

# A System for Speed and Torque Control of DC Motors with Application to Small Snake Robots

Ankur Kapoor, Nabil Simaan, Peter Kazanzides

Department of Computer Science

Johns Hopkins University

Baltimore, MD 21218

Email: {kapoor,nsimaan,pkaz}@cs.jhu.edu

**Abstract**—This paper describes a custom Low Power Motor Controller (LoPoMoCo) that was developed for a 34-axis robot system currently being designed for Minimally Invasive Surgery (MIS) of the upper airways. The robot system will include three robot arms equipped with small snake-like mechanisms, which challenge the controller design due to their requirement for precise sensing and control of low motor currents. The controller hardware also provides accurate velocity estimate from incremental encoder feedback and can selectively be operated in speed or torque control mode. The experimental results demonstrate that the controller can measure applied loads with a resolution of  $0.8N$ , even though the transmission is non-backdriveable. Although the controller was designed for this particular robot, it is applicable to other systems requiring torque monitoring capabilities.

## I. INTRODUCTION

Minimally invasive surgery (MIS) of the throat is characterized by insertion of endoscopes and multiple long tools through a narrow tube (the laryngoscope) inserted into the patient's mouth. Current manual instrumentation is awkward, hard to manipulate precisely, and lacks sufficient dexterity to permit common surgical subtasks such as suturing vocal fold tissue. This clinical problem motivated the development of a novel system for MIS of the upper airway including the throat and larynx [1] [2]. This system has a master/slave design similar to other telesurgical robots. We give a brief overview of the slave unit, focusing on its snake-like mechanism because it placed the most demands on the design of the low level controller, which is the subject of this paper.

Figure 1 shows the design of the 3 armed slave robot that works through a laryngoscope. Our design includes a laryngoscope, a base link, two similar Distal Dexterity Units (DDU) for tool/tissue manipulation, and another DDU for suction. Each DDU is a 5 degree of freedom (DoF) robot mounted on a DDU holder, which is manipulated by a 4 DoF Tool Manipulation Unit (TMU). The TMU controls the angle of approach, the rotation about and the position along the axis of the DDU holder. The TMU's are mounted on a Rotating Base Unit (RBU), permitting the system to be oriented within the throat so as to minimize collisions between DDU holders. The DDU holders are thin tubes (about 4 mm in outside diameter) providing an actuation pathway for the DDU and possibly a light-source or suction channel. Each TMU is equipped with a fast clamping device for adjusting the

axial location of the DDU. The actuation unit of each DDU is located at its upper extremity and the actuation is by super-elastic (NiTi) tubes operated in push-pull mode. This system implements actuation redundancy in the design and control of the DDU's. Each DDU and TMU has 7 and 4 actuated joints, respectively, which totals 34 actuated joints for the three armed slave robot and RBU.

The snake-like unit of the DDU is composed of a base disk, an end disk, several spacer disks and four super-elastic tubes (backbones) arranged as in Fig. 2. The central tube is the primary backbone while the remaining three tubes are the secondary backbones. The secondary backbones are equidistant from the central backbone and from one another. The central backbone is attached to both the base and end disks and to all spacer disks while the secondary backbones are attached only to the end disk and are free to slide and bend through properly dimensioned holes in the base and spacer disks. These secondary backbones actuate the snake in both push and pull modes, which makes it possible to satisfy statics of the structure while preventing buckling of the backbones - an important feature for successful reduction of diameter to 4mm or less. This makes the design of the controller challenging because it must prevent buckling by ensuring that the backbones do not get overloaded.

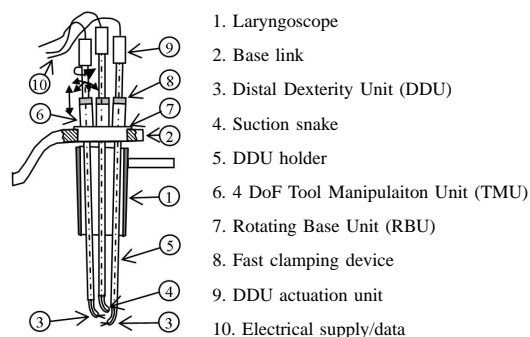


Fig. 1. Slave Robot Design

## II. DESIGN OF THE LOW LEVEL SYSTEM CONTROLLER

The requirements for controlling the surgical robot system are as follows:

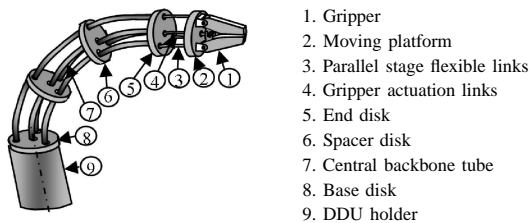


Fig. 2. The Distal Dexterity Unit (DDU)

