SOFT FOOD ROBOTIC PICK AND PLACE OPERATION WITH EMBEDDED CONTROL STRUCTURE

Shiuh-Jer Huang[1], Wei-Han Chang[1] and Janq-Yann Lin[2]
[1]*Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan*
[2]*Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan*
E-mail: sjhuang@mail.ntust.edu.tw; m9903417@mail.ntust.edu.tw; jy_lin@itri.org.tw

ICETI 2012-B1006_SCI
No. 13-CSME-25, E.I.C. Accession 3483

ABSTRACT
Robotic pick-and-place operation is planned for handling hard objects with on-off control gripper. It does not have force monitoring capability for safe grasping soft objects. Current force/torque sensor is too expensive and difficult to implement. Here, a low cost embedded control structure is designed with distributed FPGA robotic position control and gripper Arduino force control kernels. A model-free intelligent fuzzy sliding mode control strategy is employed to design the position controller of each robotic joint and gripper force controller. Experimental results show that the position and force tracking control errors of this robotic system are less than 1 mm and 0.1 N, respectively for pick-and-place different soft foods.

Keywords: gripper force control; embedded system; fuzzy sliding mode control.

OPÉRATION PAR ROBOT MANIPULATEUR PRENEUR-PLACEUR AVEC STRUCTURE DE COMMANDE INTÉGRÉE POUR ALIMENTS MOUS

RÉSUMÉ
L'opération par robot manipulateur preneur-placeur est prévue pour la manipulation d'objets durs avec commande de saisie en marche-arrêt. Il n'a pas la capacité de contrôler la force pour une saisie sans risques d'objets mous. Les senseurs actuels de force/torsion sont trop coûteux et difficiles à implantés. Nous avons conçu à faible coût une structure de commande intégrée avec circuit logique programmable de contrôle de position distribué (FPGA) et noyaux de contrôle de force Arduino. Un modèle libre de stratégie de commande intelligente par mode glissant flou est employé pour concevoir le contrôleur de position pour chaque joint robotique et contrôleur de force de saisie. Les résultats expérimentaux montrent que les erreurs de suivi de position et de force de ce système robotique sont moins que 1 mm et 0.1 N respectivement pour le chargement et le positionnement d'une variété d'aliments mous.

Mots-clés : contrôle de force de saisie ; système intégré ; commande par mode glissant flou.

NOMENCLATURE

${}^{i}A_j$	Denavit–Hartenberg transformation matrix
a, w	fuzzy membership function parameters
a_i, d_i	robotic link parameters
C^j	fuzzy consequent parameter
e_1, e_2	states control error
F	force function of voltage output of FSR sensor (N)
gs, gu	FSMC sliding variable and control voltage mapping parameters
s	sliding variable
u	motor control input
V	FSR sensor output voltage
x, y, z	end-effector Cartesian space position components
Greek symbols	
θ_i	robotic joint angle (rad)
μ	fuzzy membership function value
λ	states control error weighting factor

1. INTRODUCTION

The end-effector grasping/contact force monitoring function is an important feature for establishing robotic compliance to execute further intelligent grasping, assembly and human-interaction applications. However, multi-degree force control of achieving desired robotic compliance is known to be a complicated control problem. Most of these force control approaches adjusted the robot end-effector position in response to the measured contact force for obtaining target impedance. This is the well-known impedance force control concept [1]. This approach is different from the hybrid force control technique, which controls the position and the force separately in their controllable directions [2]. For the position based impedance force control [3], the impedance function is explicitly implemented outside the position control loop with accurate model. The adaptive techniques had been proposed to estimate environment stiffness or adjust controller gains for compensating unknown environment stiffness based on force tracking errors information [4–6]. Both hybrid motion control and impedance control have complicated control structures and strategies. They also need an expensive multi-degree torque/force sensor. It hinders the practical application.

