Proceedings of the 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP),

Development of System Supervision and Control Software for a Micromanipulation System

Ahmet Teoman Naskali
Emrah Deniz Kunt
Asıf Šabanoviç
Sabancı University
Orta Mahalle, Universite Caddesi No: 27
Tuzla 34956, Istanbul Turkey
Email: {teoman, edkunt, Asif}@sabanciuniv.edu

Gönen Eren
Galatasary University
Çıragan Cad. No:36
Ortaköy 34357, Istanbul
Telephone:(212) 227 44 80 Fax: (212) 259 20 85
Email: geren@gsu.edu.tr

***Abstract*—This paper presents the realization of a modular software architecture that is capable of handling the complex supervision structure of a multi degree of freedom open architecture and reconfigurable micro assembly workstation. This software architecture initially developed for a micro assembly workstation is later structured to form a framework and design guidelines for precise motion control and system supervision tasks explained subsequently through an application on a micro assembly workstation. The software is separated by design into two different layers, one for real-time and the other for non-real-time. These two layers are composed of functional modules that form the building blocks for the precise motion control and the system supervision of complex mechatronics systems.**

I. Introduction

Microsystems have started to gain particular importance in recent decades. From micro-electro-mechanical systems (MEMS) in mobile phones to endoscopic pills with biosensors, this technology has already become a very important part of our lives. The need for assembly of micro components in a precise manner brings out the necessity for flexible, modular and accurate mechanisms and systems that can finely pick, orientate, move and place different types of objects. The mechanical construction of these systems is a major problem but also the software control architectures and user interfaces are of critical importance for their performance. To solve some of these problems, flexible micro robot based micromanipulation systems (MMS) where automatically controlled micro robots perform the micro assembly are proposed in [1], [2]. In this approach, several computers running in parallel to control the functions of the system and the robots are used. These computers communicate with microcontrollers to perform the lower level motion control. A robot for the assembly of 3D Micro Electro Mechanical Systems (MEMS) structures is presented in [3]. Sensing devices monitor loads in a limited degrees of freedom (DOFs) and hardwares with 6 DOFs are developed by [4], [5]. A 6 DOFs robot is enhanced by [6] where the system is controlled by a program written in C++ running on a Windows based platform. This system has a motion resolution of 0.2 microns and a repeatability of +/- 2 microns, but the system does not have real-time operation. A JAVA based approach is used in [7] where microsystems development with independent Java Virtual Machine based modules and a centralized controller is studied to create a mini factory. The factory has modules communicating over a common bus and the dedicated modules can be connected to special places on a grid platform. The materials in the system are ferried around with mobile courier platforms. The system achieves 20-micron accuracy that can be upgraded to 200 nm using integrated magnetic sensors. In [8] a micro assembly module structure is developed where every module is run by a dedicated PC104 platform running Linux. Inter module communications are achieved using an RS485 bus, each module is also able to communicate with a master controller over IEEE 802.11. The software structures and architecture are generally not explained in detail in the literature. This paper covers the programming techniques and algorithms used to create a software framework for precise motion control applications over the implementation of a micro assembly workstation.

II. Design Overview

The development of the workstation includes the design of a manipulation system consisting of motion stages providing necessary travel range and precision for the realization of assembly tasks. The motion stages consist of 2 manipulators with 3 degrees of freedom (DOF) (X, Y, Z) and a sampling stage with 3 DOF's (X, Y and Rotation). The manipulator holders have been designed with adjustable angle of approach, providing 2 non-actuated DOF's. For tasks that require submicron operation, the manipulator stage can be enhanced as a fine-coarse positioning stage with the integration of piezo actuators to provide nanometer accuracies. This system is described in [9].The vision system consists of a microscope that determines the position and orientation of micro components to be assembled, with a focus and zooming system equipped with Fire-wire cameras for coarse and fine image capture and illumination systems. The electronics for the system are further described in[10]. The overall control and supervision structure therefore has the task of controlling motion stages in real-time and synchronizing their movement, adjusting the microscope and capturing the images from the cameras and presenting them to the user. It also has the task of reading the commands from the user in forms of mouse clicks on the screen, or joystick movements or scripts written and then process these commands and execute the desired motions or automated tasks. This structure is implemented as a robust real-time control system in the form of an industrial PC and a graphical user interface that permits the control of all the stages in the form of a PC. The system is able to perform robust motion control of its manipulators with sub micron accuracy that translates to maximum one encoder pulse control of the stages. The MMI computer performs data presentation, image capture, image processing whereas the RT computer operates to carry out trajectory calculations and motion control.

