Available online at www.sciencedirect.com
ScienceDirect

Computer Standards & Interfaces 30 (2008) 36 – 51

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Feasibility and infrastructural study of AR interfacing and intuitive simulation on 3D nonlinear systems

Seungjun Kim [a], N.P. Mahalik [b,*], Anind K. Dey [a], Jeha Ryu [c], Byungha Ahn [c]

[a] Human-Computer Interaction Institute (HCII), Carnegie Mellon University (CMU), USA
[b] Department of Industrial Technology, California State University - Fresno, USA
[c] Department of Mechatronics, Gwangju Institute of Science and Technology (GIST), Republic of Korea

Received 24 October 2005; received in revised form 9 March 2007; accepted 2 July 2007
Available online 20 July 2007

Abstract

High-fidelity simulations with intuitive user interfaces expedite real-world tasks. Augmented Reality (AR), enables 3D computer-generated entities to coexist in users' view of their real workspace. In this paper, the AR methodology is combined with custom-built 3D simulators of two nonlinear systems that show different motion characteristics: robot and spacecraft. The target systems are modeled on the basis of system parametric datasets, and their motions are specified and graphically represented according to characteristics of applied control algorithms. In our proposed system architecture, an AR process is incorporated with a 3D graphic engine and other functional modules for system analysis, simulation and control. Experiments demonstrate a feasibility study as well as intuitive 3D simulation performance and interactive AR interface features.

Keywords: Augmented reality; Interface; 3D simulation; Robot; Spacecraft

1. Introduction

With the high growth in the fields of software engineering and graphics, novel GUI (Graphical User Interface) technologies have been used to design intuitive HMIs (Human–Machine Interfaces) that can effectively connect humans with workspaces and equipment. Well-designed interfaces not only can help users take maximum benefit of their computing systems but also can enhance their real-world tasks. Due to the diversification of products, shortened product lives, flexible adaptation of manufacturing systems in the workplace is crucial. It follows that simulation of these systems before introducing them in the workplace is indispensable. Simulations are key for ensuring high-fidelity to target systems frequently used in the manufacturing and industrial sectors.

Research on both simulators and interfaces have been conducted for many different types of systems. In the industrial and aerospace arenas, the extension of this systems research to real environments is necessary but so sensitive to cost and errors that pre-simulation has been considerably stressed. Michel et al. demonstrated a MVC (Model–View–Control) approach for the GUI design of a cleaning mobile robot simulator in [1]. Chen et al. developed a gesture-speech based HMI for a rehabilitation robot [2]. These efforts made us note that an *intuitively-interactive* human interface in user's observation and control will help evaluate the reliability and applicability of a high-fidelity simulation task. In addition, the plant operations in virtual environments have been performed by user-centered management and control [3]. It proved the how VR (Virtual Reality) technologies can be exploited in designing human–machine interfaces beyond animation-based simulation. Specification requirements of a simulation environment for rapid control prototyping (RCP) of real-time distributed control of mechatronic systems have been also developed by Mahalik et al [4,5], which has let us remind the fact that high-fidelity simulation can expedite real-world tasks. In the aerospace arena, spacecraft simulators have been widely developed for the

* Corresponding author.
E-mail addresses: sjunikim@cs.cmu.edu, sjuni.kim@gmail.com (S. Kim), nmahalik@csufresno.edu, nmahalik@yahoo.co.uk (N.P. Mahalik), anind@cs.cmu.edu (A.K. Dey), ryu@gist.ac.kr (J. Ryu), bayhay@gist.ac.kr (B. Ahn).

0920-5489/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2007.07.007

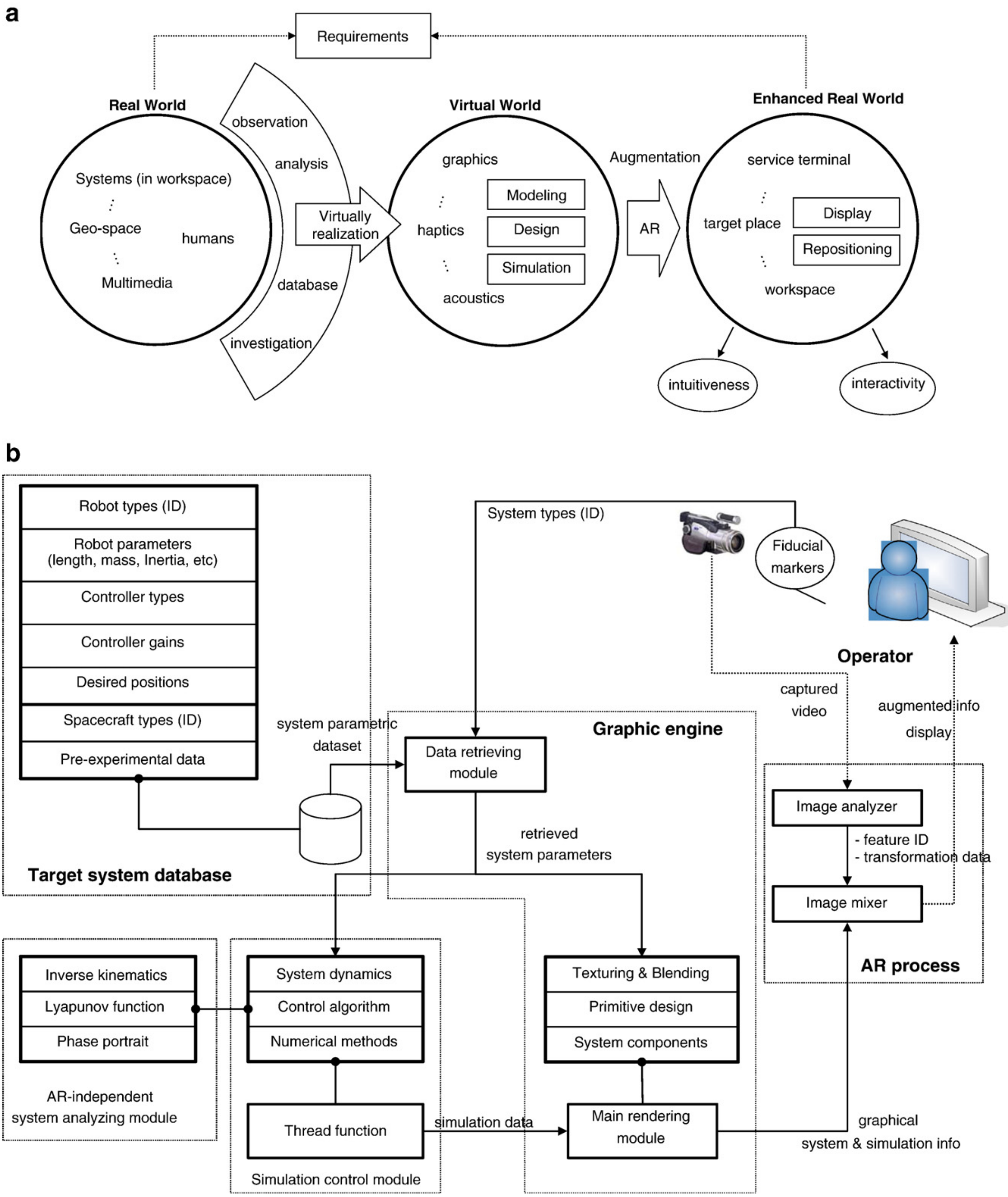


Fig. 1. System architecture.

performance test of structure modules as well as the reduction of development expenses and risks, e.g. SKT (Satellite Tool Kit), SSF (Spacecraft Simulation Framework) of ICS (Interface and Control System, Inc), DynaWiz XSV package for multi-body satellite, HSSS (High Speed Spacecraft Simulator), etc. We can also note that software infrastructure of a simulator need to be constructed coinciding with the hardware components of the target systems.

However, most present simulators separate computing systems from real workspaces. For example, we may need to simulate custom-built 3D virtual (computer-generated) robots visually abreast with real ones for the intuitive evaluation of

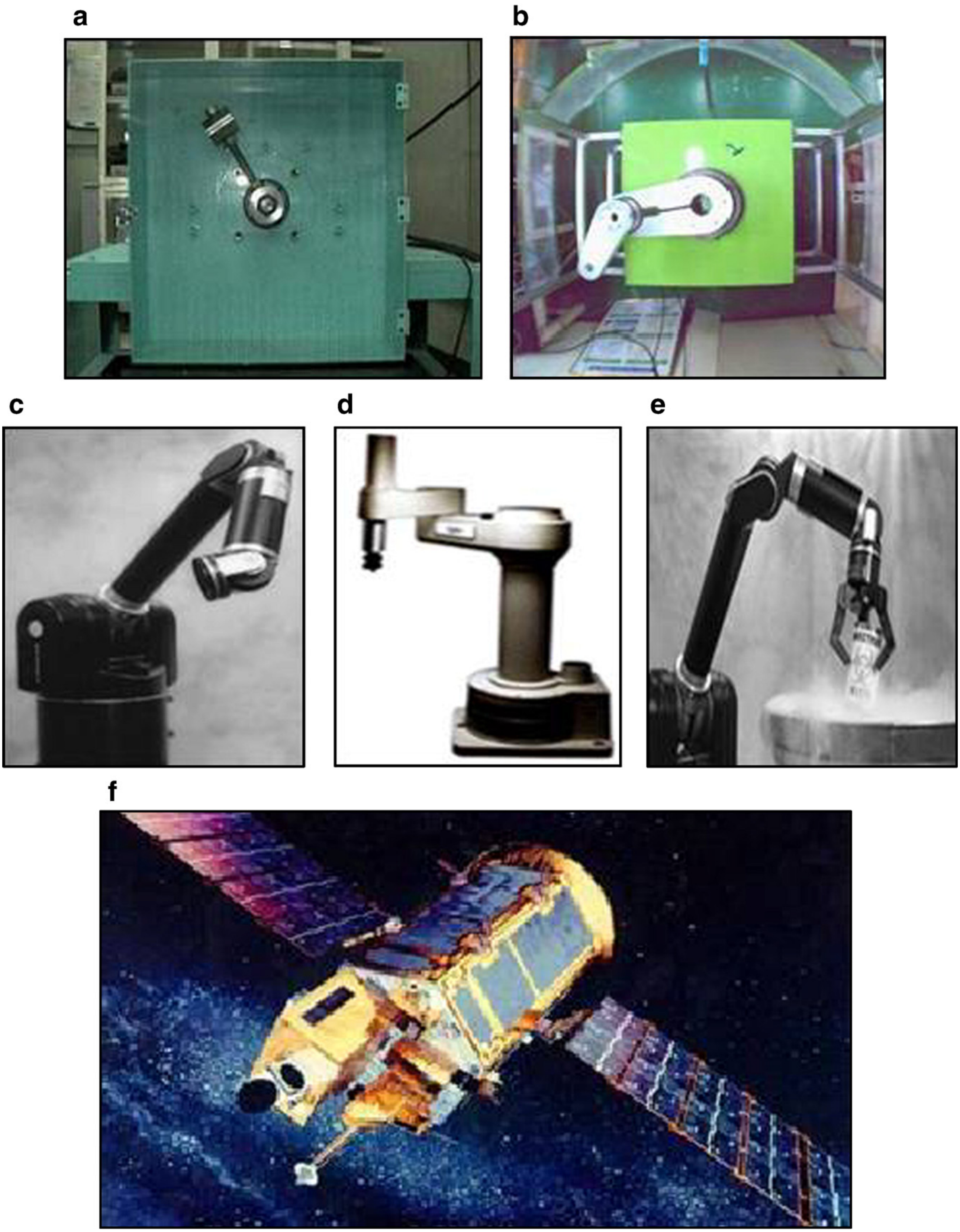


Fig. 2. Target models: (a) Custom-built one-link manipulator (b) 2-link SCARA type robot (c) 2-link vertical type robot (WAM ARM of Barrett Technology, Inc) (d) 3-link SCARA type robot (e) 3-link elbow type robot (f) Spacecrafts of gas-jet and momentum-exchange actuator type (KOMPSAT— Korean Multi-purpose Satellite).

