

Reale Systeme im virtuellen Labor, Erfahrungen mit dem Einsatz von Remote-Praktikumsversuchen in der Lehre der FernUniversität Hagen

kevi Workshop „Referenzmodelle netzbasierten Lehrens und Lernens
- virtuelle Komponenten der Präsenzlehre“

Andreas Bischoff, Christof Röhrig

Univ. Prof. Dr-Ing. H. Hoyer
Lehrgebiet Prozesssteuerung
und Regelungstechnik
Fachbereich ET+IT
FernUniversität Hagen
Kontakt: andreas.bischoff@fernuni-hagen.de

Teleoperation of Laboratory Experiments

Andreas Bischoff and Christof Röhrig, FernUniversität Hagen. Laboratory experiments are important components in natural science and engineering education. Their aim is the application and verification of theory on real systems and to learn the handling of technical plants. Teleoperation of laboratory experiments helps to overcome restrictions in time and place. The paper describes a platform-independent approach to teleoperation of laboratory experiments via Internet which is developed and tested at the University of Hagen in cooperation with the universities of Bochum and Dortmund.

1 Einleitung

Üblicherweise werden an Hochschulen mit großem Aufwand an Geld und Zeit ähnliche Laborexperimente aufgebaut. Zugleich ist oftmals durch die zunehmende Spezialisierung keine vollständige Abdeckung einzelner Fächer in einem Labor möglich. Hinzu kommt, dass diese Versuche Studierenden nur eingeschränkt und ausschließlich während der eigentlichen Versuchsdurchführung zur Verfügung stehen. An der FernUniversität Hagen, die als einzige Universität im deutschsprachigen Raum ausschließlich über Fernlehre unterrichtet, besteht ferner die Sondersituation, dass etwa 80% der Studierenden berufstätig sind und somit überwiegend in den Abendstunden

und am Wochenende das Studienangebot nachfragen. Laborversuche waren bislang in die Fernlehre schwer zu integrieren und für die Studierenden immer mit Reiseaufwand verbunden. Eine Lösung dieses Problems ist die Simulation der Laborexperimente und die Darstellung in virtueller Realität [2]. Die hier vorgestellte Alternative liegt in der Teleoperation von realen Laborexperimenten über das Internet. Mitte der neunziger Jahre wurde von den ersten Implementierungen fernbedienter regelungstechnischer Labore berichtet [3] [4]. In [5] wird der Bedarf an derartigen teleoperierten Laboren in den USA analysiert. Dabei werden auch die Gesichtspunkte der Finanzierung und Vermarktung berücksichtigt.

Am Lehrgebiet für Prozesssteuerung und Regelungstechnik der FernUniversität Hagen wurde ein fernbedientes regelungstechnisches Labor entwickelt [1]. Das Projekt „Reale Systeme im virtuellen Labor“ wurde in Kooperation mit den Lehrstühlen für Elektrische Steuerung und Regelung der Universitäten Bochum (Prof. Unbehauen, Prof. Schmid) und Dortmund (Prof. Kiendl) durchgeführt. Das Projekt wurde im Rahmen der Ausschreibung „Neue Medien in der Hochschullehre“ vom Kompetenznetzwerk Universitätsverbund MultiMedia des Landes Nordrhein-Westfalen gefördert. Es stehen derzeit drei regelungstechnische Versuche zur hochschulübergreifenden Nutzung zur Verfügung. Es wurde besonders darauf Wert gelegt, eine technische Grundstruktur zu entwickeln, die es gestattet, weitere teleoperierte Laborexperimente relativ einfach zu realisieren. Das Experiment der Ruhr-Universität Bochum behandelt ein optisches zweiachsiges Nachlaufsystem. Für eine weiterführende Beschreibung des Experiments wird auf [6] verwiesen. Von der Universität Dortmund wird das Experiment „Fuzzy-Logik basierte Prognose von Störungen am Beispiel eines 3-Tank Systems“ angeboten. In anderen Bundesländern sind weitere Projekte dieser Art initiiert worden [7], [8].

2 Anforderungen an teleoperierte Laborexperimente

Die sinnvolle Durchführung von Laborexperimenten über das Internet stellt nachstehende Anforderungen an die Realisierung:

- *Teleaktion:* Das Experiment muss mit allen wesentlichen Handlungen durchführbar sein. Im einfachsten Fall sind dazu Parameteränderungen des Regelalgorithmus und Änderungen des Sollwertes ausreichend. Sind während des Experiments Manipulationen am Aufbau notwendig, müssen entsprechende Aktoren vorgesehen werden, die diese Aktionen durchführen. Die Durchführung des Experiments sollte möglichst einfach und intuitiv wie bei einem lokal durchgeführten Experiment ablaufen. Lange

Einarbeitungszeiten und umständliche Bedienung verringern die Akzeptanz beim Benutzer. Die Reaktion auf eine Aktion sollte kurz sein, um Fehlbedienungen durch Wiederholung von Aktionen zu vermeiden.

- *Telepräsenz:* Die Studierenden sollen durch Übertragung von sensorischen Informationen das Gefühl bekommen, im entfernten Labor präsent zu sein. Um dem Benutzer das Gefühl eines echten Laborexperimentes zu vermitteln, ist es notwendig, möglichst viele Eindrücke aus dem Labor zu übertragen. Mit den heute üblichen Multimedia-PCs sind nur visuelle und akustische Informationen darstellbar. Diese geben bei vielen in der Regelungstechnik eingesetzten Experimenten einen hinreichend guten Eindruck vom realen Experiment. Es ist aber auch denkbar, weitere Sinnesmodalitäten, wie die des haptischen Erfassens, d.h. des kinästhetischen und taktilen Fühlens, zu übertragen. Dies ist jedoch nur mit zusätzlichem Hardwareaufwand beim Bediener möglich. In [9] wird ein PC-basiertes System vorgestellt, das neben visuellen und akustischen auch kinästhetische und taktile Informationen darstellen kann.
- *Aufzeichnung und Übertragung von Messdaten:* Zur Analyse der Ergebnisse des Experiments müssen alle wichtigen Daten der Regelung und der Strecke aufgezeichnet werden. Die aufgezeichneten Messdaten können dann wahlweise zur Laufzeit des Experiments oder danach übertragen werden. Eine Übertragung und Darstellung der Messwerte in Echtzeit während der Laufzeit des Experiments ist nur bei entsprechender zur Verfügung stehender Bandbreite sinnvoll. Die Übertragung der Messdaten nach dem Experiment entkoppelt die Datenrate der Aufzeichnung von der Übertragung und ist bei langsamen Verbindungen vorzuziehen. Eine Speicherung der Messwerte in einem plattformunabhängigen Format ermöglicht eine spätere Offline-Analyse mit leistungsfähigen Programmen von verschiedenen Fremdanbietern wie z.B. MatlabTM.
- *Sicherheit:* Eine wichtige Anforderung auf der Serverseite ist die Sicherheit des Experimentaufbaues und des Server-Rechners. Der Versuchsaufbau muss vor jeglicher Benutzeraktion geschützt werden, die eine Beschädigung hervorrufen könnte. Aus diesem Grund müssen alle Kommandos, die an die Fahrzeugregelung übermittelt werden, analysiert werden, um gefährliche Einstellungen zu verhindern.
- *Synchrone Kommunikation:* Typischerweise werden Laborpraktika in Gruppen durchgeführt. Dies fördert die Kompetenz zur Problemlösung im Team, welche eine wesentliche Anforderung an die Fähigkeiten eines Ingenieurs ist. Um die Teamfähigkeit der Studierenden zu fördern, soll eine

