23

Web-Based Remote Manipulation of Parallel Robot in Advanced Manufacturing Systems

Dan Zhang, Lihui Wang and Ebrahim Esmailzadeh

1. Introduction

During the last decade, the Web has gained widespread acceptance in both academic and business areas. The Web is used by many as a medium of sharing data, information, and knowledge. Today, it is widely used for development of collaborative applications to support dispersed working groups and organizations because of its platform, network and operating system transparency, and its easy-to-use user interface – Web browser. In addition to the Web technology, Java has brought about a fundamental change in the way that applications are designed and deployed. Java's "write once, run anywhere" model has reduced the complexity and cost traditionally associated with producing software on multiple distinct hardware platforms. With Java, the browser paradigm has emerged as a compelling way to produce applications for collaboration over the Internet. As business grows increasingly diversified, the potential of this application is huge. Targeting distributed, real-time monitoring and control in manufacturing sectors, a framework with high efficiency for cyber collaboration is carefully examined.

The objective of this research is to develop a Web-based digital shop floor framework called *Wise-ShopFloor* (Web-based integrated sensor-driven e-ShopFloor) for distant shop floor monitoring and control. The *Wise-ShopFloor*, with an appropriate architecture for effective data communication among a dispersed engineering team, can serve real-time data from bottom up and can function as a constituent component of e-manufacturing. The framework is designed to use the popular client-server architecture and VCM (view-control-model) design pattern with secured session control. The proposed solutions for meeting both the user requirements demanding rich data sharing and the real-time constraints are: (1) using interactive Java 3D models instead of bandwidth-consuming camera images for visualization; (2) transmitting only the sensor data and control commands between models and device controllers for monitoring and control; (3) providing users with thin-client graphical interface for navigation; and (4) deploying control logic in an application server. A

Source: Industrial-Robotics-Theory-Modelling-Control, ISBN 3-86611-285-8, pp. 964, ARS/plV, Germany, December 2006, Edited by: Sam Cubero

proof-of-concept prototype system is developed on top of the framework to demonstrate one of its potential applications on shop floor monitoring and control. It utilizes the latest Java technologies, including Java 3D and Java Servlets, as enabling technologies for system implementation. Instead of camera images, a physical device of interest is represented by a Java 3D model with behavioural control nodes embedded. Once downloaded from an application server, the Java 3D model works on behalf of its counterpart showing behaviours for visualization at a client side, but remains alive by connecting with the physical device through low-volume message passing.

This chapter presents the basis of the framework for building web-based collaborative systems that can be used for distributed manufacturing environments. It first outlines related work, followed by the concept and architecture of the framework. The *Wise-ShopFloor* concept is then demonstrated through a typical case study on device modelling, monitoring, and control. The benefits enabled by the framework include quick responses by reduced network traffic, flexible monitoring by sensor-driven 3D models, and interactive control by real-time feedback.

2. Business Service Management

Business firms generate revenues and profits through effective execution of business processes. During the last two decades, business processes have become increasingly automated, requiring IT service to support business operations. Delivering reliable and consistent levels of business and IT service is critical to the operations of business firms. In recent years, many business organizations have switched from a technical-focused IT management model to a business-oriented IT management framework that links technical capabilities to organizational needs. Using IT management tools to deliver real-time service level management not only meets service goals, but also generates greater business value for the organization [1, 2].

Recent advances in IT technologies have made a variety of new business processes possible. The proposed Web-based digital shop floor framework is an example of this type of new business processes that could not exist before. It uses new IT technology to improve the performance of business operations. This business process application is expected to result in significant increases in productivity and revenues.

The goal of our combined web-based and sensor-driven approach is to significantly reduce network traffic with Java 3D models, while still providing end users with an intuitive environment. The largely reduced network traffic also makes real-time monitoring, control, inspection, and trouble-shooting practical for users on relatively slow hook-ups such as modem connections. Participating in the collaborative system, users not only can feel reduced network traffic by real-time interactions with quick responses, but also can obtain more flexible control of the real world. In the near future, open-architecture devices (such as OpenPLCs and Open-CNC Controllers, etc.) will have web servers and Java virtual machines embedded. This will make the proposed *Wise-ShopFloor* framework more efficient for real-time monitoring and control.

3. Research Background

Initially, parallel kinematic machines (PKMs) were developed based on the Stewart platform that is a 6 DOF prismatic parallel mechanism with extensible legs. Commercial hexapods including VARIAX of Giddings & Lewis, Ingersoll hexapod, Tornado of Hexel, Geodetic of Geodetic Technology Ltd., are all based on this structure. To overcome the problems in hexapods with extensible legs, such as stiffness and heat [3,4], recently, hexapods with fixed-leg lengths have been envisioned, for example, Hexaglide of the Swiss Federal Institute of Technology [5], LINAPOD of Stuttgart University [6], and HexaM of Toyoda [7]. Hexapods with extensible telescopic legs are not suitable for linear motors, but the fixed-leg length hexapods are. Hexapods with revolute joints were also reported in the literature, for example, DELTA robot, which can reach an acceleration of up to 20g in some area of the workspace [8].

The hexapods with fixed-leg lengths are sometimes termed as sliding-leg hexapods, because sliding of the fixed-length legs along their guideways drives the moving platform. There are basically three configurations in terms of guideway angle, vertical, horizontal and angular. In the vertical configuration, gravitational force is in the moving direction. While weight does not contribute to friction, motors have to overcome weight in the upward movement. In the horizontal configuration, gravitational force is perpendicular to the moving direction and weight would fully contribute to friction. Angular configuration is in between in terms of gravitational force and friction force.

