

# Zeitlich-parallele Kopplung von diskreten Simulationssystemen mit Virtual-Reality-Systemen

Steffen Straßburger<sup>\*</sup>  
IFF Magdeburg

Thomas Schulze<sup>†</sup>  
University Magdeburg

## Zusammenfassung

Die Kopplung von kommerziellen diskreten Simulationssystemen mit Virtual-Reality(VR)-Systemen eröffnet neue Möglichkeiten im zeitlichen Zusammenspiel des Produkt- und Prozessdesigns, in der Einrichtung von virtuellen Trainingszentren und somit zur Verkürzung von Produktanlaufzeiten. Bisher wird diese Kopplung nur vereinzelt genutzt. Die Ursachen hierfür liegen in der notwendigen Bewahrung der Eigenständigkeit der verwendeten Tools. In diesem Beitrag wird sich auf die Problematik der Synchronisation, als eine der wichtigen Grundaufgaben bei der Kopplung zwischen diskreten Simulations- und VR-Systemen, beschränkt. Bestehende Synchronisationstechniken werden hinsichtlich ihrer Eignung für diese Kopplung untersucht und es wird ein Synchronisationsverfahren auf der Basis selbst-adaptierender Puffergrößen beschrieben.

## 1 Motivation

Die Visualisierung von simulierten Prozessen ist seit vielen Jahren eine erfolgreiche und nützliche Ergänzung der Simulationstechnik. Die meisten kommerziellen Simulationssysteme offerieren die entsprechenden Möglichkeiten, um die modellierten und simulierten Systeme zu visualisieren. Damit hat sich über der Zeit ein breites Spektrum von Zielen der Kopplung von Simulation und Visualisierung entwickelt. Dabei strecken sich die Ziele von der einfachen post-run Visualisierung simulierter Prozesse bis hin zum Trainieren und damit Interagieren in simulationsbasierten Virtual-Reality(VR)-Welten. Die Kopplung von Simulationen mit VR-Welten hat bisher eine starke Ausprägung bei den Trainingssimulatoren gefunden. Beispielhaft sind hier die Flugsimulatoren genannt. Hierbei handelt es sich üblicherweise um spezielle Hard- und Softwaresysteme.

Die Modellierung und Simulation in den Applikationsgebieten Logistik and Produktion erfolgt ausschließlich mit kommerziellen diskreten Simulationssystemen. Visualisierungen der simulierten Prozesse sind fester Bestandteil in den entsprechenden Simulationsprojekten. Kopplungen von Simulation mit VR-Systemen wurden in diesem Anwendungsgebiet bisher nur sporadisch eingesetzt [Kib02], [Ber02]. Eine Kopplung in dieser Applikationswelt ermöglicht neue Zugangsformen von komplexen Interaktionen zwischen dem Modellnutzer und den simulierten Objekten. Damit ergeben sich neue Möglichkeiten im zeitlich parallelen Zusammenspiel des Produkt- mit dem

---

<sup>\*</sup> Fraunhofer Institute for Factory Operation and Automation IFF, Department Virtual Development

<sup>†</sup> Fakultät für Informatik, Wirtschaftsinformatik

Prozessdesign, in der Einrichtung von virtuellen Trainingszentren und somit zur Verkürzung von Produktanlaufzeiten [Dor04].

Die Ursachen für die bisher nur vereinzelte Nutzung liegen in der Eigenständigkeit der verwendeten kommerziellen Tools zur Simulation und Visualisierung und in der damit unzureichenden Interoperabilität zwischen den Systemen. Eine stärkere Verbreitung dieser Kopplung ist nur möglich, wenn bei den zu erarbeitenden Lösungen diese Eigenständigkeit der Tools gewahrt bleibt.

Eine Hauptaufgabe bei der Kopplung von Simulation mit VR ist die Umsetzung der Synchronisation der beiden Tools, d.h. der Zeitfortschritt in beiden Tools muss koordiniert erfolgen. Existierende Vorgehensweisen zur Synchronisation sind für diese Kopplung nur bedingt geeignet, da Nutzeraktionen innerhalb der VR auch den Systemzustand der Simulation betreffen können, die beiden Tools eigenständig und auf unterschiedlichen Plattformen operieren und eine „flüssige“ Bildgenerierung auf der VR-Seite gewährleistet werden muss. Die vielfach bisher in diesen Bereichen zur Anwendung kommende Methode der gepufferten Visualisierungsanweisungen wurde von den Autoren modifiziert, um diesen speziellen Anforderungen zu entsprechen. Diese Methode ist durch eine adaptive Regelung der Puffergröße gekennzeichnet.

