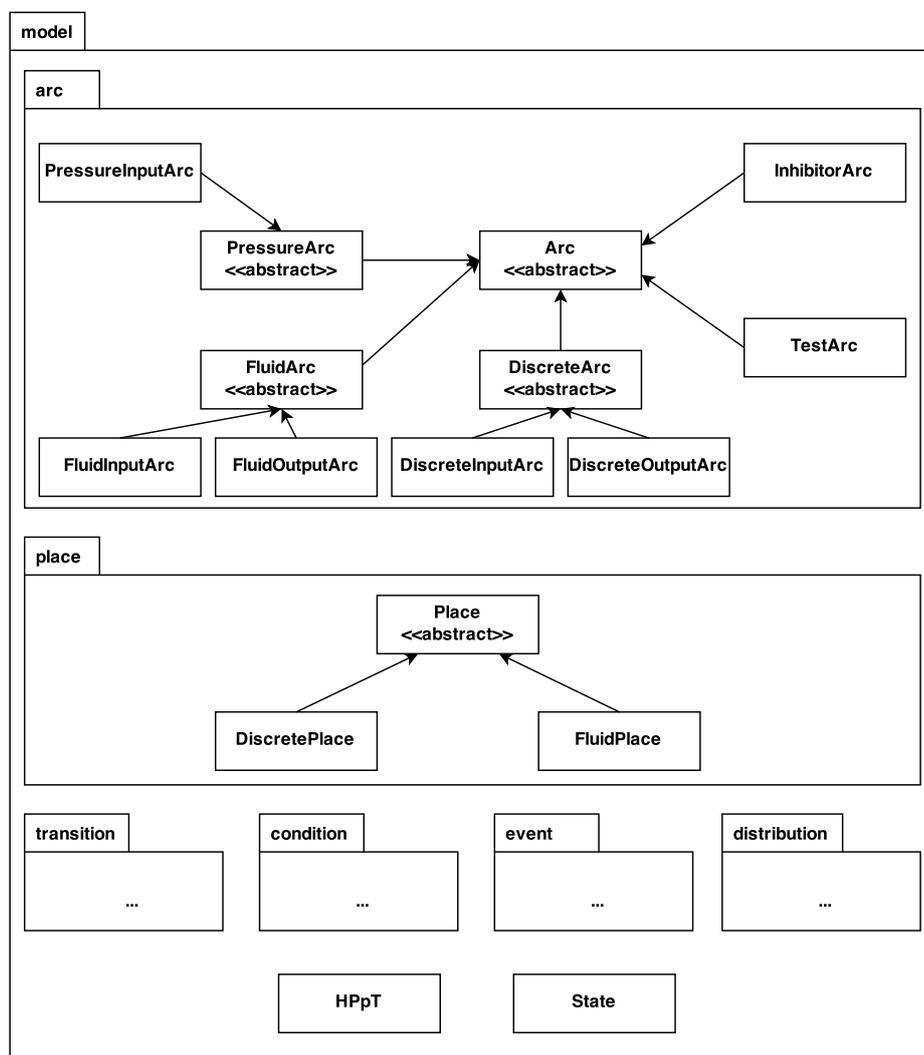


Abbildung 5.1: model-Paket

Neben den Paketen für die einzelnen Elemente umfasst das `model`-Paket noch Pakete für die Wahrscheinlichkeitsverteilungen der stochastischen Transitionen (`distribution-`

Paket), die bei der Simulation auftretenden Ereignisse (**event**-Paket), sowie die bei einem Test auf Zuverlässigkeit formulierten Bedingungen (**condition**).

5.3.2 View

Abbildung 5.2 zeigt das **view**-Paket. Die Instanzen der **view**-Klassen stellen die Fenster der Benutzeroberfläche dar und sind von der Klasse **JFrame** abgeleitet.