Since machining operation requires five axes at most, new configurations with less than six parallel axes would be more appropriate. Development on new configurations is mainly on three axes PKMs. Examples include Triaglide [9], Tetrahedral Tripod [10], and Tricept of SMT Tricept [11]. Three axis PKMs can be combined with 2 axis systems, such as x-y stage, to form five axis machines. The Wise-ShopFloor is designed to provide users with a web-based and sensor-driven intuitive shop floor environment where real-time monitoring and control are undertaken. It utilizes the latest Java technologies, including Java 3D and Java Servlets, as enabling technologies for system implementation. Instead of camera images (usually large in data size), a physical device of interest (e.g. a milling machine or a robot) can be represented by a scene graph-based Java 3D model in an applet with behavioural control nodes embedded. Once downloaded from an application server, the Java 3D model is rendered by the local CPU and can work on behalf of its remote counterpart showing real behaviour for visualization at a client side. It remains alive by connecting with the physical device through low-volume message passing (sensor data and user control commands). The 3D model provides users with increased flexibility for visualization from various perspectives, such as walk-through and fly-around that are not possible by using stationary optical cameras; whereas the largely reduced network traffic makes real-time monitoring, remote control, on-line inspection, and collaborative trouble-shooting practical for users on relatively slow hook-ups (e.g. modem and low-end wireless connections) through a shared *Cyber Workspace* [12].

By combining virtual reality models with real devices through synchronized real-time data communications, the *Wise-ShopFloor* allows engineers and shop floor managers to assure normal shop floor operations and enables web-based trouble-shooting – particularly useful when they are off-site.

Figure 1 shows how it is linked to a real shop floor. Although the *Wise-ShopFloor* framework is designed as an alternative of camera-based monitoring systems, an off-the-shelf web-ready camera can easily be switched on remotely to capture unpredictable real scenes for diagnostic purposes, whenever it is needed. In addition to real-time monitoring and control, the framework can also be extended and applied to design verification, remote diagnostics, virtual machining, and augmented virtuality in construction. It is tolerant to hostile, invisible or non-accessible environments (e.g. inside of a nuclear reactor or outside of a space station).

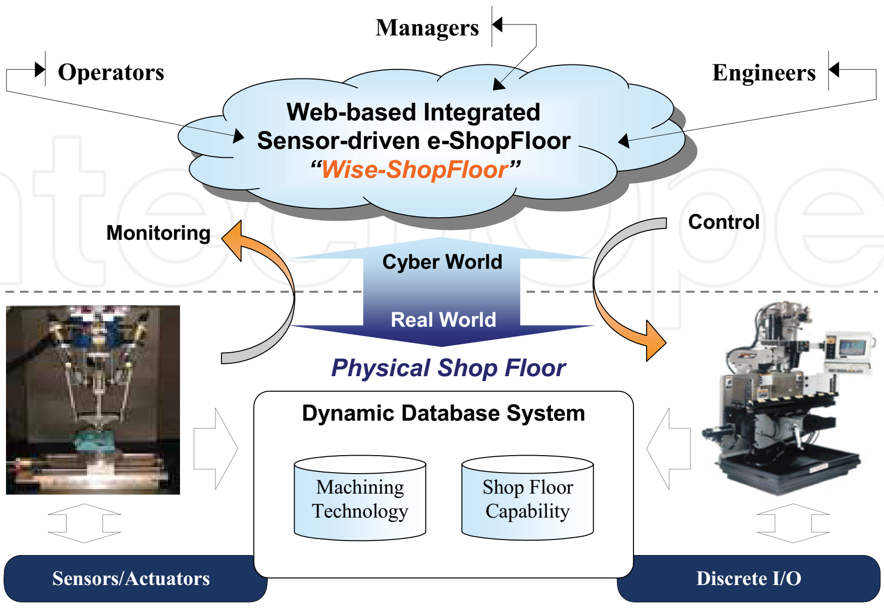


Figure 1. Concept of Wise-ShopFloor

4. Architecture Design

As shown in Figure 2, the framework is designed to use the popular client-server architecture and VCM (view-control-model) design pattern with built-in secure session control. The proposed solutions for meeting both the user requirements of rich visual data sharing and the real-time constraints are listed below.

1. Using interactive scene graph-based Java 3D models instead of bandwidth-consuming camera images for shop floor visualization;
2. Transmitting only the sensor data and control commands between models and device controllers for remote monitoring and control;
3. Providing users with thin-client graphical user interface for shop floor navigation; and
4. Deploying major control logic in a secured application server.

