

TRANSPORT SYSTEMS

UDC 007.52 (045)

DOI: 10.18372/1990-5548.56.12939

¹Yu. M. Kemenyash,²M. O. Omelchenko³V. P. Hotsyanivskyy

MULTIFUNCTIONAL ROBOTIC COMPLEX BASED ON A SINGLE-BOARD COMPUTER

^{1,2,3}Aviation Computer-Integrated Complexes Department, Educational & Scientific Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine

E-mails: ¹lindysik999@gmail.com, ²max.omel4enko@yandex.ua, ³sttt912@yahoo.com

Abstract—The purpose of the work is to create a modular multifunctional robotic complex based on the single-board computer Raspberry Pi and Arduino microcontroller. In order to solve the problems of fast transfer and processing of data, a client/server software code has been developed that minimizes potential losses. The server is universal and can modify its code for a specific task by spending a minimum amount of time. An image processing algorithm is created that reduces the amount and time for correctness checking that affects the speed of cascade construction. The fastest and optimal algorithm for cascade construction is developed, which allows to reduce the time of repair by 3 times. Optimized software communication between a single-board computer and a microprocessor for faster processing of information.

Index Terms—Damping; ribs; oscillation of liquid; tank.

I. INTRODUCTION

According to the National Association of Market Participants, the world's 10,000 workers, in 2015, had an average of 69 industrial robots. Robotics is an applied science that deals with the design development construction operation and use of robots, as well computer systems for their control, touch (based on the output signals of sensors) feedback and information processing of automated technical systems.

It is aimed at the creation of robotic systems designed to automate complex technological processes and operations, including those performed under non-deterministic conditions, to replace a person during heavy, tedious and dangerous work. Robotics is actively distributed in the workplace, where work replaces people to perform routine work. For example, Bosch has created a robotic assistant APAS (Fig. 1) that helps a person during repetitive actions.



Fig. 1. Assistant APAS

Symbotic manufactures self-propelled robots that move around the warehouse and carry boxes from the general container to sorting tapes for further delivery [1].

Caterpillar plans to develop remotely controlled machines and fully autonomous heavy robots by 2021. IBM's Texas Holders Keyboard Factory, which is 100 percent robotic.

Works such as HOSPI (Fig. 2) are used as couriers in hospitals.



Fig. 2. Robot HOSPI

Automation and robotics will affect not only the business and professions of people, but also the world economy and politics. It's important to understand that the changes associated with these approaches will not happen all over the world. The most active distribution will be automation only in the developed countries – the USA, Canada, the EU countries – and some developing countries – for example, China and India.

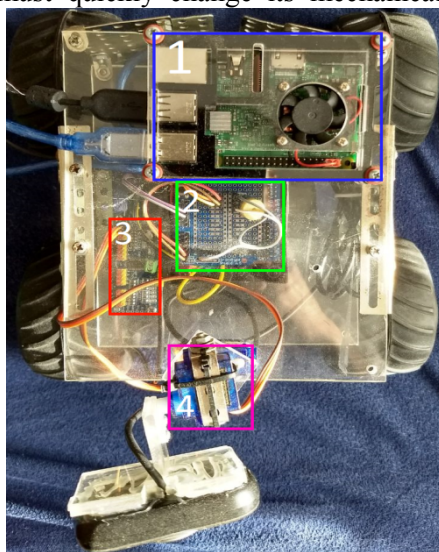
A good illustration can be the GlobalLogic project in the field of warehouse logistics, where the load of goods on pallets carry out work. To teach them to correctly place various sizes, weights, strengths and contents of boxes (otherwise the goods may suffer or fall during transportation), GlobalLogic engineers have developed a complex mathematical model and algorithms for loading goods on pallet. In general, automation is a broad area and it covers a lot of technologies. It uses machine learning, object recognition and emotions, neural networks and so on.

It raises the question of creating robots that are much functional and materially available. It is also expedient to use and select parts and software development.