Actually, many robotic operations with force monitoring requirement do not need a too complicated hybrid motion control structure. Many applications only need to monitor the grasping or contact force at specified positions instead of fully position/force hybrid motion control. Hence robotic end-effector motion control and gripper force control can be designed based on its own control kernel individually and operated in sequence with a switching signal. However, a special robot gripper should be designed with embedded control structure for contact force control. Here, the grasping force error and robot position error were monitored directly by using an individual fuzzy sliding mode controller (FSMC), respectively. The 5 DOF Mitsubishi robot is manipulated to a specified position by FPGA based control system first. Then the FSMC force controller is activated by a signal from FPGA to monitor the gripper grasping force based on Arduino embedded control system.

2. SYSTEM STRUCTURE AND GRIPPER DESIGN

The retrofitted FPGA robotic control structure has an embedded Atera Stratix system-on-a-programmable-chip (SOPC). User can define the micro-processor specification under the graphic interface, integrate the digital logic circuits coded with Verilog, or VHDL hardware languages. The main functions of FPGA hardware circuits are motor optical encoder decoding, limit switch detecting, pulse width modulation (PWM)

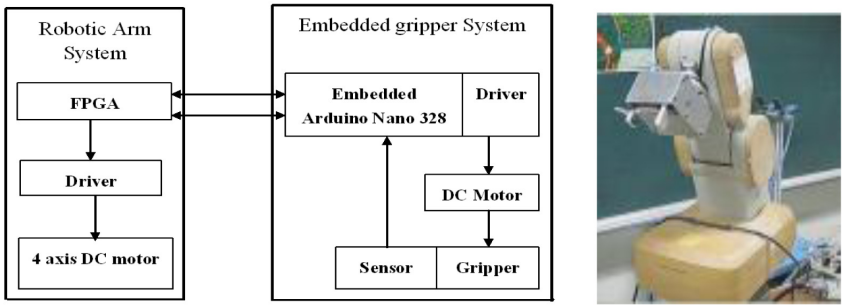


Fig. 1. The embedded robotic and gripper control system structure.

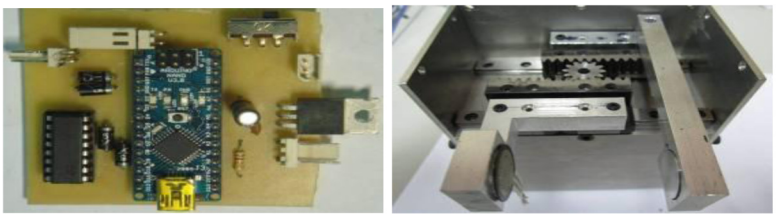


Fig. 2. Embedded control gripper mechanism and servo control circuit.

generating. The functions of the Nios II micro- processor software programs are the communication with Arduino by using a switching signal, robotic inverse kinematics calculation, robotic motion trajectory planning, and robotic motion control schemes.

The embedded gripper system includes two gripping jaw pieces, force sensitive resistive (FSR) sensor, one DC servo motor driving mechanism with Arduino control kernel for grasping force monitoring. The distributed control structure can monitor the end-effector position and grasping force individually in sequence. It has the advantages of using low cost gripper module and sensor, and easy implementation. The overall system structure is shown in Fig. 1.

Gripper is the object grasping device installed in robotic end-effector. Its motion DOF and complexness depend on the working functions specification. Multi-finger robot hand [7, 8] was designed to grasp or operate objects with complicate shape for simulating human hand operation. However, it needs dexterous mechanism design and complicate control and sensor structure. It cannot be implemented on simple embedded control system. Although, two fingers or parallel jaw gripper has function limitation, it is easy to design and implement for most of the industrial pick and place automation operations. Here, an embedded control two parallel jaws gripper is designed for frangible fruits or soft object pick-and-place application. Arduino Nano 328 is chosen as the embedded control kernel for driving the DC servo motor and monitoring the contact force of FSR force sensor installed in gripper jaw. The servo control circuit and gripper mechanism are shown in Fig. 2.