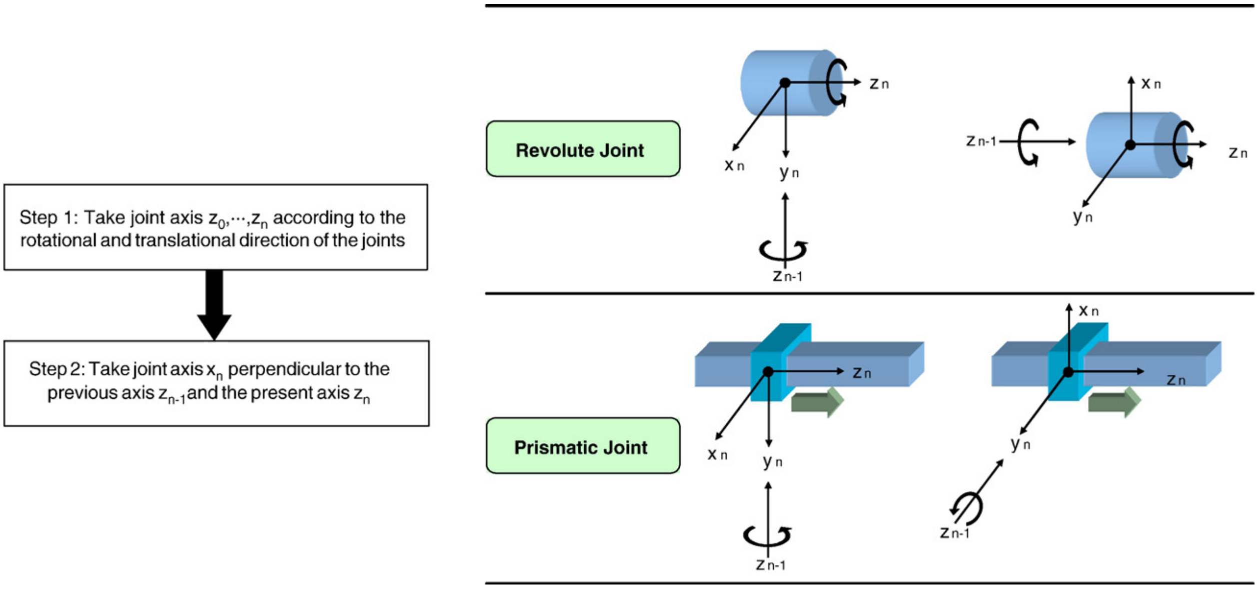

Step 1: Take joint axis $z_0,\cdots,z_n$ according to the rotational and translational direction of the joints
Step 2: Take joint axis x_n perpendicular to the previous axis z_{n-1} and the present axis z_n
Revolute Joint
Prismatic Joint
z_n
x_n
y_n
z_{n-1}

Fig. 3. Denavit–Hartenberg frame assignment.

their working in a real factory setting. In order to make the virtual objects coexist seamlessly in a user's view of the real environment, the simulators in the computing space need to be pulled into the real world. To satisfy this need, we propose the use of Augmented Reality (AR) [6–13]. Navab's work in AR registered 3D models of pipelines with the user's view of a factory under consideration, taking into account the 2D factory floor plans and the structural properties of industrial pipelines [9,10]. Rosenthal et al. have shown needle biopsies using a 3D AR guidance system [11]. Schwald and Blandine have presented an AR-based training and assistance system for the maintenance of industrial areas [12]. In a recent effort, Gelenbe et al. [13] mixed the virtual domain with the AR domain in real-time, in order to examine how novel simulated conditions can interact with a real system's operation. They inserted synthetic moving objects into live video in real-time. In keeping with these related works, we believe that a variety of 2D or 3D information conveyed by virtual images will aid the user's perception of target systems and the user's real-world tasks.

In our research, we leverage interactive 3D simulation using AR methodology for two different types of nonlinear systems: Robots and Spacecraft. In other researches, virtual objects are superficially simulated in the form of prescripted animation or recorded video only focusing on attractive display. In this paper, however, virtual models are designed using parametric datasets of the target systems for high-fidelity simulation. These models are simulated based on their dynamics and control algorithms. To perform target system

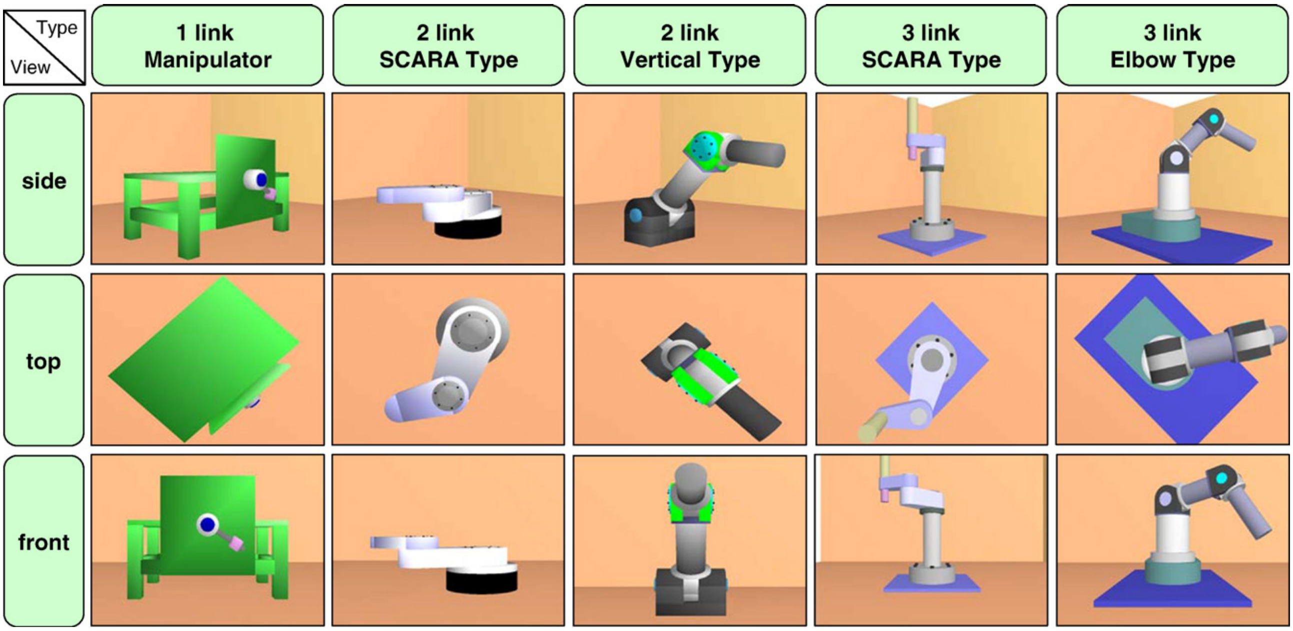


Fig. 4. Virtual (computer-generated) robots.

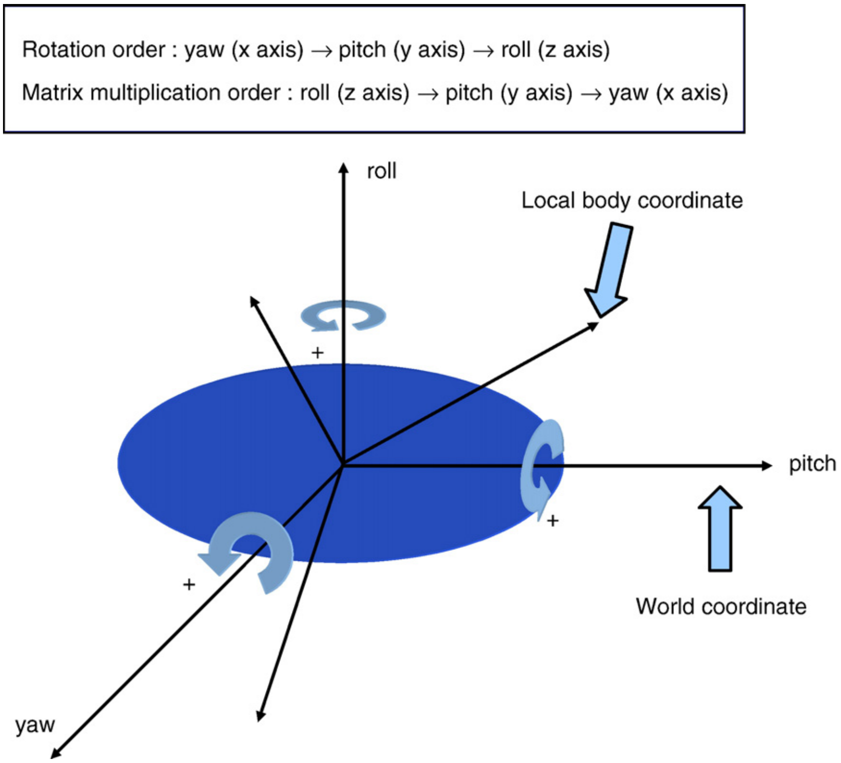

Rotation order : yaw (x axis) → pitch (y axis) → roll (z axis)
Matrix multiplication order : roll (z axis) → pitch (y axis) → yaw (x axis)
roll
Local body coordinate
+
pitch
+
World coordinate
+
yaw

Fig. 5. Relationship between roll–pitch–yaw angles and graphic transformations.

simulation using AR methodology, we have done the following:

- Target model selection and 3D graphical representation: system component analysis for five types of robotic systems & two types of spacecrafts, and 3D graphic engine design
- High-fidelity simulation: dynamics & kinematics modeling, and real-time simulation utilizing numerical method & thread function
- AR interfacing & experimental validation: noise filtering, conformant marker configuration, control & stability tests, 2D & 3D information visualization, and end-user interaction

The organization of this paper is as follows. The next section describes target models and the proposed system architecture by functional modules. Section 3 presents systematic and graphic modeling of the target systems. It also includes the technical implementation including numerical methods, noise filtering and collaboration tests in AR environment. The last section deals with experimental demonstration for validation. The paper ends with conclusion and future work.