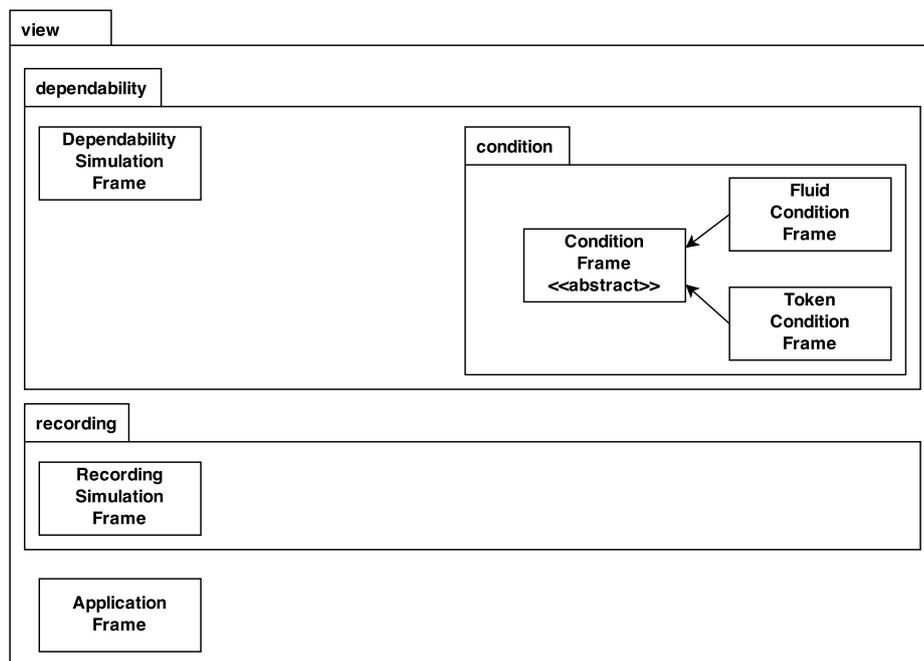


Abbildung 5.2: view-Paket

Ein Objekt der Klasse **ApplicationFrame** z. B. stellt das Hauptfenster des Programms dar. Es sind Buttons verfügbar mit denen ein HPpT geladen und Simulationen der beiden Simulationsarten gestartet werden können. Außerdem stehen zwei Textbereiche bereit in denen zum einen ein geladenes HPpT angezeigt wird und zum anderen Statusbenachrichtigungen wie Fehlermeldungen ausgegeben werden können.

5.3.3 Controller

Abbildung 5.3 zeigt das **controller**-Paket, das die Klassen umfasst, deren Instanzen die Behandlung der Benutzerinteraktionen steuern.

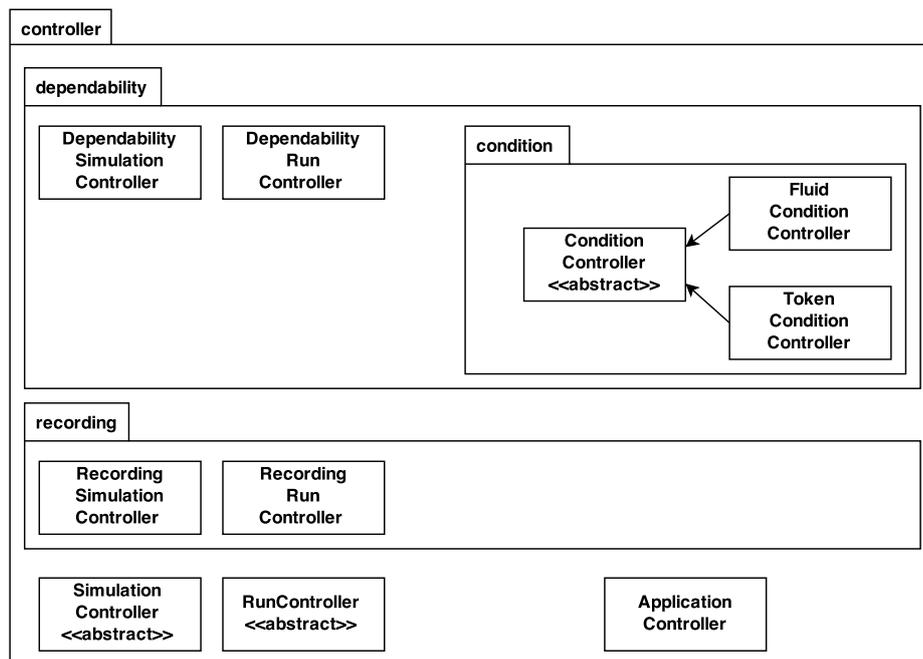


Abbildung 5.3: controller-Paket

Eine Benutzerinteraktion, wie z. B. ein Klick auf den Button „Simulation starten“ im Fenster `DependabilitySimulationFrame`, ruft eine Methode im entsprechenden Controller, in diesem Fall `simulate()` im `DependabilitySimulationController` auf, die dann die Benutzerinteraktion ausführt.

Die Klasse `Application` verwendet als einzige Klasse das Singleton Entwurfsmuster, damit auf ihre Instanz von jedem Objekt aus zugegriffen werden kann, um Benachrichtigungen auszugeben.

5.4 Ablauf einer Simulation

Um das Zusammenspiel der Klassen zu erläutern, betrachten wir den Ablauf einer Simulation, die die Zuverlässigkeit eines HPPt-Modells testen soll. Wir beginnen die Betrachtung zu dem Zeitpunkt nachdem der Benutzer bereits das HPPt geladen hat und alle Parameter und Bedingungen für die Simulation eingegeben hat und auf den Button „Simulation starten“ klickt. Wie bereits beschrieben, führt dies zunächst zum Aufruf der Methode `simulate()` im `DependabilitySimulationController`.