Wenn in den folgenden Ausführungen der Begriff Simulation verwendet wird, so ist dieser Begriff auf diskrete Simulationssysteme zu beschränken. Diese Einengung ist zulässig, da diskrete Simulationssysteme in vielen Applikationsgebieten dominieren.

Im folgenden Abschnitt wird eine Klassifizierung der Kopplungsmöglichkeiten zwischen Simulation und Visualisierung vorgenommen, um die Kopplungsform Simulation mit VR entsprechend einzuordnen. Anschließend erfolgt die Spezifikation der Anforderungen aus VR-Anwendungen an die Synchronisation und es werden unterschiedliche Synchronisationsformen diskutiert und hinsichtlich ihrer Eignung für die Kopplung von Simulation mit VR untersucht. Im folgenden Abschnitt wird die Methode der adaptiven Puffergröße erläutert und in den Kontext mit einer Applikation gestellt.

## 2 Klassifikationsaspekte bei der Kopplung

Zur Beschreibung der verschiedenen Kopplungsformen zwischen Simulationen und Simulation wird eine Klassifikation vorgeschlagen (Tabelle 1). Diese Klassifikation beruht auf vier unterschiedlichen Merkmalen, die die Bereiche zeitliche Parallelität, Interaktion, Hardware und Eigenständigkeit des Visualisierungstools abdecken. Für jedes Merkmal werden zwei alternative Ausprägungen verwendet. Diese Klassifizierung wurde vorgenommen, um die Kopplung zwischen Simulation und VR einordnen zu können.

Aus der Menge der theoretisch möglichen Kombinationen der Merkmalsausprägungen werden gegenwärtig im Bereich der diskreten Simulation die in Tabelle 2 aufgelisteten Kopplungsformen genutzt.

Die Kopplung von Simulation mit VR bedingt eine *zeitliche Parallelität* (Concurrent) zwischen der Simulation und der Visualisierung, denn die Simulation beeinflusst die Visualisierung und die Nutzeraktionen beeinflussen über das VR-System die Simulation. In [Dor04] wird eine Kopplung zwischen einer diskreten Simulation und einem VR-System beschrieben, aber eine Beeinflussung der Simulation von Seiten des Nutzers in der Cave ist nicht möglich.

Tabelle 1: Klassifikationsmerkmale und ihre Ausprägung

<b>Merkmal</b>	<b>Ausprägung</b>	
<b>Zeitliche Parallelität</b> Zeitliche Parallelität zwischen Simulation und Visualisierung	<b>Concurrent</b> Simulation und Visualisierung laufen zeitlich parallel	<b>Post-run</b> Die Visualisierung läuft zeitlich nach der Simulation
<b>Interaktion</b> Interaktionen zwischen Simulation und Visualisierung	<b>Bidirektional</b> Simulation und Visualisierung reagieren gegenseitig auf Befehle des anderen Tools	<b>Unidirektional</b> Nur die Visualisierung reagiert auf Befehle der Simulation
<b>Hardware-Plattform</b> Hardware-Plattform auf denen die Simulation und die Visualisierung operieren	<b>Monolithisch/homogen</b> Simulation und Visualisierung laufen auf einer Plattform	<b>Verteilt</b> Simulation und Visualisierung operieren auf unterschiedlichen Hardware-Plattformen
<b>Eigenständigkeit des Visualisierungstools</b>	<b>Integriert</b> Visualisierungstool ist in das Simulationstool integriert	<b>Extern</b> Visualisierungstool arbeitet unabhängig vom Simulationstool

Ein weiteres entscheidendes Merkmal für die Kopplung zwischen Simulation und VR ist die bidirektionale Richtung der *Interaktion*. Die Arbeiten von [Fra04] und [Ber02] weisen dieses Merkmal auf, doch ist der Umfang der Interaktionen von der VR-Seite aus stark eingeschränkt.