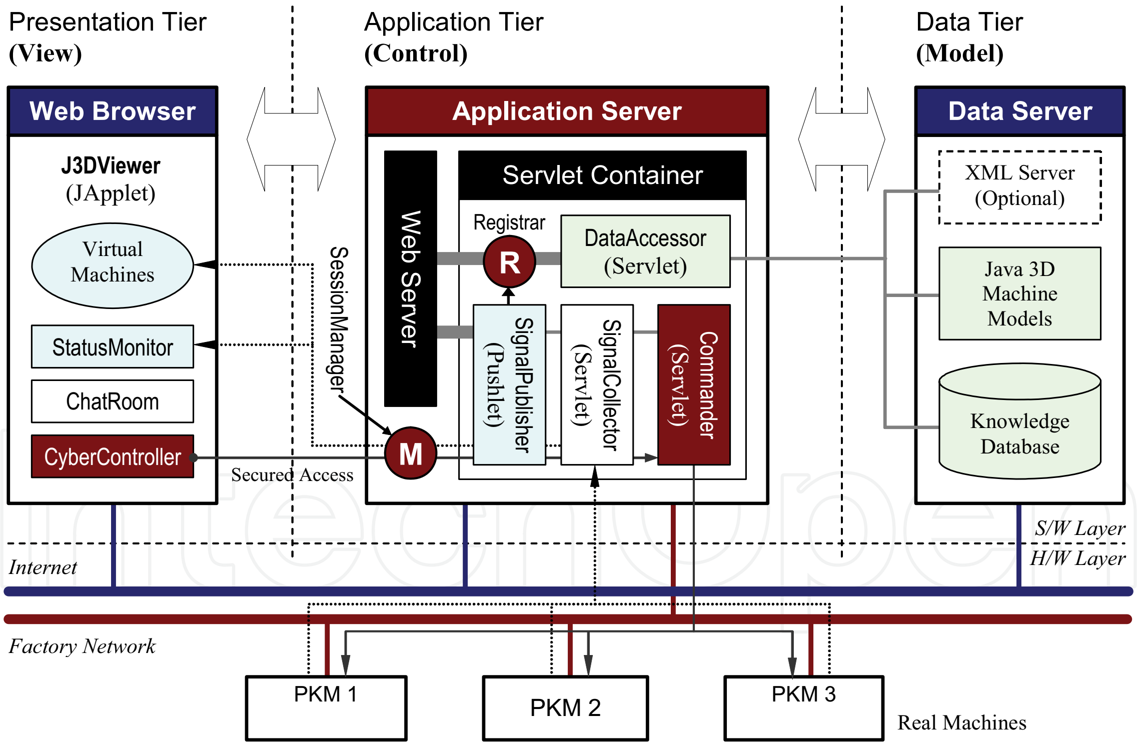
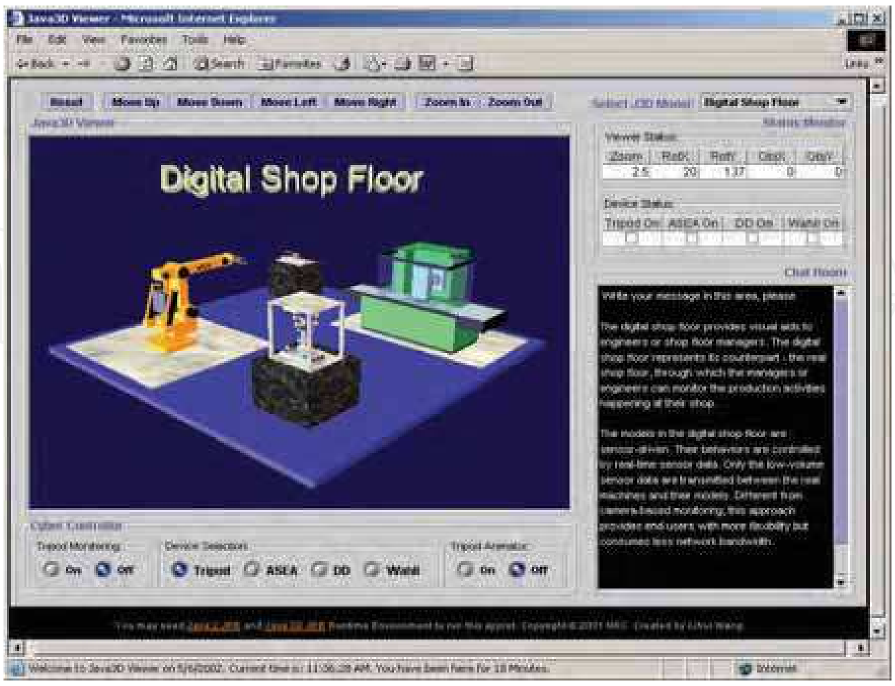


Figure 2. Architecture of Wise-ShopFloor

The mid-tier application server handles major security concerns, such as session control, viewer registration, data collection/distribution, and real device manipulation, etc. A central *SessionManager* is designed to look after the issues of user authentication, session control, session synchronization, and sensitive data logging. All initial transactions need to pass through the *SessionManager* for access authorization. In a multi-client environment – the *Wise-ShopFloor*, different clients may require different sets of sensor data for different models. Constrained by network security, a Java 3D model residing in an applet is not allowed to communicate directly with a real device through socket communication. It is also not efficient to have multiple clients who share the same model talking with the same device at the same time. The publish-subscribe design pattern is adopted to collect and distribute sensor data at the right time to the right client efficiently. As a server-side module, the *SignalCollector* is responsible for sensor data collection from networked physical devices. The collected data are then passed to another server-side module *SignalPublisher* who in turn multicasts the data to the registered subscribers (clients) through applet-servlet communication. A *Registrar* is designed to maintain a list of subscribers with the requested sensor data. A Java 3D model thus can communicate indirectly with sensors no matter where the client is, inside a firewall or outside. The JMF (Java Media Framework) is chosen for the best combination between applets and servlets. For the same security reasons, a physical device is controllable only by the *Commander* that resides in the application server. Another server-side component called *DataAccessor* is designed to separate the logical and physical views of data. It encapsulates JDBC (Java Database Connectivity) and SQL codes and provides standard methods for accessing data (Java 3D models, knowledge base of the devices, or XML documents). The knowledge base is found helpful for device trouble-shooting, while XML will be used for high-level data communication in future extensions.

