A CURRENT-SENSING BASED CONTROLLER OF BRUSHED DC MOTORS FOR ROBOTIC APPLICATIONS

ARUN DAYAL UDAI
Ph. D Scholar, Department of Mechanical Engineering, IIT Delhi, INDIA, arun_udai@yahoo.com

SUBIR KUMAR SAHA
Department of Mechanical Engineering, IIT Delhi, INDIA, saha@mech.iitd.ac.in

This paper proposes an inexpensive low power hardware drive solution for the control of a brushed DC motor which can be used for compliant robotic tasks. The system has been designed for simultaneous driving of four DC motors where each can be controlled in terms of velocity, position and maximum torque. Full speed USB 2.0 communication support of a microcontroller allowed computationally expensive tasks to be performed on a PC, whereas the motors were driven directly in position/torque control mode. The controller can find its application in compliant control of a robotic manipulator performing tasks like grinding, deburring, polishing, master-slave systems or assembly operations.

I. INTRODUCTION

Most of the domestic and specialized industrial operations demand a controlled force/torque (F/T) to be delivered by its actuators so that they may not damage the environment with which they are interacting. A robot running only on position control applies its full capacity to follow a trajectory when obstructed by any object existing in its workspace. However, robots that can sense an obstruction can intelligently control their driving torque to perform compliant robotic tasks. This requires forces to be controlled using an explicit force/torque sensor fitted either at the wrist [1] or at the joint [2] or at the robot-base [3]. The arrangement enforces inherent trade-off between the wrist-sensor stiffness and the manipulator's response for force and position [4]. A robot with a wrist-sensor is not capable to detect any obstruction on the way of other links. Solution to this is a joint-torque sensing arrangement with which contact with any other link can be detected. Advantages of joint-torque sensory feedback over wrist-sensor were discussed in [5]. Various other approaches, e.g., in [6], [7], etc., have been proposed in the past that do not depend explicitly on the force/torque sensor feedback. Introducing a joint-torque sensor makes the joint construction complex, heavier and costly. It also reduces the structural stiffness of the joints. Approaches to joint-torque estimation based on physical modeling and its runtime parameters were attempted in the past [8 – 10] for synchronous motor drives.

The proposed work implements a methodology to estimate the torque of a brushed DC motor based on its current measurements during runtime. This eliminates the need for an explicit force/torque sensing device at the joint or at the end-effector which is an expensive and complex solution. Note that the brushed DC motors are generally used in compact low power robots, industrial automation systems, medical equipment, desktop electronic devices, etc. due to their simplicity in control and compact designs. Brushed DC motors are mathematically predictable as their output torque varies linearly with the current flowing through the armature, and its direction can be simply reversed by reversing the armature current. Thus, the entire control scheme is converged to precise sensing and controlling of the current only. Current based techniques for control of brushed DC motors have been attempted in the past using power amplifiers [11], [12] and [13]. The model proposed by [11] has inherent disadvantage that it reduces the driving voltage

Proc. of the 3rd IFToMM International Symposium on Robotics and Mechatronics (ISRM 2013)
Edited by I-Ming Chen, Marcelo Ang, Guilin Yang, Kin Huat Low and Albert Causo

ISBN: 978-981-07-7744-9 :: doi:10.3850/978-981-07-7744-9_009

across the motor and thereby reducing the speed of its armature. This causes a reduced performance and inaccurate manipulator trajectory. The method in [12] used a current chopper drive using a microcontroller to directly control the power input to the motor and thereby controlling the speed of the motor. In [13], current sensing was only used for preventing the controller against over-current and the speed of the motor was controlled in both directions by regulating the Pulse Width Modulation (PWM) duty cycles in both directions of armature current. Table I lists a survey of commercially available brushed DC motor controllers with their technical features. Most of them are limited either by type of communication, feedback they provide, access to control parameters, or by the type of control they offer. It may be observed that most of these controllers are designed for velocity or position control applications and current information was used mostly for over-current protection. However, there are few motion controllers available from manufacturers like Galil Motion Control, Advanced Motion Controls (AMC), Copley Controls, Elmo Drives etc. which can do simultaneous position/velocity and current control but their applications are limited to industry due to their high cost. Moreover, complete access to internal firmware is still limited, which limits dynamic gain changing which may be required in case of adaptive controllers. This motivated us to design an inexpensive brushed DC motor controller which would provide an access to all the control parameters for both position and current control with high-speed communication. Being able to control both position and current is a perquisite for implementing any force/position control algorithm.