II. PROBLEM STATEMENT

The basis of the problem is to create a client-server project that will be able to process information in real time and in large numbers. It will also have a large bandwidth for sharing with the server, but will be simple and multitasking.

In the work of the compressor can include various manipulators of the sensors external and internal modules, depending on the work performed. The complex must quickly change its mechanical



part due to breakage or alteration of the landscape and the environment. If necessary, the software part can be completely replaced, optimized or supplemented with features that are not added to the main software without spending much time compiling, rebooting and setting the parameters [2].

The reaction of the system under a certain condition must be rapid. When processing images should use the optimal algorithm which will be sure to determine the specified object or objects without downloading the system. When using an external periphery, the maximum processing speed of information should be.

That is, the system should be universal, multifunctional and modular.

Multifunctional robotic complex on the basis of a single-board computer has the form (Fig. 3).

The given complex consists of:

- single-board computer;
- microcontroller, driver for engines and expansion cards;
- servo driver type PCA9685;
- MG996 actuators;
- system of optical observation;
- mechanical part.

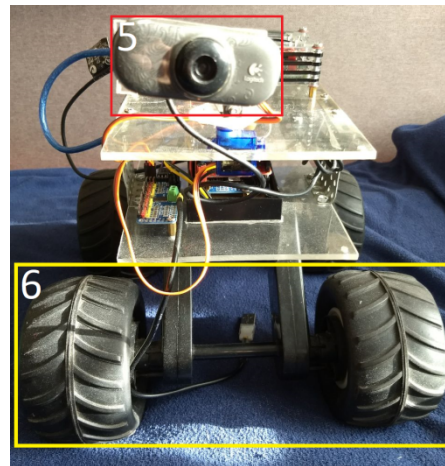


Fig. 3. Robotic complex

III. PROBLEM SOLUTION

The basis of the work of the complex will be a one-payment computer. These devices are capable of turning into control units with virtually any technique [3].

Supported interfaces UART, SPI, I2S, I2C, GPIO. The board features full-sized USB 2.0 ports, an HDMI interface for outputting images and a Micro-USB connector. Operating temperature range from minus 25 to plus 70 degrees Celsius. These

computers are small in size (up to 10x12 cm), have a low cost and support many operating systems that are created on the Linux kernel, so they can be rebuilt for better performance.

There are a large number of single-board computers:

- ROCK64 4K USB 3.0;
- Tinker Board S;
- ODROID-XU4 Octa-Core;
- Banana Pro;

- LeMaker HiKey 960;
- Asus Tinker Board;
- LattePanda;
- Raspberry Pi 3 Model B +;
- QOTOM OEM.

We use Raspberry Pi 3 as it is the best option for the development of a robotic system.

The following packages are required from the software in addition to the operating system:

- Python 2.7 / 3.6;
- tkinter is a graphical library that allows you to create programs with a window interface. This library is an interface to the popular programming languages and tools to create graphical applications tcl / tk;
- Python lib glob, os, urllib, serial, numpy;
- network socket – the name of the software interface for the exchange of data between the processes of the interconnected network;
- OpenCV – library functions and algorithms of computer vision, image processing and numerical algorithms for general purpose open source. The library provides tools for processing and analysis of image content, including recognition of objects in photos (faces and figures of people, text), tracking the movement of objects, image conversion, the use of machine learning and identify common elements in various images.

To use any manipulator, driver, sensor, or sensor, a real time-related system is required. A single-board computer does not have such a possibility therefore it is necessary to use a microcontroller.

Microcontroller or microcomputer – made in the form of microcircuit specialized microprocessor system including a microprocessor, memory units for storing program code and data, I / O ports and blocks with special functions (counters, comparators, ADCs, etc.): NODEMCU; PARTICLE PHOTON; ESP8266; TEENSY; BEAGLEBONE; MSP430; STM32; Arduino.

Arduino has the most optimal configuration and meets the requirements of the specification.