TABLE I  
MAXON DC MOTOR SPECIFICATIONS

Rated Power	0.75W
Nominal Voltage	12V
No load current	6mA
Starting current	106mA
Terminal resistance	114Ω
Terminal inductance	0.92mH
Total diameter	10mm

- 1) Control the three 7-axis DDU's, three 4-axis TMU's and one RBU, for a total of 34 actuated joints.
- 2) Avoid buckling the snake portion of the DDU by preventing motors from exceeding strain limits of the secondary backbones.
- 3) Initialize (home) the robot with little or no motion to avoid buckling and to allow re-homing when deployed in the patient.
- 4) Measure and limit the interaction forces with the environment.

The surgical robot system is a small-scale design and does not require high joint speeds. Therefore, it is possible to use small, low-power motors with high gear reductions. We selected the Maxon RE Series brush DC motor with integrated planetary gearhead and (magnetic) incremental encoder for the DDU and possibly the TMU and RBU as well. Some key motor specifications are shown in Table I. Note that the stall current is 106 mA and the no-load current is only 6 mA. We decided to prevent overloading the snake backbones by controlling (or at least limiting) the motor currents. We are still evaluating methods for measuring interaction forces. The easiest solution is to add external force sensors, but this has the potential disadvantage of increasing system size. Once we determine the force measurement resolution required for our clinical application, we will evaluate whether we can adequately estimate interaction forces from the motor current feedback. At a minimum, we anticipate that the motor current feedback will give us sufficient information about the backbone tensions to develop control laws that resolve its actuation redundancy (3 secondary backbones providing 2 dof). Because the motors contain incremental encoders, our controller must also obtain feedback from an absolute position sensor, such as a potentiometer. Finally, the design must be scalable to handle 34 axes with minimal wiring complexity.

We investigated several commercially-available motion controllers and amplifiers and found several solutions that satisfied

most of the requirements, but none that satisfied all of them. One major difficulty was that even the small (low power) amplifiers were rated at several Amps and would therefore provide motor current feedback signals that were scaled to our range. This would not provide sufficient resolution for our measurements. We considered solutions that combined off-the-shelf components with custom components, but eventually decided that the resulting wiring complexity for the 34 axis robot system would lead to reliability and maintenance problems. We therefore designed a custom board that provides all I/O hardware and power amplification to control up to four axes. With this arrangement, each 7-axis DDU requires two boards, each 4-axis TMU requires one board and the 1-axis RBU requires one board, for a total of 10 boards (9 if the RBU is controlled from an unused DDU channel).

For simplicity, we selected a centralized architecture in which all control computations are performed on a single PC and motor I/O is accomplished via the custom boards installed in the PC (to further simplify our design task, we chose the ISA bus). Figure 3 shows the custom controller board, called the Low Power Motor Controller (LoPoMoCo). The entire 34-axis controller consists of 9 LoPoMoCo boards and a single board computer installed in a PICMG backplane. Each controller board contains a high-density 50-pin (SCSI-2) connector for motor power and feedback signals for 4 axes. To simplify cabling, we also designed a small 2-layer board that interfaces 4 Maxon motors/encoders and 4 analog sensors to the 50-pin cable (see Figure 3). The following sections give further details about the LoPoMoCo.

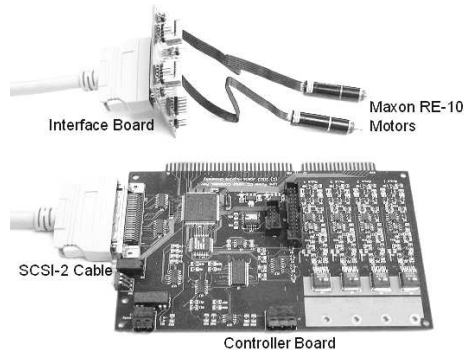


Fig. 3. Low Power Motor Controller (LoPoMoCo)

#### A. Power Amplifier Section

One key requirement is to sense small motor currents (less than 100mA), which can be difficult in a noisy environment. Common sources of noise include:

- Brush noise due to the make/break characteristic of brush commutation.
- Amplifier noise due, for example, to pulse width modulation (PWM) designs.
- Torque ripple due to gearing effects.

We therefore chose to build a custom amplifier that is properly scaled for our expected motor currents, includes low-pass filtering of sensed current and uses a linear design.