Table I: Features of Commercially Available Brushed DC Motor Controllers

Make	Braking	Closed Loop Velocity/Position Control	Current Control	Channels	Communication
RobotQ	Yes	Both	No	1/2	Serial (RS232)
Scorpion XL	No	Position	No	1	TTL Serial, I2C, SPI
PARALLAX	No	Velocity	No	1	RS232/ USB
Rover 5	No	Velocity	Sensing for overload detection, no feedback	4	Digital
MAXON	Yes	Both	Mostly included in the controller, but not accessible/controllable	1/2/4	RS232/ Ethernet/ USB
MD22	No	Predefined modes	No	2	I2C
Robo Claw BASIC MICRO	Yes	Velocity	No (Over-current check)	2	TTL Serial
BasicATOM Nano BASIC MICRO	No	Velocity	No	4	TTL Serial
Roboteq	No	Both	No	1/2	TTL Serial/ Analog
Sabertooth Dimension Engineering	Yes	Velocity	No	1/2	TTL Serial/ Analog
Pololu SMC	No	Velocity	No	2	RS232
SyRen10/20 Dimension Engineering	Yes	Velocity	No	1	TTL Serial/ Analog
MegaMoto	Yes	Velocity	Current feedback on any one of the channels	2	TTL Signals
Cytron MD30B	No	Velocity	Current limiting	1	TTL Signals

This paper is organized as follows: Section II explains the requirement of the proposed controller, whereas section III provides the development details of the controller. Section IV contains the calibration techniques used for the controller design, followed by the testing of the controller in section V. Finally, conclusions are given in section VI.

II. REQUIREMENT OF THE PROPOSED CONTROLLER

For the development of a current-sensing based controller, a target robot, namely an old MA3000 was chosen which is shown in Figure 1. The robot's controller was disconnected in order to be able to connect the proposed controller to the robot. The key requirement of the proposed controller was taken as to control the end-effector's force without introducing any external joint or end-effector force sensor. It may be noted that except the mechanical arm with the motors of the robot, no other electrical/electronic components of the original robot was used. All the actuators had gears with reduction ratio below 50:1, which allowed the joints to be back drivable with small friction. Links are coupled to motors through gears and belt system.

One of the key challenges was to sense and precisely control small current in noisy environment. Common sources of noise included commutation brush noise, PWM jitter, torque ripple due to gearing and transmission, etc. The controller board therefore was designed with four copper layers having two inner power planes carrying the control power supply of 5V to obtain stable power source for Digital-to-Analog converter (DAC) and Analog-to-Digital (ADC) reference, as illustrated in Figure 2. One percent tolerance resistances were used to obtain precise current limiting characteristics. For the widely varying current control requirement of 0.05A to 0.4A for each motor, a driver chip from Texas Instruments, i.e., DVR8814 [14] was chosen. This driver chip allows two motors to be driven with different maximum current by just varying the sensing resistances and applying the required driving voltages. Two chips were used in the scheme to control four motors as the intention was to control the four working DC motors of the original five Degree-of-Freedom (DOF) MA3000 robot.