Tabelle 2: Mengen von Ausprägungsmerkmalen mit Beispielen zur Kopplung

<b>Ausprägungsmerkmale</b>	<b>Beispiele</b>
Post-run / Unidirektional / Monolithisch / Extern	Visualisierungstool Proof Animation [Wol04]
Post-run / Unidirektional / Verteilt / Extern	Kopplung von SLX-Simulationen mit Visualisierungen in einer Cave [Dor04]
Concurrent / Unidirektional / Monolithisch / Integriert	Visualisierungen in den Tools eMPlant [eMP04] und ARENA [ARE04]
Concurrent / Unidirektional / Monolithisch / Extern	Visualisierungstool Concurrent Proof [Wol04]
Concurrent / Bidirektional / Verteilt / Extern	Kopplung von eMPlant-Simulationen mit VR [Fra04], [Ber02]
Concurrent / Bidirektional / Monolithisch / {Integriert , Extern}	Trainingssimulatoren zur Pilotenausbildung

Die Simulationssysteme operieren überwiegend auf Windows-basierten PC-Plattformen und die VR-Systeme residieren mehrheitlich auf Unix- bzw. Linux-basierten

Workstations. Aus diesem Grund ist die zu untersuchende Kopplung durch eine verteilte Arbeitsweise auf inhomogenen *Hardware-Plattformen* gekennzeichnet.

Simulations- und VR-Systeme sind komplexe eigenständige Softwaresysteme, die bei einer Kopplung auch ihre *Eigenständigkeit* erhalten müssen. Für eine erfolgreiche Kopplung von Simulation mit VR-Systemen ist die Beibehaltung der Eigenständigkeit der Systeme eine entscheidende Voraussetzung. Existierende VR- oder Simulationssysteme müssen mit anderen existierenden VR- oder Simulationssystemen gekoppelt werden. Dieser Umstand bedingt eine Eigenständigkeit der verwendeten Tools.

Damit sind allgemeine Merkmale spezifiziert, die an eine Kopplung von diskreter Simulation und VR gestellt werden. Der folgende Abschnitt beschreibt nun spezielle Anforderungen.

### **3 Spezielle Anforderungen an die Kopplung aus VR-Anwendungen**

Unter Virtual-Reality versteht man eine vom Computer generierte 3D-Umgebung, in die man als Nutzer eintauchen kann und diese künstliche Welt als Realität wahrnehmen kann. Dabei ist man als Benutzer Bestandteil dieser Welt und man kann mit ihr interagieren. (nach [VDI03]).

Es existieren Lösungen, die für sich in Anspruch nehmen, Simulation mit VR zu verbinden [Whi02],[Kib02]. Diese Anwendungen basieren auf der Nutzung von 2D-Interfaces am Computer, wie Monitor, Keyboard und Maus. Mit diesen Features werden 3D-Modelle der simulierten Umgebung betrachtet. Die verwendeten Interaktionen unterscheiden sich nicht grundsätzlich von den Standard-Interaktionen in einer 2D-Repräsentation der simulierten Umgebung [Dor04]. Erste Ansätze zur Kopplung von Simulationen mit VR-Systemen im produktionstechnischen Anwendungsfeld werden in [Dor04], [Ber02] und [Fra04] beschrieben.

Bei einer Kopplung von Simulationen mit VR-Anwendungen sind drei Aufgaben zu lösen. Das *Mapping* zwischen Simulations- und VR-Modell und der *Datenaustausch* zwischen den beiden Modellen sind die ersten beiden Aufgaben. Diese Aufgaben können mit den bekannten Vorgehensweisen gelöst werden.

Die dritte Aufgabe ist die *Synchronisation* zwischen den beiden zeitbehafteten Systemen. An die Synchronisation werden durch die Berücksichtigung von Interaktionen in beiden Richtungen, besondere Anforderungen gestellt. Wünschenswert ist eine Identität zwischen Simulations- und Visualisierungszeit, damit Interaktionen zu dem Zeitpunkt, an dem sie ausgelöst wurden auch bei dem anderen System berücksichtigt werden können. Durch den Nutzer vorgenommene Veränderungen in der VR-Welt zum Zeitpunkt  $t$  müssen sich auch in der Simulationswelt zum Zeitpunkt  $t$  niederschlagen.

Eine Identität der Zeiten in den beiden kann nicht erreicht werden. Es muss eine Ungleichheit zwischen den Zeiten akzeptiert werden. Im Allgemeinen ist die Simulationszeit größer als die Visualisierungszeit, da die Simulation die Visualisierung hauptsächlich beeinflusst. Bedingt durch die bidirektionalen Interaktionen, muss das Simulationsmodell in der Lage sein, auf Interaktionen (externe Ereignisse) zu reagieren, deren Zeitstempel kleiner oder gleich der aktuellen Zeit ist.