The versatility of the system depends on its simplicity and functionality, therefore, the client/server programs using the TCP/IP protocol are additionally used. Then creating a certain local area network using 802.11AC, you can update the software remotely. Using an external WIFI module with a 5G frequency gives 802.11AC speeds equal to 866 mb/s per channel, as with a 2.4G 802.11N speed, the speed will be 150 mb/s the need to change the detection of certain objects programmatically changes the file with cascades.

This makes it possible to create cascades on the main server and simply transfer them to the nabolic

system. Images are needed in large numbers and not in the RGB model.

For universal processing, you can use the PIL library:

```
import urllib.request
import os
import numpy
from PIL import Image
def download_and_resave_images():
    neg_images_link = 'http://somelink'
    neg_image_urls =
urllib.request.urlopen(neg_images_link).read().decode()
    pic_num = 1
    for i in neg_image_urls.split("\n"):
        try:
            urllib.request.urlretrieve(i, "neg/"+str(pic_num+a)+".jpg")
            imP = Image.open("neg/"+str(pic_num+a)+".jpg")
            if not imP.mode == 'RGB':
                imP = imP.convert('RGB')
            imP.save("neg/"+str(pic_num+a)+".jpg")
            pic_num += 1
        download_and_resave_images()
```

import urllib gives you the ability to upload images at such a very high speed. The OpenCV library does not always correctly handle uploaded images as they sometimes have a non-standard model. The image load is checked if it differs from the standard model then the image must be replaced by the standard imP = imP.convert ('RGB').

This will reduce the percentage of errors from 15–20% to 0%, which contributes to the stability of the processing system and the creation of a cascade.

After that, all images change their model with RGB to GRAYCOLOR to facilitate processing (Figs 5 and 6).

```
import cv2
pic_num=1
while True:
    try:
        img =
cv2.imread("neg/"+str(pic_num)+'.jpg',cv2.IMREAD_GRAYSCALE)
        resized_img= cv2.resize(img,(200,200))
        cv2.imwrite("neg/"+str(pic_num)+'.jpg',resized_img)
        pic_num+=1
```



Fig. 5. Non-resize

All images should be of the same size using the function `cv2.resize()`, the resulting image is created to create a cascade.



Fig. 6. Resize

To manage, you need to create a server that will be the basis of a single-board computer:

```
import glob
def scan():
    return
glob.glob('/dev/ttyS*')+glob.glob('/dev/ttyUSB*')+glob.glob('/dev/tty
ACM*')
if __name__ == '__main__':
    for name in scan():
        ard_ser=serial.Serial(name,9600)
server_UP =
socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server_UP.bind(('ip',port))
server_UP.listen(1)
conn,adr=server_UP.accept()
while (1):
    Data=conn.recv(128)
    if len(Data)>0:
        ard_ser.write(Data)
```

When connecting a microcontroller via UART-USB it is not necessary to know its addresses. Since the operating system can change the address after restarting or shutting down, it should be specified every time. At the same time, this must be done instantly. Def `scan()` function: defines the name of the port where the microcontroller is located. Using the serial library, `serial.serial(name, bod)` creates a connection to the bod speed, which will depend on the task and the amount of information.

The `socket.socket()` function creates a socket for exchanging information through the created IP, and the port must be manually set or generated for increased security.

All data is transmitted in “String” and serial-port arduino accepts char so the *Parsing* feature created is optimized and processing this information as quickly as possible:

```
for(int k=stp;k<len;k++)
{ S_web_lr=S_web_lr+inc_Mass[k];}
```

```
Web_lr=S_web_lr.toInt();
Web_lr = map(Web_lr, -100, 100, lrmIn, lrmAx);
Web_pwm.setPWM(0, 0, Web_lr);}
```

The created client transmits the direction and speed that is produced by pulse-width modulation and the position of the servo drives on which the image-receiving device or controller is mounted.

