МОДЕЛИРАНЕ И УПРАВЛЕНИЕ НА РОБОТ ВЪВ ВИРТУАЛНА ЛАБОРАТОРИЯ ЧРЕЗ ИНТЕРНЕТ

Симеон Цветанов, Тодор Стоилов

Резюме: *Докладът представя изграждането на експериментална лаборатория за следене движението на робот. Лабораторията е достъпна чрез глобалната мрежа Интернет. Така се реализира виртуален достъп на отдалечени потребители до ресурсите за управление на робота. Роботът е управляван за придвижване на видео камера и наблюдение в технологична среда. Представено е системното програмно осигуряване реализиращо функциите на виртуалната лаборатория, показани са резултати от работата и функционирането на робота.*

MODELLING AND CONTROL OF ROBOT IN VIRTUAL LABORATORY OVER THE INTERNET

Simeon Tsvetanov, Todor Stoilov

Abstract: *The paper presents the design and exploitation issue of experimental laboratory for monitoring and control of robot model. The laboratory is available over the Internet. This approach allows remote users to obtain access to the distributed software resources for control. The robot model is controlled to move video camera for exploration the work scene. The software suite is developed to support functionality of a virtual laboratory interfacing the distant user control influences to Matlab base software performing robot simulations. The virtual laboratory is applied for learning and education*

Съвременна тенденция в развитие на информационните услуги в Интернет е отдалечения достъп до изчислителни и технологични ресурси [6 7]. Този достъп позволява сложни технически устройства да се ползват от голям кръг потребители. Направлението на информационните услуги разработвани в Интернет за дистанционно ползване на технически средства е наречено "Виртуална лаборатория" [8]. Методите на виртуалната лаборатория широко се прилагат в електронното и дистанционното обучение [9].

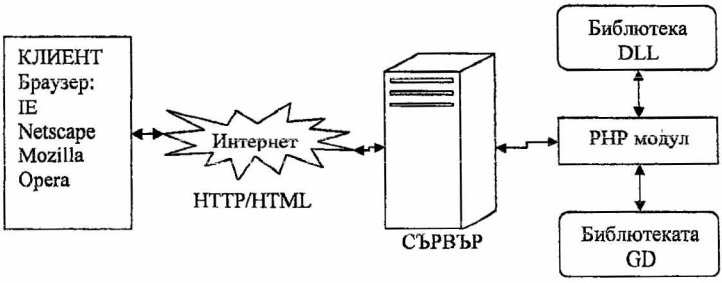
В настоящия доклад е представена разработка, която реализира дистанционно управление на робот. При управлението роботът придвижва видео камера и съгласно получаваните команди чрез глобалната мрежа изследва работни обекти като променя ъгъла на наблюдение, мащаба на изображението, оценява координати на местоположение и др.

Годишник на ТУ – София, т. 54, кн. 2 / 2004 г.
Proceedings of the TU of Sofia, vol. 54, № 2 / 2004

Архитектура на виртуалната лаборатория

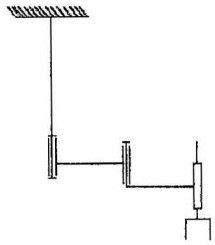
Аналитичното движение на робота се дефинира чрез програмен модел в средата на *Matlab*. Реализира се интерактивно изпълнение на модела с различни входни данни, предавани от външни файлове. Тези файлове се поддържат от сървърна система, която поддържа отдалечен интерфейс с потребителя чрез интернет. Наборът от сървърските програми представя резултатите от моделирането и управлението на робота в графичен и интерактивен вид. Сървърските програми се реализират на технологията PHP. Интерпретаторът PHP използва графичната библиотека GD за реализиране на графична функционалност при поддържане на интефейса за управление на робота, както и за представяна на неговото състояние. Отдалеченият потребител на виртуалната лаборатория използва обикновен браузър: *Internet Explorer*, *Mozilla*, *Netscape* и др.