Bearbeitung der Praktikumsversuche im Team ermöglicht werden. Die Einführung solcher kollaborativen Tele-Experimente erfordert technische Lösungen beim Simultanzugriff auf das Experiment und bei der Kommunikation. Um fernbedienbare Experimente einer Gruppe von Studierenden zur Verfügung zu stellen sind synchrone Kommunikationswege erforderlich. Ein Versuchsbetreuer muss in der Lage sein Details des Versuchsaufbaues und der Durchführung zu erläutern. Außerdem ist eine Zusammenarbeit der verteilten Gruppenmitglieder erforderlich und erwünscht.

- *Skalierbarkeit der übertragenen Datenrate:* Die Studierenden der FernUniversität nutzen unterschiedliche Zugänge zum Internet. Diese reichen von der Benutzung einfacher Modems, über ISDN, bis zur Nutzung eines Rechners mit Ethernetanschluss (Computerraum einer Hochschule oder eines Studienzentrums). Die zur Verfügung stehende Bandbreite sollte möglichst effektiv zur Erzeugung von Telepräsenz genutzt werden. Zugleich muss sichergestellt werden, dass auch bei stark eingeschränkter Bandbreite eine Versuchsdurchführung möglich ist, wenngleich dies eine reduzierte Qualität der Telepräsenz bedeutet. Dabei kann eine augmentierte Realität hilfreich sein, bei der das Videobild, z.B. durch virtuelle Realität, ergänzt wird. Die Video- und Audiodaten sollten unter Berücksichtigung physiologischer Eigenschaften der menschlichen Sinne komprimiert werden, um die begrenzte Bandbreite besser ausnutzen zu können.
- *Plattformunabhängigkeit der Client-Software:* Die Studierenden der FernUniversität Hagen nutzen unterschiedliche Soft- und Hardwareplattformen, wobei PCs mit dem Betriebssystem Windows den weitaus größten Teil einnehmen. Es erscheint sinnvoll, die Benutzung nicht auf Windows-PCs zu beschränken, sondern auch andere Betriebssysteme und Hardwareplattformen zu unterstützen, um die Benutzer anderer Systeme nicht auszuschließen und um bei zukünftigen Entwicklungen nicht eingeschränkt zu sein.
- *Keine Kosten durch zusätzliche Hardware oder Software:* Die Laborexperimente sollen mit der üblichen Computerausstattung bei den Studierenden durchführbar sein. Nach Möglichkeit soll die Installation von zusätzlichen Softwarekomponenten vermieden werden. Wenn zusätzliche Software notwendig ist, sollte die Installation einfach und ohne spezielle Betriebssystemkenntnisse durchführbar sein. Sicherheit des Versuchsaufbaus: Um die Sicherheit der Versuchsanlage nicht zu gefährden, sind unzulässige Aktionen, die zu gefährlichen Anlagenzuständen führen können, abzufangen. Die Versuchsanlage muss sich nach einem Experiment

immer wieder automatisch in einen reproduzierbaren Anfangszustand überführen lassen. Zugriffsverwaltung: Nur autorisierte Nutzer sollten Zugang zum Experiment erhalten. Weiterhin ist es notwendig, den exklusiven Zugriff eines Nutzers zu sichern, um einen ungestörten Experimentablauf zu gewährleisten. Dies ist z.B. möglich, indem potentielle Nutzer in eine Warteschlange eingereiht werden und nach dem FIFO-Prinzip Zugang bekommen. Besser ist jedoch eine Terminverwaltung, bei der Nutzer Versuchszeit an freigegebenen Terminen selbst reservieren können. Einfache Erweiterbarkeit des Systems: Um Anpassungen am vorhandenen System einfach durchführen oder um weitere Experimente hinzufügen zu können, sollte die Software modular konzipiert werden. Eine objektorientierte komponentenbasierte Struktur ist dabei hilfreich. Die einzelnen Komponenten können dann mittels Vererbungsmechanismus angepasst und das gesamte System erweitert werden.

3 Systemarchitektur

3.1 Entwurfsidee

Grundlegende Entwurfsidee ist die Nutzung des Internets als Kommunikationsstruktur und die Nutzung eines Web-Browsers als Benutzerinterface. Der Web-Browser stellt ein Werkzeug zur Verfügung, mit dem sowohl Information ausgetauscht werden kann als auch Software als Java-Applet ablauffähig ist. Ein Kommunikationsserver, der einen Web-Server und einen Applikationsserver enthält, ist die Schnittstelle zwischen dem Benutzer auf der einen und dem Experiment auf der anderen Seite. Die Software wurde als Client/Server-Architektur entworfen und ist in der Programmiersprache Java™ [10] (im folgenden Java) implementiert. Die Internet-Protokolle dienen zum standardisierten Austausch der Daten zwischen Client und Server. Die Implementierung von Client und Server in Java ermöglicht die gemeinsame Nutzung von Programmmodulen und entspricht dem von Sun Microsystems™ eingeführten Entwicklungsparadigma „Write Once, Run Anywhere™“. Die Plattformunabhängigkeit der Programmiersprache Java wird durch die Interpretation von Bytecode in der Java Virtual Machine (JVM) erreicht. Um die Sicherheit des Clients nicht zu gefährden, laufen die Java-Applets in der sogenannten Sandbox des Web-Browsers mit eingeschränktem Zugriff auf die Systemressourcen ab. Die dadurch beschränkten Möglichkeiten der Client-Software sind beim Entwurf zu berücksichtigen. Die Java-Applets werden beim Darstellen der Web-Seite vom Server nachgeladen und gestartet, sodass eine Installation auf dem Client nicht notwendig ist. Auf diese Weise ist es auch nicht notwendig, bei Weiterentwicklungen der Software ein Update auf dem Client durchzuführen. Die JVM ist Bestandteil gängiger Web-Browser, sodass die

Nutzung des Labors nicht auf eine Plattform beschränkt ist. Bild 1 zeigt die Kommunikationsstruktur zwischen Client und Server.