To visualize and create a client, you need:

```
import socket
from tkinter import *
```

There are necessary libraries to implement feedback from the server.

```
def but_fow_clicked():
    PWM_r=PWM_R.get()
    client_sock.send(Direc+str(PWM_r)+'_'+str(PWM_l))
```

The function `but_fow_clicked()` is an example of implementing an optimal code for manual control of the complex.

```
def but_connect_clicked():
client_sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
client_sock.connect(('IP',PORT))
client_sock.send('pass')
```

When calling `but_connect_clicked()`, a client-server pair is created to manually manage the whole complex with a correctly specified port password and IP.

The processing of frothing in this robotic system is based on the OpenCV software.

Cascades are used to find the object, which can be created and trained separately from the work system.

```
sock_cascade = cv2.CascadeClassifier('cascade.xml')
cap= cv2.VideoCapture(0)
While True:
    ret, img =cap.read()
    gray =cv2.cvtColor(img,0)
    sock= sock_cascade.detectMultiScale(gray)
    for (x,y,w,h) in sock:
        Func_call()
```

When searching for multiple objects, the function `cv2.CascadeClassifier('cascade_name.xml')` is used.

Image processing is in real time so the function of choosing `cv2.VideoCapture(0)` is used.

This will delay the output of the image on the client, but this will not affect the processing speed. When the object is found, `Func_call()` causes other subfunctions that control the behavior of the system.

This algorithm allows you to identify a few objects simultaneously, which makes the complex a lot of functional.

Creating a cascade depends on the size and number of images. With 5000 images, 200x200 px. and the twist and width of the 20 px step. it takes about 14 hours. The learned cascade can be transferred to the above-said server and replaced by an existing one. But when using the keys, `-precalcValBufSize <size>` and `-precalcIdxBufSize <size>` time is shortened to 5 hours, that is almost 3 times.

IV. CONCLUSION

In this article a modular robotic complex was developed based on the single-board computer Raspberry Pi and microcontroller Arduino and the problems of its creation were solved. The client/server type software code is developed for fast transmission and processing of data without any losses. The server is versatile because it can modify

its program code for a specific task by spending a minimum amount of time.

The created image processing algorithm reduces the amount and time for the correctness check that affects the speed of cascade construction.

A fast and optimal algorithm for cascade construction is created which allows to reduce the time of repair by 3 times. This allows you to accurately identify an object or objects with large input data. Optimized software communication between a single-board computer and a microprocessor for faster processing of information.

REFERENCES

- [1] Owen Bishop, "Robot Builder's Cookbook," MIT Press, 2000, pp. 84–90.
- [2] Doug Williams, *PDA Robotics: Using Your Personal Digital Assistant to Control Your Robot*, NT Press, 2006, Proc., 284 p.
- [3] Bruno Siciliano, Oussama Khatib, "Springer Handbook of Robotics," *Proc.*, 2008, pp. 160–168.

Received February 05, 2018

Kemenyash Yuriy. Assistant Professor.

Aviation Computer-Integrated Complexes Department, Educational & Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine

Education: National Aviation University, Kyiv, Ukraine, (1996).

Research area: Identification of Complex Systems.

Publications: more than 30 papers.

E-mail: lindysik999@gmail.com

Omelchenko Maxym. Student.

Aviation Computer-Integrated Complexes Department, Educational & Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine.

Research area: Automation, robotic complex.

Publications: 3.

E-mail: max.omel4enko@yandex.ua

Hotsyanivskyy Vladimir. Student.

Aviation Computer-Integrated Complexes Department, Educational & Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine.

Research area: Automation, robotic complex.

Publications: 3.