There have pressure resistance, inductance, capacitor, electro-optic and piezoelectric types sensors for using in robotic haptic sensing [9]. Caldwell et al. [10] employed multi FSR sensors to develop the tactile perception for sensing object hardness and shape. Here, FSR is chosen as the force sensor for measureing gripper grasping force during soft object operation. Since, the FSR voltage output is not linear with respect to applied force, an off-line calibration should be done to find the mapping function between measured output voltage and contact force. After 30 samples experiments with standard weight, a transfer function had been found based on Matlab curve fitting tool box:

$$F = 3.99V^7 - 51.62V^6 + 261.83V^5 - 654.42V^4 + 835.6V^3 - 503.6V^2 + 138.3V + 2.13. \quad (1)$$

3. ROBOT INVERSE KINEMATICS

In order to execute the manipulator pick-and-place positioning control in workspace, the inverse kinematics should be investigated. Generally, the end-effector working position or motion path in Cartesian space are converted into control variables in joint space coordinates for control purpose by using the inverse kinematics and Denavit–Hartenberg transformation matrix. Since most of the assembly or pick-and-place operations are planned on a horizontal plane of the working space, the end-effector orientation can be specified as orthogonal and point down to the X–Y horizontal plane. Then the Denavit–Hartenberg transformation matrix of the end-effector with respect to the reference inertia coordinate is

$$ {}^{\text{ref}}T_{\text{tool}} = {}^{0}A_1 \cdot {}^{1}A_2 \cdot {}^{2}A_3 \cdot {}^{3}A_4 \cdot {}^{4}A_5 = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & -1 & 0 & y \\ 0 & 0 & -1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2} $$

Based on the Mitsubishi Movemaster RV-M2 robot link parameters and forward kinematics calculation, the Denavit–Hartenberg transformation matrix can be derived and described by using the robotic D–H parameters a_i and θ_i. The joint angle θ_i corresponding to each specific Cartesian position can be solved by comparing the D–H matrix components and some trigonometric functions operations based on following steps:

$$ \begin{aligned} &\text{Step 1}: && \theta_1 = \theta_5 = A\tan 2(y,x) \\ &\text{Step 2}: && b = \pm\sqrt{(x^2+y^2)} \\ &\text{Step 3}: && \theta_3 = \cos^{-1}\left(\frac{b^2+(d_1-d_5-z)^2-a_1^2-a_1(b-a_1)-a_2^2-a_3^2}{2a_2a_3}\right) \\ &\text{Step 4}: && \theta_2 = A\tan 2\frac{(a_2+a_3C_3)(d_1-d_5-z)-(a_3S_3)\cdot b}{(a_2+a_3C_3)\cdot b+a_3S_3\cdot(d_1-d_5-z)} \\ &\text{Step 5}: && \theta_4 = -\theta_2-\theta_3. \end{aligned} \tag{3} $$

This approach can reduce the trigonometric functions calculation from 17 times to 7 comparing with that of traditional inverse kinematics. The computer time on the Nios II SOPC can be reduced from 4.5 to 2.5 ms for increasing the system closed loop frequency.

4. FUZZY SLIDING MODE CONTROL

Since, multi degree of freedom robotic control system has nonlinear and complicated dynamics behavior, it is difficult to establish an appropriate dynamic model for the model based controller design, especially for the onboard microprocessor. Here the sliding mode concept [11] is combined with fuzzy control strategy to design a model-free fuzzy sliding mode controller (FSMC) for robotic motion and force control. The design process is briefly described as following paragraphs.

A sliding surface on the phase plane is defined as

$$ s(t) = \left(\frac{d}{dt}+\lambda\right)e_1 = e_2 + \lambda e_1, \tag{4} $$

where $e_i = x_{id} - x_i$ are defined as the state control errors. The sliding variable, s, will be used as the input signal for establishing a fuzzy logic control system to approximate the specified perfect control law,