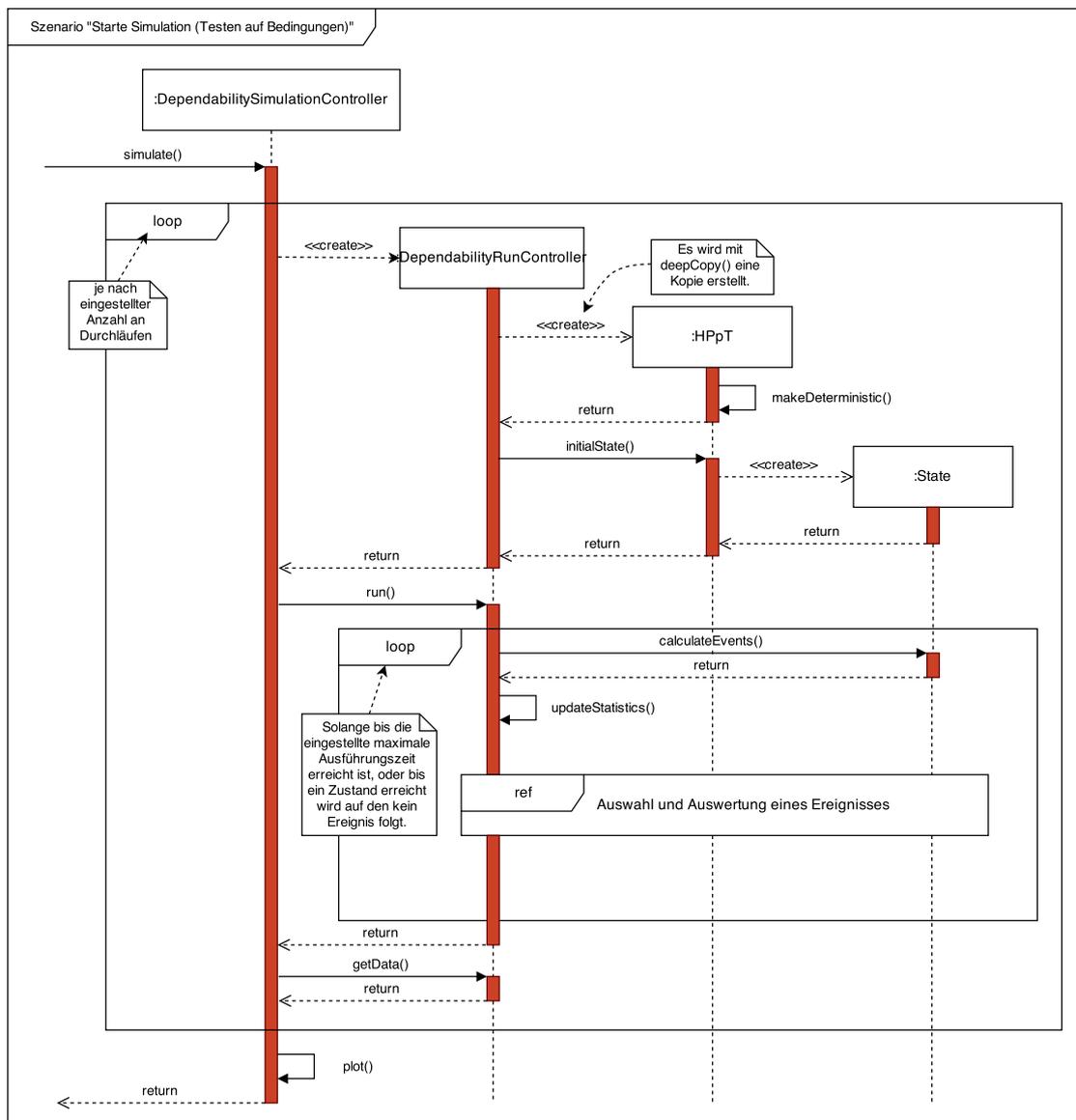


Abbildung 5.4: Szenario „Simulation starten“

Das Sequenzdiagramm in Abbildung 5.4 beschreibt den weiteren Verlauf der Simulation. Je nachdem welche Anzahl an Durchläufen der Benutzer eingestellt hat, wird nun die äußere Schleife durchlaufen.

Bei jedem Durchlauf wird ein neues `RunController`-Objekt erstellt, welches wiederum eine Kopie des geladenen `HPpT`-Objekts erstellt. An dieser Kopie werden anschließend durch den Aufruf von `makeDeterministic()` für alle stochastischen Transitionen Zufallszahlen als Schaltzeiten bestimmt. Da bei jedem Durchlauf eine neue Kopie des `HPpT`-Objekts erstellt wird, sind die Zufallszahlen bei jedem Durchlauf anders. Nachdem das `HPpT`-Objekt vorbereitet wurde, wird durch den Aufruf von `initialState()`

der Startzustand erstellt und durch den Aufruf von `run()` der Durchlauf gestartet.

Die innere Schleife ist im Wesentlichen die Umsetzung der in Kapitel 4 beschriebenen Simulation: Für den aktuellen Zustand wird die Ereignismenge durch den Aufruf von `calculateEvents()` berechnet und nur die Ereignisse mit minimaler verbliebener Zeit weiter betrachtet. Bis zu dieser Zeit werden dann durch den Aufruf von `updateStatistics()` Daten für die Auswertung gesammelt. Konkret werden bei dieser Simulationsart die vorher vom Nutzer formulierten Bedingungen in einzelnen Zeitschritten überprüft. Anschließend wird unter den verbleibenden Ereignissen analog zu Abschnitt 4.4 das mit minimaler Ordnung ausgewählt und ausgewertet. Der dadurch ermittelte Folgezustand wird zum aktuellen Zustand für den nächsten Durchlauf der inneren Schleife.

Die Schleife wird solange durchlaufen, bis ein Zustand erreicht wird für den die Ereignismenge leer ist oder die maximale Ausführungszeit erreicht wird.