2. System overview

The proposed overall system configuration is as shown in Fig. 1(a). Real world elements interfaced with human, target models are selected for representation in a virtual world. Computer graphics represent the complicated configuration of realistic models of virtual characters in the 3D domain. Most AR work has been devoted on augmenting visual aspects using graphics and display equipment rather than other sensation aspects such as touch or sound, using haptics or acoustics. In our test-bed, the main operations from graphical modeling to AR-incorporated simulation are performed through three main functional modules: target system database, graphic engine, and AR process. There are two additional computing modules used for simulation control and result analysis, as presented in Fig. 1(b).

As shown in Fig. 1(b), the target system database component produces a parametric dataset for the target system. In this paper, we consider the following target systems: five types of robot systems and two types of spacecraft systems with differering joint configuration and actuator types, respectively. For the robot systems, a typical dataset is composed of robot type identifier, robot parameters, controller properties and target positions of links. Graphical models and simulation control properties are determined by retrieving the appropriate datasets. For the spacecraft systems, the actuator type and pre-experimental data are taken into consideration. The graphical spacecraft models can be animated according to pre-stored experimental data. Examples f the target models are shown in Fig. 2.

Video captured by cameras is analyzed in the AR process component, using the ARToolkit, regarded as one of the reliable *de facto* in prototyping AR applications [14], to detect area of fiducial features. Fiducial features refer to patterned markers which the system has been well trained to recognize and track. Appropriate target models and datasets are automatically used, when matching fiducial features are detected. The 3D spatial relationships between the camera and the features are based on 4 × 4 transformation matrix set from which the position and pose of virtual models are derived. The dataset retrieved is sent to the simulation control module and graphics engine. The simulation control module contains system dynamics and control

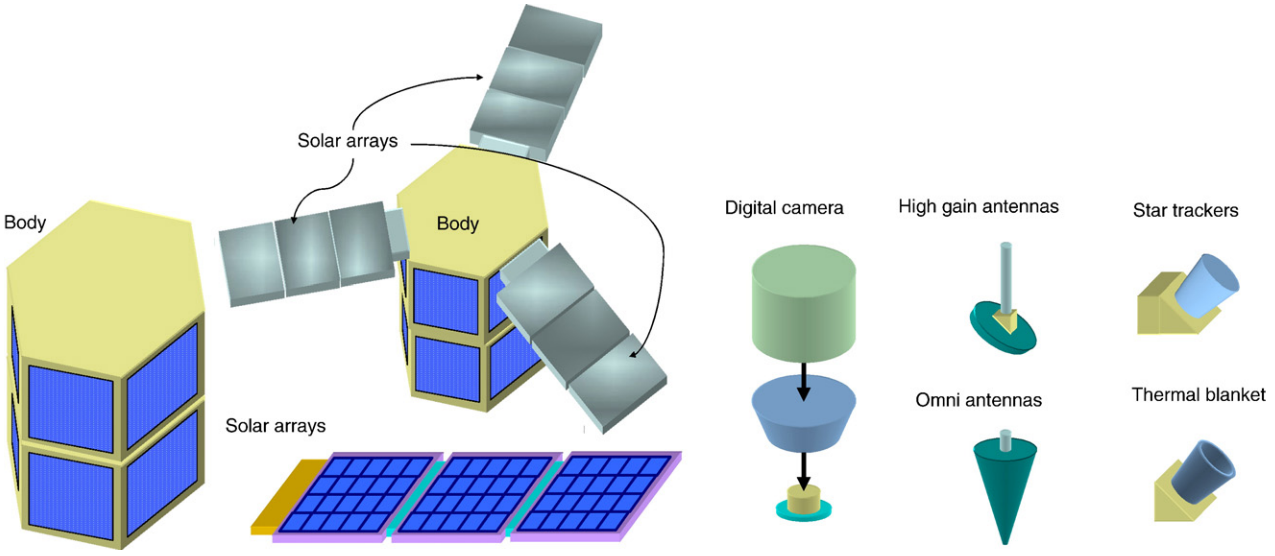


Fig. 6. Graphic components of the spacecraft.

algorithms. To find numerical solutions for differential equations of system dynamics, the 2nd order Runge–Kutta method is applied. Virtual models can be simulated independently of the rendering frequency of AR process. The graphics engine facilitates graphical modeling of primitives and system components based on system parameters such as system type, link length, load existence, etc. This engine creates all graphical effects including annotation and information panels.

Finally, the AR process provides mixed images, merging virtual models with the real environment. By arranging fiducial features, user can interact with virtual target models in the real environment scene. 3D experimental simulations are controlled with a mouse, by analyzing its location with respect to the spatial relationship between world and camera screen coordinate are analyzed. In addition, useful functions for robot system analysis such as inverse kinematics and stability test using the Lyapunov function and phase portraits are available independently on the AR service.

3. Modeling and implementation

3.1. Robotic system

The dynamic modeling and control algorithms is presented in the sequel. The motions are generated on the basis of the system dynamics and kinematics. We used the ideal Euler–Lagrange equations for robot dynamics Eq. (1).

$$\text{Matrix form}: \ \mathbf{D}(\mathbf{q})\ddot{q} + \mathbf{C}(\mathbf{q},\dot{q})\dot{q} + \mathbf{g}(\mathbf{q}) = \tau \Rightarrow \sum_j d_{kj}(\mathbf{q})\ddot{q}_j + \sum_{i,j} c_{ijk}(\mathbf{q})\dot{q}_i\dot{q}_j + \phi_k(\mathbf{q}) = \tau_k, \tag{1}$$

$$\text{where } c_{kj} = \sum_{i=1}^{n} c_{ijk}(\mathbf{q})\dot{q}_i = \sum_{i=1}^{n} \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\} \times \dot{q}_i, k = 1,\ldots,n$$

$\mathbf{D}(\mathbf{q})$ is the robot inertia matrix which is a symmetric and positive definite matrix ($\mathbf{q}$ is the vector of the joint angles). $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ represents the Coriolis and centrifugal forces ($\dot{q}_i^2$ are called centrifugal and $\dot{q}_i\dot{q}_j$ where $i \neq j$ are called Coriolis terms). The terms c_{ijk} are known as Christoffel symbols. $\mathbf{g}(\mathbf{q})$ is the gravitational forces which can be derived from the potential energy, and $\boldsymbol{\tau}$ is the vector of the input torques. By deriving the friction force, we can give provide a modified input torque to compensate for friction.

$$F(\dot{\mathbf{q}}) = (\mathrm{F}_v|\dot{\mathbf{q}}| + F_c)\mathrm{sgn}(\dot{\mathrm{q}}) \tag{2}$$

where F_v=diag(f_{v1}, f_{v2},..., f_{vn}) denotes the viscous friction coefficient matrix and F_c=diag(f_{c1}, f_{c2},..., f_{cn}) denotes the Coulomb friction coefficient matrix. Two kinds of control algorithms such as PD controller Eq. (3) and output feedback controller Eq. (4) developed by Ailon [15] were used. Here the constant vector $\mathbf{v}$ is updated only according to the present positional state of the robot links.

$$\tau = K_p(\mathbf{q_d} - \mathbf{q}) + K_d(\dot{q}_{\mathbf{d}} - \dot{q}) \tag{3}$$

$$\tau = -S(\mathbf{q}_1 - z) \Leftarrow \dot{z} = -S(z - \mathbf{q}_1) - Rz + v \tag{4}$$

In addition, the Lyapunov stability for target robot systems can be tested in AR independent simulation modules. The Lyapunov candidate functions are given as Eq. (5) and Eq. (7) corresponding to each controller. The function V in quadric form is always positive definite and its derivative is negative semidefinite.

$$V = \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{D}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}(\mathbf{q}_{\mathrm{d}} - \mathbf{q})^{\mathrm{T}}\mathrm{K}_{\mathrm{p}}(\mathbf{q}_{\mathrm{d}} - \mathbf{q}) \Rightarrow \dot{\mathrm{V}} = -\dot{\mathbf{q}}^{\mathrm{T}}K_d\dot{q} \leq 0 \tag{5}$$

$$H_r(\mathbf{q},\dot{q},z) = \frac{1}{2}[\dot{\mathbf{q}}^{\mathbf{T}}\mathbf{D}(\mathbf{q})\dot{\mathbf{q}} + (\mathbf{q} - \mathrm{z})^T\mathrm{S}(\mathbf{q} - \mathrm{z}) + (\mathrm{z} - \mathrm{R}^{-1}\mathrm{v})^T\mathrm{R}(\mathrm{z} - \mathrm{R}^{-1}\mathrm{v})] + \mathrm{U_g}(\mathbf{q}) \tag{6}$$

$$\text{where} \quad \mathrm{g}(\mathbf{q})^{\mathrm{T}} = \frac{\partial U_g(\mathbf{q})}{\partial \mathbf{q}}$$

$$V(\mathbf{q},\dot{q},z)=H_r(\mathbf{q},\dot{q},z)-H_r(\overline{q},0,\overline{z})\Rightarrow\dot{V}(\mathbf{q},\dot{q},z)=-\dot{z}\dot{z}^T\leq 0 \tag{7}$$

3.2. Spacecraft system

The kinematics of spacecraft is expressed in Eq. (8) where $\varpi=[\varpi_1\ \varpi_2\ \varpi_3]^{\mathrm{T}}$ denotes the angular velocity of spacecraft expressed in the spacecraft frame which is attached to the center of mass, $R\in\mathrm{SO}(3)$ (SO(3) stands for the Special Orthogonal group of order 3), and $S(\varpi)\in\mathrm{SS}(3)$ (SS(3) denotes the set of Skew Symmetric matrices of order 3) [16]. The dynamics of the spacecraft is derived as shown in Eqs. (9) and (11) according to its actuator type, gas-jet or reaction wheel.