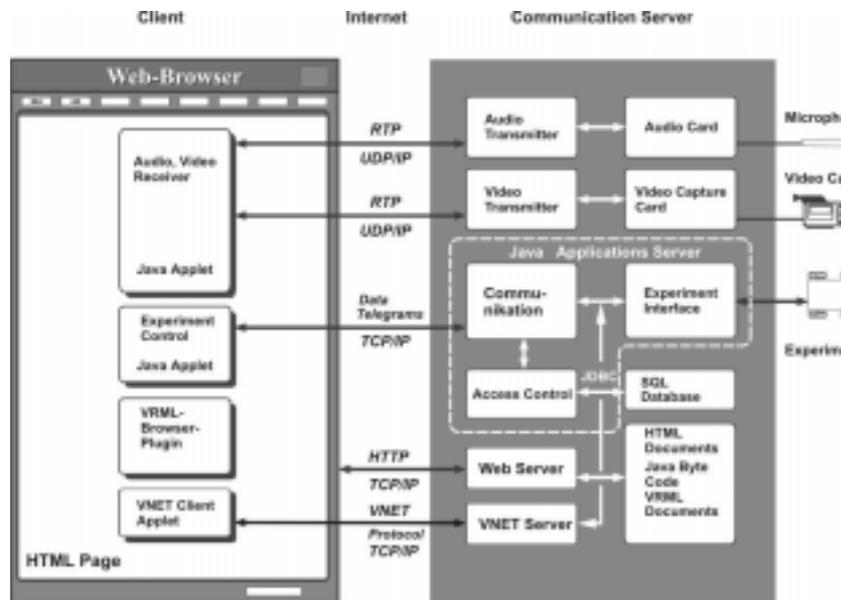


Bild 1: Kommunikationsstruktur

Als Betriebssystem des Servers kommt aufgrund der Stabilität und Skalierbarkeit Linux zum Einsatz. Durch die Implementierung des Applikationsservers in Java ist eine Portierung auf andere Betriebssysteme leicht möglich. Die Client-Software besteht aus zwei Java-Applets, die in eine Web-Seite eingebettet sind. Die Applets werden zusammen mit der Web-Seite mittels HTTP (Hypertext Transfer Protokoll) vom Web-Server geladen und gestartet. Ein Applet dient zur Erzeugung der Telepräsenz, das zweite Applet zur Steuerung des Experiments. Die Telepräsenz zur Unterstützung der Versuchsdurchführung wird mittels Video- und Audioübertragung aus dem Labor erzeugt. Es hat sich gezeigt, dass sich damit ein guter Eindruck vom realen Versuch erzeugen lässt. Das Telabor an der FernUniversität Hagen wird zum Lehren der Grundlagen der Regelungstechnik eingesetzt. Die Experimente beinhalten einfache lineare Regler, meist PID-Regler, mit fester Struktur und variablen Parametern. Es

wurden andere Ansätze vorgestellt, bei denen der Algorithmus des Reglers von den Studierenden frei programmiert und auf den Server geladen wird [11] oder der Algorithmus auf dem Client-Rechner ausgeführt wird [12]. In der Implementierung an der FernUniversität Hagen wurde darauf bewusst verzichtet, da es hier notwendig ist, die Stabilität des Regelalgorithmus zu überprüfen, um den Versuchsaufbau nicht zu gefährden. Weiterhin kann der erforderliche Reglertakt nicht über das Internet realisiert werden, sodass eine Realisierung des Reglers im Client nicht möglich ist.

3.2 Kommunikationsstruktur

Bei der Kommunikation zwischen Client und Server werden mehrere unabhängige Informationskanäle verwendet. Der zentrale Steuerungskanal sendet Befehle an das Experiment und empfängt Statusinformationen vom Experiment. Weitere Kanäle werden zur Erzeugung der Telepräsenz verwendet, wobei Video- und Audioströme im Server erzeugt und an den Client gesendet werden. Weiterhin werden die Messdaten des Experiments aufgezeichnet und für die spätere Analyse auf dem Server gespeichert. An die Informationskanäle werden unterschiedliche Anforderungen gestellt. Der Steuerungskanal sendet und empfängt nur geringe Datenmengen, die jedoch unverfälscht mit möglichst geringer Verzugszeit zwischen Client und Server ausgetauscht werden müssen. Die Messdaten, die im Experiment aufgezeichnet wurden, müssen ebenfalls unverfälscht übertragen werden. Die Übertragungszeit spielt dabei aber eine untergeordnete Rolle. Die Video- und Audioströme dagegen enthalten große Datenmengen, die effizient und mit möglichst geringer Verzugszeit zum Client übertragen werden müssen. Gehen dabei vereinzelt Daten verloren oder werden diese verfälscht, ist dies für den Gesamteindruck der Telepräsenz unerheblich.

Das grundlegende Übertragungsprotokoll im Internet ist das IP (Internet Protocol) mit den darauf aufbauenden Protokollen TCP (Transmission Control Protocol) und UDP (User Datagram Protocol). Das IP ermöglicht die Datenübertragung in einem teilweise gestörten Netz. Es werden mehrere Übertragungswege zwischen den Kommunikationspartnern zugelassen, die während der Übertragung, z.B. bei Teilausfall des Netzes, wechseln können. Diese Eigenschaft führt jedoch zu einer nicht zugesicherten und wechselnden Übertragungsdauer der Datenpakete. Der Verlust von Datenpaketen wird nicht erkannt, da im IP der Empfang nicht quittiert wird. Datenverluste treten im Internet z.B. durch Hardwaredefekte aber meistens durch Überlastung der Router auf. Bei Überlastung werden Datenpakete verworfen und nicht weitergeleitet. Das IP wird nicht einzeln verwendet, sondern von einer Transportschicht mit den Protokollen TCP oder UDP vollständig abgedeckt. Das TCP überträgt die Information gesichert mit Bestätigung des Empfangs. Es stellt

sicher, dass die Daten unverfälscht und in der Reihenfolge ihrer Versendung empfangen werden. Gehen Datenpakete bei der Übertragung im Netz verloren, werden diese wiederholt übertragen, welches zu einer verlängerten Übertragungszeit führt. Weiterhin erzeugt das Protokoll einen hohen Overhead, der besonders bei großen Datenmengen störend wirkt. Um große Datenmengen zu übertragen, wird deshalb das UDP verwendet. Dieses Protokoll ist jedoch ein unsicheres Protokoll mit geringem Overhead, welches meist im Zusammenhang mit anderen überlagerten Protokollen verwendet wird. Auf der Transportschicht befindet sich die Applikationsschicht, welche entweder aus vordefinierten Protokollen wie HTTP (Hypertext Transfer Protocol) oder FTP (File Transfer Protocol) besteht oder bei Bedarf vom Anwender selbst erstellt wird. Die Programmierschnittstelle zur Transportschicht stellen sogenannte Sockets dar, welche im Berkley UNIX eingeführt wurden. Sockets nehmen die Grundidee von UNIX auf, wo jegliche Ein-/Ausgabe auf eine Datei-Ein-/Ausgabe abgebildet wird. Ein Socket wird dabei über eine Internetadresse und eine Portnummer definiert, die beide rein abstrakte Nummern sind und keine physikalische Repräsentation haben. Java besitzt eine plattformunabhängige Programmierschnittstelle zur Netzwerkprogrammierung (java.net-Paket). Die Sockets der Transportschicht werden dabei mit Datenströmen (Klassen `InputStream`, `OutputStream`) verbunden, die komfortabel interpretiert werden können. Weiterhin werden wichtige Protokolle der Anwendungsschicht wie HTTP und FTP von Java direkt unterstützt. Java unterstützt die Netzwerkprogrammierung auf einer noch höheren Ebene. Mit RMI (Remote Method Invocation) wird die Kommunikation objektorientierter verteilter Anwendungen unterstützt. Im Gegensatz zu CORBA (Common Object Request Broker Architecture) unterstützt JMI nur Applikationen, die in Java implementiert sind. JMI bietet dafür die Funktionalität, dass neben dem Austausch der Objektdaten auch der Objectcode über das Netzwerk übertragen werden kann. RMI benötigt für diese Funktionalität einen Kommunikationsoverhead, der die Anwendung bei begrenzter Bandbreite nicht ratsam erscheinen lässt.