Nachdem die innere Schleife verlassen wurde, werden mit `getData()` die Daten des Durchlaufs ausgelesen. Nach dem Abschluss der äußeren Schleife, werden die gesammelten Daten dann ausgewertet und durch `plot()` der Graph erstellt und angezeigt.

5.5 Persistenz

Während Objekte wie Status, Ereignisse und Bedingungen nur zur Laufzeit erstellt und verwendet werden, möchten wir das HPPt-Modell auch über die Ausführungszeit des Programms hinaus erhalten. Ein HPPt-Modell soll daher nach dem Programmstart geladen werden können.

Im Gegensatz zum FST kann mit dem Simulator selbst sogar gar kein HPPt-Modell erstellt werden. HPPts müssen mit einem Texteditor erstellt und dann mit dem Simulator geöffnet werden. Als Dateiformat wurde ein XML-basiertes Format entworfen.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <HPPt name="HPPt-Reservoir">
3   <Places>
4     <FluidPlace id="Pc1" initialFluid="0.0" boundary="2000.0" groundarea="50.0"/
5     >
6     <DiscretePlace id="Pd1" initialTokens="1"/>
7     <DiscretePlace id="Pd2" initialTokens="1"/>
8   </Places>
9   <Transitions>
10    <FluidTransition id="TF1" nominalFlowRate="1.0"/>
```