$$\dot{R}=S(\varpi)R\quad\Leftarrow\quad S(\varpi)=\begin{bmatrix}0 & \varpi_3 & -\varpi_2\\ -\varpi_3 & 0 & \varpi_1\\ \varpi_2 & \varpi_1 & 0\end{bmatrix} \tag{8}$$

3.2.1. Gas-jet actuator type spacecraft

J is the central inertia matrix, $\tau=[\tau_1\ \tau_2\ \tau_3]^T$ is the applied torque, and $\xi=[\varphi\ \theta\ \psi]^T$ is the roll–pitch–yaw (RPY) angles.

$$\dot{\xi}=\gamma(\xi)\varpi\quad\Leftarrow\dot{\varpi}=J^{-1}S(\varpi)J\varpi+J^{-1}\tau \tag{9}$$

3.2.2. Momentum-exchange actuator type spacecraft

$\Lambda_s=\mathrm{diag}[I_1\ I_2\ I_3]$ is the inertia matrix excluding the wheels, B is the moment of inertia of each wheel with respect to the rotation axis, $\Omega=[\Omega_1\ \Omega_2\ \Omega_3]^T$ is the angular velocity vector of reaction wheels, $L=\Lambda_s+2\mathrm{diag}[A\ A\ A]$ where A is the moment of inertia with respect to the other two principal axes, and J_i is the inertia matrix of ith wheel. Lastly, we applied an output controller as shown in Eq. (10) [17,18]. A selected triplet $\{K, M, N\}$ represents the controller matrices in diagonal form and v is a feed forward term. All of the experimental data for spacecraft is already stored in the target system database after MATLAB simulations. To show situational simulations that differ from the robot system, the graphical spacecraft are simply animated on the basis of this pre-experimental data.

$$\tau=-\gamma(\xi)KN[\xi-r]\quad\Leftarrow\quad\dot{r}=-(N+M)r+N\xi+v \tag{10}$$

$$\dot{\xi}=\gamma(\xi)\varpi\quad\Leftarrow\quad\dot{\varpi}=L^{-1}S(\varpi)\left(\left[\Lambda_s+\sum_{i=1}^{3}J_i\right]\varpi+B\Omega\right)+L^{-1}\tau \tag{11}$$

3.3. Numerical method for simulation: Runge–Kutta method

The 2nd order Runge–Kutta method [19] is applied to find numerical solutions as far as system dynamics are concerned. The system states x_{n+1} after a time step of h can be obtained by the previous state x_n at time t_n as per Eq. (12) where $n\geq 0$. The average slope $F(t_n, x_n; h)$ between time t_n and t_{n+1} is used.

$$x_{n+1}=x_n+hF(t_n,x_n;h)$$

$$x_{n+1}=x_n+\frac{h}{2}[f(t_n+x_n)+f(t_n+h,x_n+hf(t_n,x_n))] \tag{12}$$

3.4. Graphical components

For the virtual representation, we have used OpenGL which enables a perspective view, light effects, materials as well as

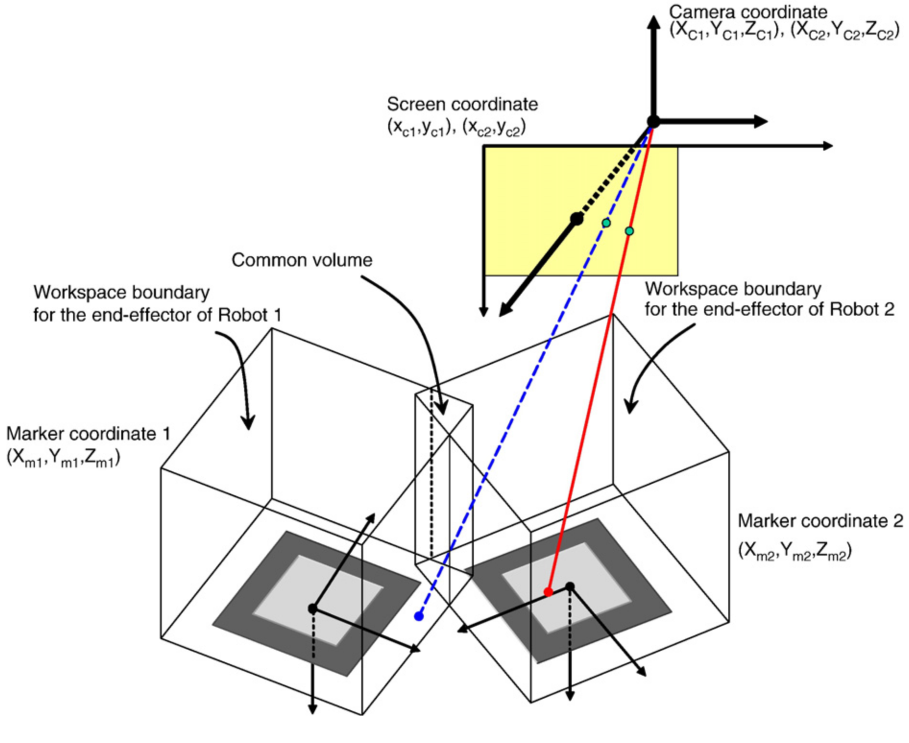


Fig. 7. Relationship of coordinate frames for robot collaboration in AR environment.

detailed object modeling in 3D domain. The rendering [20] is mainly processed by hardware, which is much faster than performing this in software. Note that the design of virtual objects by using the combination of vertices, lines and faces can create significant redundancy in modeling software [20]. For this reason, we have modeled several graphic primitives which can be frequently rendered so that we just have to consider the combination of their coordinate frames simply calling these primitives.

Fig. 3 shows the Denavit–Hartenberg (DH) frame assignment of robot joints and the procedure, which is conventionally used for selecting frames of reference. We have followed this DH frame assignment also in graphical configuration. It allows us to easily analyze the 3D objects by joints despite complex transformation executions.

Using joint features and the graphic primitives, five kinds of robot systems are designed in the graphic engine. Fig. 4 shows the gallery of the virtual robots in three different views: front, top and side. The virtual robots are composed of a base frame, links, joints and cylindrical loads. Their outlines are flexibly modifiable according to the retrieved information such as masses and lengths. In the AR independent simulator which is the former version of the proposed simulator in this paper, 2D graphs were used for state reporting [21]. Link positions, angular velocities, input torques and stability test results can be monitored using those graphs. In that case, the variations of simulation time and window size have to be carefully specified for proper visualization. In the AR-based simulator, however, a 2D panel is provided for reporting system features and annotation panels for each robot joint. To preserve global awareness of the real environment scene, a blending technique discussed in [7] is applied to the panels, and their viewing volumes are independent of each other.

Most robot systems have a prop base so that the relative motion of joints can be easily perceived. In the case of a spacecraft, however, its body is floating in 3D void space. For an intuitive visualization, the relationship between the world and local coordinate frames has to be well defined. Specifically, three rotational motions (roll–pitch–yaw) of the spacecraft must be supported around an arbitrary orientation. Hence, the order of transformations in graphics has to be arranged in the reverse order as the real rotations for the appropriate animation of the spacecraft motion as shown in Fig. 5. As a supplement, the present angles are visually reported on the three circular planes which aid in understanding the spacecraft motion.

The graphical components of the virtual spacecrafts are selected after analyzing the outline of the target satellite

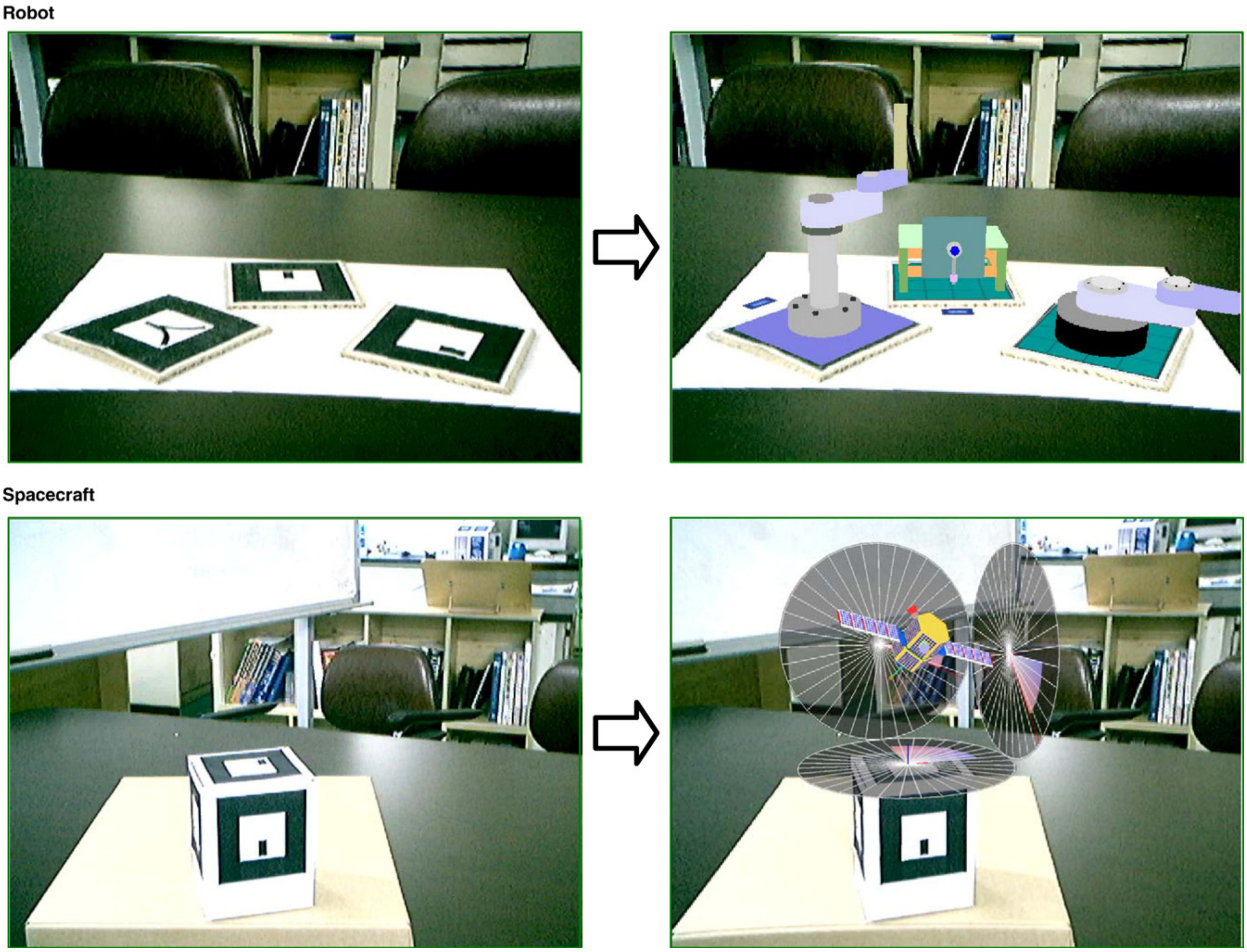


Fig. 8. Pattern configuration of real scene.