Die Simulationszeit kann im Simulator angehalten werden, wenn sich die beiden Zeiten zu weit voneinander entfernen. Im Gegensatz dazu schreitet die Visualisierungszeit proportional zur realen Zeit voran und ein „Anhalten“ der Visualisierungszeit ist nicht möglich.

Es ist für VR-Anwendungen nicht akzeptabel, wenn die Simulationszeit kleiner als die Visualisierungszeit ist. „Rücksprünge“ im Visualisierungsmodell werden vom Nutzer nicht akzeptiert.

Für die Kopplung von Simulation mit VR sind Synchronisationsverfahren zu verwenden, die keine oder nur eine geringe Zeitdifferenz zwischen den beiden Systemen gewährleisten. Die Größe dieser Differenz muss abhängig von den Übertragungszeiten im Netzwerk und abhängig von dem Faktor der zeitlichen Parallelität der Visualisierungszeit mit der Realzeit gewählt werden können. Eine Visualisierung im Zeitraffertempo lässt den Nutzer eine größere Differenz zwischen den beiden Zeiten nicht erkennen. Im Gegensatz dazu, muss bei einer Visualisierung im Zeitlupentempo, eine kleinere Differenz gewährleistet werden.

Im folgenden Abschnitt werden bestehende Synchronisationsverfahren hinsichtlich ihrer Verwendung für die Kopplung von Simulation mit VR untersucht.

## **4 Synchronisationsprinzipien für die zeitlich parallele Kopplung von Simulation und Visualisierung**

Synchronisation ist eine sehr wichtige Eigenschaft für das Zusammenarbeiten von zwei zeitbehafteten Systemen, die eigene Paradigmen für den Zeitfortschritt haben. Die diskrete ereignisorientierte Simulation versucht im Allgemeinen so schnell wie möglich eine Zeitfortschreibung zu erzielen, wobei die Zeit in nicht-äquidistanten Zeitabschnitten voranschreitet. Im Gegensatz dazu ist die Visualisierung durch einen quasi-kontinuierlichen Zeitfortschritt in äquidistanten und Echtzeit-proportionalen Zeitabschnitten gekennzeichnet. Das Ziel der Synchronisation ist ein quasi gleicher Zeitwert in beiden Systemen.

Bezeichnet man mit  $T$  die reale Zeit, mit  $t_{SIMU}(T)$  die Simulationszeit zum realen Zeitpunkt  $T$ , mit  $t_{VISU}(T)$  die Visualisierungszeit zum realen Zeitpunkt  $T$  und mit  $\Delta t(T)$  die Differenz zwischen Simulations- und Visualisierungszeit zum realen Zeitpunkt  $T$ , so gilt die folgende Gleichung:

$$t_{SIMU}(T) = t_{VISU}(T) + \Delta t(T)$$

Die Visualisierungszeit  $t_{VISU}(T)$  lässt sich nach der folgenden Gleichung berechnen, wobei  $c$  ein Proportionalitätsfaktor der Visualisierungszeit zur realen Zeit  $T$  ist

$$t_{VISU}(T) \approx c * T$$

Zur Synchronisation von zeitbehafteten Komponenten in verteilten Systemen wurden in den letzten Jahren Methoden entwickelt. Hierzu zählen spezielle Methoden für die verteilte Simulation, wenn die Komponenten nur Simulationen sind, und mehr allgemeine

Methoden, wie sie zur Kopplung von heterogenen Komponenten in der High Level Architecture HLA verwendet werden.

Die existierenden Methoden werden wie folgt klassifiziert:

- Synchronisation durch Echtzeit

Die Zeit schreitet in beiden Systemen gleichmäßig vorwärts (siehe Abbildung 1). Vielfach arbeiten dann die beiden Systeme Echtzeit-proportional. In diesem Fall strebt  $\Delta t$  gegen Null. Dieser Ansatz ist typisch für sehr enge Kopplungen zwischen Simulation und Visualisierung, wenn zwischen den beiden Systemen nur unidirektionale Interaktionen getauscht werden. Dieser Ansatz ist für die Kopplung von Simulationen mit VR nicht geeignet, da die Koordinierung der Echtzeit-proportionalen Zeitfortschritts im Simulationssystem zusätzlichen Overhead im Simulator bedingt.