Die integrierte Multi-User-Virtual-Reality-Umgebung nutzt die ebenfalls Java-basierte Client-Server Infrastruktur des VNET-Paketes [23]. Details der Einbindung in das Kommunikationskonzeptes finden sich in [24].

3.3 Versuchssteuerung

Zur Kommunikation zwischen dem Steuerungs-Applet und dem Kommunikationsserver wurde ein eigenes Applikationsprotokoll entwickelt, welches auf TCP aufsetzt. Dieses Protokoll wurde zuerst in einem Kommunikationsmodul implementiert, welches eine vorhandene Software zur

Erstellung von Kursen für einen mobilen Roboter erweitert [13]. Das Kommunikationsmodul wird in der aktuellen Software im Java-Applet und im Java-Applikationsserver weiterverwendet. Der Java-Applikationsserver besteht aus einem generischen Teil der Zugriffsverwaltung und dem Kommunikationsmodul sowie einem applikationsspezifischen Teil der Versuchsschnittstelle. Die Telegramme sind universell aufgebaut und enthalten die Telegrammart mit weiteren Parametern. Die Befehle und Parameter werden als ASCII-Strings übertragen, welche leicht in andere Datentypen gewandelt werden können. Auf diese Weise wurde ein weitgehend generischer Protokollaufbau erreicht, der auch ohne Umwandlung in eine Log-Datei geschrieben werden kann. Das Applikationsprotokoll ist generisch und kann für alle Versuche verwendet werden. Die Umsetzung der Telegramme geschieht im Modul Versuchsschnittstelle, welches eine physikalische Verbindung zum Experiment besitzt. Das Experiment wird in der Regel von einem eigenen Echtzeitrechner gesteuert. Auf einen eigenen Echtzeitrechner kann verzichtet werden, wenn im Server die Echtzeiterweiterung RT-Linux zum Einsatz kommt, da damit ein fester Takt für die Regelung erzeugt werden kann.

3.4 Telepräsenz

Um den Studierenden das Gefühl eines realen Experiments zu vermitteln, wird ein Video- und ein Audiostrom aus dem Labor an den Client gesendet. Der Empfang und die Darstellung der Medienströme wird durch Java Media Framework (JMF) realisiert [14]. JMF ist ein API (Application Programming Interface), um Video, Audio und andere zeitbasierte Datenströme in Java zu integrieren. JMF ist eine Standarderweiterung der J2SETM (Java 2 platform, Standard Edition) und ist bis jetzt noch in keinem Web-Browser integriert. JMF ist als reine Java-Version verfügbar, die sowohl auf dem Client als auch auf dem Web-Server installiert werden kann. Bei einer Installation auf dem Web-Server werden vom Java-Applet benötigte JMF-Klassen vom Web-Server geladen, es besteht somit keine Notwendigkeit, JMF auf dem Client zu installieren. Zur Geschwindigkeitssteigerung und zur Unterstützung weiterer Medientypen werden sogenannte „Performance Packs“ angeboten. Sie bestehen aus „native Code“ und müssen auf dem Client installiert werden. Diese Pakete sind zur Zeit der Verfassung des Beitrages für die Plattformen Windows, Solaris und Linux verfügbar. Die von Sun Microsystems in J2SE eingeführte Technologie „Java Optional Package Installation“ lässt die manuelle Installation von JMF auf der Seite des Clients entfallen, sobald diese Technologie in die Web-Browser integriert wird.

Zur Übertragung von Medienströmen über das Internet verwendet JMF das Real-Time Transport Protokoll (RTP) [15]. RTP nutzt als Transportprotokoll meist

UDP und ermöglicht die Übertragung von mit Zeitstempel versehenen Daten. UDP wird verwendet, da der Overhead von TCP für die Übertragung von Medienströmen zu gross ist und ein Nachsenden von verlorenen Paketen zu viel Zeit in Anspruch nehmen würde. Verlorene Pakete werden in diesem Protokoll ignoriert. Das bedeutet, dass der Empfänger auch dann die Medienströme darstellen können muss, wenn einzelne Pakete verloren gehen. Weiterhin werden nicht alle Datenpakete in einem äquidistanten Zeitraster empfangen, da die Router im Internet einzelne Pakete zwischenspeichern und zeitverzögert senden. Der so entstehende Effekt wird Jitter genannt. Um den Einfluss des Jitters zu beheben, werden die Daten im Empfänger gepuffert, nach ihrem Zeitstempel sortiert und dann in äquidistanten Zeitabständen zeitversetzt abgespielt. RTP nutzt zur Übertragung von Steuerungs- und Statusinformationen das Real-Time Control Protocol (RTCP). RTCP überträgt z.B. die Qualität (Datenverlust, Jitter) des Medienstroms beim Empfang und wird dazu genutzt, den Datendurchsatz des Senders anzupassen. RTP selbst definiert nicht die Art und Weise, wie die Medien codiert werden. Die Codierung wird auch als CODEC (CODer/DECoder) bezeichnet und ist notwendig, da unkomprimierte Medienströme zu viel Bandbreite benötigen würden. Über das sieben Bit breite Feld Payload Type wird im RTP das verwendete CODEC dem Empfänger mitgeteilt. Standardisierte CODECs, die im Zusammenhang mit RTP verwendet werden, sind in [16] definiert. Es wird aber auch eine dynamische Zuordnung von benutzerdefinierten CODECs zum Payload-Type zugelassen. JMF unterstützt häufig eingesetzte standardisierte CODECs und lässt sich damit im Zusammenhang mit anderen Produkten einsetzen. Für jeden Medienstrom wird ein eigener RTP-Kanal benötigt. Werden mehrere Medienströme gleichzeitig verwendet (z.B. Video und Audio), müssen diese synchronisiert werden. Dies geschieht über die Zeitstempel der Datenpakete im RTP.