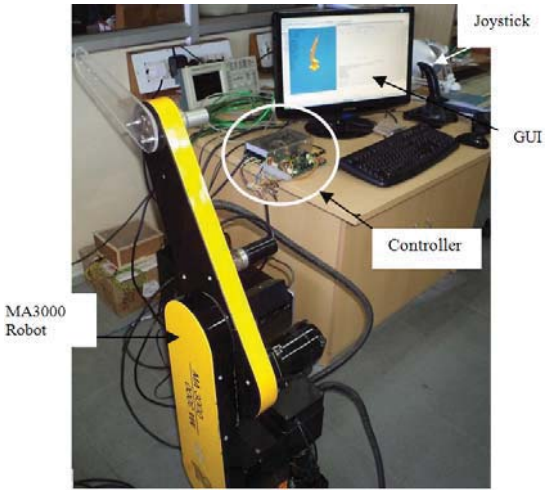


Figure 1. Target Robot MA3000

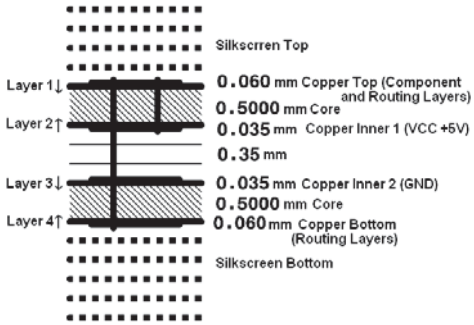


Figure 2. Layers of the controller board

III. DEVELOPMENT OF THE CONTROLLER

Different stages of the development of the controller for current-based force control of a robot are explained below.

Architecture of the Controller

Each motor requires three microcontroller pins including one PWM pin for controlling its speed. Also, to communicate with an 8-bit DAC device it also requires additional eight output pins in the microcontroller apart from eight ADC input pins for taking inputs for current and potentiometer feedback from each motor. With these constraints in the number of input/output pins, ADC and USB feature for communication, a 64-pin microcontroller with USB communication was utilized. These controlled the speed and torque of a DC motor using PWM signals and current control, respectively. As shown in Figure 3, an 8-bit USB microcontroller AT90USB1287 [15] was used which has 128k bytes of flash memory and runs at 16 MHz clock speed for on-board processing, communication and sensor interfacing. The microcontroller was interfaced with an 8-bit quad voltage output DAC device TLC7226 [16] for precise voltage generation of motor driver for current control reference inputs. The microcontroller was also interfaced with two DRV8814 motor driver chips which have dual H-Bridge current control with braking mode inputs. Winding current control inputs of the driver chip were supplied with voltage outputs of the DAC. The direction control of the motors was done by phase inputs of the driver where the velocity control was done by 8-bit PWM signals generated by the microcontroller and fed to the motor enable inputs.

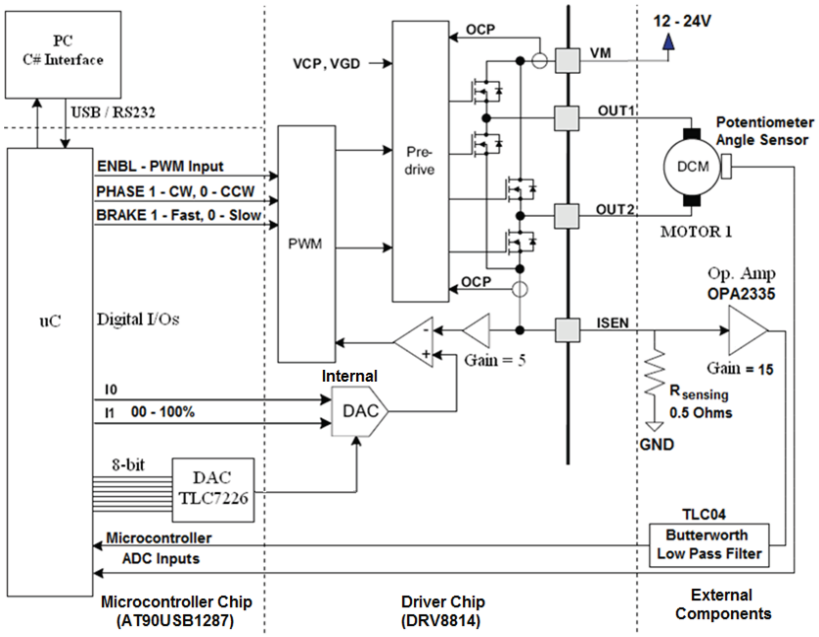


Figure 3. Scheme for the proposed robot controller

Based on the scheme of the proposed robot controller, a PCB was fabricated and assembled, as shown in Figure 4, whose specifications are shown in Table. II. The developed controller has been featured with special functionalities, namely, position and velocity control, braking, and current limiting for torque control. These are explained next.