Although the global behaviours of Java 3D models are controlled by the server based on real-time sensor signals, users still have the flexibility of monitoring the models from different perspectives, such as selecting different 3D machine models, changing viewpoint, and zooming, through *J3DViewer* at the client side. Authorized users can submit control commands through *CyberController* to the application server. The *Commander* at server-side then takes over the control for real device manipulations. Another client-side module *StatusMonitor* can provide end users with a view of run-time status of the controlled device. For the purpose of collaborative trouble-shooting, a *ChatRoom* is included in the framework for synchronized messaging among connected users.

A proof-of-concept prototype is developed on top of the framework to demonstrate its application on remote monitoring and control. Figure 3 shows one snapshot of the web user interface of the prototype. A more detailed discussion from device modelling to control is provided in Section 6 through a case study of a Tripod test bed.



Zoom	RotX	RotY	ObjX	ObjY
2.5	20	137	0	0

Write your message in this area, please

The digital shop floor provides visual aids to engineers or shop floor managers. The digital shop floor represents its counterpart - the real shop floor, through which the managers or engineers can monitor the production activities happening at their shop.

The models in the digital shop floor are sensor-driven. Their behaviors are controlled by real-time sensor data. Only the low-volume sensor data are transmitted between the real machines and their models. Different from camera-based monitoring, this approach provides end users with more flexibility but consumes less network bandwidth.

Figure 3. Web user interface for remote monitoring and control

5. Shop Floor Security

According to an NCMS report [13], there is a growing consensus that linking shop floor hardware to the Internet will become the backbone technology for collaborative manufacturing. However, a major concern of implementing Internet or Web-based collaborative manufacturing systems is the assurance that proprietary information about the intellectual property owned by the organization or information about the company's operations is available only to authorized individuals. Any web-based collaborative systems must accommodate privacy of the individuals and organizations involved in collaborative activities. Gathering and processing information about the activities of individuals or groups while managing or operating processes or devices via computer networks can provide a great deal of detail concerning the ways in which the individuals interact as well as process-related information. In a highly competitive manufacturing environment, the information about the operations of or the information provided by individuals or organizations should only be shared by those involved. Clearly, it is also important to avoid security disasters for hardware at shop floor level. Web-based remote monitoring and control typically involve sharing information in the form of detailed run-time operations, as well as real-time and mission-critical hardware controls. For general acceptance of the *Wise-ShopFloor*, the secrecy of the proprietary information must be properly maintained. For security, our approach depends on the familiar security infrastructure built into the Java platform. This security architecture consists of byte-code verification, security policies, permissions, and protection domains. In addition to the security infrastructure, other security and privacy issues are considered in the framework for implementation, including digital rights management for information access and sharing, data encryption, and process confidentiality protection.

Figure 4 shows how a remote end user can get access indirectly to the real shop floor without violating shop floor security. All data communication between the end user and a shop floor device goes through the application server, and is processed by a server-side module before passing the data onto its receiver. As mentioned in Section 4, only the server-side modules are allowed to collect sensor data or manipulate devices within their limits. On the other hand, all end users are physically separated from the real shop floor by using segmented networks (Intranet/Internet, and Factory Network) with the application server as a gateway.

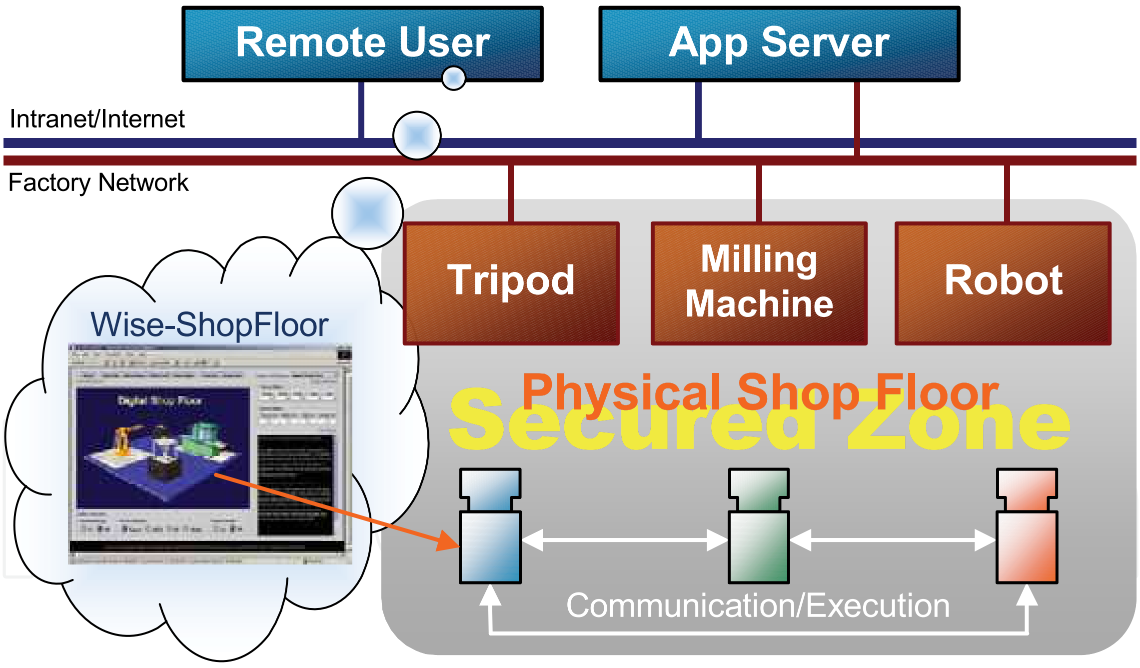


Figure 4. Indirect secure access to physical shop floor

6. Implementation

This section describes how a physical device is modelled, monitored, and controlled. The Tripod is a parallel kinematic machine developed at IMTI's lab [14]. Instead of camera images, the Tripod is modelled by using the scene graph-based interactive Java 3D with behavioural control nodes embedded. Once downloaded from the application server, it behaves in the same way as its physical counterpart for remote monitoring and control at client-side, facilitated by the model-embedded kinematics and sensor signals of the real Tripod.