We chose OPA547 (or OPA548) operational amplifier (op amp) from the Burr-Brown division of Texas Instruments because it provides the following features:

- Operation at supply voltages up to  $\pm 30\text{V}$ .
- 500 mA continuous current and 750 mA peak current (3A and 5A for OPA548).
- A control input to set the maximum motor current.
- A thermal shutdown feature with associated status flag.
- A control input to disable the amplifier.

The DDU motors operate at 12V and use up to 106 mA, so OPA547 is an appropriate choice. For higher-power motors, we can upgrade to the pin-compatible OPA548. The ability to specify the motor current limit is also desirable because it allows us to operate the amplifier in different control modes, such as speed or torque control.

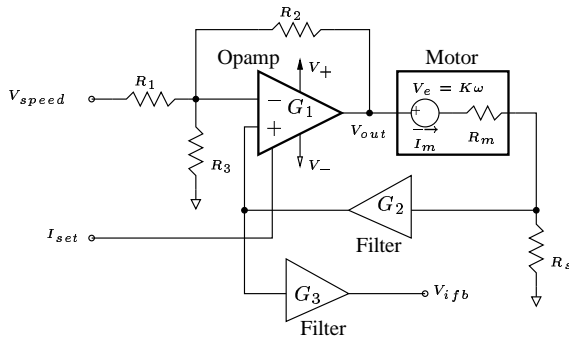


Fig. 4. Power Amplifier Design

A simplified schematic of one channel of our linear amplifier is shown in Figure 4. It has two analog control inputs: one for the motor speed and one for the motor current limit. The motor speed control voltage,  $V_{speed}$ , is connected to the inverting input of the power op amp, which has a gain  $G_1 = -R_2/R_1$ . For the motor current limit, the control voltage is converted (circuit not shown) to a current,  $I_{set}$ , which specifies the motor current limit,  $I_{lim}$ , by the equation  $I_{lim} = KI_{set}$ , where  $K$  is 5000 for OPA547 and 15000 for OPA548. We chose component values to set our current limit range to  $150\text{mA}$ . There is also a digital control input to enable/disable the amplifier and a digital output to reflect the motor's status (not shown in Figure 4). The motor current is sensed by measuring the voltage across resistor  $R_s$ . This voltage is filtered and amplified (gain  $G_2 > 0$ ) before being applied to the positive feedback terminal of the op amp. This circuit implements speed control using armature voltage feedback [3], as described below. The second order active filter is required to reduce the current feedback noise, which would otherwise be amplified by the feedback circuit and cause oscillations in motor speed. We introduce another active filter to adjust the voltage range (gain  $G_3$ ) and provide additional low-pass filtering before performing an A/D conversion of the current feedback signal,  $V_{ibf}$ .

The basic idea for speed control is that when  $V_{in}$  is constant, the back emf of the motor,  $V_e$  (proportional to the motor speed) should remain constant for all motor currents  $I_m$ :

$$\frac{dV_e}{dI_m} = 0 \quad (1)$$

We use a simple motor model that consists of a resistor  $R_m$  and a voltage source  $V_e$  (we neglect the motor inductance, which is small). Applying Kirchoff's laws to the circuit of Figure 4 results in the following equations:

$$V_{out} = I_m R_m + V_e + I_m R_s \quad (2)$$

$$\frac{V_{out} - G_2 I_m R_s}{R_2} + \frac{V_{in} - G_2 I_m R_s}{R_1} + \frac{-G_2 I_m R_s}{R_3} = 0 \quad (3)$$

Note that this is a static analysis that does not consider the effect of the filter cutoff frequency. Differentiating equations 2-3 with respect to  $I_m$  and combining the results yields:

$$\frac{dV_e}{dI_m} = G_2 R_s R_2 \left( \frac{1}{R_1} + \frac{1}{R_3} \right) - R_m + R_s (G_2 - 1) = 0 \quad (4)$$

Resistor values  $R_1$  and  $R_2$  are determined from the desired amplifier gain, which depends on the range of  $V_{in}$  and the rated motor voltage. Solving equation 4 for  $R_3$  yields the following equation:

$$R_3 = \frac{R_2 G_2}{\frac{R_m}{R_s} - \frac{G_2 R_2}{R_1} - (G_2 - 1)} \quad (5)$$