Position and Velocity Control

While an ADC input pin of the microcontroller reads the potentiometer's voltage for the angle sensing of the motor, its velocity was controlled by a PWM output pin. An 8-bit PWM of 31.25 kHz was applied through the microcontroller's output compare pins. Higher the duty cycle of the PWM higher will be the speed of the motor shaft. The ADC clock was prescaled at 100 kHz with which it takes 13 clock cycles [15] to sample one data from the single ended input. The speed of ADC sampling was adjusted to have a proper balance between the sampling, communication and processing. Velocity of the shaft was calculated by calculating the ratio of the change in ADC value to the sampling time. An alternative way to estimate the velocity could be to measure the back electro-motive-force (emf) across the DC motor when the PWM signal is LOW. It is directly proportional to the back emf voltage [13]. As each ADC conversion with any voltage input to the ADC pin takes almost 13 clock cycles to process, calculating the velocity by taking derivative was less computationally expensive, as can be seen from the calculation of velocity using optical encoders and microcontrollers [20]. Following the methodology of [20], difference in ADC input from the potentiometer readings were used in place of encoder counts.

Table II. Sprecifications of the Developed Controller

Parameter	Value
Number of Channels	4
Maximum current per channel	2.5 Amperes (at 24 Volts)
Voltage range per channel	8 Volts – 45 Volts
Control supports	Closed loop control for position/ velocity and current
Braking mode	Regenerative (slow braking), reverse current braking (fast braking)
Mode of communication	USB 2.0, RS – 232 Serial
Speed of communication	Maximum: 12 MHz (full speed USB)
Position feedback	0 – 5 V with 8 bit resolution
Measured voltage for Current feedback	0 – 5 V with 8 bit resolution
Command	8 bit for both current and velocity
Firmware	Upgradable for future modifications with Open Source
Operating Temperature	-45^{o}C to 85^{o}C (with air cooling)
Safety	Thermal shutdown 150^{o}C; Overcurrent trip; Under voltage lockout.

Braking

As shown in Figure 5, two different modes of current decay modes, namely, fast decay and slow decay, were exploited for braking. In fast decay mode, once the chopping current reaches the

threshold limit, the H-bridge reverses state to allow winding current to flow in the reverse direction. As the winding current approaches zero, the bridge is disabled to prevent any reverse current flow. In slow decay mode, winding current is re-circulated by enabling both the low-side Field Effect Transistors (FET) in the bridge. Fast decay mode allows instant braking of the DC motor and prevents any overshooting of angular position in case of position control.