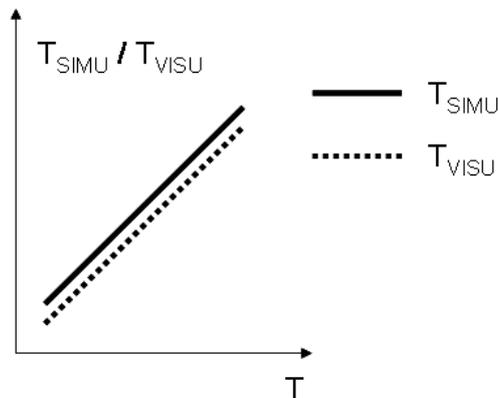


Abbildung 1: Zeitlicher Verlauf bei der Echtzeit-Synchronisation

- Synchronisation durch eine zentrale Instanz

Auch bei dieser Form schreitet die Zeit in beiden Systemen quasi-gleichmäßig vorwärts. Der Zeitfortschritt wird bei einer zentralen Instanz beantragt und diese Instanz gewährt den einzelnen Systemen einen gemeinsamen Zeitfortschritt. Die Kommunikation zur Synchronisation zwischen Simulation und Visualisierung erfolgt nur über diese zentrale Instanz. In diesem Fall strebt  $\Delta t$  gegen Null. Die von der High Level Architecture (HLA) offerierten Synchronisationsmethoden fallen in diese Kategorie [Rit98]. Im speziellen Fällen kann auch auf eine zentrale Instanz verzichtet werden, dann synchronisieren sich die Systeme direkt auf der Basis von bekannten Synchronisationsalgorithmen aus der verteilten Simulation.

Vorteilhaft bei diesem Ansatz ist die identische Zeit in beiden Systemen. So werden Interaktionen aus der Visualisierung ohne Verletzung der Kausalität in der Simulation berücksichtigt. Gegen die Verwendung dieses Verfahrens bei der Kopplung von Simulation und VR sprechen der Aufwand zur Implementierung dieser Synchronisationsalgorithmen in bestehende kommerzielle Simulations- und VR-Systeme, und die Möglichkeit des „Wartens“ des VR-Systems auf die Freigabe des eigenen Zeitfortschritts durch das Simulationssystem. Dieses Warten wird vom Nutzer in einer

VR-Umgebung nicht akzeptiert. Aus diesen Gründen ist dieser Ansatz in der vorliegenden Form für die Kopplung von Simulation und VR nur bedingt geeignet.

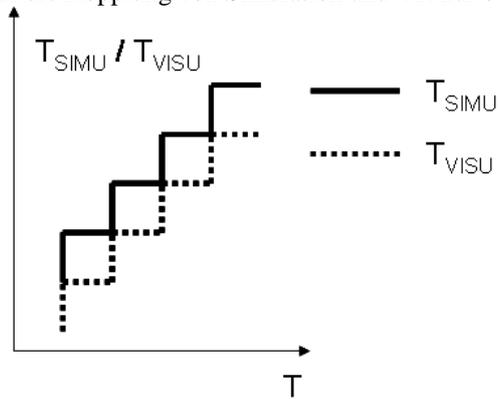


Abbildung 2 : Zeitlicher Verlauf bei der Synchronisation durch eine zentrale Instanz

- Synchronisation durch Pufferung

Die Simulation sendet zeitbehaftete Visualisierungsbefehle an die Visualisierung. Diese Befehle werden in einem Puffer gespeichert, und die Visualisierung liest diese Befehle aus dem Puffer aus. Die Schreibaufufe in den Puffer sind synchrone Aufrufe. Wenn ein Aufruf infolge eines gefüllten Puffers nicht ausgeführt werden kann, dann „wartet“ die Simulation solange, bis der aktuelle Pufferinhalt kleiner als die maximale Pufferkapazität ist. Die Simulationszeit ist immer größer als die Visualisierungszeit, d.h.  $\Delta t(T)$  ist immer größer Null. Dieses Verfahren wird beispielsweise bei der Kopplung des Visualisierungstools Concurrent Proof [Wol04] verwendet.