This produces the same equation as [3] if  $G_2$  is set to 1.  $R_m$  is obtained from the motor data sheet or by measuring the motor resistance. There is some flexibility in selecting the value of the sense resistor,  $R_s$ , but it is important to consider the resulting voltage drop. A good rule of thumb is to set  $R_s$  to about 10% of the value of  $R_m$ . Since resistance values are positive, the denominator of equation 5 must be positive which, given  $R_s$ , limits the value of  $G_2$  as follows:

$$G_2 < \frac{\frac{R_m}{R_s} + 1}{\frac{R_2}{R_1} + 1} \quad (6)$$

## B. Interface Section

Our interface section (Fig. 5) is built around an Altera FLEX10KE field programmable gate array (FPGA) and includes a Maxim MAX547 octal, 13-bit digital to analog converter (DAC) and a Maxim MAX125 data acquisition system (DAS), which includes a 14-bit analog to digital converter (ADC). The DAC provides the 4 motor speed and 4 motor current limit control voltages that drive 4 power amplifier channels. The DAS processes the 4 motor current feedbacks from the amplifier sections as well as the 4 analog feedbacks (e.g., from absolute position sensors such as potentiometers). The MAX125 is particularly suited for this application because it is a 2x4-channel, simultaneous-sampling DAS with a fast conversion time ( $3\mu\text{s}/\text{channel}$ ). Thus, we are able to simultaneously sample all 4 motor current feedback signals,

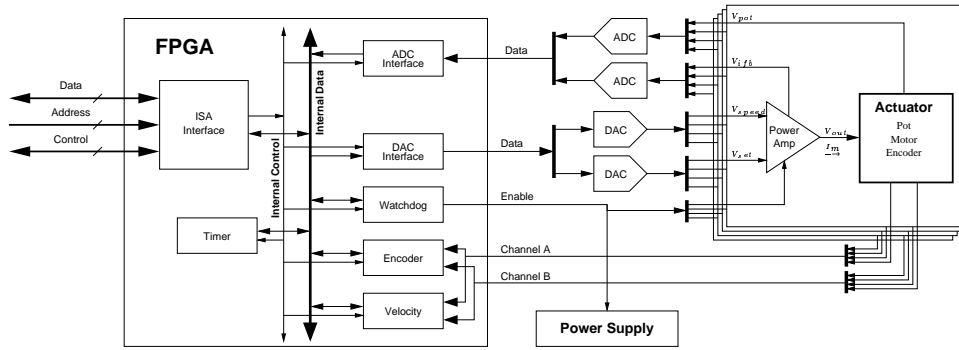


Fig. 5. Block diagram of LoPoMoCo

convert them to digital format, and then repeat the process for the 4 analog sensor feedback signals.

The FPGA is clocked by the 8.33 MHz ISA bus clock and implements the following functions (Fig. 5):

- 1) The interface between the ISA bus and all other components on the board, such as the DAC and DAS (ADC).
- 2) Position measurement from quadrature decoding of 4 incremental encoder feedbacks, maintained in 24-bit counters.
- 3) Precise velocity measurement from the incremental encoder feedbacks, as described below.
- 4) A watchdog timer that can be programmed with a resolution of  $122.93\mu s$  up to a total period of  $125.88ms$ . The watchdog must be refreshed within the programmed period; otherwise, it initiates a shutdown of the power amplifier section via the 4 amplifier enable signals and/or a global shutdown circuit.
- 5) An interval timer that can be programmed with a resolution of  $30.73\mu s$  up to a total period of  $31.48ms$ . It can interrupt the PC using any one of 8 interrupt numbers.

These features all fit within an EPF10K30E FPGA, with many gates available for additional features. Furthermore, the FLEX10KE family includes higher-density pin-compatible chips. The FPGA program can be easily modified because it is stored in a serially reprogrammable configuration EEPROM (Atmel AT17LV512A).

### C. Velocity Measurement

For many control schemes, it is valuable to accurately estimate the joint velocities. In a compact design such as the snake robot, there is little room for tachometers or additional hardware. For systems that use incremental encoders, the two most common hardware techniques to estimate velocity are to measure time between transitions of the pulses (1/DT method) [4] or to measure the number of encoder pulses in a given time interval (DP/DS method) [5]. Each of these methods has limitations, however. The 1/DT method works well at low encoder frequencies, but has poor resolution at high frequencies because the time between pulses is short. Conversely, the DP/DS method works well at high frequencies, but has poor resolution at low frequencies because the number of pulses in a given time period is small. A typical solution is

to implement both methods and incorporate logic to select the most accurate velocity estimate at any given time [6]. In terms of practical implementation, it is also important to consider that the 1/DT method will overflow at very low encoder frequencies and the DP/DS method will overflow at very high frequencies.