Системната архитектура на виртуалната лаборатория е представена на фиг.1.



Фиг. 1. Системна архитектура на виртуалната лаборатория

Описание на робота



Използван е робот с три степени на свобода. В конкретния случай роботът е тип "SCARA", фиг.2.

Фиг. 2. Робот с три степени на свобода

Този прототип може да бъде усложняван. За практически приложения това може да е робот-ръка на космическа совалка. Местенето на робота, придвижването от него на обекти в пространството и извършване на работни операции се извършва от изпълнително звено-хващач. В настоящата разработка в хващача е поставена видео камера. Чрез нея се наблюдава работната сцена от гледна точка на хващача, за да се осъществи обратна връзка. Посетителят на сайта има възможност да управлява робот чрез визуална обратна връзка.

Всяка точка в пространството има три степени на свобода. Те се отчитат чрез координатите *x, y, z*. При задаване на положението на хващача, управлението на робота трябва да изчисли необходимото взаимоположение между отделните стави (ставните координати), при което хващачът ще бъде на желаното място. Това изчисление е изпълнявано чрез решаване на обратната задача на кинематиката.

Потребителят на виртуалната лаборатория се ориентира за местоположението на робота по динамично генерираното изображение от сървъра. Това изображение представлява снимка на работната сцена на робота. Потребителят променя състоянието на робота чрез задаване на три променливи, съответсващи на желаното положение на камерата/хващача в пространството. Тези данни се обработват от сървъра и той изпраща ново изображение за състоянието на робота, като се изчисляват текущите ставни координати и се решава обратната кинематична задача.

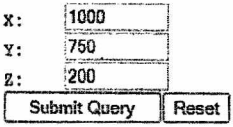
Данните от потребителя се събират чрез HTML форма. Потребителят въвежда информация в полетата на формата и ги изпраща на сървъра чрез натискане на бутон *submit*. Браузерът форматира данните и изпраща заявка към web-сървъра. За да може web-сървъра да започне работа и да изпълни програма, web-браузерът трябва да пакетира данните, въведени от потребителя и да изпрати HTTP заявка към web-сървъра. HTTP заявката се състои от: URL за страницата или скрипта, до които потребителят иска да получи достъп; въведените във формата данни и всякаква допълнителна информация.

Последната стъпка при изпълнение на приложението е генериране на отговор за успешно извършена операция, който се връща към браузера.

Програмиране от страната на клиента

Въвеждането на желаните координати *x, y, z* , се осъществява от три полета, които са реализирани с три INPUT елемента, с NAME атрибути d_x, d_y, и d_z. Интерфейсът е реализиран с web страница, имаща стандартна HTML/HEAD/BODY структура и полета, съдържащи се в елемента FORM.

Прието е, че първоначално роботът се намира в "Home position". За това при зареждане на страницата елементите имат зададени стойностти. Максимално опростен интерфейс за въвеждане на данните е показан на фиг. 3.



Фиг. 3. Опростен интерфейс за въвеждане на данни

При натискане на *Submit* данните се пращат, сървърът генерира изображение, но изображението замества формата на екрана и за да върне данните от формата, потребителят трябва да използва бутона BACK. В нито един момент от времето потребителят не може да види едновременно изображението и данните.

Подобряване функционалността на интерфейса с виртуалната лаборатория

Web страницата дава възможност на потребителите да въвеждат координатите и височината на хващача на робота. В разработката е реализиран подобрен интерфейс чрез модифициране на формата за въвеждане на данни.

Модифицирането на форма е процес, чрез който данните, въведени от потребителя, се превеждат в друг формат преди да бъдат предоставени на сървъра. Когато данните от форма се предоставят на техния сървър, те трябва да бъдат форматирани по начин, по който сървърът може да ги разбере. Този формат обикновено има следния вид:

<url>?<field1=value>?<field2=value>...