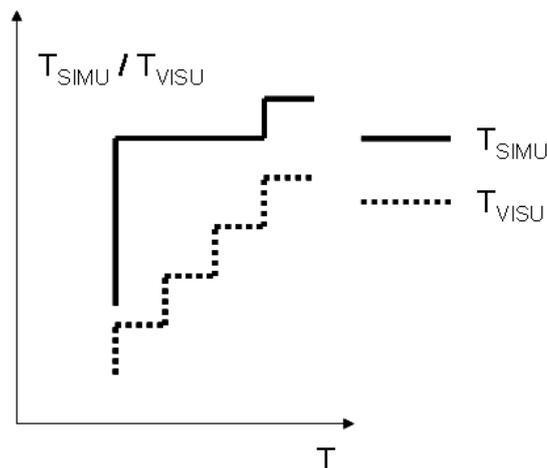


Abbildung 3: Zeitlicher Verlauf bei der klassischen Puffer-Synchronisation



die dann das Simulationsmodell beeinflussen. Ein wesentliches Merkmal der klassischen Pufferstrategie ist, dass der Puffer Animations-Kommandos mit einer beliebig weiten Zeitspanne enthält. Der Puffer wird nicht durch die Zeit begrenzt, sondern nur durch die Anzahl von Kommandos. Die gepufferten Kommandos können somit eine Zeitspanne vom Millisekundenbereich bis hin zu Tagen umfassen.

Für die Anwendung zur Kopplung von Simulation mit VR-Systemen kann dieser klassische Ansatz nicht verwendet werden. Bei der Kopplung mit VR-Systemen müssen die Interaktionen aus VR-Welt hinreichend genau im Simulator berücksichtigt werden. Somit ist eine Pufferstrategie erforderlich, die zum einen Visualisierungskommandos innerhalb einer kleinen Zeitspanne enthält, zum anderen aber auch eine hinreichende Anzahl von Visualisierungskommandos zur Umsetzung einer flüssigen und kontinuierlichen Visualisierung enthält.

Aus diesen Überlegungen heraus wurde die Strategie einer selbst-adaptiven Puffergröße entwickelt. Kernpunkte sind eine konfigurierbare Puffergröße in Zeiteinheiten und die maximal erlaubte Zeitdifferenz zwischen zwei Visualisierungskommandos. Die Zeiteinheit entspricht der Zeiteinheit in der Simulation bzw. in der VR. Beim Start der Kopplung wird eine initiale Puffergröße festgelegt. Das Grundprinzip dieser Strategie ist in Abbildung 5 aufgezeigt. Die initiale Puffergröße ist auf 5 Zeiteinheiten und die maximale Zeitdifferenz ist auf 3 Zeiteinheiten festgelegt. Die Zeiteinheit in den beiden Systemen entspricht Sekunden.

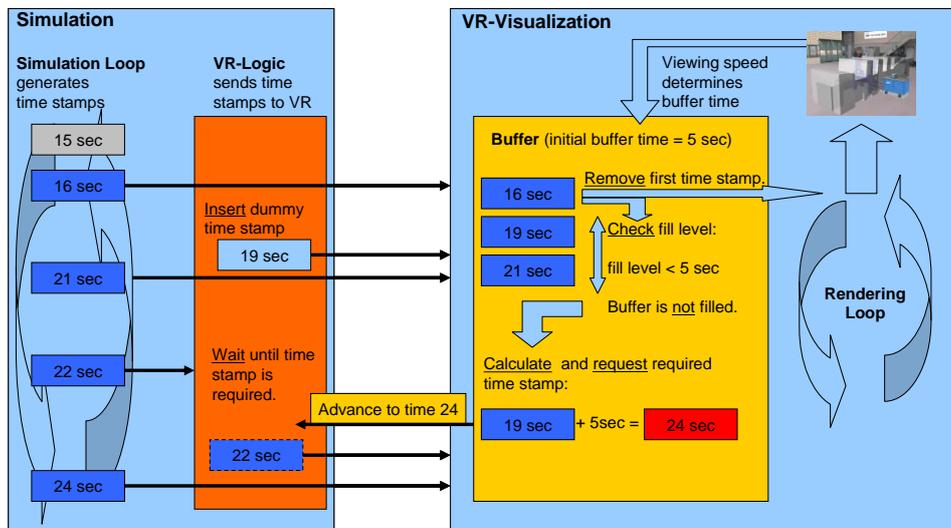


Abbildung 5: Adaptiver Pufferalgorithmus

Die Puffergröße ist einstellbar in Abhängigkeit von den Latenzzeiten im Netzwerk zur Datenübertragung und der Leistungsfähigkeit des Visualisierungstools. Die maximale Zeitdifferenz beeinflusst die Generierung von Visualisierungskommandos. Eine „dummy“ Anweisung wird in den Anweisungsstrom eingefügt, wenn die Zeitdifferenz zwischen zwei realen Visualisierungskommandos diese maximale Zeitdifferenz

überschreitet. Somit wird ein konstanter Fluss von Visualisierungskommandos gewährleistet, auch wenn die Ereigniszeitpunkte in der Simulation weit auseinander liegen.