KOMPSAT model as shown in Fig. 6. The main component is a body frame where three solar arrays are attached and other specific components are arranged according to the actuator types being used. In the AR-based simulator, the graphical nameplate of each component is activated by keyboard interaction.

Table 1
1-link manipulator parameters

Link length(l)	0.2 m
Link mass(m)	1.8 kg
Output feedback controller gains(R,S)	30,900
Target position(θ)	1.57 rad(+90°)

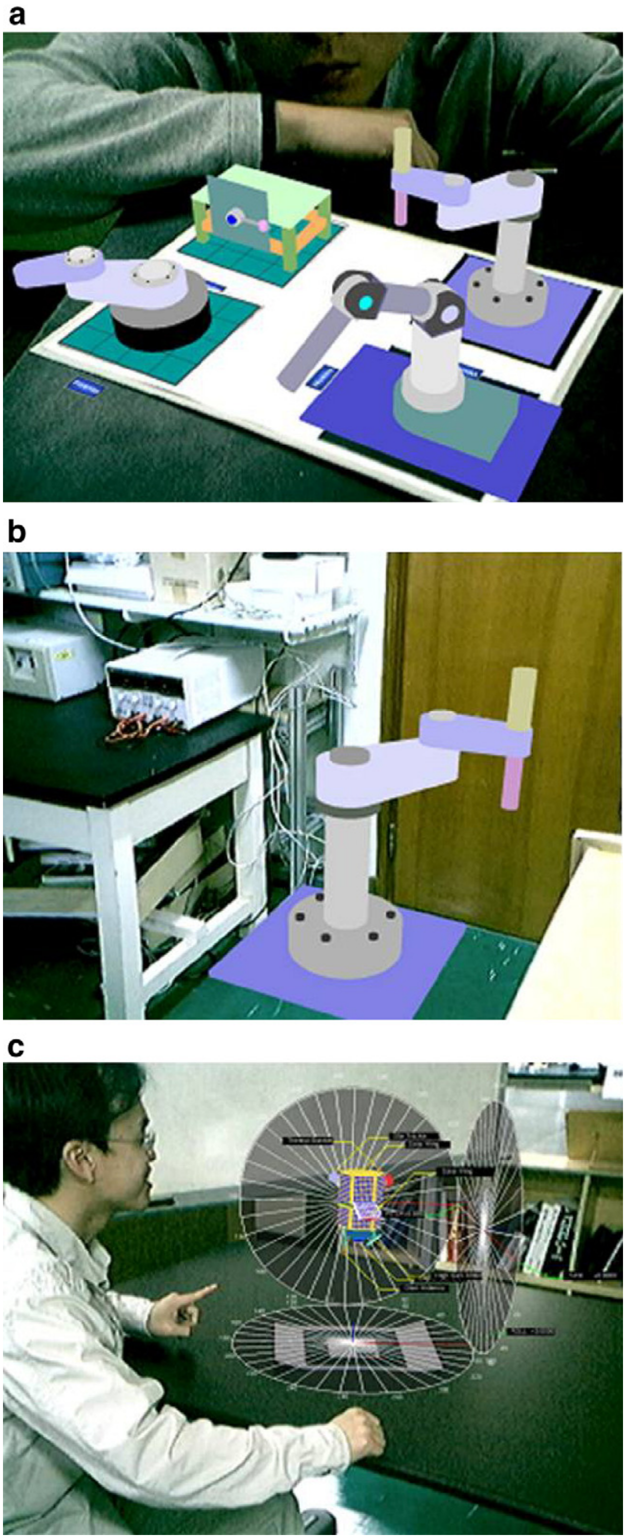


Fig. 9. Various target places: (a) Robots on the desk (b) Robots on the laboratory floor (c) Spacecraft on the center of floor.

Several interaction modules for the flexible viewpoint navigation have been implemented in the formal spacecraft simulator which has no AR functionalities. The relationship between object orientation and viewpoint follows Eq. (13) where x, y, z denotes the eye position of the observer and x_0, y_0, z_0 denotes the orientation of the coordinate frame for virtual objects. θ and φ represent angular movements of the viewpoint in a spherical coordinate frame.

$$\begin{aligned} &x = r\cos(\theta)\cos(\phi) + x_0; y = r\cos(\theta)\cos(\phi) + y_0;\\ &z = r\sin(\theta) + z_0 \end{aligned} \tag{13}$$

3.5. Noise filtering

We observe simulations mostly in the static view state after dynamically configuring the markers. The trembling of augmented virtual models in the static view state can disturb a user's observation. The cause of the trembling is that the four vertex data detected from the marker vary slightly. They affect the elements of transformation matrices (T_{cm}) from the marker coordinates to the camera coordinates. A noise filter is applied to resolve this problem. We give the elements the average of several previous values if the variation is within a constraint. If the variation crosses this constraint, the motion of the marker is regarded as being in the dynamic state and the filtering stops.

3.6. Analysis of coordinate frames for robot collaboration

The virtual objects are drawn in the marker coordinate frames. A virtual object in a frame is independent. Fig. 7 shows the relationships between the camera coordinates and two marker coordinates.

The base of a robot is fixed on the marker orientation, with its end-effector able to move in a limited fashion within a boundary. For collaboration between two robots, there has to exist a common volume where end-effectors of both Robot 1 and Robot 2 can reach. Even if it is given, however, it is hard to know that both end-effectors have arrived at a particular point in the volume because of the respective representations in their own marker coordinate frames. Therefore, the present positions of the end-effectors have to be analyzed in a common coordinate frame. A simple and reliable procedure for collaboration in AR process can be performed as follows.

- Step 1: Retrieve the present coordinates of end-effectors in the marker coordinate systems: (X_{m1}, Y_{m1}, Z_{m1}) and (X_{m2}, Y_{m2}, Z_{m2})

- Step 2: Retrieve the coordinates of the end-effectors by using T_{cm} in the camera coordinate systems: (X_{C1}, Y_{C1}, Z_{C1}) and (X_{C2}, Y_{C2}, Z_{C2})
- Step 3: Determine whether the difference between the coordinates of the two cameras is within a reliable constraint.
- Step 4: If so, then begin the proposed collaboration, such as load transfer.

Careful extraction of relevant coordinates and matrices in program codes is necessary because many of rotations and translations affect the present positions of end-effectors during animation. The 2D screen coordinates cannot guarantee that the end-effectors are exactly arrived at the same position but the 3D camera coordinates are more reliable.

4. Demonstration

4.1. Pertinent configuration of markers

In marker-based AR experiments, different target models can be effectively visualized with different fiducial marker configurations [8]. In the case of robot systems, a patterned marker

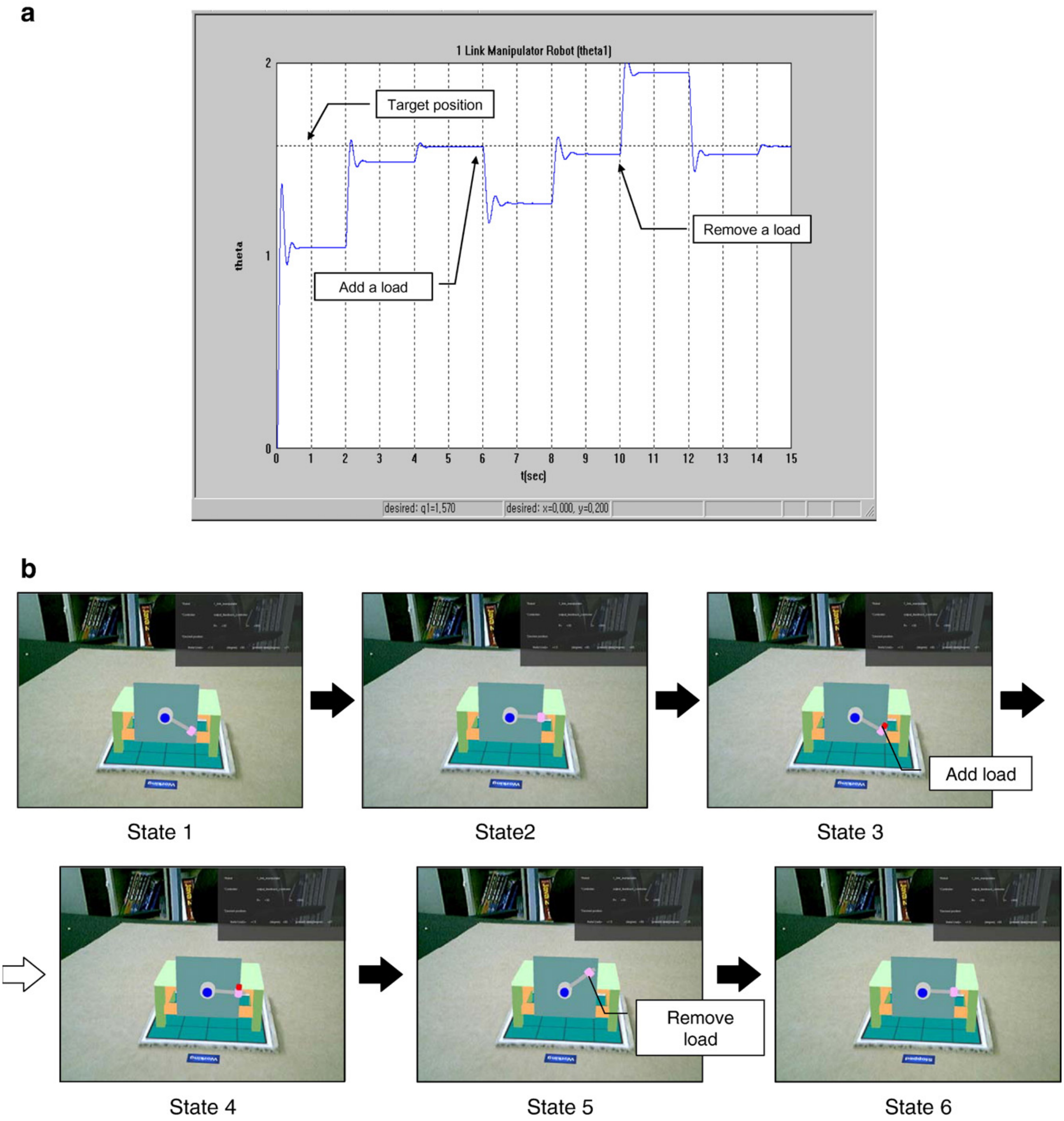
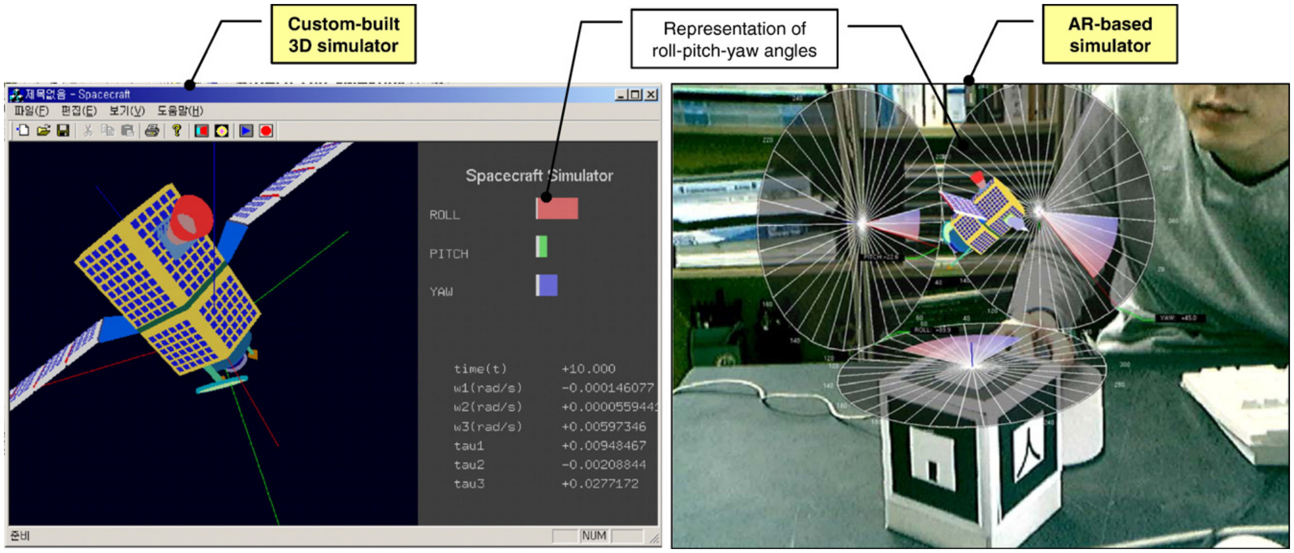