6.1 Java 3D Modelling for Tripod

Java 3D is designed to be a mid to high-level fourth-generation 3D API [15]. What sets a fourth-generation API apart from its predecessors is the use of scene-graph architecture for organizing graphical objects in the virtual 3D world. Unlike the display lists used by the third-generation APIs (such as VRML, OpenInventor, and OpenGL), scene graphs can isolate rendering details from users while offering opportunities for more flexible and efficient rendering. Enabled by the scene-graph architecture, Java 3D provides an abstract, interactive imaging model for behaviour and control of 3D objects. Because Java 3D is part of the Java pantheon, it assures users ready access to a wide array of applications and network support functionality [16]. Java 3D differs from other scene graph-based systems in that scene graphs may not contain cycles. Thus, a Java 3D scene graph is a directed acyclic graph. The individual connections between Java 3D nodes are always a direct relationship: parent to child. Figure 5 illustrates a scene graph architecture of Java 3D for the Tripod. This test bed is a gantry system, which consists of an x-table and a Tripod unit mounted on a y-table. The end effecter on the moving platform is driven by three sliding-legs that can move along three guide-ways, respectively.

As shown in Figure 5, the scene graph contains a complete description of the entire scene with a virtual universe as its root. This includes the geometry data, the attribute information, and the viewing information needed to render the scene from a particular point of view. All Java 3D scene graphs must connect to a *Virtual Universe* object to be displayed. The *Virtual Universe* object provides grounding for the entire scene. A scene graph itself, however, starts with *BranchGroup* (BG) nodes (although only one BG node in this case). A *BranchGroup* node serves as the root of a sub-graph, or branch graph, of the scene graph. The *TransformGroup* nodes inside of a branch graph specify the position, the orientation, and the scale of the geometric objects in the virtual universe. Each geometric object consists of a *Geometry* object, an *Appearance* object, or both. The *Geometry* object describes the geometric shape of a 3D object. The *Appearance* object describes the appearance of the geometry (colour, texture, material reflection characteristics, etc.). The behaviour of the Tripod model is controlled by *Behaviour* nodes, which contain user-defined control codes and state variables. Sensor data processing can be embedded into the codes for remote monitoring. Once applied to a *TransformGroup* node, the so-defined behaviour control affects all the descending nodes. In this example, the movable objects (X-Table, Y-Table, and Moving Platform) are controlled by using three control nodes, for on-line monitoring/control and off-line simulation. As the Java 3D model is connected with its physical counterpart through the control nodes by low-volume message passing (real-time sensor signals and control commands, etc.), it becomes possible to remotely manipulate the real Tripod through its Java 3D model (see also [17]).