Der aktuell Fortschritt der Simulationszeit wird durch die Visualisierung dominiert, wie in Abbildung 6 aufgezeigt. Die Visualisierung verfolgt die Strategie, dass der Puffer immer mit dem nächsten Visualisierungskommando für die eingestellte Puffergröße gefüllt ist. Das bedeutet, dass bei konstanter Puffergröße in Zeiteinheiten, sich eine unterschiedliche Anzahl von Kommandos im Puffer befinden kann. Wenn die Visualisierung ein Kommando aus dem Puffer entfernt, wird die restliche Zeitspanne im Puffer berechnet. Ergibt sich ein Wert, der kleiner ist als die Puffergröße, wird an die Simulation ein Anforderungskommando gesendet („Advance to time ...“). Dieses Kommando enthält den notwendigen Zeitfortschritt.

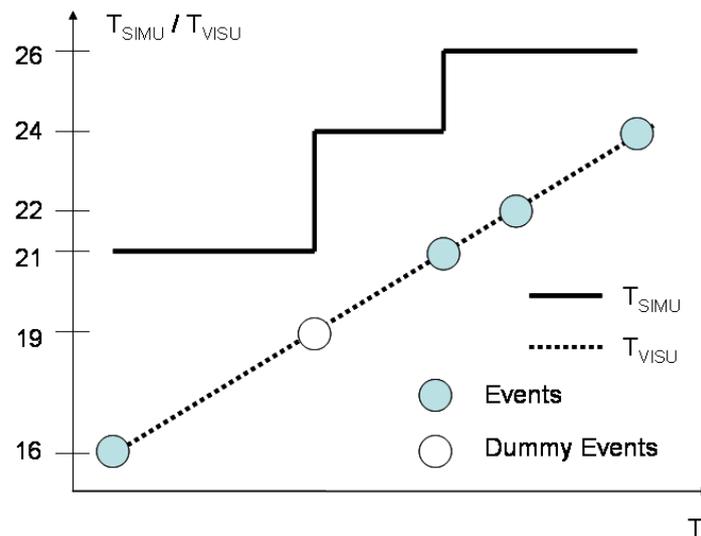


Abbildung 6 : Zeitverlauf beim adaptierenden Algorithmus

Wird beispielsweise zum Zeitpunkt  $t_{VISU}(T)=20$  im VR-System eine Interaktion mit Rückwirkung auf die Simulation ausgelöst, so beträgt die aktuelle Zeit in der Simulation  $t_{SIMU}(T)=24$ . Die Berücksichtigung der Interaktion in der Simulation erfolgt quasi ohne Zeitverlust an realer Zeit  $T$ , aber die Zeitdifferenz beträgt 4 Zeiteinheiten. Wurde als gemeinsame Zeiteinheit Sekunden festgelegt, so reagiert die Simulation mit einer Verzögerung von 4 Sekunden. Eine Verzögerung in dieser Größenordnung wird für die vorgesehenen Applikationen akzeptiert.

Die adaptive Pufferstrategie regelt die Puffergröße in Abhängigkeit von der Visualisierungsgeschwindigkeit. Bei einem Anstieg der Visualisierungsgeschwindigkeit wächst die Puffergröße selbständig mit. Wird beispielsweise ausgehend von einer Puffergröße von 5 Sekunden die Visualisierungsgeschwindigkeit verdoppelt, so wird eine Puffergröße von 10 Sekunden eingestellt. Andererseits wird bei sinkender Visualisierungs-

geschwindigkeit die Puffergröße verringert. Im Fall einer Zeitlupenbetrachtung mit halber realer Zeit wird die Puffergröße auf 2,5 Sekunden korrigiert.