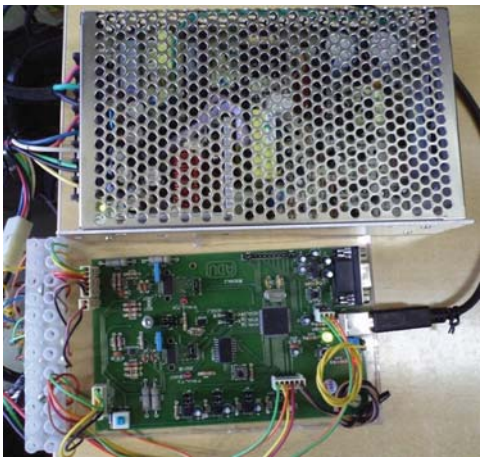


Figure 4. Assembled PCB with power supply

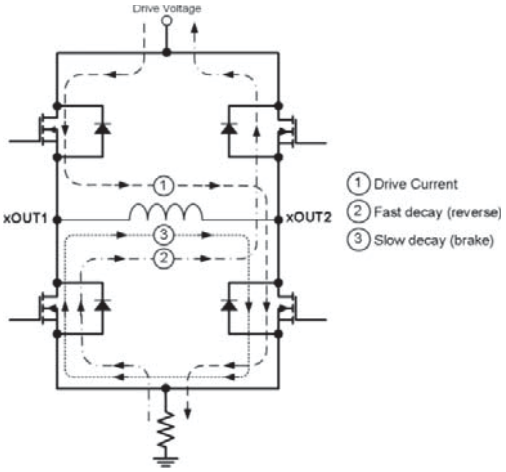


Figure 5. Braking modes in H-Bridge

Current Limiting for Torque Control

The current was limited by current chopping mechanism of the driver chip DRV8814. When the H-Bridge was enabled, winding current through the motor rises at a rate depending on the applied voltage and the inductance of the winding. When the current reaches the threshold, its flow through bridge was disabled until the beginning of the next internal PWM cycle. The threshold can be set by the DAC input to the reference of the driver. The maximum driving current through the motor can be given by (1) as follows:

$$I_{CHOP} = \frac{V_{REF}}{5R_{sensing}} \tag{1}$$

where V_{REF} is the reference voltage to the driver and $R_{sensing}$ is the low side current sensing resistance. The values of V_{REF} and $R_{sensing}$ were taken as 2.5V and 0.5 ohms, respectively, thus limiting the maximum current to 1 A with 100% DAC scaling inputs, i.e., I0 and I1, of Figure 3. Keeping the internal DAC at 100%, reference voltage, V_{REF} was made to vary from 0 – 2.5V using an 8-bit quadruple DAC TLC 7226, to vary the current through the motor. A reference diode LM336 was used to supply a constant 2.5V to the DAC chip. Voltage across the current sensing resistor was picked up by a low offset voltage (5μV) and zero drift operational amplifier, OPA2335 at a constant gain of 15. So, a variation of 0.03V to 0.3V across current-sensing resistance during normal operation of the DC motor was amplified to maximum of 4.5V that fell well within the input limit of the microcontroller ADC pin. The sensed voltage was made to pass through a low pass fourth-order Butterworth filter TLC04 [18] to filter out the noise that may have caused due to DC motor brush, PWM jitter, torque ripple etc. With this current measurement, a

closed-loop servo was formed to run the motor in current control mode. It controlled the maximum torque of a motor.

IV. CALIBRATION AND IDENTIFICATION

In order to be able to use the developed current-sensing controller for torque control, an appropriate calibration was needed. For that, an experimental setup was made to measure the torque at a joint, namely the fourth joint, and the corresponding currents drawn by the DC motor was measured. This step also proved the satisfactory functioning of the developed controller for its current-sensing, and limiting it to a specified value. The stalling torque of the motor was measured using an ATI Force/Torque (F/T) sensor fitted at the joint. The motor was allowed to rotate for ½ revolutions before stalling so as to reduce the effects of any static friction in the shaft. Special precautions were taken to obtain the minimum torque required to stall the motor.
Stalling currents Vs. commanded torques are plotted in Figure 6, where the stalling currents are along y-axis. Note that the commanded torques are in 0-255 scale along x-axis. Since the torque commands were transferred through byte stream of USB interface and their range is between 0 to 255 values. The slope of the fitted straight line using curve fitting tool of the MATLAB software gave the value of motor constant K_m. The line does not pass through the origin which may be attributed to the friction in the motor shaft. Any required torque can then be obtained from the current (I) Vs. torque (T_c) relationship given by (2).