```

10     <PressureTransition id="TP1" />
11     <GeneralTransition id="TG1">
12         <Exp lambda="0.01"/>
13     </GeneralTransition>
14     <DeterministicTransition id="TD1" priority="1" weight="1.0" firingTime="
        120.0"/>
15 </Transitions>
16 <Arcs>
17     <PressureInputArc id="A1" from="Pc1" to="TP1" crossSectionArea="0.01"/>
18     <FluidOutputArc id="A2" from="TF1" to="Pc1" weight="1.0" share="1.0"
        priority="1"/>
19     <TestArc id="A3" from="Pd1" to="TF1" weight="1.0"/>
20     <DiscreteInputArc id="A4" from="Pd1" to="TD1" weight="1.0"/>
21     <TestArc id="A5" from="Pd2" to="TP1" weight="1.0"/>
22     <DiscreteInputArc id="A6" from="Pd2" to="TG1" weight="1.0"/>
23 </Arcs>
24 </HPpT>

```

Auflistung 5.1: Beispiel für ein HPpT im XML Format

Auflistung 5.1 zeigt das in Kapitel 3 vorgestellte HPpT-Beispiel im XML-Format. Jedes Element eines HPpT wird als ein XML-Element bestehend aus Start- und End-Tag abgebildet. Die ihm zugeordneten Eigenschaften werden zu Attributen des jeweiligen XML-Elements. Je nach seiner Art ist ein Element dann Kindelement einer der Listen Places, Transitions, oder Arcs, die wiederum Kindelemente des umschließenden HPpT-Element sind. Das komplette XML-Schema liegt dem Programm bei.

Für das Parsen der Dateien wird die Java-Schnittstelle Java Architecture for XML Binding (JAXB) verwendet. Dafür wurden in den `model`-Klassen Java-Annotationen hinzugefügt. Die XML-Datei wird beim Öffnen durch den Parser in ein Java Objekt der Klasse `HPnG` umgewandelt. Die Umkehrung ist theoretisch auch möglich wird aber bisher im Programm nicht benötigt. // Nach dem Laden des Petri-Netzes wird dieses noch initialisiert, um Listen für Mengen wie z. B. $\mathcal{O}^F(P_i^e)$ zu berechnen. // Der Inhalt der XML-Datei wird nach einem erfolgreichen Laden des Petri-Netzes zwar zur Übersicht im Hauptfenster angezeigt, kann aber dort nicht mehr verändert werden.

5.6 Generierung der Zufallszahlen

Für jede stochastische Transition wird bei jedem Durchlauf eine zufällige Schaltzeit entsprechend der ihr zugeordneten kumulativen Verteilungsfunktion (CDF) generiert.

Um aus den durch die Java-Klasse `Random` generierten gleichverteilten Zufallszahlen andersverteilte Zufallszahlen zu generieren, wenden wir die Inversionsmethode [Hav98, S. 424 ff.] an. Hierbei benutzen wir das Inverse der kumulativen Wahrscheinlichkeitsfunktion.

Wir zeigen das Verfahren exemplarisch für die Exponentialverteilung. Diese hat die CDF $F : \mathbb{R} \rightarrow [0, 1)$,

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases} \quad (5.1)$$

Das Inverse der CDF $F^{-1} : [0, 1) \rightarrow \mathbb{R}$ ist dann:

$$F^{-1}(p) = -\frac{\log(1-p)}{\lambda}. \quad (5.2)$$

Setzt man nun in F^{-1} gleichverteilte Wahrscheinlichkeiten, also Zahlen aus $[0, 1) \subset \mathbb{R}$, ein, erhält man exponentialverteilte Zufallswerte.

5.7 Numerische Schwierigkeiten

Während wir im Kontext der Modellierung bei Berechnungen noch von exakten reellen Zahlen ausgehen konnten, müssen wir bei der Implementierung beachten, dass eine Fließkommazahl nur eine approximative Darstellung einer reellen Zahl ist. Dadurch treten einige numerische Schwierigkeiten auf.

Bei jeder Berechnung einer Zeit $\Delta\tau$ bis zu einem Ereignis wie in Abschnitt 4.3 oder eines Füllstands x'_i einer kontinuierlichen Stelle wie in Abschnitt 3.4 kommt es zu Rundungsfehlern. Folgendes Beispiel soll dies verdeutlichen:

Eine kontinuierliche Stelle P_1^c erreicht in der Zeit $\Delta\tau$ ihre obere Begrenzung $\phi_b^P(P_1^c) = 10$. Das Ereignis sei das einzige in der Ereignismenge und wird ausgewählt und ausgeführt. Bei der Entwicklung der Füllstände wird nun die entsprechende Formel für $\Delta\tau$ ausgewertet. Der neue Füllstand x'_1 ist nun aber durch die Rundungsfehler im Allgemeinen nicht gleich 10.

Um die Gefahr auszuschließen, dass $x'_1 > 10$ ist kann bei allen Tests auf das Erreichen der oberen (unteren) Begrenzung anstelle von „=“ auf „ \geq “ („ \leq “) getestet werden. Aber