Fig. 10. Controller test for 1 link manipulator: (a) 2D graph (b) 3D AR scenes (State 1–6).


Custom-built 3D simulator
Representation of roll-pitch-yaw angles
AR-based simulator
제목없음 - Spacecraft
파일(F) 편집(E) 보기(V) 도움말(H)
Spacecraft Simulator
ROLL
PITCH
YAW
준비
NUM

```
time(t)       +10.000
w1(rad/s)     -0.000146077
w2(rad/s)     +0.0000559441
w3(rad/s)     +0.00597346
tau1          +0.00948467
tau2          -0.00208844
tau3          +0.0277172
```

Fig. 11. Attitude control simulation for spacecraft.

identifies a virtual robot model to be overlaid on a live capturing image as shown in Fig. 8. For the registration of virtual spacecraft models, box-style configurations are made by using five markers (See Fig. 8). Because all five markers are indicating a single orientation in the marker coordinate system, a virtual object can be faithfully overlaid on the real video scene even if any one marker disappears from the user's view (camera's view). Such a multiple marker configuration ensures reliable augmentation in the situation of graphical occlusion interrupting image analysis which happens mostly when users are manipulating or interacting with the models.

This configuration works effectively for a number of applications. We can conceive of several testable applications by changing the target workplaces using the system features described in Section 3.4. Firstly, the robot simulator can be used for the purposes of educational aids in the classroom or in industry. An instructor can demonstrate the 3D simulation to students or engineer trainees by manipulating a fiducial marker that represents a robot model. If they are wearing video or optical see-through HMDs, the visual information of system states can be intuitively offered in any direction. Also, virtual robots can be positioned visually abreast with real robots to evaluate their working in the manufacturing line of a real factory. It will enhance the user's perception and aid the user's real-world tasks. Our collaboration test could be used to demonstrate this.

Next, the spacecraft simulator can be placed in the showcase stage by employing a box style configuration of fiducial markers. Researchers or exhibition visitors may want to observe the bottom part of the spacecraft from underneath. Although the vision system would have difficulty in detecting the upper marker, the other side markers can ensure the correct registration of the virtual spacecraft models. Most of all, the visual support of 3D spherical coordinates for reporting states will help viewers to observe the motion of floating spacecraft in 3D void space. Fig. 9 shows how virtual models can be merged in the real environment scenes.

4.2. Experimental validation

The features of AR-interfaced 3D simulations are demonstrated by several experiments.

4.2.1. Visually-intuitive simulation

To show how the characteristics of the control algorithm are reflected in the 3D simulation, the performance of the output feedback control algorithm is demonstrated for the 1-link manipulator model. The controller enforces that the robot link will reach its target destination by checking the difference

Table 2
Spacecraft parameters

Gas-jet actuator type spacecraft	
Central inertia matrix(J)	diag[2,1,2]
Controller gains(K,M,N)	diag[10,10,10], diag[2,3,2], diag[3,2,3]
Target positions(φ,θ,ψ)	π/2, π/8, π/4
Reaction wheels actuator type spacecraft	
Inertia matrix without wheels(Λ_s)	diag[2,3,2]
Moment of inertia(A,B)	0.05, 0.1
Inertia matrix of ith wheel(J_i)	diag[B,A,A], diag[A,B,A], diag[A,A,B]
Controller gains(K,M,N)	diag[10,10,10], diag[3,2,3], diag[4,8,4]
Target positions(φ,θ,ψ)	π/2, π/8, π/4

Table 3
2-link and 3-link SCARA type robot parameters

	2-link SCARA type robot		3-link SCARA type robot		
Link order	Link1	Link2	Link1	Link2	Link3
Link length(l)	0.5 m	0.3 m	0.48 m	0.26 m	0.4 m
Link mass(m)	32.7 kg	7.73 kg	32.7 kg	7.73 kg	5.0 kg
PD controller gains(K_p, K_d)	120, 15	56, 5	120, 15	56, 5	4, 6
Target positions(θ)	0.785 rad	1.57 rad	0.785 rad 0.0 rad	0.785 rad −0.785 rad	0.315 rad 0.165 rad

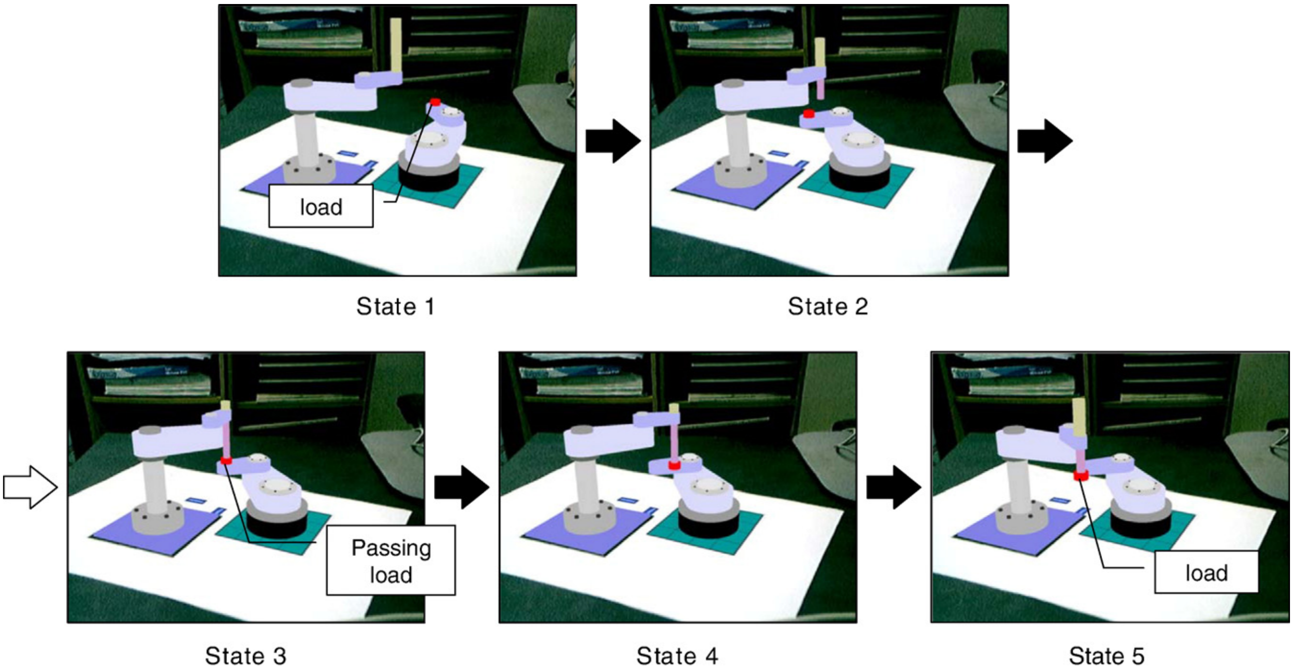


Fig. 12. Collaboration between two virtual robots — 3D AR scenes (State 1–5).

between the present link position and the target position at each timestep. The system parameters in this experiment are given in Table 1.

The AR-interfaced 3D simulation proceeds as shown in Fig. 10(b).

- State 1 ($0 \leq t < 4$): it shows the intermediate state of the controlling link to the +90° position which is our final target position by updating constant vector **v** of the controller. Whenever the position state of link converges to a steady position near the target position, the vector **v** is continuously updated so that the link can finally reach the target position.
- State 2 ($4 \leq t < 6$): The control of the link is complete. The link is positioned at the target position +90°.
- State 3 ($t = 6$): A load is added at the end of link. The link mass is changed from 1.8 kg to 2.8 kg. Fig. 10(b) presents a more intuitive visualization than (a) about this state. The link position deviates from the target position because of the load.
- State 4 ($6 < t < 10$): The link again approaches the target position. The feature of the controller that deals with mass change is shown because it works based only on information about the present position of the link.
- State 5 ($t = 10$): The load is removed. The link position deviates from the target position again in the opposite direction of state 3.
- State 6 ($10 < t \leq 15$): The link reaches the target position again.