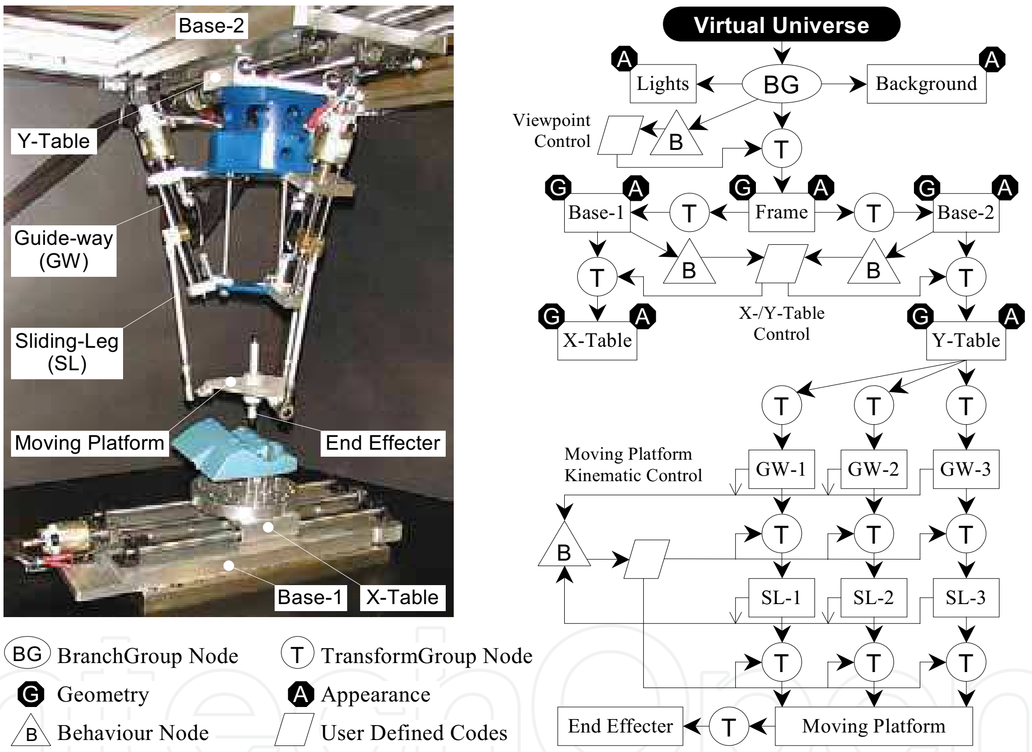


Figure 5. Java 3D scene graph architecture for Tripod

6.2 Kinematic Modelling for Tripod

Kinematics studies the geometric properties of the motion of points without regard to their masses or to the forces acting upon them. While the scene graph is the emergent standard hierarchical data structure for computer modelling of 3D worlds, kinematic models of physical devices or mechanisms that have external constraints or constraints that span interior nodes do not fit comfortably into its open-branched tree topology. In the case of our Tripod monitoring and control, models of both constrained kinematics and inverse kinematics are solved separately and embedded into the behaviour control nodes in a scene graph to calculate the motions of respective components. Typically, constraints can be expressed in a number of equations or inequalities that describe the relationships among Tripod components. Based on sensor signals collected from the real Tripod, both constrained kinematic model and inverse kinematic model of the Tripod are needed to calculate the positions and orientations of the three sliding-legs and moving platform for 3D Tripod model rendering.

For the purpose of mathematical formulation, a Tripod kinematic model is shown upside-down in Figures 6 and 7. It is a 3-dof parallel mechanism with linear motion component actuators, the type of linear motion actuated machines with fixed leg lengths and base joints movable on linear guideways (e.g. HexaM, Eclipse, Hexaglide, GeorgV, Z^3 Head). This mechanism consists of three kinematic chains, including three fixed length legs with identical topology driving by ballscrews, which connects the fixed base to the moving platform. In this 3-dof parallel mechanism, the kinematic chains associated with the three identical legs consist, from base to platform, of an actuated linear motion component (ballscrew in the case), a revolute joint (connected by a nut), a fixed length moving link, and a spherical joint attached to the moving platform. The arrangement of the structure would be subject to bending in the direction parallel to the axis of the revolute joint. The advantages of the structure are: 1) with this basic structure of parallel mechanism, it can be easily extended to 5-dof by adding two gantry type of guideways to realize the 5-dof machining; 2) with the fixed length legs, one can freely choose the variety of leg forms and materials and use linear direct driver to improve the stiffness; and 3) due to reduced heat sources, it is possible to keep the precision in a high level and to maintain a stable stiffness if compared with variable legs.

The kinematic equation for the position of the *i*th spherical joint is given as

$$\mathbf{p}_i = \mathbf{h} + \mathbf{R}\mathbf{p}'_i \tag{1}$$

where, $\mathbf{p}_i = [\, p_{ix}, p_{iy}, p_{iz} \,]^T$ is the vector representing the position of the *i*th joint in the global coordinate system O-*xyz*, $\mathbf{p}'_i$ is the vector representing the same point but in the local coordinates $C\text{-}x'y'z'$, $\mathbf{h} = [\, x_c, y_c, z_c \,]^T$ is the vector representing the position of the moving platform, and **R** is the rotation matrix of the moving platform in terms of rotation angles θ_x, θ_y, and θ_z about x, y, and z axis, respectively.

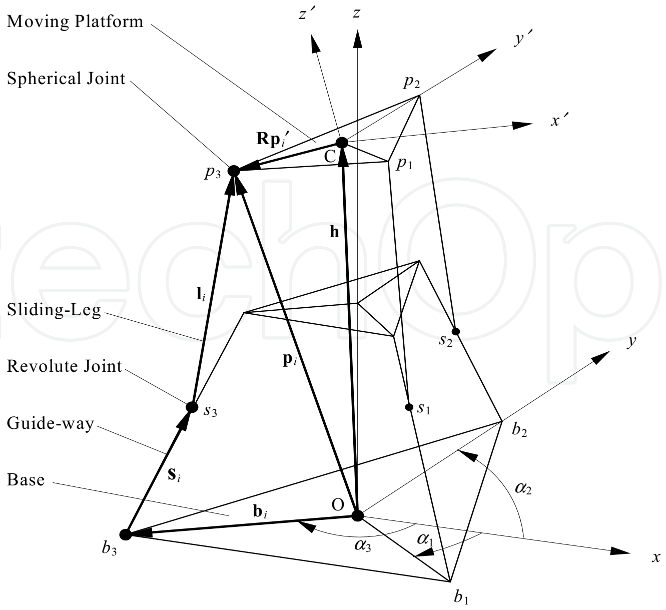


Figure 6. Tripod kinematic model

Among the six motion components of the moving platform, it is known that x_c, y_c, and θ_z are dependent variables. The constraint equations can be derived as

$$x_c = -\frac{\sqrt{3}}{3} L_p \cos\theta_y \sin\theta_z \tag{2a}$$

$$y_c = \frac{\sqrt{3}}{6} L_p \left(-\cos\theta_y \cos\theta_z + \cos\theta_x \cos\theta_z - \sin\theta_x \sin\theta_y \sin\theta_z\right) \tag{2b}$$

$$\theta_z = \arctan\left(-\frac{\sin\theta_x \sin\theta_y}{\cos\theta_x + \cos\theta_y}\right) \tag{2c}$$

While constrained kinematics of the Tripod is used for monitoring, inverse kinematics is needed for position control. Considering the *i*th sliding-leg/guide-way system, the kinematic equation of the position of the *i*th spherical joint, i.e. eq. (1), can be re-written as

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{s}_i + \mathbf{l}_i \tag{3}$$

where $\mathbf{b}_i$ is the vector representing the position of the lower end of the *i*th guide-way attached to the base, $\mathbf{s}_i$ is the vector representing the displacement along the *i*th guide-way, and $\mathbf{l}_i$ is the vector representing the *i*th sliding leg.