auch wenn $x'_1 < 10$ ist, kann es unter Umständen zu Problemen kommen. In diesem Fall wird für die das Erreichen der oberen Begrenzung ein neues Ereignis berechnet und es könnte erneut vorkommen, dass der neue Füllstand $x''_1 < 10$ ist. Im ungünstigsten Fall entsteht eine Endlosschleife. Um auch dieses Problem zu umgehen werden grundsätzlich bei allen Füllstand-Berechnungen die Ergebnisse auf zwei Nachkommastellen gerundet. Die Gefahr, dass der neue Füllstand auf zwei Nachkommastellen immer noch von der Begrenzung abweicht, ist dann sehr gering.

Bei anderen Berechnung wie z. B. Zeiten, effektive Volumenströme, etc. treten ebenfalls Rundungsfehler auf, sie haben jedoch keinen kritischen Einfluss auf den Ablauf der Simulation.

5.8 Erweiterbarkeit

Das Programm besitzt noch einige Schwächen. Es wurden zwar viele Fälle mit dem Simulator getestet, jedoch können immer noch unentdeckte Fehler existieren. In Zukunft sollten weitere Tests folgen.

Bei vielen (mehr als 1000) Simulationsdurchläufen braucht das Programm auf weniger leistungsstarken Systemen schon einige Sekunden. Gegebenenfalls könnte eine genaue Performance-Analyse aufzeigen, wie das Programm schneller gemacht werden kann. So könnte z. B. die Implementierung der Lambert-W-Funktion durch Intervallschachtelung zu Performance Einbußen führen. Eine Implementierung mit Potenzreihen wäre vielleicht effizienter.

Statt den Inhalt der geladenen XML-Datei einfach nur als Text anzuzeigen könnte das HPpT entweder wie im FST auch editierbar sein oder stattdessen sogar graphisch dargestellt werden.

In dem Simulator sind mit den Klassen `UniformDistribution`, `NormalDistribution` und `ExponentialDistribution` drei Verteilungen implementiert. Theoretisch kann aber auch jede andere Wahrscheinlichkeitsverteilung implementiert werden, indem sie die abstrakte Klasse `ProbabilityDistribution` implementiert.

Ebenso können beliebig viele zusätzliche Bedingungen erstellt werden. Hierfür muss ein neuer Controller, der von `ConditionController` erbt, ein Model, das von `Condition` erbt, und ein View, der von `ConditionFrame` erbt, implementiert werden.

6 Fallbeispiel

Nachdem wir in den vorherigen Kapiteln den Modellformalismus für HPpTs eingeführt haben, wollen wir mit diesem nun ein Fallbeispiel betrachten und es mit dem entwickelten Simulator auf Zuverlässigkeit prüfen. Dafür werden wir in 6.1 zunächst einen Fall als ein HPpT Modell beschreiben, bevor wir in 6.2 die Ergebnisse der Simulation auswerten.

6.1 Fallbeschreibung

Abbildung 6.1 zeigt unser Fallbeispiel als HPpT-Modell, das im folgenden beschrieben wird. Wir geben dabei auch die Einheiten explizit an.

Im Mittelpunkt steht der Wassertank P_1^c mit einer Kapazität von $\phi_b^P(P_1^c) = 2000 \text{ m}^3$ und einer Grundfläche von $\phi_a^P(P_1^c) = 50 \text{ m}^2$. Der Tank P_1^c versorgt über einen druckabhängigen Ausfluss auf Bodenhöhe kontinuierlich einen Verbraucher T_1^P mit Wasser. Das Ausflussrohr $\langle P_1^c, T_1^P \rangle$ hat einen Querschnitt von $\phi_a^A(\langle P_1^c, T_1^P \rangle) = 0,015625 \text{ m}^2 = 156,25 \text{ cm}^2$. Gefüllt wird der Tank P_1^c durch die Zuflüsse T_1^F , T_2^F und T_3^F , die mit stückweise konstanten Volumenströmen kontinuierlich Wasser in den Tank P_1^c pumpen. Ihre nominellen Volumenströme sind $\phi_f^T(T_1^F) = 0,18 \frac{\text{m}^3}{\text{s}}$, $\phi_f^T(T_2^F) = 0,13 \frac{\text{m}^3}{\text{s}}$ und $\phi_f^T(T_3^F) = 0,11 \frac{\text{m}^3}{\text{s}}$.

Durch diesen Zufluss und den druckabhängigen Ausfluss nähert sich das Volumen im Tank P_1^c an eine Asymptote wie in 2.5 an.

Bei normalen Bedingungen hat der Tank P_1^c einen konstanten Zufluss von $0,18 \frac{\text{m}^3}{\text{s}} + 0,13 \frac{\text{m}^3}{\text{s}} + 0,11 \frac{\text{m}^3}{\text{s}} = 0,42 \frac{\text{m}^3}{\text{s}}$. Die Asymptote liegt dann bei $1841,32 \text{ m}^3$ ein.

Untersucht werden soll nun, ob bei Ausfällen der Zuflüsse durch einen Defekt der Pumpen, das Wasser in Tank P_1^c ausreicht, um über den Zeitraum der Reparatur die Versorgung des Verbrauchers zu gewährleisten.

Dafür modellieren wir für jeden Zufluss den Defekt und die Reparatur wie hier exem-

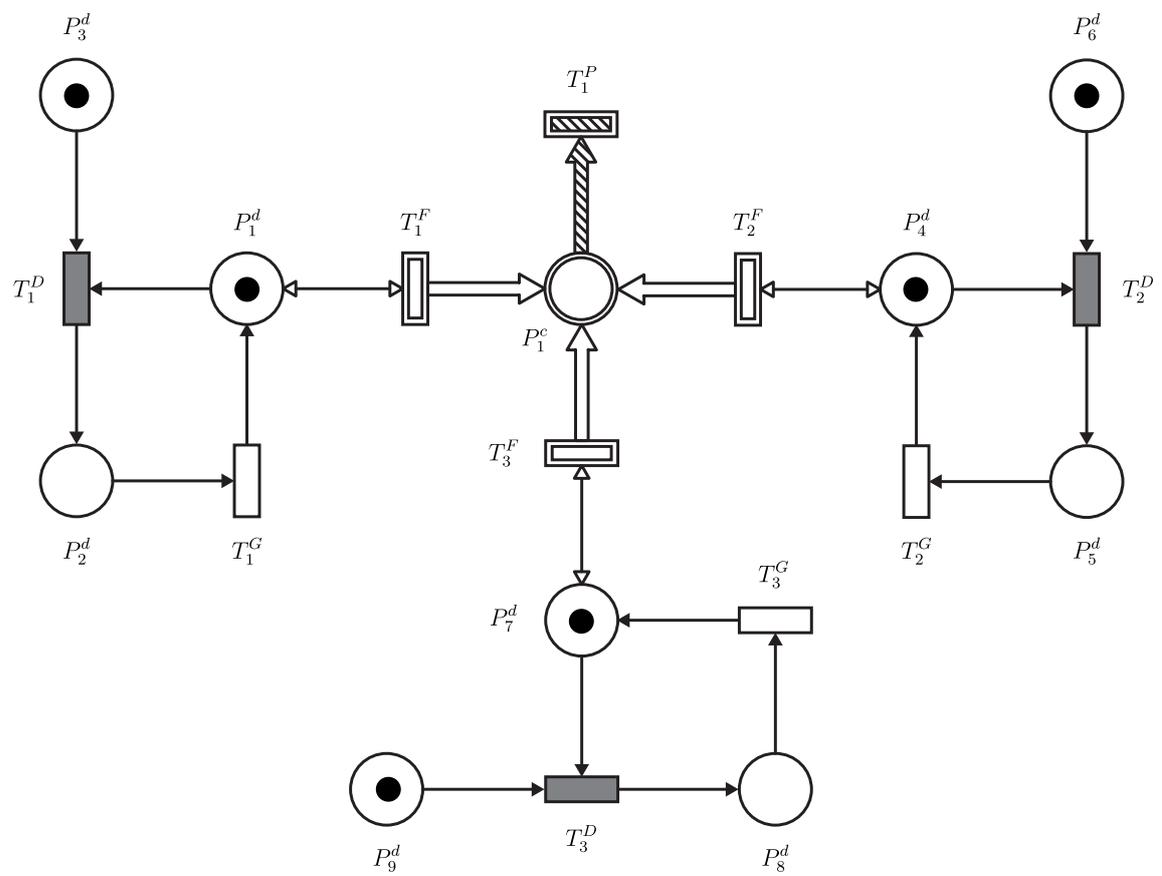


Abbildung 6.1: Fallbeispiel als HPpT Modell

plarisch für den Zufluss T_1^F beschrieben: Wir ergänzen drei diskrete Stellen P_1^d (Pumpe intakt), P_2^d (Pumpe defekt) und P_3^d (Defekt wird noch eintreten) und zwei Transitionen T_1^D (Defekt tritt ein) und T_1^G (Pumpe wird repariert) und verbinden diese mit Kanten wie es Abbildung 6.1 zeigt. Die Gewichtung der Kanten ist dabei immer 1.