The feature of the output feedback control algorithm is also well reflected in AR environment. Compared with a 2D graph-style representation, the 3D graphic simulation enhances visual intuition about robot motion. Further, AR interfacing demonstrates the extensibility of this intuitive simulation into real workspaces, e.g. the case of applying new operation mechanisms to manufacturing lines.

4.2.2. Effectively-observable simulation

In this section, we present an attitude control simulation for spacecraft with pre-experimental data. Fig. 11 shows how the simulation can be visualized in an AR environment compared with a previous 3D simulator [22]. Because spacecraft is floating in 3D void space, allowing the user to flexibly move their viewpoint in AR provides a natural way to observe the spacecraft's motion in space. It is more effective than the usual 3D simulation on a static display for virtual models, in which the viewpoint is manually controlled only by a mouse or keyboard interaction. In the AR setting, present states of roll–pitch–yaw angles are continuously displayed in 3D spherical coordinates, which also helps user's visual perception of the spacecraft motion. The 3D simulation can be enabled in an AR environment using threading, allowing virtual models to be simulated independent of the rendering frequency of the AR process. This experiment demonstrates how effective the AR interface is in helping users observe the simulation. The system parameters are given in Table 2.

Table 4
2-link SCARA type robot parameters

Link order	Link1	Link2
Link length(l)	0.39 m	0.27 m
Link mass(m)	32.7 kg	7.73 kg
PD controller gains(K_p,K_d)	100, 10	100, 10
Target positions(θ)	0.481 rad	0.328 rad

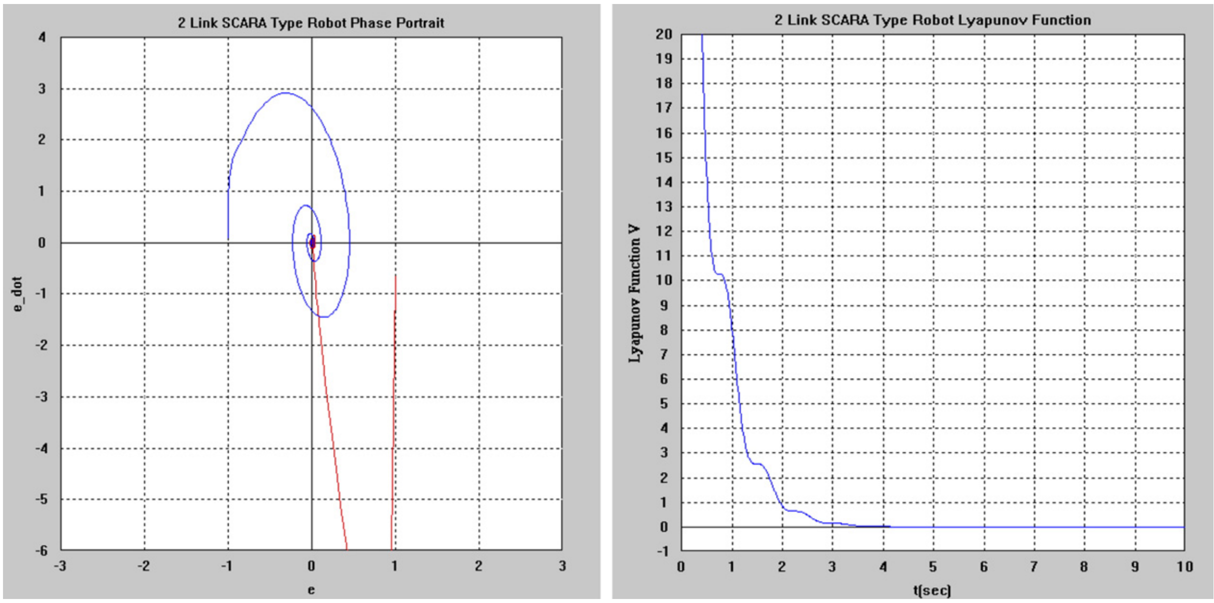


Fig. 13. Control stability test for 2 link SCARA type robot — 2D graph.

4.2.3. Collaboration of virtual robots in AR environment

In practice, there are a number of virtual objects that exist in AR applications, and mutual interaction between them can occur even in simulation. However, this mutual interaction cannot be realized as in the 3D simulation because each virtual object has its own local reference coordinate system within the AR environment. In this experiment, we considered a situation in which two robots were collaborating: 2-link and 3-link SCARA type robot (Robot 1 and Robot 2 respectively). It also validates the procedures in Section 3.6. The gains of each PD controller are given purposely to generate an oscillation for reliable collaboration in uncertain transfer timing of a load. When the transfer of the load is completed from Robot 1 to Robot 2, the target position for Robot 2 will be changed. The system parameters are given in Table 3.

Fig. 12 shows sequential scenes of the collaboration simulation.

- State 1: Robot 1 and Robot 2 start to move arbitrarily in time. A load is mounted on Robot 1.
- State 2: Robot 1 reaches its target position, however Robot 2 is still in motion. The position of the end-effector of Robot 2 is being checked to determine whether it came in contact with the present position of the load.
- State 3: The end-effector of Robot 2 is passing right on the top side of the load.

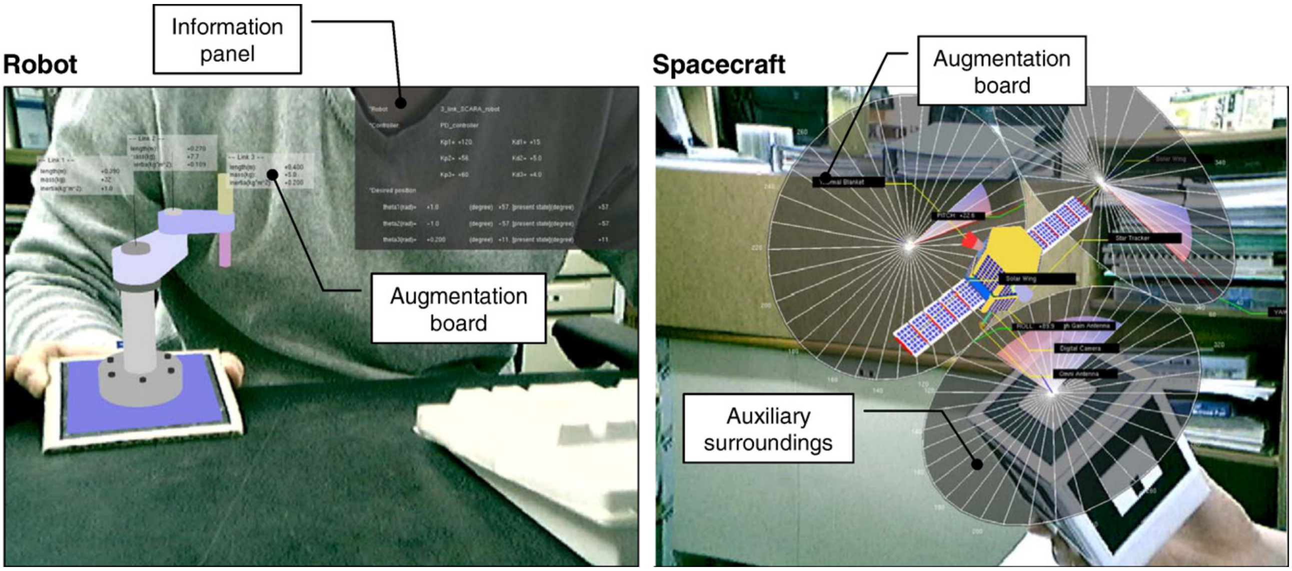


Fig. 14. Fully visualized scene.

- State 4: The load of Robot 1 is transferred to Robot 2. Then, Robot 2 is commanded to the next target position.
- State 5: Robot 2 is in motion to the next target position.

Although each virtual robot is arbitrarily positioned, it is only if their end-effectors are sharing a certain volume that collaboration is possible. This experiment demonstrates mutual interaction between virtual objects in simulation in an AR environment, and it can be extended to the case of interaction with real objects.

4.2.4. Test of control stability in AR-independent environment

As an auxiliary validation, the stability test of a 2-link SCARA type robot control is performed using Lyapunov functions and phase portrait. This stability test is a matter of concern in conventional simulation. This experiment is available regardless of AR service in our embedded simulation modules. The system parameters are given in Table 4, and Fig. 13 shows the graph of them when applying the PD controller, in which the state of the robot is converging to an equilibrium point.

4.3. Informative visuals

In AR-based simulations, three informative items are visually configured to aid users in obtaining the present simulation state: information panel, annotation board and auxiliary surroundings (See Fig. 14).

Firstly, the information panel in the right-top corner reports the system control parameters, real-time simulation states, etc. It has a fixed global coordinate system regardless of the present eye-position. Secondly, the annotation board represents information about a system component. The board, suspended on the top of a guide branch, is positioned according to the present eye-position (=camera aim pose). Although the branch is connected to the component in 3D spatial relation, the visible face of the board is forced to follow the local 2D coordinates system so that it always looks toward the eye. Thus, all the information on the board is faithfully visible all the time to the user unlike in a real environment where mutual occlusion frequently occurs between a physical board and other objects. Lastly, auxiliary surroundings such as 3D circular planes in the spacecraft simulator report the simulation states in a visually-enhanced form close to the target model.

Table 5
Summary of informative visuals

Description		Parameters displayed
Annotation	Robot links information	Length, mass, inertial, controller gains, etc
	Spacecraft components	Digital camera, high gain antennas, star trackers, omni antennas, thermal blanket, etc
Information panel		Robot type, controller type, desired link positions, present position states, etc
Simulation control button		8 operation steps: Demo – Start – Working – Pause – Paused – Continue – Stopped – Initiate sequence
3D circular planes		Roll–pitch–yaw angle representation of spacecraft

In addition, versatile computer graphics techniques such as blending, texturing, local viewport setting, etc. may be applied to enhance visual intuitiveness of a model. A blending effect helps users keep a global awareness of the surroundings in an AR environment. Users can control the simulation in progress by interacting with graphical control buttons (augmented from the typical flatness of the computer graphics workspace). Keyboard and mouse events are monitored to allow information about the system or simulation state to be made available when desired by the user. Tangible interfacing techniques, more commonly being leveraged in AR research, support more natural user-friendly interaction [6–8]. Table 5 summarized parameters displayed in the informative visuals.