III. Software Overview

The aim of the software for the SUMAW is to enable the users to perform precise micro assembly tasks intuitively and also perform automated tasks using the system. The system tasks can be separated in two parts according to their timing constraints. The interactions of the system were examined and classified according to their real-time requirements. Then the devices with dedicated controllers were examined if they required real-time communications or they were able to function as real-time systems that required non real-time references.

The real-time systems was programmed with 3 threads, one for receiving information one for sending information and one for the real-time operations. These threads need fast communication with each other which was solved using shared memory. These threads can cause load on the system and interfere with the operations of the system. To assure that the real-time task is not interrupted, one core of the intel core 2 duo processor was dedicated to real-time operations leaving the other core for communications and the operating system.The hardware platform was migrated several times during development and different IO cards were used, to facilitate this process and to harmonize the functions that access different IO cards. Wrapper functions were developed making the system portable and separating the software framework from the hardware. It was noted that multiple DOF's and different functions cause the code to become incomprehensible rather rapidly. To resolve this problem, structures for each DOF were developed to store the motion stage constants and variables, providing a semi object oriented approach. A library for control was developed to house the functions enhancing code reuse. These two programming techniques enabled the rapid development of the project and enabled the generation of understandable high level code. Positioning of the stages required very precise control, for this sliding mode control algorithm was used. To enable the usage of all aspects of the system the MMI has been developed. To provide intuitive operation point and click and joystick based manipulation was coded. In order to make the system more flexible and to be able to perform automation, scripting abilities have been developed. The system was calibrated and means of measuring the positions in the vertical axis was devised using the focus ability of the microscope. Tedious positioning tasks requiring manual labor were automated using image processing.

An ascii based communication protocol that is easy to debug was developed and the two computers were linked together with Ethernet to provide a fast yet flexible link that is platform independent.

A. RT Controller

The purpose of the real-time system is to control the 10 degrees of freedom existing on the SUMAW and control the manipulators, individually as well as grouped motion movement. The real-time system is also capable of expanding its abilities and degrees of freedom with the structured manner that the motion control code was developed depending on the availability of IO cards. The real-time computer for motion control is an industrial PC with running Slackware Linux and a real-time extension [11]. RTAI extension provides real-time abilities to the operating system by taking over interrupts and making RT threads non preemptible.

1) Threads: There are 3 main threads in the RT system: one for motion control, one for sending messages to the MMI and one for receiving messages from the MMI. These threads communicate over shared memory. This shared memory is also an effective means of transferring information from the non real-time threads to the real-time threads and vice versa. The shared memory is mapped as a C struct where every element is an array with size in accordance with the number of DOF's of the system which is design time configurable. The communication threads reside on the same processor as the OS and listen and send data over TCP/IP sockets.

All the motion control code is executed on a single dedicated processor which runs all the real-time deterministic tasks. The algorithms for the tasks are depicted in Fig.1. The 2nd processor is used by the OS and the communication threads. The real-time thread has highest priority in the system, and it is not affected by the hardware or software interrupts, assuring real-time performance.