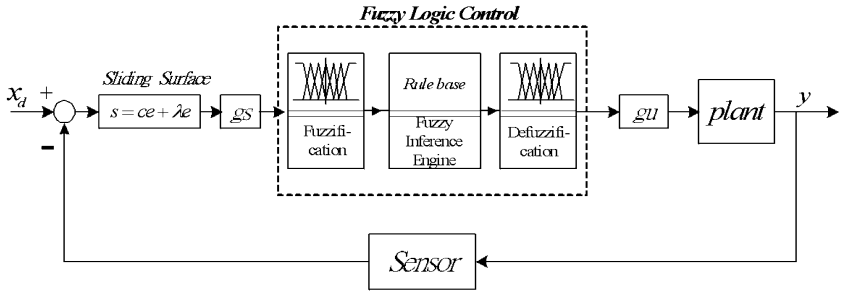


Fig. 3. Fuzzy sliding mode control block diagram.

u_{eq}. With this perfect control law, the closed loop control system has an asymptotical stability dynamic behavior [12].

$$\dot{s}(t) + \lambda s(t) = 0. \tag{5}$$

Since λ is a positive value, the sliding surface variable, s, will gradually converge to zero. Based on the definition of sliding surface variable, s, in Eq. (4), the system output error will converge to zero, too. In this study, a fuzzy system is employed to approximate the mapping between the sliding variable, s, and the control law, u, instead of model-based calculation. This control law may have certain difference with the perfect control law u_{eq}, then the following equation can be derived:

$$\dot{s}(t) = -\lambda s(t) + b(X,t)[u_{\text{eq}}(t) - u(t)]. \tag{6}$$

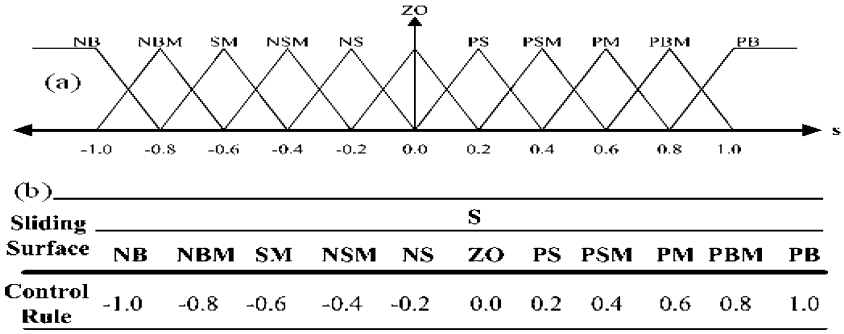
Generally, $b(X)$ is a positive constant or a positive slow time-varying function for practical physical systems. By multiplying both sides of the above equation with s gives

$$s(t)\dot{s}(t) = s(t)\{-\lambda s(t) + b(X,t)[u_{\text{eq}}(t) - u(t)]\}. \tag{7}$$

Based on the Lyapunov theorem, the sliding surface reaching condition is $s \cdot \dot{s} < 0$. If a control input u can be selected to satisfy this reaching condition, the control system will converge to origin of the phase plane. It can also be found that $\dot{s}$ increases as u decreases and vice versa in Eq. (6). If $s > 0$, then the increasing of u will result in $s\dot{s}$ decreasing. When the condition is $s < 0$, $s\dot{s}$ will decrease with the decreasing of u. Based on this qualitative analysis, the control input u can be designed in an attempt to satisfy the inequality $s \cdot \dot{s} < 0$. The relating theory about the convergence and stability of the adaptation process on the basis of the minimization of $s\dot{s}$ can be found in [13].

Here, a fuzzy logic control is employed to approximate the nonlinear function of equivalent control law, u_{eq}. The control voltage change for each sampling step is derived from fuzzy inference and defuzzification calculation instead of the equivalent control law derived from the nominal model at the sliding surface. It can eliminate the chattering phenomenon of a traditional sliding mode control. The controller design does not need a mathematical model and without constant gain limitation. The system control block diagram is shown in Fig. 3. The sliding surface variable, s, is employed as the one dimensional fuzzy input variable. The one dimensional fuzzy rules, Fig. 4(b), is designed based on the sliding surface reaching condition, $s \cdot \dot{s} < 0$.