Zum Empfang des Video- und Audio-Stroms wurde ein Java-Applet mit JMF realisiert, welches die beiden Medienströme mit einem Sessionmanager empfängt und synchronisiert. Es werden zur Zeit die CODECs H.261 (Video) und GSM (Audio) verwendet. Diese benötigen eine geringe Bandbreite bei akzeptabler Darstellungsqualität. Die ausgegebene Datenmenge ist bei GSM fest 13,2 kBit/s, während sie bei H.261 durch Größe des Videobildes, Bildwiederholungsrate und Videoqualität angepasst werden kann. Durch den Einsatz von H.261 zur Videoübertragung ist ein „Performance Pack“ von JMF notwendig.

Zur Generierung des Video- und Audiostroms werden die Mbone-Tools verwendet, da zur Zeit der Implementierung die Linux-Version von JMF noch nicht freigegeben war. Die Skalierung des Datendurchsatzes der Server erfolgt mit CGI-Skripten, die die angepassten Versionen der Mbone-Tools

parametrieren. Es wird dabei durch Anpassung der Bildrate und der Videoqualität die Datenrate des Videostroms der zur Verfügung stehenden Bandbreite angepasst. Bei sehr geringer Bandbreite oder nicht vorhandener Soundkarte kann die Erzeugung des Audiostroms abgeschaltet werden. Für eine detailliertere Beschreibung der Realisierung wird auf [17] verwiesen.

Für die Kommunikation der (möglicherweise räumlich verteilten) Praktikumsgruppe untereinander und dem Versuchsbetreuer sind typische synchrone Kommunikationstechniken wie Video- und Audiokonferenzen nicht geeignet, da sie einen hohen Ressourcenbedarfs an Kommunikationsbandbreite erfordern. Insbesondere Konferenzen in großen Gruppen (also Point to Multipoint-Konferenzen) erfordern bei n-Teilnehmern n-mal die Bandbreite eines Videostreams, wenn eine wirkliche Diskussion, d.h. nicht nur eine halbduplex Kommunikation von einem Lehrenden zu den n-Lernenden, stattfinden soll. Die verfügbare Bandbreite des Teilnehmers mit der schmalbandigsten Anbindung begrenzt für die beschriebene Form der Videokonferenz die maximale Anzahl der Kommunikationspartner. Ein weiterer Nachteil von Videokonferenzen ist die für alle Teilnehmer benötigte Hardwareausstattung. Für die ressourcenschonende Echtzeitkommunikation zwischen n-Teilnehmern hat sich der **Internet Relay Chat** als Kommunikationsform bewährt. Diese Form der Kommunikation stößt aber bei ungeübten Kommunikationsteilnehmern auf Akzeptanzprobleme. Rein textbasierte anonyme Kommunikation wird als „unnatürlich“ erlebt. Interaktive Kommunikation wird als natürlich empfunden, wenn ein visueller Eindruck des Gegenübers zur Verfügung steht. Eine Kommunikation fällt leichter, wenn eine Umgebung zur Verfügung steht, in und mit der interagiert werden kann. Eine neuartige synchrone Kommunikationsform ist der sogenannte 3D-Chat (Multi-User-Virtual-Reality-Umgebung), bei dem herkömmliche Text-Chat Techniken mit Virtual Reality Umgebungen verknüpft werden. Die Benutzer solcher Umgebungen werden in der virtuellen Realität durch sogenannte Avatare repräsentiert. Teilnehmer einer solchen VR-Konferenz sehen die Avatare anderer Benutzer in einem der realen Welt nachgebildetem Szenario. Es ist möglich solche Avatare, zusätzlich zu den Bewegungsmöglichkeiten im virtuellen Raum, mit einfachen Verhaltensmöglichkeiten wie z.B. Gesten auszustatten. Hierdurch wird den Benutzern zusätzlich die Möglichkeit nonverbaler Kommunikation eröffnet. 3D-Chats haben einen sehr viel geringeren Ressourcenbedarf an Bandbreite als Videokonferenzen weil nur Events (Avatar-Positionsänderungen, Gesten usw.) übermittelt werden müssen.

Die realisierte Multi-User-Virtual-Reality-Umgebung basiert auf der **Virtual Reality Modeling Language (VRML[22])** einem ISO-Standard für 3D-Daten im Web. Der Gegenstand des Laborexperimentes, ein Omnidirektionales Fahrzeug wird in einer virtuellen Laborumgebung ebenfalls durch einen Avatar

repräsentiert. Alle Benutzer können während des Versuches die Bewegungen des Fahrzeuges sowohl im Videobild als auch in der virtuellen Realität verfolgen. Eine Echtzeitdarstellung ist über eine Modemverbindung mit 56kBit/s möglich, verzichtet der Benutzer auf eine Video- und Audiodatenübertragung ist eine Verbindung über GSM mit einem Mobiltelefon (9600 Bit/s) ausreichend.

3.5 Zugriffsverwaltung

Mit der Anmeldung zum Praktikum erhalten die Studierenden eine Benutzerkennung mit Paßwort sowie ein Zeitkontingent für jeden gewünschten Versuch. Mit diesen Daten können Versuchszeiten bis zur Höhe des Zeitkontingents reserviert werden. Um eine größtmögliche Flexibilität zu erhalten, erfolgt die Reservierung von den Studierenden selbst. Nicht benötigte Versuchszeit kann freigegeben und zu einem späteren Zeitpunkt genutzt werden. Das entsprechende Benutzerinterface wurde als Java-Applet implementiert. Der Benutzer erhält im Applet einen Terminkalender, bei dem freigeschaltete Termine markierbar sind (Bild 2).