2) Control Library: A library to house control functions has been generated and also used in various other projects. The primary functions of the library are: $PIDControl$, $PIVelocityControl$, $DisturbanceObserver$, $SlidingModeControl$, $SlidingModeVelocityControl$, where Disturbance Observer is a function used to calculate the disturbance to the system so that compensation can be applied. The Sliding Mode Control is formulated as below:

$$u_k = u_{k-1} + (GBT_s)^{-1}((DT_s + 1)\sigma_k - \sigma_{k-1}) \quad (1)$$

where u_k is discrete control input, $G = \{\lambda \ 1\}$ with λ being a positive constant, B is the input matrix, T_s is sampling time, D is a positive constant and σ_k is the sliding mode manifold. The control structure (1) is suitable for implementation since it requires measurement of the sliding mode function and the value of the control applied in the preceding step. Thus (1) is used as control structure as discrete sliding mode for each DOF. The derivation of the control law in equation (1) is given in [9].

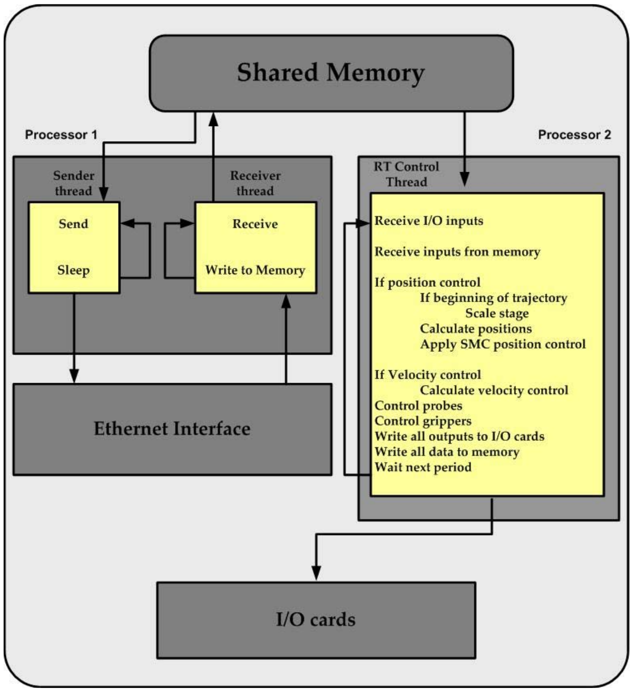

Figure 1. Thread Structure.

3) Trajectory generation: Trajectory generation for the stages is necessary for the manipulators to follow linear trajectories. To achieve a jerk free and smooth trajectory a trigonometric jerk model [12] was applied. A three part S-curve was formed consisting of the acceleration, constant velocity and deceleration phases, where the linear motion stages acceleration limits determines the sinus characteristics during the acceleration and deceleration phases and the velocity limit determined the speed at the constant velocity phase.

B. MMI

The man machine interface (MMI) of the system is the graphical user interface (GUI), which is in the form of a windows program that is used with a mouse, a keyboard and a joystick Fig.3. The man machine interface is written in C#.

1) Modes of Operation: The MMI has 2 modes of position control Fig.2. By entering the positions, by mouse clicks or by joystick movements. The system is also capable of performing some positioning tasks semi-automatically. To avoid stiction an algorithm with repetitive small pushes has been developed Fig 4. A fully automated pushing technique has also been implemented using an atomic force microscope probe. In this approach, determination of the positions of the particles are determined by image processing and the manipulation is realized using the probe. Details of this technique can be found in [13].