Eleven fuzzy rules are employed in this control system to obtain appropriate dynamic response and control accuracy. The input membership functions are scaled into the range of -1 and $+1$ with equal span. Hence a scaling factor gs is employed to map the sliding surface variable, s, into this universe of discourse. A scaling factor gu is employed to adjust the value of control voltage. Membership functions of fuzzy input and output variables, and the fuzzy rules of the FSMC are shown in Figs. 4a and 4b, respectively.



(b)

Sliding Surface	S										
	NB	NBM	SM	NSM	NS	ZO	PS	PSM	PM	PBM	PB
Control Rule	-1.0	-0.8	-0.6	-0.4	-0.2	0.0	0.2	0.4	0.6	0.8	1.0

Fig. 4. (a) Sliding variables fuzzy membership functions; (b) joints fuzzy control parameters and fuzzy control rules.

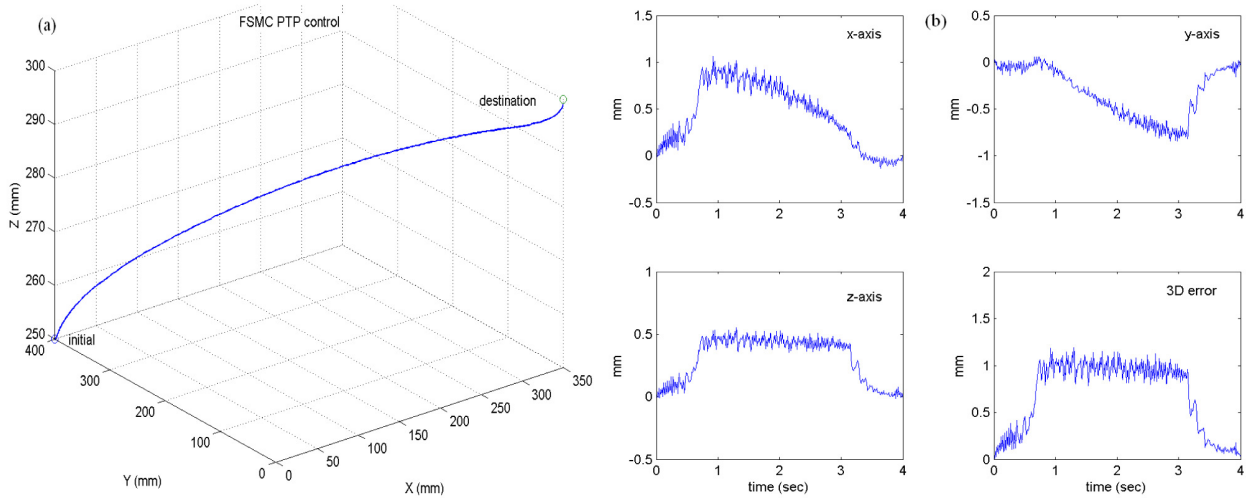


Fig. 5. (a) PTP motion response in Cartesian space, and (b) position tracking error in X, Y and Z directions and 3D contouring error.

The membership function used for the fuzzification is of a triangular type. The function can be expressed as

$$\mu(s) = \frac{1}{w}(-|s-a|+w), \tag{8}$$

where w is the distribution span of the membership function, s is the fuzzy input variable and a is the parameter corresponding to the value 1 of the membership function. The height method is employed to defuzzify the fuzzy output variable for obtaining the control voltage of each joint control motor (which is a nonlinear function derived from the fuzzy inference decision and defuzzification operation)

$$u = \frac{\sum_l^m \mu^j \cdot C^j}{\sum_l^m \mu^j} \equiv \sum_1^m \phi_j C^j, \tag{9}$$

where m is the rules number and C^j is the consequent parameter.