Auf den Stellen P_1^d und P_3^d liegt zunächst eine Marke. Die Transition T_1^D hat dann Erlaubnis und wird nach einer festgelegten Zeit schalten und somit den Defekt eintreten lassen. Dadurch erhält die Transition T_1^G Erlaubnis und wird abhängig von ihrer Verteilungsfunktion irgendwann schalten und die Pumpe reparieren. Die Marke an Stelle P_3^d kann nicht erneuert werden, ein erneuter Defekt ist in diesem Modell nicht vorgesehen.

Für dieses Beispiel betrachten wir den Fall, dass nach einer Stunde alle drei Pumpen in Abständen von jeweils 30 Minuten ausfallen. Wir setzen $\phi_d^T(T_1^D) = 3600 \text{ s} = 1 \text{ h}$, $\phi_d^T(T_2^D) = 5400 \text{ s} = 1,5 \text{ h}$ und $\phi_d^T(T_3^D) = 7200 \text{ s} = 2 \text{ h}$.

Für die Reparatur der Pumpen nehmen wir an, dass diese jeweils zwischen einer und zwei Stunden dauert und nehmen eine Gleichverteilung an, mit der kumulierten Verteilungsfunktion:

$$F(x) = \begin{cases} 0 & \text{falls } x < 3600 \text{ s}, \\ 1 & \text{falls } x \geq 3600 \text{ s}, \\ \frac{x-3600 \text{ s}}{3600 \text{ s}} & \text{sonst.} \end{cases} \quad (6.1)$$

Ferner nehmen wir an, dass der Verbraucher einen bestimmten Mindestdruck braucht, der dann gegeben ist, wenn der Füllstand im Wassertank größer als 300 m^3 ist. Wir stellen daher die Bedingung für die Zuverlässigkeit des Modells, die wir im Folgenden testen werden.

6.2 Auswertung

Abbildung 6.2 zeigt das Ergebnis der Simulation mit 10000 Durchläufen, ausgeführt auf einem System mit Windows 8, 16 GB RAM und einem Intel Core i5-4570 x64-basierten Prozessor mit 3,2 GHz. Geprüft wurde die Bedingung über eine maximale Ausführungszeit von 6 Stunden (18000 s) und in Zeitschritten von 1 Minute. Die Simulation dauerte 7,7 Sekunden.

Zu sehen ist, dass nach ca. 2,5 Stunden (9000 s) die Wahrscheinlichkeit, dass der Füllstand von P_1^c größer 300 m^3 ist von 100% abrupt auf ca. 45% abfällt. Zu die-



Abbildung 6.2: Ergebnis der Simulation

sem Zeitpunkt sind alle drei Pumpen bereits ausgefallen. Die erste könnte mit einer Wahrscheinlichkeit von 50% wieder repariert sein, die anderen beiden sind aber noch sicher defekt. Je nachdem, wie schnell die erste Pumpe repariert wird, kann es also sein, dass dadurch die Bedingung noch erfüllt werden kann. Mit fortschreitender Zeit wird es immer wahrscheinlicher, dass die erste oder auch die zweite Pumpe repariert wurden und somit auch, dass die Bedingung erfüllt ist. Nach ca. 3,5 Stunden (12600 s) ist die Wahrscheinlichkeit wieder auf 100% ansteigen. Zu diesem Zeitpunkt sind mindestens die ersten beiden Pumpen wieder repariert und der Füllstand ist in jedem Fall wieder auf über 300 m^3 angestiegen.

Je nachdem wie kritisch die Zuverlässigkeit des Tanks für den Verbraucher ist, kann dieses Ergebnis nun zu weiteren Simulationen oder Konsequenzen führen. Wenn z. B. die Ergebnisse vor dem Hintergrund, dass der Ausfall aller drei Pumpen in Abständen von 30 Minuten unwahrscheinlich ist, hingenommen werden, können weitere Simulationen wahrscheinlichere Fälle, wie z. B. den Ausfall von 2 der 3 Pumpen überprüfen. Wenn die Ergebnisse auf Grund von drastischen Konsequenzen eines Ausfalls nicht hingenommen werden können, könnten Modifikationen, wie z. B. eine Notfallpumpe, in Betracht gezogen werden.

7 Zusammenfassung

Mit dieser Arbeit wurde der Modellformalismus der HPnGs um druckabhängige Transitionen erweitert und es wurde ein Simulator für das neue Modell entwickelt. Dadurch ist es nun möglich kritische Infrastrukturen, z. B. Wasserversorgungen, mit druckabhängigen Ausflüssen auf ihre Zuverlässigkeit zu untersuchen. Angewendet haben wir den Simulator an unserem Fallbeispiel, womit wir gezeigt haben wie das Modell bei kritischen Infrastrukturen eingesetzt werden kann, um ihre Zuverlässigkeit zu testen.

Schwierigkeiten lagen unter anderem darin, dass schon bei einfachen Fällen, wie dem Tank mit zusätzlichem konstantem Zufluss 2.2, die Formeln sehr kompliziert wurden. Dadurch bedingt wurden andere interessante Fälle wie druckabhängige Zuflüsse wegen noch komplizierteren Ketten- oder Ringbeziehungen, z. B. bei mehreren kontinuierlichen Stellen, die durch druckabhängige Transitionen verbunden sind, schnell ausgeschlossen. Die zusätzlichen Möglichkeiten wirken unwesentlich im Vergleich zu der zusätzlichen Komplexität.