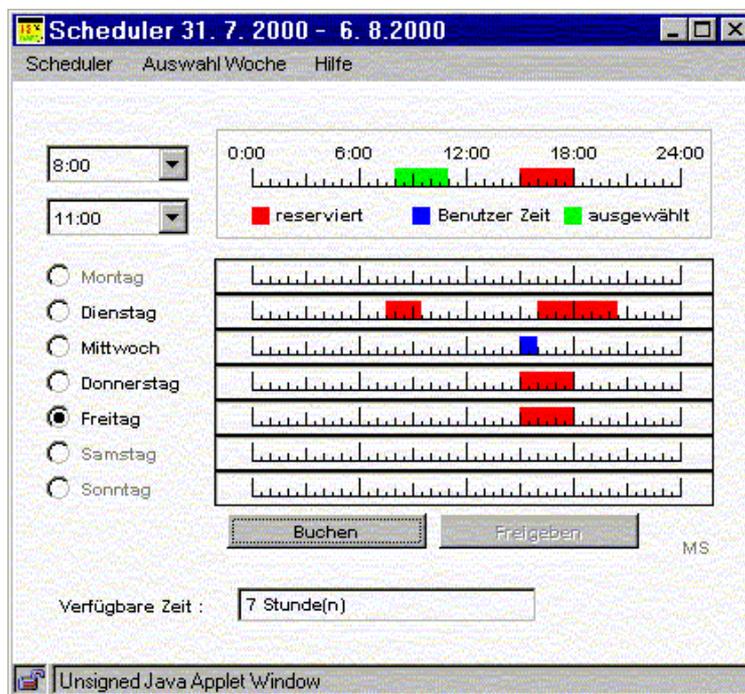


Bild 2: Terminreservierung

Das Applet kommuniziert mit einem Server, der die Daten in einer SQL-Datenbank ablegt. Die Schnittstelle zwischen der Datenbank und dem Java-Server wurde mit JDBC (Java Database Connectivity) [18] implementiert. JDBC ist ein API, welches eine plattformunabhängige Schnittstelle zur Datenbankprogrammierung realisiert. Es sind JDBC-Treiber für eine große Anzahl von Datenbanken von unterschiedlichen Anbietern verfügbar. Die Administration der Datenbank erfolgt ebenfalls mit Web-Browser und Java-Applet. Für weitere Details der Zugriffsverwaltung wird auf [19] verwiesen.

4 Beispielversuch: Omnidirektionales Fahrzeug

4.1 Versuchsaufbau

Der Versuch, der im Verbund von der FernUniversität Hagen angeboten wird, behandelt die Antriebsregelung eines omnidirektionalen Fahrzeuges (Bild 3).



Bild 3: Omnidirektionales Fahrzeug

Das Fahrzeug ist mit vier Mecanum-Antriebseinheiten ausgestattet und besitzt dadurch alle drei Freiheitsgrade in der Ebene. Ein derart ausgestattetes, mobiles System wird durch das gezielte Ansteuern aller Räder ohne Lenkeinrichtung

allseitig frei beweglich, sodass jede beliebige Kombination aus Vorwärts-, Seitwärts- und Drehbewegung möglich ist. Das Fahrzeug erhält somit trotz seiner Flurgebundenheit die Mobilität eines Luftkissenfahrzeuges. Bild 4 zeigt ein Mecanum-Rad in zwei Ansichten sowie ein Fahrzeug, welches mit vier derartigen Rädern ausgestattet ist. Die Antriebseinheit besteht aus einem Rad, auf dessen Umfang acht bzw. zwölf Rollen in einem Winkel von 45° zur Orientierung angeordnet sind. Diese Rollen sind so geformt, dass die Silhouette der Mecanum-Antriebseinheit einen Kreis bildet. Das Rad der Antriebseinheit besteht aus Gusseisen bzw. Stahl, wobei die Rollen aus Gummi hergestellt sind, um eine möglichst große Bodenhaftung zu gewährleisten.

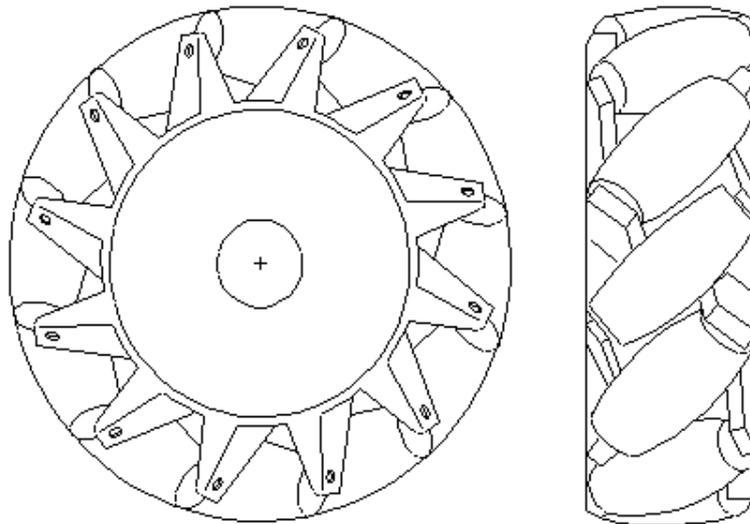


Bild 4: Mecanum-Räder

Das Fahrzeug wird von einem VME-Bus-Rechner mit dem Echtzeitbetriebssystem pSOS+ gesteuert. Der Rechner ist über ein Funk-Ethernet mit dem Lehrgebiets-LAN verbunden. Die Mecanum-Räder werden mittels elektronisch kommutierter Servomotoren angetrieben. Die Servosteller enthalten integrierte Stromregler und werden somit mit drehmomentproportionalen Stellgrößen angesteuert. Die in C++ implementierte Steuerungssoftware enthält zwei verschiedene Regelungsstrukturen. Die erste Struktur ist eine Reglerkaskade mit PD-Lageregler und unterlagertem PID-Drehzahlregler. Der Drehzahlregeltakt beträgt $0,003\text{ s}$ der überlagerte Takt des Bahninterpolators und

der Lageregelung 0,015 s. Diese Struktur wird im Grundlagenpraktikum eingesetzt. Die zweite Struktur enthält die in [20] entwickelte Verkopplungsregelung, die vertiefenden Experimenten vorbehalten bleibt.

4.2 Versuchsdurchführung

Im Grundlagenpraktikum besteht die Aufgabe der Studierenden darin, zuerst die Struktur des Modells des Fahrzeuges aufzustellen und dann mittels aufgezeichneter Sprungantworten die Parameter des Modells zu bestimmen. Aufbauend auf dem Modell werden dann mit unterschiedlichen Entwurfsverfahren die Parameter der Reglerkaskade bestimmt und in das Fahrzeug geladen.