5. Conclusion and future work

In this paper, we have presented a feasibility study of using AR-based 3D simulations for two target systems, robots and spacecraft. Compared with ordinary 3D simulations, the AR-based simulations enhance users' ability to interact with the models in the simulation intuitively, since the models are placed in the users' actual workspace. Information about the motion of models due to embedded dynamics and kinematics bear augment these AR simulations. Providing a reliable characterization of a model's motion using an AR methodology provides added benefits beyond what is available in the usual 3D modeling of a model's visual appearance. In our experimental validation, we have demonstrated the how AR-based simulations can be supported using appropriate fiducial marker configurations and supplementary visuals.

We believe AR interfacing technology together with context-aware computing technology will take a significant and initiative role in future ubiquitous computing environment. Both techs have been showing us the most reliable feasibility in seamlessly implanting information (whether it is visible or invisible) in our physical environment; it is because in UbiComp environment the information should be intelligently determined and intuitively interacted with users in order to be offered at a right place in the most appropriate form in a particular time. In this point of view, in this paper, we have showed an infrastructural approaching method for evaluating the feasibility of AR-based simulation by focusing on mechanical systems and physical workspace.

Additionally, we have been concurrently conducting research in other prospective AR application domains, as an extension of our conceptual development schematic: geospace and multimedia. In these domains, the functional modules and end-user interaction must be re-implemented for each application. Indeed, the subsystem modules in an AR test-bed vary considerably according to the type application and its requirements; what may be the most important and complex technical aspect of one application can turn out to be completely trivial in other applications. The use of AR is consequently still nascent despite a wealth of noteworthy applications and its many valuable features. Accordingly, our future work will be devoted to specifying how AR can be explicitly applied to real-world applications, in keeping with our main purpose of conceptual development towards building an enhanced real-world.

Acknowledgments

This work is mainly supported by the Ministry of Information and Communication (MIC) South Korea through the Realistic Broadcasting IT Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST). In addition, some portion of this work was partially supported by the Ministry of Science and Technology (MOST) of South Korea through two projects at GIST (Gwanju Institute of Science and Technology): 'The developments of HMI (Human-Machine Interface) system in virtual environments' and 'Spacecraft attitude and orbit precision control — A new methodology for spacecraft control: Theory and Applications'. The authors gratefully acknowledge the contribution of Prof. A. Ailon in Ben Gurion University of the Negev, Israel, and Dr. Eunseok Choi in SAIT (Samsung Advanced Institute of Technology).

References

[1] B. Michel, S. Peter, T. Jean-Yves, Smart interface-to-device architecture for HMI, Proceedings of Systems, Man, and Cybernetics 1999, IEEE SMC '99 Conference, vol 6, 1999, pp. 1063–1068.

[2] S. Chen, Z. Kazi, M. Beitler, M. Salganicoff, D. Chester, R. Foulds, Gesture-speech based HMI for a rehabilitation robot. Southeastcon'96. Bringing Together Education, Science and Technology, Proceeding s of the IEEE (1996) 29–36.

[3] M. Endo, S. Koide, S. Misono, S. Suzuki, Development of human–machine interface composed of virtual reality and interface agent on process plant operation, Systems, Man, and Cybernetics 1999, IEEE SMC '99 Conference Proceedings, vol 5, 1999, pp. 636–641.

[4] N.P. Mahalik, S.D. Kim, B.H. Ahn, Specification requirement of temporal HILS environment for RCP of distributed mechatronic systems, IEEE Int'l Conference on Mechatronics, ICM/HIMA, Taiwan, 2005.

[5] N.P. Mahalik, J.H. Ryu, B.H. Ahn, Simulation integrated management layer for real-time embedded DCN, Computer Standards & Interfaces 28 (5) (2006) 508–522.

[6] S.J. Kim, J.E. Cha, J.P. Kim, J.H. Ryu, S.E. Eom, N.P. Mahalik, B.H. Ahn, A novel test-bed for immersive and interactive broadcasting production using augmented reality and haptics, IEICE Trans. on Information and Systems: Special Sect. on Artificial Reality and Telexistence E89-D (1) (2006) 106–110.

[7] S.J. Kim, H.K. Kim, S.E. Eom, N.P. Mahalik, B.H. Ahn, A reliable new 2-stage distributed interactive TGS system based on GIS database and augmented reality, IEICE Trans. on Information and Systems: Special Sect. on Artificial Reality and Telexistence E89-D (2006) 98–105.

[8] G.G. Yun, S.J. Kim, S.E. Eom, B.H. Ahn, 3D augmented reality map with fingertip interaction, HCI 2004 Proceedings, pp. 598–604, 2004.

[9] N. Navab, Developing killer apps for industrial augmented reality, Computer Graphics and Applications, IEEE 24 (3) (May/June, 2004) 16–20.

[10] N. Navab, et al., Scene augmentation via the fusion of industrial drawings and uncalibrated images with a view to markerless calibration, Proc. 2nd Int'l Workshop Augmented Reality (IWAR 99), IEEE CS Press, Los Alamitos, Calif., 1999, pp. 125–133.

[11] M. Rosenthal, A. State, J. Lee, G. Hirota, J. Ackerman, K. Keller, E.D. Pisano, M. Jiroutek, K. Muller, H. Fuchs, Augmented reality guidance for needle biopsies: A randomized, controlled trial in phantoms, Proc. Medical Image Computing and Computer-Assisted Intervention (2001) 240–248.

[12] B. Schwald, B. de Laval, An augmented reality system for training and assistance to maintenance in the industrial context. Journal of WSCG, 11 (1), WSCG'2003, Plzen, Czech Republic, 3–7 Feb., 2003.

[13] E. Gelenbe, K. Hussain, V. Kaptan, Enabling simulation with augmented reality, in: M.C. Calzarossa, E. Gelenbe (Eds.), MASCOTS 2003, LNCS 2965, 2004, pp. 290–310.

[14] ARToolKit, http://www.hitl.washington.edu/artoolkit.

[15] A. Ailon, Analysis and synthesis of an output feedback for an uncertain robot model with flexible joints, Proc. IFAC Conference on System Sturcture and Control, Nates France, 1995, pp. 370–375.

[16] P.E. Crouch, Spacecraft attitude control and stabilization: applications of geometric control theory to rigid body models, IEEE Transactions on Automatic Control 29 (4) (1984) 321–331.

[17] A. Ailon, B. Grinberg, R. Segev, M. Gil, A simple output-based stabilizing controller for attitude regulation of a rigid spacecraft with gas-jet actuators, The 2001 American Control Conference, USA, 2001, June.

[18] A. Ailon, R. Segev, B. Grinberg, M. Gil, Velocity-free controllers for spacecraft attitude regulation: some further applications and results, 2001, IEEE Conf. on Control applications, Mexico, September 2001.

[19] Kendall Atkinson, Elementary Numeric Analysis, 314–316.

[20] S.J. Kim, Implementation of a PC-based 3-D robot simulator using OpenGL, Master thesis, Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, 2000.

[21] S.J. Kim, E.S. Choi, B.H. Ahn, The development of 3-D robot simulators using OpenGL, Proceeding of the KIEE Summer Annual Conference 2000, Muju, Korea, July 2000.

[22] S.J. Kim, S.W. Lee, W.S. Jeong, B.H. Ahn, PC-based 3D graphic spacecraft simulator using OpenGL. 2002 International Conference on Control, Automation and Systems, Muju Korea, 2002.

Dr. Seungjun Kim is working as a Post-doc Researcher in Human-Computer Interaction Institute (HCII) at Carnegie Mellon University (CMU). He received his BS (1998) in Electrical & Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST), an MS (2000) and PhD (2006) in Mechatronics from Gwangju Institute of Science and Technology (GIST), South Korea, Postdoctoral research at GIST Technology Initiative (GTI) in the year 2006. His research interest is in the integration of Augmented Reality interfacing method for prototype 3D applications and Augmented Reality in context-aware environment.

Dr. N P Mahalik is a faculty member in the Department of Industrial Technology, California State University-Fersno, USA. He received his bachelor, master, and PhD degrees in year 1989, 1993 and 1998, respectively. To his credit Dr. Mahalik has received the prestigious National Overseas Scholarship (1994) and Brain-Korea-21 fellowship (2001) award from the Indian and Korean government, respectively, for pursuing research especially in the field of system integration during his 16 years of academic career. Dr. Mahalik has guided 5 PhDs and 14 ME/MS Theses. His research area has been on Software for Control, Networked Control, Micromechatronics, Automation, Micro Systems, Fault Detection and Isolation, Augmented Reality. He has written/edited 5 books and more than 80 articles and reports. He is also editor in chief of IJAAC and guest editor of several international journals. He was a visiting professor to MSTU, Russia and GIST, S. Korea. He has been a member/associate of NAIT (USA), ISA (USA), MF (UK), IEE (UK), and ISTE (India).

Professor Anind K. Dey is an Assistant Professor in the Human-Computer Interaction Institute (HCII) at Carnegie Mellon University (CMU). He received his BS (1993) in Computer Engineering from Simon Fraser University, an MS (1995) in Aerospace Engineering from Georgia Tech, and an MS and PhD (2000) in Computer Science from Georgia Tech. He performs research at the intersection of human-computer interaction and ubiquitous computing, and has published over 50 articles in this area.

Prof. Jeha Ryu is a Professor in the Department of Mechatronics, GIST. He received his BS (1982), MS (1984) and PhD (1991) degree from Seoul National University, Korea Advanced Institute of Science & Technology and University of Iowa (USA), respectively, in Mechanical Engineering. He has received the best education award and president award and has been teaching advanced control, dynamics, automation, virtual environment and haptics. His research interests include parallel manipulator kinematics/dynamics/control/optimal design, haptic device for VR interface, computer haptics/VR dynamic simulation, vehicle dynamics/control, hardware-in-the-loop simulation. Prof. Ryu is a member of ASME, KSME, KSAE and IEEE. More than 120 of his research articles and reports have been published.

Prof. Byungha Ahn is a Senior Professor in the Department of Mechatronics, GIST. He was Director of the System Integration Laboratory with additional administrative position as Vice-President at the institutional level. After serving in the Air Force Tactical Flight Wing as Pilot and subsequently as Chief during 1966–1981, Prof. Ahn preferred to join the academic field since then. Prof. Ahn received his PhD in Industrial Engineering in the year 1980 and has achieved many distinctions, awards, publications and patents to his credit. He is the auditor of the Society of Korea Industrial Engineering; member, Society of Korea Human Factors Engineering; member, Board of Directors, Society of Korea CALS/EC. His research interests include data acquisition/fusion for intelligent transport systems; industrial application of information technology; microtelemanipulation; air traffic control; systems management and reliability analysis. More than 120 of his research articles and reports have been published.