$$I = 0.0031T_c - 0.29 \tag{2}$$

It was observed that the stalling current varied almost linearly with the torque with an error of ±0.05A, as expected for a DC motor. Next, the current measurement was calibrated against precise torque measurements using a F/T sensor. Figure 7 shows the variation of measured torques with commanded torques in 0-255 scale. The variation here is also linear with a maximum variation of 0.15 Nm. The equation of the fitted line of actual torque, represented by T_a, Vs. the commanded torque, represented by T_c, is given by (3)

$$T_a = 0.01654\ T_c - 0.66818 \tag{3}$$

Equation (3) allows one to obtain any torque command required to attain a particular stalling torque. Since at the torque at the output shaft of a DC motor is proportional to its armature current, the motor constant at joint 4 can then be easily identified. This step was carried out as the motor specifications were not available. Actually, the robot is very old (procured in 1980's) and manufacturer of the motors (Bodine Electric) could not provide any data of those motors because they are no longer in production.
For the above calibration purpose, an 8-bit torque and velocity commands for each motor were sent through the USB interface using a Graphical User Interface (GUI) developed in C#. The GUI can specify the limit of the stalling current value to the motor, and sets the duty cycle of the PWM outputs to the motor driver for speed regulation. The GUI is designed to take input from Joystick and communicate to the controller board through the PC's USB port. The controller board in turn drives the motors. Note that other motors can be calibrated in a similar manner and their motor constants be identified. However, they were not carried out here mainly because the emphasis of the present paper is to report the development of a current-sensing controller and its satisfactory functioning.

V. TESTING OF CONTROLLER

For the purpose of torque estimation of a moving link based on current-sensing approach, the last link of the original MA3000 was replaced with an aluminum link of known mass, i.e., 0.263 Kg with its center of gravity located at 0.015m from the axis of rotation (Fig. 1 shows a similar link of perspex). This was made to rotate from lowest vertical position to top vertical position, i.e., from 180 degrees to 0 degree. Current measurements were then taken using the developed controller. Using Figs. 6 and 7, and the motor constant K_m, as identified in section IV, actual torques were estimated as

$$T_e = K_m I + T_f \quad (4)$$

where I is the current through the armature sensed using the developed controller. The term T_f is the coulomb friction torque when the armature shaft is static and remains constant once it starts rotating. This is suited for slow speed rotating shafts [21] and is assumed for the proposed model. Since, there is a rotation about one axis only, the ideal torque to perform the task against gravity can be calculated as

$$T_i = mgd \sin\theta \quad (5)$$

where m is the mass of the link, g is acceleration due to gravity, d is the distance of the centre of gravity from the axis of rotation and θ is the vertical angle of inclination of the link. Figure 8 shows the variation of the ideal torque T_i and the estimated torque T_e based on the current-sensing. Note that the current signals are noisy as they were tapped before the Butterworth Low Pass Filter, as shown in Figure 3. This was so as no provision was made in the PCB to tap the current signal after the filter. This will be taken care off in the future versions of the controller. Greater deviation of the torque during start may be attributed to frictional torque being dominant at slow speed.

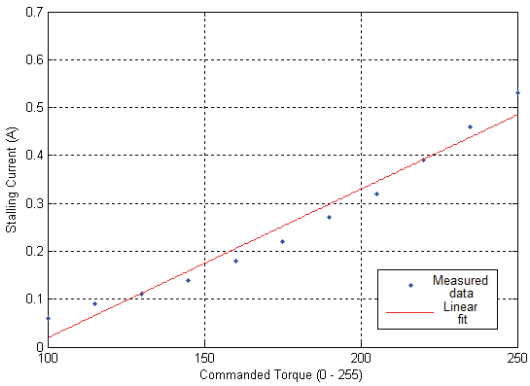


Figure 6. Stalling current (I) Vs. Commanded Torque (T_c)