Diese Vorgehensweise wird deshalb auch als Methode der selbst-adaptierenden Puffergröße bezeichnet. Die Puffergröße stellt sich selbständig auf veränderte Bedingungen durch Netzwerkverzögerungen und Visualisierungsgeschwindigkeiten ein. Dadurch wird die Verzögerung zwischen Simulation und VR auf ein vertretbares Minimum begrenzt unter Bewahrung eines hohen Grades der Interaktivität innerhalb der VR-Welt.

## 6 Ausblick und Zusammenfassung

In einem Pilotprojekt wurde dieses Verfahren erfolgreich eingesetzt. Für eine Applikation aus dem Bereich Produktion und Logistik wurde ein SLX-basiertes Simulationsmodell auf einem PC mit einer VR-Welt in einer Cave gekoppelt. Das Simulationsmodell ist in der Lage, auf ausgewählte Interaktionen des Nutzers in der Cave zu reagieren. Die nicht vermeidbaren Konsistenzprobleme zwischen den beiden Welten wurden durch die Wahl einer initialen Puffergröße von 5 Sekunden beschränkt. Der dadurch bedingte systematische Fehler bei der Nutzerinteraktion mit der Simulation hatte vernachlässigbare Auswirkungen auf die Simulationsergebnisse.

Der adaptierende Pufferalgorithmus auf der Seite der Simulation wurde gekapselt, so dass dieses Verfahren nicht an das entwickelte Modell gebunden ist. Auf der Cave-Seite wurden die notwendigen Teile in Java implementiert, so dass auch hier eine Portierung in andere VR-Systeme möglich ist.

Die nächsten Forschungsarbeiten in diesem Projekt fokussieren zum einen auf eine Portierung dieser Vorgehensweise zur Nutzung in anderen kommerziellen Simulationstools und zur Implementierung dieser Pufferstrategie in weiteren VR-Systemen. Ein anderer Schwerpunkt ergibt sich aus der Erweiterung des Umfangs der zulässigen Interaktionen aus der VR-Welt heraus, die dann im Simulator berücksichtigt werden müssen.

Zukünftige Untersuchungen könnten außerdem die Nutzung von State-Saving und Rollback-Fähigkeiten von SLX zur Lösung von Konsistenzkonflikten zwischen VR und Simulation beinhalten.

Die vorgestellte Methode zur selbst-adaptierenden Puffergröße hat sich als pragmatischer und robuster Ansatz zur interaktiven zeitlich-parallelen Kopplung von Simulation und VR erwiesen und kann die Basis für eine breite Anwendung der Kopplung von kommerziellen Simulationstools mit VR in der Praxis bilden.

## 7 Literatur

[ARE04] Rockwell Software Inc. URL: <http://www.arenasimulation.com/>

[Ber02] J. Bergbauer. *Entwicklung eines Systems zur interaktiven Simulation von Produktionssystemen in einer virtuellen Umgebung*. Dissertation Universität Clausthal 2002.

[Dor04] Denis V. Dorozhkin, Marco Lemessi, Gordon D. Rehn and Judy M. Vance. *Integrating Operations Simulation Results with an Immersive Virtual Reality*

*Environment*. Proceedings of the 2004 Winter Simulation Conference, R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds. pp. 1713-1719

- [eMP04] Tecnomatix Technologies Ltd. URL: <http://www.technomatix.com>
- [Fra04] Ronny Franke. *Kopplung von diskreter Simulation und interaktiver 3D-Visualisierung*. Diplomarbeit an der Fakultät für Informatik, Universität Magdeburg. 2004.
- [Kib02] Deogratias Kibira and Chuck Mc Lean. *Virtual Reality Simulation of Mechanical Assembly Production Line*. Proceedings of the 2002 Winter Simulation Conference, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charness, eds. pps 1130-1137.
- [Rit98] Klaus-Christoph Ritter, U. Klein, S. Straßburger, M. Diessner. *Web-basierte Animation verteilter Simulationen auf Basis der High Level Architecture (HLA)*. In Preim, B. and P. Lorenz (Ed.), Proceedings der Tagung Simulation und Visualisierung. 5.-6.3.1998, Magdeburg. SCS European Publishing House.pp. 41-52.
- [VDI03] VDI Richtlinie3633:Part 11 203. *Simulation und Visualisierung*.
- [Whi02] L. Withman, Madhavan, V., Malzahn, D. and Twomey, J. *Virtual Reality Model to Aid Case Learning*. In Proceedings of the Industrial Engineering Research Conference, May 2002.
- [Wol04] Proof Animation. URL: <http://www.wolverinesoftware.com/>