2) Scripting: Some of the C# functions used in the system are also available for scripting. The scripting language has the same syntax as C# and it has the ability to use standard Microsoft .NET framework as well as the functions developed

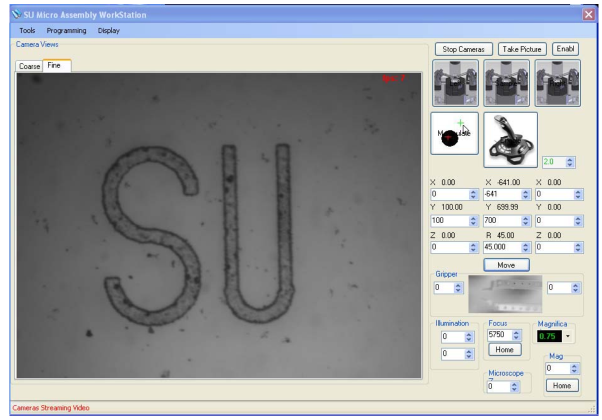

Figure 2. The MMI Screen.

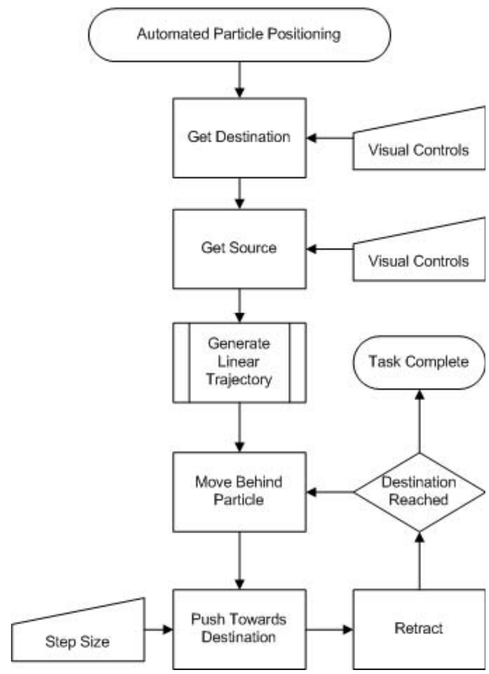

Figure 4. Semi-Automated Particle Pushing Algorithm.

for the workstation enabling writing scripts of different complexity levels by different levels of users. The code is compiled and finally it is executed to run the script during runtime.

3) Vision based Depth Estimation: Focusing on objects to be able to view them can be a tedious task in microsystems where depth information is not known apriori. The focus function of the microscope involves the movement of the focus lens in the vertical axis. With the change of focus, the microscope parameters stay the same, only the zone which is viewed moves in the vertical axis. Therefore, if it is know that a given object is in focus, then the focus parameter (position of focusing motor) can be used to determine its position in the Z axis. The algorithm for auto depth extraction from focus is:

- Move microscope focus to home position
- Move motor with micron intervals downwards, and record