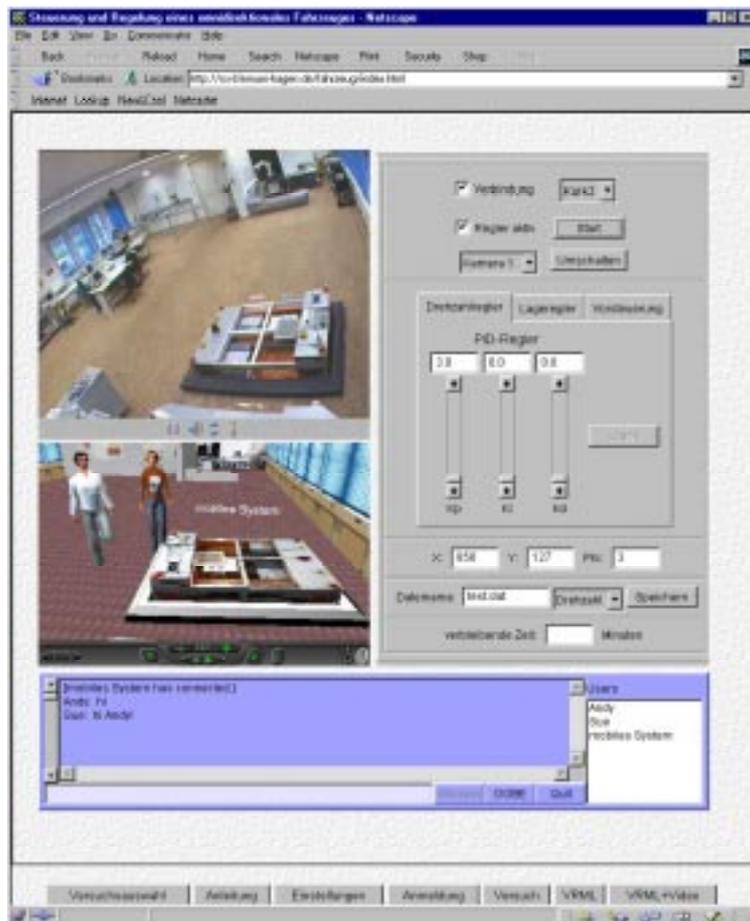


Bild 5: Steuerung des omnidirektionalen Fahrzeuges

Es stehen nun mehrere Testkurse zur Verfügung, die abgefahren werden können, um die Regelung zu testen. Die Kommunikation zwischen Fahrzeug und Server geschieht mittels Datentelegrammen über TCP. Es wird der gleiche Telegrammaufbau wie zwischen Java-Applet und Server verwendet. Für eine weiterführende Beschreibung des Versuchs wird auf [21] verwiesen.

Während einer Bewegung werden Messwerte im Fahrzeugrechner im Reglertakt gespeichert und auf Anforderung über einen eigenen Datenkanal mittels TCP an den Server übertragen. Von den Studierenden kann dabei ein Dateiname vorgegeben werden, der eine bessere Übersicht über die Daten ermöglichen soll.

Die Daten können mittels der Download-Funktion des Web-Browser zur späteren Analyse heruntergeladen werden. Auf eine Echtzeitübertragung der Daten während des laufenden Versuchs wird verzichtet, da das Funk-Ethernet die dabei auftretende Datenrate nicht übertragen kann.

4.3 Datenanalyse

Zur Analyse der Daten wurde ein Programm in Java entwickelt, welches sowohl offline als Java-Applikation als auch online als Java-Applet eingesetzt werden kann. Das Java-Applet ist aufgrund der Sicherheitsrestriktionen in seinem Funktionsumfang eingeschränkt. Es kann nicht auf die Ressourcen des Clients zugreifen. Damit ist das Speichern der Messdaten auf dem lokalen Dateisystem und der Ausdruck der Graphiken nicht möglich. Das Applet dient deshalb zur Anzeige und Überprüfung der Daten während des laufenden Versuches. Es lädt die Messdaten vom Kommunikationsserver mittels HTTP. Dabei werden die komfortablen Möglichkeiten der Netzwerk-API von Java genutzt. Es kommt die Klasse `java.net.URL` zum Einsatz, mit der sowohl eine Dateiauswahl über den Index des Datenverzeichnisses als auch der Download der Dateien selbst realisiert wurde. Das Programm stellt aus den eingelesenen Messdaten Graphiken in eigenen Fenstern dar. In den Fenstern ist dann eine manuelle oder automatische Skalierung, die Auswahl von Farben und das Einfügen eines Rasters möglich. Dies ist hilfreich bei einem späteren Ausdruck der Messwerte für die Versuchsausarbeitung. Neben der grafischen Darstellung der Messwerte ermöglicht das Applet eine Datenanalyse. Es können dabei sowohl die Daten des geschlossenen Regelkreises, als auch die Daten der Sprungantwort der Strecke analysiert werden. Die Analyse des geschlossenen Regelkreises ergibt Kennwerte zur Regelabweichung wie maximale Regelabweichung, Mittelwert der quadratischen Regelabweichung und Mittelwert des Betrages der Regelabweichung. Die Analyse der Sprungantwort der Strecke ergibt ein einfaches Modell mit PT1Tt-Approximation, bei dem die Verzugszeit, Anstiegszeit und der Übertragungsbeiwert bestimmt werden. Mit diesen Daten kann dann durch das Applet ein Reglerentwurf nach Ziegler und Nichols durchgeführt werden. Bild 6 zeigt das Analyse-Applet mit Messdaten in zwei Fenstern.

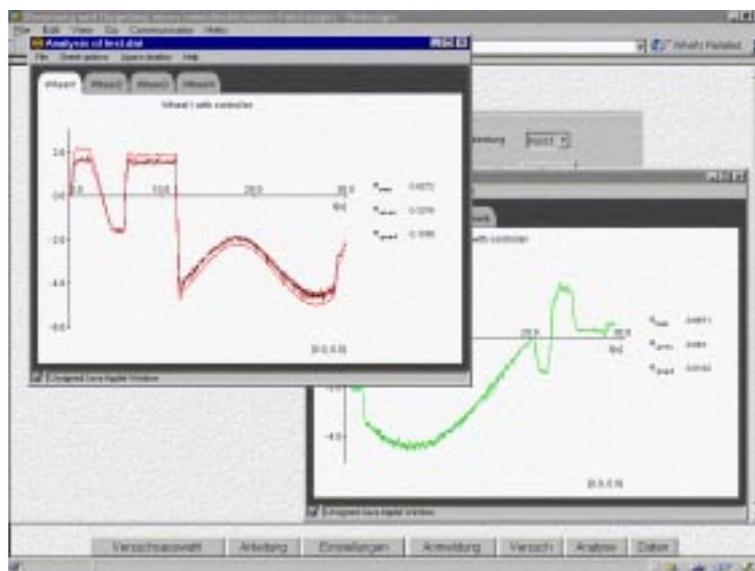


Bild 6: Datenanalyse

5 Erfahrungen

5.1 Didaktische Probleme

Beim Einsatz im Laborpraktikum wurde festgestellt, dass der Betreuung der Studierenden ein hoher Stellenwert zukommt. Die Betreuung erfolgt synchron zum Experiment mittels der vorgestellten Multi-User-Virtual-Reality-Umgebung. Andernfalls erfolgt die Betreuung mittels E-Mail, was allerdings während des laufenden Experiments umständlich und zeitintensiv ist. Gruppenarbeit ist durch den Einsatz der vorgestellten Multi-User-Virtual-Reality-Umgebung möglich, notwendig und wurde von den Benutzern der Praktikumsversuche ausdrücklich gewünscht. Online-Praktikumsversuche erhöhen den Bearbeitungsaufwand für die Studierenden, die Zeit- und Ressourcenersparnis rechtfertigt aber diesen Aufwand.

5.2 Technische Probleme

Ein häufiger Grund für technische Probleme ist ein limitierter Internetzugang der Benutzer vom Arbeitsplatz aus, bzw. über Provider wie z.B. AOL. Während unsere berufstätigen Benutzer (ein hoher Anteil der Studierenden der FernUniversität studiert im Teilzeitstudium) teilweise mit Limitierungen verursacht durch Firewalls ihrer Arbeitgeber zu kämpfen haben, werden Benutzer einiger Online-Dienste, wie beispielsweise AOL nur über HTTP-Proxys mit dem WWW verbunden, eine wirkliche Verbindung mit dem Internet besteht nicht. Die umfangreichen Funktionalitäten eines ferngesteuerten Praktikums sind nur zu realisieren, wenn der volle Umfang der TCP/IP bzw. UDP Internetprotokollfamilie zur Verfügung steht. Das durch das WWW bekannte Hypertext Transfer Protocol (HTTP) reicht hierfür alleine *nicht* aus.