E-mail: sttt912@yahoo.com

Ю. М. Кеменяш, М. О. Омельченко, В. П. Хоцянівський. Багатофункціональний робототехнічний комплекс на базі одноплатного комп'ютера

Розглянуто створення модульного багатофункціонального роботизованого комплексу на базі одноплатного комп'ютера Raspberry Pi та мікроконтролера Arduino. Для вирішення задач швидкої передачі та обробки даних розроблено програмний код клієнт-серверного типу який мінімізує можливі втрати. Сервер є повністю універсальним і може змінювати свій програмний код під конкретну задачу витрачаючи на це мінімальний час. Створено алгоритм обробки зображень який зменшує об'єм та час на перевірку на коректність що впливає на швидкість побудови каскадів. Розроблено максимально швидкий та оптимальний алгоритм побудови каскадів що дає змогу зменшити час опрацювання у три рази. Оптимізовано програмний зв'язок між одноплатним комп'ютером та мікропроцесором для більш швидкої обробки інформації.

Ключові слова: роботизований комплекс; одноплатний комп'ютер; сервер.

Кеменяш Юрій Михайлович. Старший викладач.

Кафедра комп'ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційно-діагностичних систем, Національний авіаційний університет, Київ, Україна.

Освіта: Національний авіаційний університет, Київ, Україна, (1996).

Напрямок наукової діяльності: ідентифікація складних систем.

Кількість публікацій: більше 30 наукових робіт.

E-mail: lindysik999@gmail.com

Омельченко Максим Олегович. Студент.

Кафедра комп'ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційно-діагностичних систем, Національний авіаційний університет, Київ, Україна.

Напрямок наукової діяльності: автоматизація, робототехнічний комплекс.

Кількість публікацій: 3.

E-mail: max.omel4enko@yandex.ua

Хоцянівський Володимир Петрович. Студент.

Кафедра комп'ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційно-діагностичних систем, Національний авіаційний університет, Київ, Україна.

Напрямок наукової діяльності: автоматизація, робототехнічний комплекс.

Кількість публікацій: 3.

E-mail: sttt912@yahoo.com

Ю. М. Кемениш, М. О. Омельченко, В. П. Хоцяновский. Многофункциональный робототехнический комплекс на базе одноплатного компьютера

Рассмотрено создание модульного многофункционального роботизированного комплекса на базе одноплатного компьютера Raspberry Pi и микроконтроллера Arduino. Для решения задач быстрой передачи и обработки данных разработан программный код клиент-серверного типа, который минимизирует возможные потери. Сервер является полностью универсальным и может менять свой программный код под конкретную задачу, тратя на это минимальное время. Создан алгоритм обработки изображений, который уменьшает объем и время на проверку на корректность, которая влияет на скорость построения каскадов. Разработан максимально быстрый и оптимальный алгоритм построения каскадов, что позволяет уменьшить время обработки в 3 раза. Оптимизирована программная связь между одноплатным компьютером и процессором для более быстрой обработки информации.

Ключевые слова: роботизированный комплекс; одноплатный компьютер; сервер.

Кемениш Юрий Михайлович. Старший преподаватель.

Кафедра авиационных компьютерно-интегрированных комплексов, Учебно-научный институт информационно-диагностических систем, Национальный авиационный университет, Киев, Украина.

Образование: Киевский авиационный университет, Киев, Украина, (1996).

Направление научной деятельности: идентификация сложных систем.

Количество публикаций: более 30 научных работ.

E-mail: lindysik999@gmail.com

Омельченко Максим Олегович. Студент.

Кафедра авиационных компьютерно-интегрированных комплексов, Учебно-научный институт информационно-диагностических систем, Национальный авиационный университет, Киев, Украина.

Образование: Национальный авиационный университет, Киев, Украина.

Направление научной деятельности: искусственный интеллект.

Количество публикаций: 2.

E-mail: max.omel4enko@yandex.ua

Хоцяновский Владимир Петрович. Студент.

Кафедра авиационных компьютерно-интегрированных комплексов, Учебно-научный институт информационно-диагностических систем, Национальный авиационный университет, Киев, Украина.

Образование: Национальный авиационный университет, Киев, Украина.

Направление научной деятельности: искусственный интеллект.

Количество публикаций: 2.

E-mail: sttt912@yahoo.com