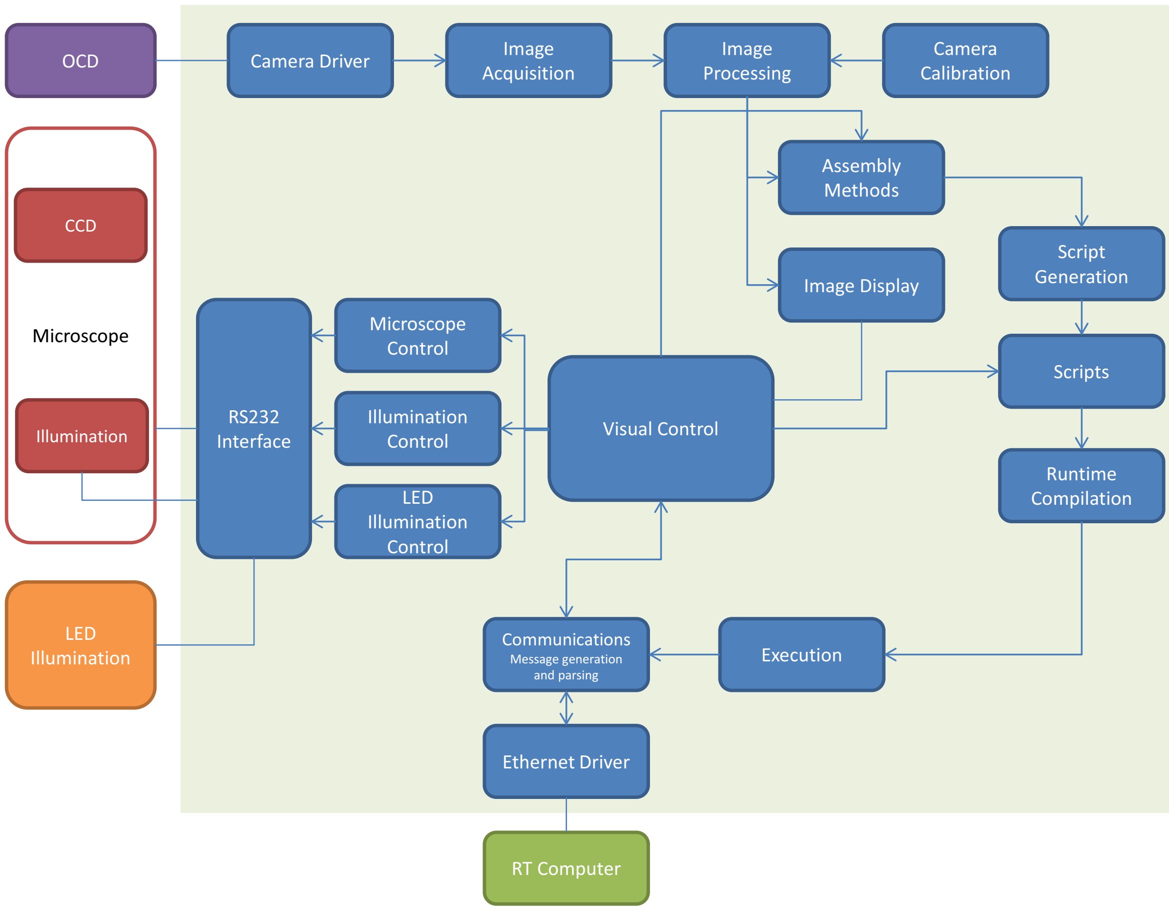

Figure 3. MMI Structure.

sharpness of entire image

- Peaks in sharpness are assumed to be the Z positions of the objects and planes.

The plane positions are recorded for further access. This algorithm operating on the entire image does not yield accurate results for probes since the probe is tilted for manipulation purposes and defining the plane of the probe is not possible. To overcome this a version of the algorithm that works on locating the tip of the probe selected by the user was developed.

IV. Experiments

The motion control system was configured to run with at a control frequency of 10kHz. It was observed that when all 10 degrees of freedom are following trajectories grouped in 3's, the control frequency maintained its real-time characteristics. Position control for the motion stages was achieved to the resolution of one encoder pulse (7 nm) and piezo stages were operated at a precision of 1 nanometer. Manual manipulation tasks, semi-automated manipulation and automated manipulation tasks were performed using the workstation. The system was configured with tungsten probes for automatically pushing polymethylmetacrilate (PMMA) particles using the mouse interface. The system can be equipped with a micro gripper to perform pick an place tasks using the joystick. Then the particles were positioned in a pattern according to predefined assembly procedure. The operators reported that the the MMI is user friendly and intuitive to use.

The workstation was equipped with a force sensing probe and a micro gripper. The core real-time algorithms and components of the system were utilized and a task specific image processing and vision MMI was developed to extract and estimate the membrane properties of zebrafish embryos [14].

V. Conclusion

In this paper a novel micro assembly workstation is presented. The workstation uses standard hardware (Intel x64 based CPU and motherboard) and runs on a Linux operating system. A dedicated processor approach is used to obtain real-time performance whereas most of the previously developed systems do not support real time or depend on specific hardware. The software structure created for the workstation is designed to be modular and expandable. Wrapper interfaces are used to read and write data from/to IO cards. This enhances the portability of the system. Communication between the real-time components and the MMI is established over TCP/IP. Additionally multiple graphical user interfaces or real-time systems can be used over several computers which communicate between each other enabling expandability of the processing power. Image processing functions were implemented to detect objects and to perform tasks on the system. The workstation includes basic image processing and automated assembly functions. Experimental results show that the workstation is capable of performing precise motion control. Furthermore the system can be adopted to perform a variety of different tasks including micro parts placement operations and biological manipulations.