Since $\mathbf{s}_i = s_i \mathbf{u}_i^s$ and $\mathbf{l}_i = l_i \mathbf{u}_i^l$, where $\mathbf{u}_i^s$ and $\mathbf{u}_i^l$ are the direction vectors of the *i*th guide-way and the *i*th leg, respectively, the actuator displacement s_i can be solved considering that the leg length is a constant

$$\left| \mathbf{p}_i - \mathbf{b}_i - s_i \mathbf{u}_i^s \right| = l_i \tag{4}$$

where l_i is the length of the *i*th sliding leg. For given θ_x, θ_y, and z_c, dependent variables x_c, y_c, and θ_z can be determined by eqs. (2a) - (2c), then **h** and **R** are fully defined. With this, $\mathbf{p}_i$ can be determined by eq. (1), and subsequently s_i can be solved using eq. (4). The true solution of eq. (4) should be the one closer to the previous value, that is

$$s_i = \left(s_i^{(k)}, \quad \min_{k=1,2} \left| s_i^{(k)} - s_i(j-1) \right| \right) \tag{5}$$

where *j* stands for the *j*th step. In practice, the initial value of s_i is provided by an encoder. For the sake of brevity, interested readers are referred to [18] for further details of the Tripod kinematics.

6.3 Remote Monitoring and Control

Web-based remote device monitoring and control are conducted by using the *StatusMonitor* and *CyberController*, which communicate indirectly with the device controller through an application server. In the case of Tripod monitoring and control, they are further facilitated by the kinematic models, to reduce the amount of data travelling between web browsers and the Tripod controller.

Figure 7. CAD model of Tripod

The required position and orientations of the moving platform are converted into the joint coordinates s_i ($i = 1, 2, 3$) by the inverse kinematics for both Java 3D model rendering at client-side and device control at server-side. The three sliding-legs of the Tripod are driven by three 24V DC servomotors combined with three lead screws. Each actuator has a digital encoder (1.25 μm/count) for position feedback. The position data s_i ($i = 1, 2, 3$) of the sliding-legs are multicast to the registered clients for remote monitoring, while only one user at one time is authorized to conduct remote control. A sampling rate of 1 kHz is used for the case study. Figure 8 shows how the Tripod is manipulated from one state to another within the proposed *Wise-ShopFloor* framework.

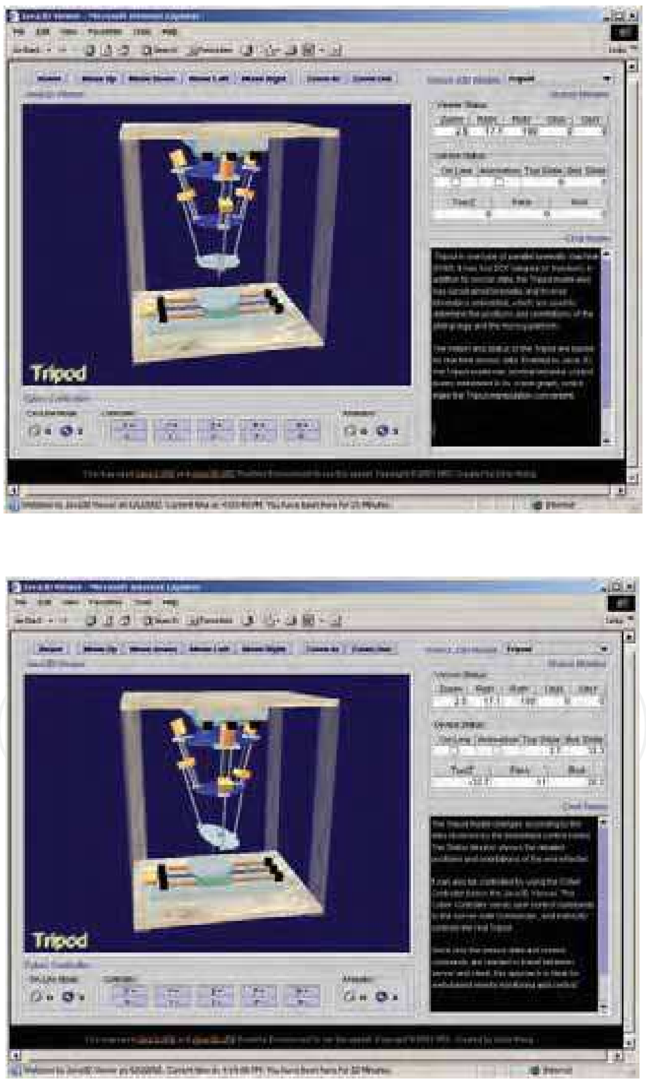


Figure 8. Web-based remote monitoring and control

6.4 Managerial Implications

The *Wise-ShopFloor* is a business process that is based on new IT technology to execute business processes. It leverages the IT management tools to deliver reliable and secured transmission of data between the end users and real shop floors. It provides not only an efficient mechanism for real-time monitoring and control in manufacturing, but it also improves a manufacturing firm's business performance. The implementation of this client-server architecture is likely to result in significant increases in productivity and revenues.