За формата от фиг. 3 форматът е следния.

```
http://127.0.0.1/project/ro/ro_action.php?d_x=1000&
```

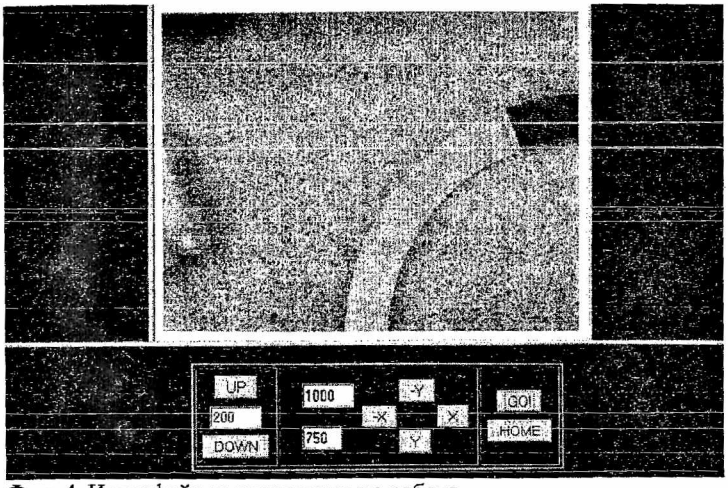
d_y=750&d_z=200

където <field1>, <field2>, и т.н. са NAME атрибутите на елементите на формата. Съответните стойностите са VALUE атрибутите на тези елементи с малки промени, направени от браузъра.

Направените промени позволяват:

- потребителят да модифицира размера на изображението чрез промяна на мащаба (zooming). Намаляването на мащаба се реализира като пропорционално намаляване на размера, което позиционира част от областта (повече или по-малко) в същото по размер изображение. Увеличаването на мащаба, се дефинира като увеличаване на размера като така поставя по-голяма област в същото по размер изображение.
- добавяне на бутони, за да може потребителят да настрои прецизно централните координати. Това позволява потребителят лесно да прави бързи промени.

Изображенията, получени от работата на робота, са представени на фиг.4.

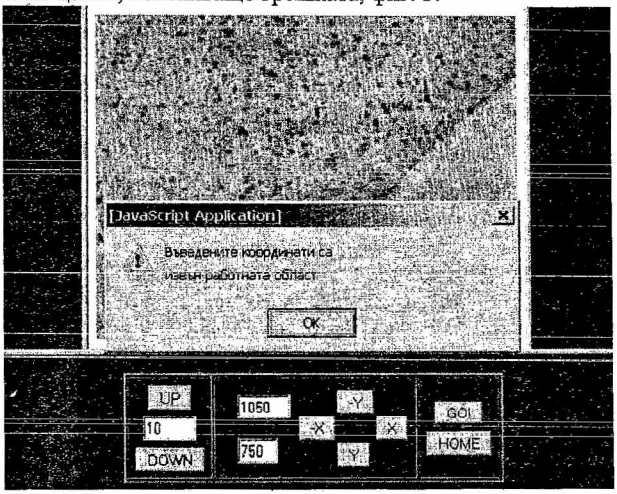


Фиг. 4. Интерфейс за управление на робота.

При клиентската програма е използвана JavaScript програма, която позволява да се следи движението на робота и да се възпроизвеждат негови предишни състояния.

Програмиране от страната на сървъра

След като данните са изпратени на сървъра, променливите са достъпни за PHP скрипта на сървърската програма. В случая това са $d_x, $d_y, $d_z, за *x*, *y* и *z* съответно. За тези данни трябва да се реши обратната задача на кинематиката и да се прати изображение, имитиращо снимка. Ако въведените координати са невалидни, се праща изображение, съответсващо на текущото положение на робота и съобщение, обясняващо грешката, фиг. 5.



Фиг. 5. Проверка на достоверността на входните данни

Решаването на обратната задача на кинематиката се изпълнява в среда *Matlab*. Взаимодействието между сървърските PHP програми и средата *Matlab* се реализира чрез междинните файлове *in.txt*, и *error.txt*.