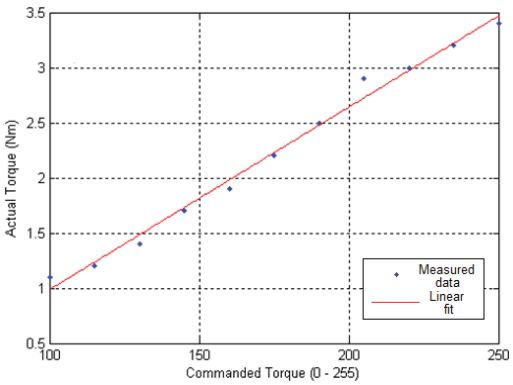


Figure 7. Actual torque (T) Vs. Commanded Torque (T_c)

In order to check the functioning of all four channels, they were tested separately for torque control of the last link connected at joint 4. They were found to be performing satisfactorily and consistently. From Figure 8, the maximum variation in estimated torque is 0.002 Nm with reference to the ideal torque, which can be considered as acceptable for practical application with force control. Next, to test the developed controller, for the motor at joint 2 and 3 currents of motors 2 and 3 were also recorded, which are shown in Figure 9. As axis one moves the link perpendicular to the gravitational force no variation current can be observed and thus not reported. Plots for the torques at these joints can also be estimated provided the motor constants are known using the

methodology provided in section IV for the motor at joint 4. However, these constants are generally available in the manufacturer's specifications. They were not reported here as the emphasis of this paper is more on the functioning of the control hardware, which has been demonstrated successfully.

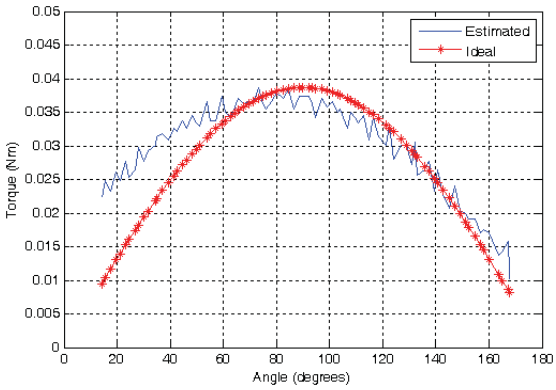
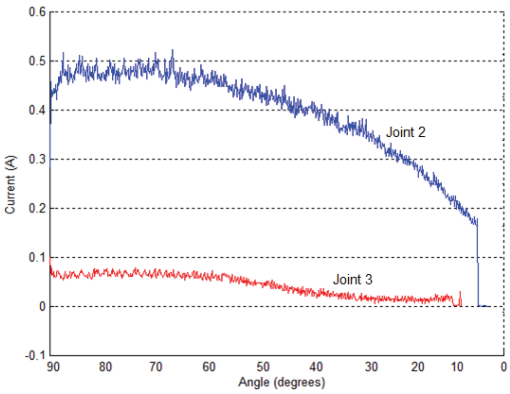


Figure 8. Ideal and estimated torques at Joint 4.



Figire 9. Current variations at Joint 2 and 3.

VI. CONCLUSIONS

This paper proposed an inexpensive integrated architecture of a controller which can be used for combined position and force control of a robot's end-effector. It used a current-sensing technique to estimate the joint torques without the need for an explicit force/torque sensor fitted to the joint or to the wrist. Besides, its features of continuous current control and thereby controlling the torque during runtime make it different and unique from any of the existing controller in the market as pointed out in Table I. The controller is useful for any robotic system performing compliant manipulation tasks. Control schemes as suggested by [22] or [23] can be used for the motion control of the end-effector of a robot based on the proposed current-sensing controller. Also models of friction at different shaft speeds and coil temperature may be included in the proposed model for joint torque estimation. This will be done in future and will be reported separately.

ACKNOWLEDGEMENTS

The authors sincerely acknowledge the technical inputs given by Mr. D. Jaitly of Mechatronics Lab. at IIT Delhi and Mr. K. Sujit of BIT, Mesra which led to the successful realization of this work. The research assistantship provided to the first author from the sponsored project entitled "Adaptive Force control of an Industrial Robot (KUKA KR6) equipped with Force/Torque Sensor," funded by BRNS/BARC, Mumbai is also acknowledged.