Ein vollwertiger Internetzugang ist erforderlich um alle Funktionalitäten von Online-Praktikumsversuchen zu nutzen.

Weitere technische Probleme entstehen durch nicht garantierte Übertragungsbandbreiten im Internet (Quality of Service) und die Notwendigkeit der Installation von Browser-Plugins bei den entfernten Benutzern (Java Media Framework, VRML-Plugin). Solche Probleme können durch umfangreiche Dokumentation und Betreuung gemildert werden.

6 Zusammenfassung und Ausblick, übertragbare Konzepte

Der Beitrag beschreibt eine plattformunabhängige Lösung für den teleoperierten Zugriff auf Laborexperimente. Die Studierenden benötigen für die Bedienung lediglich einen Web-Browser mit JVM, JMF und VRML-Browser-Plugin. Die entwickelten Methoden und Werkzeuge sind generisch und werden bereits in verschiedenen Laborversuchen an der FernUniversität Hagen und an den Universitäten Bochum und Dortmund eingesetzt. Es wurden weitgehend alle Anforderungen aus Abschnitt 2 erfüllt. Das Konzept ist aufgrund der Implementierung in Java plattformunabhängig und frei von Installationen auf dem Client. Lediglich durch den Einsatz von JMF müssen Einschränkungen in Kauf genommen werden, welche allerdings zukünftig durch die Integration neuer Java-Technologien in die Web-Browser entfallen werden. Die FernUniversität Hagen ist Partner im BMBF-Projekt LEARNET („Lernen und Experimentieren an realen technischen Anlagen im Netz“ [25]). Die LEARNET-Projektpartner entwickeln weitere Praktikumsversuchsaufbauten, die über das Internet ferngesteuert werden können. Die vorgestellte Multiuser-Virtual-Reality-Umgebung wird für die „Plattform 2001“ der FernUniversität eingesetzt [26]. Virtuelle Seminarveranstaltungen und eine „Virtuelle Cafeteria“ werden mit dieser Umgebung an der FernUniversität Hagen realisiert.

Literaturverzeichnis

[1] Reale Systeme im virtuellen Labor:
<http://prt.fernuni-hagen.de/rsvl/>

[2] C. Schmid: A Remote Laboratory Using Virtual Reality on the Web, *Simulation* 73 (1), S. 13-21, 1999

[3] B. Aktan, C. A. Bohus, L. A. Crowl, M. H. Shor: Distance Learning Applied to Control Engineering Laboratories, *IEEE Transactions on Education* 39 (3), S. 320-326, 1996

- [4] J. Henry: Controls Laboratory Teaching via the World Wide Web, In Proc. ASEE Annual Conference, Washington, USA, 1996
- [5] A. Bhandari, M. H. Shor: Remote-Access Engineering Educational Laboratories: Who, What, When, Where, Why, and How? In Proc. American Control Conference, Chicago, USA, 2000
- [6] T. Junge, C. Schmid: Web-based Remote Experimentation Using a Laboratory-scale Optical Tracker, In Proc. American Control Conference, Chicago, USA, 2000
- [7] K. Schilling, H. Roth, O. Rösch: Mechatronik-Experimente in Virtuellen Labors, Künstliche Intelligenz 2/00, S. 41-46, 2000
- [8] H. Fricke, C. Köppel, A. Oulianov, S. Steingrad: Laborausbildung im Internet - ein Verbundprojekt des Landes Baden-Württemberg, In Proc. 44th International Scientific Colloquium Technical University of Illmenau, Illmenau, 1999
- [9] H. Baier, M. Buss, F. Freyberger, J. Hoogen, P. Kammermeier, G. Schmidt: Distributed PC-based Haptic, Visual and Acoustic Telepresence System - Experiments in Virtual and Remote Environments, In Proc. IEEE Virtual Reality Conference, Houston, USA, 1999
- [10] K. Arnold, J. Gosling, D. Holmes: The Java™ Programming Language Third Edition, Addison Wesley, 2000
- [11] Y. Piguet, D. Gillet: Java-based Remote Experimentation for Control Algorithms Prototyping, In Proc. American Control Conference, San Diego, USA, 1999
- [12] J. W. Overstreet, A. Tzes: An Internet-Based Real-Time Control Engineering Laboratory, IEEE Control Systems Magazine 19 (5), S. 19-34, 1999
- [13] C. Röhrig, A. Jochheim: Remote Control of Laboratory Experiments, In Proc. 19th World Conference on Open Learning and Distance Education, Wien, Österreich, 1999
- [14] Sun Microsystems: Java Media Framework
<http://www.java.sun.com/products/java-media/jmf/>

- [15] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: RTP: A Transport Protocol for Real-Time Applications, Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, 1996
- [16] H. Schulzrinne: RTP Profile for Audio and Video Conferences with Minimal Control, Request for Comments (Proposed Standard) 1890, Internet Engineering Task Force, 1996
- [17] C. Röhrig, A. Jochheim: The Virtual Lab for Controlling Real Experiments via Internet. In Proc. IEEE International Symposium on Computer-Aided Control System Design, Hawaii, USA, 1999
- [18] M. Fisher, R. Cattell, G. Hamilton, S. White, M. Hapner: JDBC/TM API Tutorial and Reference, Second Edition: Universal Data Access for the Java/TM 2 Platform, Addison Wesley, 1999
- [19] C. Röhrig, A. Jochheim: Java-based Framework for Remote Access to Laboratory Experiments, In Proc. IFAC/IEEE Symposium on Advances in Control Education, Gold Coast, Australien, 2000
- [20] A. Jochheim: Reglerentwurf für Mehrgrößensysteme unter der Nebenbedingung vorgegebener Ausgangsgrößenverkopplung, Dissertation, FernUniversität Hagen, 1994
- [21] A. Jochheim, C. Röhrig: The Virtual Lab for Teleoperated Control of Real Experiments. In Proc. IEEE Conference on Decision and Control, Phoenix, USA, 1999
- [22] <http://www.web3d.org>
- [23] White, S. and J. Sonstein (1998). VNet.
<http://www.csclub.uwaterloo.ca/~sfwhite/vnet/>.
- [24] A. Bischoff, C. Röhrig: A Multiuser Environment for Remote Experimentation in Control Education, IBCE'01 (Internet based Control Education 2001), Madrid, 2001
- [25] Lernen und Experimentieren an realen technischen Anlagen im Netz,
<http://www.esr.ruhr-uni-bochum.de/LearNet/>
- [26] <http://www.fernuni-hagen.de>