The gain scheduling parameter is used to map the corresponding variables into this nominal range. These mapping parameters are specified as gs, and gu for the sliding variable and control voltage, respectively. This approach is a novel gain scheduling 1D fuzzy sliding mode control structure. The values of these parameters are not critical for this gain scheduling fuzzy sliding mode controller. They can be roughly determined by

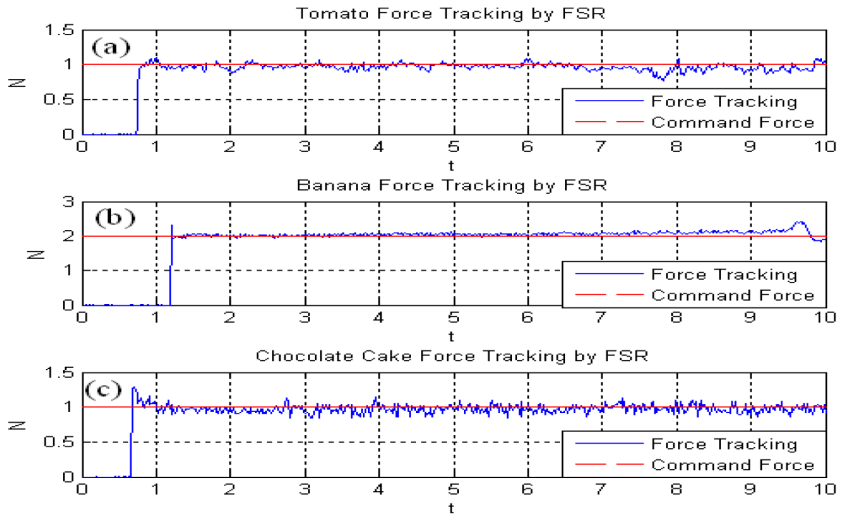


Fig. 6. The grasping force control monitoring for (a) small tomato, (b) banana and (c) chocolate cake.

simple experimental tests. Then the same values can be employed for different joint motion control and grip force control with appropriate steady state accuracy. This control strategy can switch automatically between end-effector positioning control and grip force control by a switching communication signal between FPGA and Arduino control kernels.

5. EXPERIMENTAL RESULTS

In order to achieve desire motion specification and avoid the collision in the motion environment, an appropriate controller should be designed to monitor the end-effector motion trajectory. The multi axis manipulator is planned to execute point to point (PTP) motion control purpose in this study. The trapezoid speed curve motion trajectory is planned for PTP motion. Since the SOPC system is employed to implement this robotic servo control system, the control system cannot provide large computation ability for the model-based controller. Here, a model-free 1D fuzzy sliding mode controller is designed for each joint to control this Mitsubishi RV-M2 5 DOF robotic system and monitor the gripper grasping force, respectively. In order to evaluate the transient and steady sate control performances, the following experiments were performed. The sampling frequency in these experiments was 100 Hz. Since the sliding variable s is divided into 11 fuzzy subsets from –1 to +1 with equal interval 0.2, a parameter gs is used to regulate the sliding variables into that range. A parameter gu was used to adjust the control input. These parameters are chosen as 15 and 4 (joint motion control) and 12.5 and 0.8 (grasp force control) for gs and gu, respectively. If these parameters are varied within 50% and 200% of the original specified values, the control system performance is not changed significantly.

Case (A): End-effector positioning control with step position change

For the pick-and-place application, the FSMC control strategy is employed to monitor the robot end-effector to move from a point to another point. The specified trajectory for the robotic end-effector is a trapezoid speed curve with a constant acceleration and deceleration interval for each joint, and it is moving from (0, 400, 250) mm to (350, 0, 300) mm in Cartesian space with 4 sec total motion time. The motion trajectory in Cartesian space and the position error in each coordinate axis are shown in Figs. 5a and 5b, respectively. The overall position trajectory tracking error is less than 1.0 mm. The destination steady state position error is 0.03 mm. It is accurate enough for industrial pick-and-place applications.