ACKNOWLEDGEMENTS

This project has been supported by the T.R. Prime Ministry SPO (State Planning Organization) grant (2003k120530). The firs and second authors would also like to thank for the support provided by Yousef Jameel Scholarship. The fourth author would like to thank Galatasaray University Research Project no:11.401.003 for supporting him.

REFERENCES

[1] H. Woern, J. Seyfried, S. Fahlbusch, S. Buerkle, and F. Schmoeckel, "Flexible microrobots for micro assembly tasks," *Micromechatronics and Human Science*, pp. 135–143, 2000.

[2] S. Fatikow and U. Rembold, "An automated microrobot-based desktop station for micro assembly and handling of micro-objects," *Emerging Technologies and Factory Automation*, vol. 2, pp. 586–592, Nov 1996.

[3] N. Dechev, W. L. Cleghorn, and J. K. Mills, "Micro assembly of 3-d mems structures utilizing a mems microgripper with a robotic manipulator," *International Conference on Robotics and Automation*, 2003.

[4] D. Dao, T. Toriyama, J. Wells, and S. Sugiyama, "Silicon piezoresistive six-degree of freedom force- moment micro sensor," *Sensors and Materials*, vol. 15, pp. 113–135, 2003.

[5] P. Estevez, J. Bank, M. Porta, P. S. J. Wei, M. Tichem, and U. Staufer, "6 dof force and torque sensor for micro-manipulation applications," *Sensors and Actuators: A Physical*, vol. in press, 2012.

[6] N. Dechev, L. Ren, W. Liu, W. L. Cleghorn, and J. K. Mills, "Development of a 6 degree of freedom robotic micromanipulator for use in 3d mems micro assembly," *International Conference on Robotics and Automation*, pp. 281– 288, 2004.

[7] T. Gaugel, M. Bengel, and D. Malthan, "Building a mini-assembly system from a technology construction kit," *Assembly Automation*, vol. 24, no. 1, pp. 42–48, February 2004.

[8] R. H. Heikkila, I. T. Karjalainen, J. J. Uusitalo, A. S. Vuola, and R. O. Tuokko, "Possibilities of a microfactory in the assembly of small parts and products - first results of the m4-project," *International Symposium on Assembly And Manufacturing*, pp. 166–171, 2007.

[9] E. Kunt, A. Naskali, I. Khalil, and A. Sabanovic, "Design and development of workstation for microparts manipulation and assembly," *Turkish Journal of Electrical Engineering and Computer Science*, vol. 19, no. 5, 2011.

[10] E. D. Kunt, K. Cakir, and A. Sabanovic, "A workstation for micro assembly," *Mediterranean Conference on Control and Automation*, July 2007.

[11] L. Dozio and P. Mantegazza, "Linux real time application interface (rtai) in low cost high performance motion control," *Motion Control*, 2007.

[12] K. D. Nguyen and I. C. T. Ng, "On algorithms for planning s-curve motion profiles," *International Journal of Advanced Robot Systems*, vol. 5, no. 1, pp. 99–106, 2008.

[13] H. Bilen and M. Unel, "Micromanipulation using a micro assembly workstation with vision and force sensing," *International Conference on Intelligent Computing*, 2008.

[14] H. Bilen, M. A. Hocaoglu, E. A. Baran, M. Unel, and D. Gozuacik, "Novel parameter estimation schemes in microsystems," *International Conference on Robotics and Automation*, pp. 2394–2399, 2009.