REFERENCES

[1] Khatib O., "A unified approach for motion and force control of robot manipulators: The operational space formulation," IEEE J. of Robotics and Automation, 3(1), pp: 43-53, 1987.

[2] Albu-Schäffer A., Haddadin S., Ott C., Stemmer A., Wimböck T., Hirzinger G., "The DLR lightweight robot: design and control concepts for robots in human environments," Industrial Robot: An International Journal, 34(5), pp. 376-385, 2007.

[3] Ott C., Nakamura Y., "Base force/torque sensing for position based Cartesian impedance control," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), pp. 3244-3250, 2009.
[4] Roberts R., Paul R., Hillberry B., "The effect of wrist force sensor stiffness on the control of robot manipulators," IEEE International Conference on Robotics and Automation (ICRA 1985), Vol. 2, pp. 269- 274, 1985.
[5] Aghili F., "Joint torque sensory in robotics," Industrial Robotics: Programming, Simulation and Applications, Low Kin Huat (Ed.), InTech Open, 2006, pp: 23-46.
[6] Tanie K., Yokoi K., Kaneko M., and Fukuda T., "A position sensor based torque control method for a DC motor with reduction gears," Proceedings of IEEE International Conference on Robotics and Automation (ICRA 1988), Vol. 3, pp. 1867-1870, 1988.
[7] Kaneko K., Komoriya K., Ohnishi K., and Tanie K., "Accurate torque control for a geared DC motor based on an acceleration controller," International Conference on Industrial Electronics, Control, Instrumentation, and Automation, Vol. 1, pp. 395-400, 1992.
[8] Chung S. K., Kim H. S., Kim C. G., and Youn M. J., "A new instantaneous torque control of PM synchronous motor for high-performance direct-drive applications," IEEE Transactions on Power Electronics, 13(3), pp. 388-400, 1998.
[9] Mohammed O. A., Liu S., and Liu Z., "Physical modeling of PM synchronous motors for integrated coupling with Machine drives," IEEE Transactions on Magnetics, 41(5), pp. 1628-1631, 2005.
[10] Petrovi V., and Stankovic A. M., "Modeling of PM synchronous motors for control and estimation tasks," 40th IEEE Conference on Decision and Control, Vol. pp.2229-2234, 2001.
[11] Kapoor A., Simaan N., and Kazanzides P., "A system for speed and torque control of DC motors with application to small snake robots," IEEE Mechatronics & Robotics (MechRob 2004), Germany, Sept. 2004.
[12] Castagnet T., and Nicolai J., "Digital control of brushed DC motor," ST Application Note, 1999.
[13] Brushed DC motor control using LPC2101 – (AN10513)," Philips, Application Note, 2007.
[14] DRV8814 Datasheet, Texas Instruments, October, 2010.
[15] AT90USB1287 Datasheet, ATMEL Corporation, 2009.
[16] TLC7226 Datasheet, Texas Instruments, April, 2009.
[17] OPA2335 Datasheet, Texas Instruments, July, 2003.
[18] TLC04 Datasheet, Texas Instruments, March, 1995.
[19] MAX232 Datasheet, MAXIM Integrated, 2010.
[20] Dunworth A., "Digital instrumentation for angular velocity and acceleration," IEEE Transaction on Instrumentation Measurement, Vol. IM-18, pp. 132–138, 1969.
[21] Canudas, C., Astrom, K., Braun, K., "Adaptive friction compensation in DC-motor drives," IEEE J. of Robotics and Automation, Vol. 3(6), pp.681-685, 1987.
[22] Kosuge K., Takeuchi H., and Furuta K., "Motion control of a robot arm using joint torque sensors," IEEE Transactions on Robotics and Automation, 6(2), pp.258-263, 1990.
[23] Zhang G., and Furusho J., "Control of robot arms using joint torque sensors," IEEE International Conference on Robotics and Automation (ICRA 1997), 18(1), pp. 48-55, 1997.