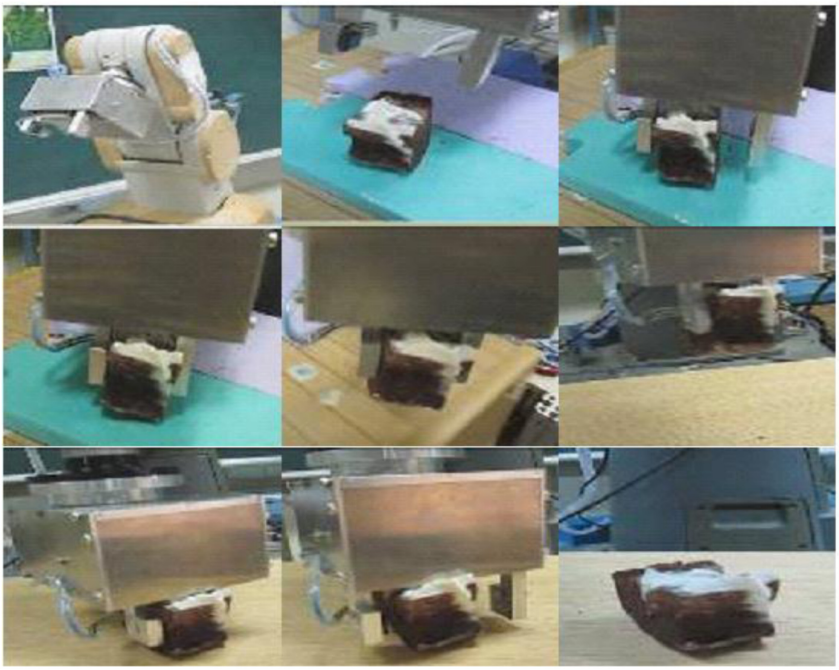


Fig. 7. Sequential pictures for demo the chocolate cake pick and place operation.

Case (B): Gripper force control for grasping soft foods

Before integrating this embedded control gripper into robot end-effector, the appropriate grasping force and FSMC control parameters are tested by experiments. The gripper is planned to grasp and pick up soft chocolate cake, small tomato, and banana without damage the object. The experimental results of contact force monitoring are shown in Figs. 6a, 6b and 6c for tomato, banana, and chocolate cake, respectively. The appropriate grasping force can be found based on gripper force/position relationship and an intelligent program. It can be observed that this FSMC control spends less than 0.1 sec to settle down the specified contact force with steady state error less than 0.1 N for grasping frangible fruits and chocolate cake.

Case (C): Integrating gripper into robot end-effector for pick and place operation

For the following soft foods pick-and-place operations, the robot end-effector is planned to move from the basis coordinate origin (0, 0, 0) to a point (450, 0, 250) mm first, and then perpendicular down to the object picking position (450, 0, 180) mm with gripper jaw perpendicular to horizontal plane for grasping object. Then, the robot FPGA controller sends a signal to gripper Arduino controller for activating the grasping force monitoring control function. After the grasping force converged, the Ardunio kernel sends a signal to start the FPGA robot arm motion controller for driving the end-effector moving up to (450, 0, 250) mm and then moving to another position (0, 450, 250) mm. Finally, the robot arm is manipulated to a specified object place down position (0, 450, 150) mm for commanding the gripper to release the object. Nine pictures of these sequent operations are shown in Figs. 7, 8a and 8b for pick-and-place chocolate cake, small tomato and banana, respectively.

It can be observed that this embedded robotic control system can effectively execute the soft objects pick-and-place operations without surface damage or object broken.

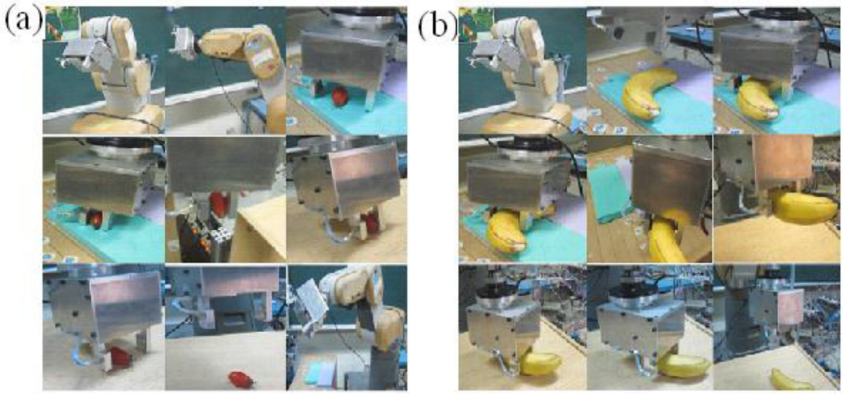


Fig. 8. Sequential pictures for demo (a) small tomato and (b) banana pick and place operation.