Im Hinblick auf die aktuelle Forschung an Modellformalisten wie dem der HPnGs konnte gezeigt werden, dass eine Erweiterung um nicht-lineare Übergänge möglich ist aber schon einfache Erweiterungen mit großen Anpassungen verbunden sein und zu komplizierten Formeln führen können.

In Zukunft könnte versucht werden das in dieser Arbeit beschriebene Modell so zu erweitern, dass druckabhängige Zuflüsse mit den oben beschriebenen Ketten- oder Ringbeziehungen möglich sind. Es sollte jedoch vermieden werden, dass mit jeder Erweiterung die Komplexität des Modells stark ansteigt. Bemühungen sollten daher dahin gehen einen Formalismus zu finden in dem allgemeine nicht-lineare Übergänge behandelt werden können.

Abkürzungsverzeichnis

CDF Kumulative Verteilungsfunktion

DES Diskrete Ereignis Simulation

FST Fluid Survival Tool

HPnG Hybrides Petri-Netz

HPpT Hybrides Petri-Netz mit druckabhängigen Transitionen

JAXB Java Architecture for XML Binding

XML Extensible Markup Language

Abbildungsverzeichnis

2.1	Tank mit druckabhängigem Ausfluss	3
2.2	Tank mit konstantem Zufluss und druckabhängigem Ausfluss	7
2.3	Zeitliche Entwicklung des Volumens (1. Fall)	10
2.4	Zeitliche Entwicklung des Volumens (2. Fall)	10
2.5	Zeitliche Entwicklung des Volumens (3. Fall)	10
3.1	Elemente der erweiterten HPnGs	12
3.2	Beispiel für ein HPPt mit druckabhängiger Transition	14
3.3	Entwicklung der Füllstände (1. Fall)	17
3.4	Entwicklung der Füllstände (2. Fall)	18
3.5	Entwicklung der Füllstände (3. Fall)	19
3.6	Entwicklung der Füllstände (4. Fall)	20
4.1	Fallunterscheidungen für die Ereignisse der kontinuierlichen Stellen	27
4.2	Fallunterscheidungen für die Ereignisse der Kanten (kein druckabhängiger Ausfluss)	29
4.3	Fallunterscheidungen für die Ereignisse der Kanten (druckabhängiger Ausfluss)	31
5.1	model-Paket	37
5.2	view-Paket	38
5.3	controller-Paket	39
5.4	Szenario „Simulation starten“	40
6.1	Fallbeispiel als HPPt Modell	46
6.2	Ergebnis der Simulation	48

Tabellenverzeichnis

4.1	Behandlung der Ereignisse	33
-----	-------------------------------------	----

Auflistungsverzeichnis

5.1	Beispiel für ein HPpT im XML Format	41
-----	---	----

Algorithmenverzeichnis

1	Anpassung der effektiven Volumenströme	21
2	Ereignissimulation	24

Literaturverzeichnis

- [ACG⁺15] Alberto Avritzer, Laura Carnevali, Hamed Ghasemieh, Lucia Happe, Boudewijn R. Haverkort, Anne Koziolk, Daniel Menasche, Anne Remke, Sahra Sedigh Sarvestani, and Enrico Vicario. Survivability evaluation of gas, water and electricity infrastructures. *Electronic Notes in Theoretical Computer Science*, 310, 2015. Proceedings of the Seventh International Workshop on the Practical Application of Stochastic Modelling (PASM). 1
- [DA05] René David and Hassane Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer-Verlag Berlin Heidelberg, 2005. 1
- [Fou15] The Apache Software Foundation. Commons math version 3.5. <http://commons.apache.org/proper/commons-math/>, 2015. 36
- [Gia10] Douglas C. Giancoli. *Physik: Lehr- und Übungsbuch*. Pearson Deutschland GmbH, 2010. 4
- [GR13] Marco Gribaudo and Anne Remke. Hybrid petri nets with general one-shot transition. Vordruck zur Leistungsbewertung, 2013. 1, 12, 13, 14, 16, 21, 22, 23, 26, 29, 32, 33, 35, 36
- [GRH13] Hamed Ghasemieh, Anne Remke, and Boudewijn R. Haverkort. Analysis of a sewage treatment facility using hybrid petri nets. In *7th International Conference on Performance Evaluation Methodologies and Tools, Value-Tools '13, Torino, Italy, December 10-12, 2013*, 2013.
- [Hav98] Boudewijn R. Haverkort. *Performance of Computer Communication Systems*. John Wiley & Sons, Inc., 1998. 43
- [Map13] Maplesoft. Maple version 17.0, 2013. Waterloo ON, Canada. 8
- [Pos13] Björn F. Postema. Fluid survival tool: A model checker for hybrid petri nets. Masterarbeit, University of Twente, Niederlande, 2013. 35

-
- [Ric15] Yann Richet. Jmathplot. <https://github.com/yannrichet/jmathplot>, 2015. 36
- [vdG14] Daniel van de Giessen. Discrete-event simulation of hybrid petri nets with general one-shot transitions. Bachelorarbeit, University of Twente, Niederlande, 2014. 35