Частта от сървърската PHP програма, взаимодействаща с файла in.txt е даден на фиг.6

```
$body = quotemeta($d_x_r) .
            "\t" . quotemeta($d_y_r) .
            "\t" . quotemeta($d_z); //конкатенация
$in = fopen( "in.txt", "w");//отваря in.txt за запис
fwrite($in, $body);//запис в файла
```

фиг. 6. Взаимодействиео между сървърските PHP програми и средата *Matlab*.

Табулацията се поставя защото файла ще се ползва от *Matlab*.

След като данните бъдат записани в файла *in.txt*, се зарежда външен модул посредством функцията dl(). Функцията приема като параметър името на файл, зависещо също и от платформата (за Windows - това е dll). В случая като външен модул се зарежда файла *nevolq.dll*.

Част от създадения с текстовия редактор на *Matlab* файл *nevolq.m*, който след това е компилиран до *nevolq.dll* , е показана на фиг.6 (За да се компилира m-файл, се прави запис "*mex-setup*" в командния прозорец на *Matlab* , след което се следват инструкциите).

```
Cor=load('in.txt');% зарежда от файла in.txt
X = Cor(1);
Y = Cor(2);
Z = Cor(3);
.....
Error = [ER1 ER2 ER3];
save error.txt Error -ASCII;% Запис в файла error.txt
```

фиг. 7. Част от файла *nevolq.m*, изясняваща взаимодействието между сървърските PHP програми и средата *Matlab*.

Обработка и изпращане на изображението

Всяко изображение се адресира чрез целочислен идентификатор, подобен на указателите при файловете и базите данни. Този идентификатор се връща при отваряне на изображението и се предава на всички следващи извиквания на функции, които ще работят с изображението. След това трябва да се изведе изображението. Има два начина да се направи това: да се прати директно към браузера или да се запише като *.png файл (съществуват и много други формати). И в двата случая се използва фукцията Imagepng(): ако се зададе име на файл, изображението ще се запише в този файл; в противен случай, то ще бъде изпратено директно към браузера.

Изображението, което се визуализира в браузера, е с размер 400x300 пиксела и е част от изображението, записано във файла scene.png, представляващ цялата работна сцена. GD библиотеката предоставя разнообразни възможности за обработка на изображения, като например една от функциите, които е използвана ImageCopyResized(), копираща правоъгълна част от едно изображение в друго.

Изводи

Реализирана е системната и програмна стуктура на виртуална лаборатория. Осъществено е управление на робот чрез предаване на данни в глобална мрежа. Виртуалната лаборатория позволява да се управлява положението на видео камера, като динамично се задават желани посоки на наблюдение, мащабиране на изображенията, управление движението на робота. Виртуалната лаборатория е използвана за дистанционно обучение.

Литература:

1. Castagentto, J., Rawar, H., Shumann, S., Scollo, C., Veliath, D. **Professional PHP Programming.** Wrox Press, 2000.
2. http://www.php.net
3. Уилям, П., Шърмън, Е. ***Dynamic HTML in Action***, Microsoft Press, 1998.
4. **Matlab Reference Guide.** The Math Works, Inc., August 2002
5. Круглински, Д., Шепърд, Д., Уингол, С. **Програмиране с Microsoft Visual C++6.0**, СофтПрес – ООД, 1999.
6. www.virlab.com
7. www.chem.ox.ac.uk/vrchemistry/
8. www.jhu.edu/~virtlab/virtlab.html
9. www.dlcoursefinder.com

За авторите

проф. дтн инж. Тодор Стоилов
инж. Симеон Цветанов,
Институт по Компютърни и Комуникационни Системи
1113 София, Ул Акад. Г. Бончев бл. 2
todor@hsi.iccs.bas.bg, simeon@hsi.iccs.bas.bg