6. CONCLUSIONS

An embedded FPGA and Arduino double kernels control structure is constructed for robotic positioning and grasping force control, respectively. This embedded control structure is implemented on a retrofitted Mitsubishi 5 DOF robot. Both control kernels are switched in sequence with an activating signal for monitoring the end-effector position and grasping force individually. It can be employed in most of the pick-and-place applications. 1D model-free fuzzy sliding mode controller was designed for each joint to execute intelligent end-effector motion control and gripper force control, respectively. This control structure has low cost and model-free advantages for achieving good transient and steady state responses. The experimental results show that this FSMC intelligent control system can effectively monitor the specified robotic end-effector positioning with tracking error less than 1.0 mm and steady state error less than 0.2 mm, and the gripper can dexterous pick and place fragile fruits and chocolate cake with grasping force error less than 0.1 N. This low cost double kernels embedded control structure can be employed for industrial robotic pick-and-place or assembly operations with force monitoring requirement.

ACKNOWLEDGEMENT

The authors would like to thank the Industrial Technology Research Institute for part of the financial support of this research.

REFERENCES

1. Hogan, N., “Impedance control: An approach to manipulators, Parts I, II and III”, *ASME Journal of Dynamic System, Measurement and Control*, Vol. 107, No. 1, pp. 1–24, 1985.
2. Raibert, M.H. and Craig, J.J., “Hybrid position and force control of robot manipulators”, *ASME Journal of Dynamic System, Measurement and Control*, Vol. 102, No. 2, pp. 126–133, 1981.
3. Lawrence, D.A., “Impedance control stability properties in common implementation”, in *Proceedings of IEEE International Conference on Robotics and Automation*, Denver, CO, USA, pp. 1185–1190, 24–29 April, 1988.
4. Colbaugh, R. and Engelmann, A., “Adaptive compliant motion of manipulators: Theory and experiments”, in *Proceedings of IEEE International Conference on Robotics Automation*, San Diego, CA, USA, pp. 2719–2726, 8–13 May, 1994.
5. Lin, C.H. and Lin, C.P, “Integral backstepping control for a PMLSM using adaptive RNNUO”, *International Journal of Engineering and Technology Innovation*, Vol. 1, No. 1, pp. 53–64, 2011.
6. Viba, J., Fontaine, J. and Kruusmaa, M., “Motion control optimization of robotic fish tail”, *Journal of Vibroengineering*, Vol. 11, No. 4, pp. 607–616, 2009.

7. Jacobsen, S.C., "The Utah/MIT dexterous hand: Work in progress", *International Journal of Robotic Research*, Vol. 3, No. 4, pp. 21–50, 1984.
8. Michelman, P., "Precision object manipulation with a multifingered robot hand", *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 1, pp. 105–113, 1998.
9. Webster, J.G., *Tactile Sensors for Robotics and Medicine*, John Wiley & Sons, 1988.
10. Caldwell, D.G., Buvsse, A. and Shou, W., "Multi-sensor tactile perception for object manipulation/identification", in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, USA, Vol. 3, pp. 1904–1911, 7–10 July, 1992.
11. Edwards, C. and Spurgeon, S.K., *Sliding Mode Control – Theory and Applications*. Taylor & Francis, London, Bristol, 1998.
12. Wang, L.T. and Chen, C.C., "A combined optimization method for solving the inverse kinematics problem of mechanical manipulator", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, pp. 489–499, 1991.
13. Hwang, G.C. and Lin, S.C., "A stability approach to fuzzy control design for nonlinear systems", *Fuzzy Sets Systems*, Vol. 48, No. 3, pp. 279–287, 1992.