7. Conclusions

This chapter presents the *Wise-ShopFloor* framework and describes detailed three-tier architecture. The goal of the web-based approach is to reduce network traffic with Java 3D models, while still providing users with intuitive environments. Participating in the *Wise-ShopFloor*, users not only can feel reduced network traffic by real-time interactions, but also can obtain more flexible control of their real shop floors. The application in modern manufacturing system is demonstrated for its feasibility and the promise of this novel approach to the growing distributed shop floor environments. As decentralization of business grows, a large application potential of this research is anticipated, in addition to remote real-time monitoring and control.

8. References

Cao, J., Li, M.L., Zhang, S.S. and Den, Q. N. 2004. Composing Web Services Based on Agent and Workflow. Grid and Cooperative Computing, Part 1. Berlin: Springer-Verlag Berlin, pp. 948-955.

Zeng, L. Benatallah, B., Ngu, A. H. H., Dumas, M., Kalagnanam, J., and Chang, H. 2004. QoS-Aware Middleware for Web Services Composition", IEEE Transactions on Software and Engineering, May. 30(5): 311-327.

G. Pritschow, "Research and Development in the Field of Parallel Kinematic Systems in Europe", Parallel Kinematic Machines – Theoretical Aspects and Industrial Requirements, edited by Boër, C.R., Molinari-Tosatti, L, and Smith, K.S., pp.1-16, Springer-Verlag, (1999).

J. Tlusty, J. Ziegert, and S. Ridgeway, "Fundamental Comparison of the Use of Serial and Parallel Kinematics for Machine Tools", Annals of the CIRP, Vol. 48/1, pp. 351-356, (1999).

M. Honegger, A. Codourey, and E. Burdet, "Adaptive Control of the Hexaglide, a 6 DOF Parallel Manipulator", Proceedings of the 1997 IEEE

International Conference on Robotics and Automation, Vol. 1, pp. 543-548, (1997).

G. Pritschow, and K.-H. Wurst, "Systematic Design of Hexapods and Other Parallel Link Systems", Annals of the CIRP, Vol. 46/1, pp. 291-295, (1997).

M. Suzuki, K. Watanabe, T. Shibukawa, T. Tooyama, and K. Hattori, "Development of Milling Machine with Parallel Mechanism", Toyota Technical Review, Vol. 47 No. 1, pp. 125-130, (1997).

F. Pierrot, "From Hexa to HexaM", International Parallelkinematik-Kolloquium IPK'98, ETH Zurich, pp. 75-84, (1998).

H.K. Tönshoff, C. Soehner, and H. Ahlers, "A New Machine Tool Concept for Laser Machining", Proceedings of International Seminar on Improving Machine Tool Performance, San Sebastian, pp.199-124, (1998).

B.S. El-Khasawneh, and P.M. Ferreira, "The Tetrahedral Tripod", Parallel Kinematic Machines – Theoretical Aspects and Industrial Requirements, edited by Boër, C.R., Molinari-Tosatti, L, and Smith, K.S., pp. 419-430, Springer-Verlag, (1999).

Kochan, A., "Parallel Robots Perfect Propellers", Industrial Robot, Vol. 23, No. 4, pp. 27-30, (1996).

L. Wang, B. Wong, W. Shen and S. Lang, "Java 3D Enabled Cyber Workspace", *Communications of the ACM*, Vol.45, No.11, pp. 45 – 49, 2002.

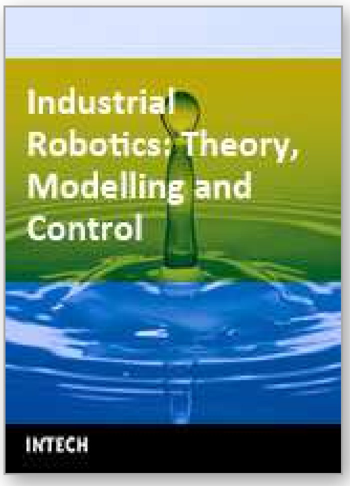
NCMS, "Factory-Floor Internet: Promising New Technology or Looming Security Disaster", *Manufacturing In Depth*, National Center for Manufacturing Sciences, November, 2001.

Dan Zhang, L. Wang and Sherman Y. T. Lang, 2005, Parallel Kinematic Machines: Design, Analysis and Simulation in an Integrated Virtual Environment, Transactions of the ASME Journal of Mechanical Design, Vol.127, Issue 7, pp. 580-588

J. Barrilleaux, *3D User Interfaces with Java 3D*, Manning Publications Co., 2001.

H. Sowizral, K. Rushforth and M. Deering, *The Java 3D API Specification*, Addison-Wesley, 2001.

L. Wang, F. Xi, D. Zhang and M. Verner, "Design Optimization and Remote Manipulation of a Tripod", International Journal of Computer Integrated Manufacturing, Vol.18, No.1, pp.85-95, 2005



Industrial Robotics: Theory, Modelling and Control

Edited by Sam Cubero

ISBN 3-86611-285-8
Hard cover, 964 pages
Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

How to reference

Dan Zhang, Lihui Wang and Ebrahim Esmailzadeh (2006). Web-Based Remote Manipulation of Parallel Robot in Advanced Manufacturing Systems, Industrial Robotics: Theory, Modelling and Control, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech, Available from: http://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/web-based_remote_manipulation_of_parallel_robot_in_advanced_manufacturing_systems

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821