We implemented both velocity estimation methods in the FPGA so that the higher-level software can obtain either or both estimates. Equations 7 and 8 give the quantization errors in measurement for the 1/DT and DP/DS methods, respectively, assuming a 1-bit counting error:

$$\% \text{ Error} = \frac{f_{encoder}}{f_{DT} - f_{encoder}} \quad (7)$$

$$\% \text{ Error} = \frac{1}{f_{encoder} \times T_{DS}} \quad (8)$$

Here,  $f_{DT}$  is the frequency of the clock signal used for the 1/DT method and  $T_{DS}$  is the period over which encoder pulses are accumulated for the DP/DS method. To keep quantization error less than 1% and to avoid overflowing an  $n$ -bit counter, the range of encoder frequencies for the 1/DT and DP/DS methods are given by equations 9 and 10, respectively. These ranges are graphically depicted in Figure 6 for different values of  $f_{DT}$  and  $T_{DS}$ , assuming 16-bit counters ( $n = 16$ ).

$$\frac{f_{DT}}{101} \leq f_{encoder} \leq \frac{f_{DT}}{2^n - 1} \quad (9)$$

$$\frac{1}{0.01 \times T_{DS}} \leq f_{encoder} \leq \frac{2^n - 1}{T_{DS}} \quad (10)$$

### D. Design Verification and Calibration

The LoPoMoCo was implemented as a 6-layer board with surface-mount components on both sides. We used 1% tolerance resistors to achieve precise gains, but also verified the accuracy of the motor current feedback using a fixed  $50\Omega$  load resistor instead of a motor (to eliminate the effect of motor noise). We connected a digital multi-meter, set to measure current, in the load resistor and verified that our digitized current feedbacks were within  $0.05mA$  of the multi-meter measurements. We therefore concluded that calibration of motor current feedback was not required.

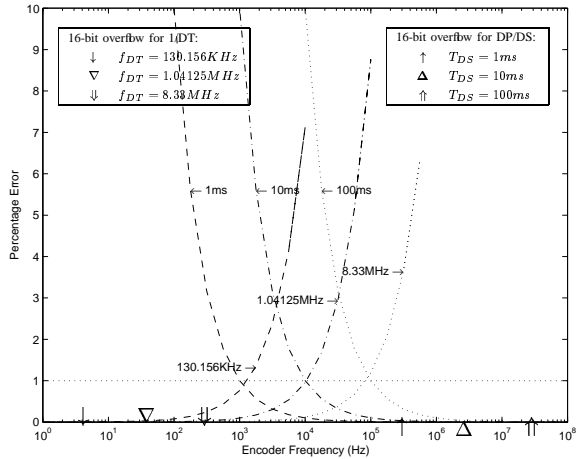


Fig. 6. Velocity Measurement Design

We verified the performance of the OPA547 motor current limit using the  $50\Omega$  resistor. Figure 7 shows the measured current feedback (horizontal axis) that corresponds to the current limit specified via the DAC output (vertical axis). This figure illustrates that the current limit control works well except at small currents (less than about  $20mA$ ), but that calibration is required to handle the offset between the theoretical current limit (calculated from component values and the equation in the OPA547 data sheet) and the measured current limit. We calibrated the motor current limit by fitting piecewise linear segments to the positive and negative motor current data in Figure 7.

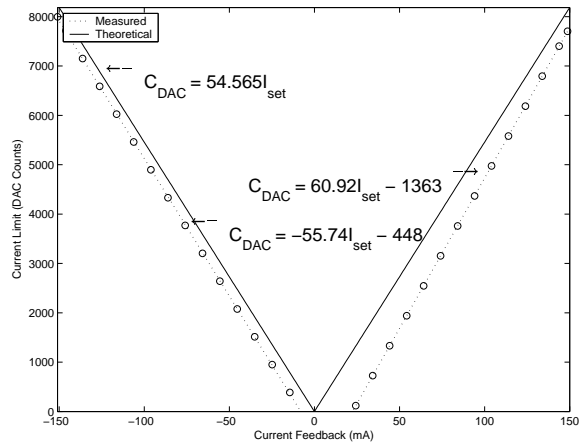


Fig. 7. Current Limit Calibration

### III. EXPERIMENTAL RESULTS AND DISCUSSION

We built an experimental model of a snake actuation unit to test the performance of our system. This section describes the experimental setup and the tests that were performed to evaluate the accuracy of first measuring and then controlling the forces applied by the motor.

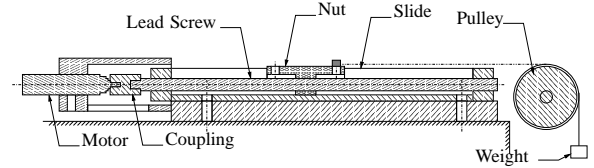


Fig. 8. Experimental Setup

#### A. Experimental Setup

The backbones of the snake-like units are actuated by lead screw that converts motor rotations into translations. Due to the high gearing that such a system provides, it becomes non-back-drivable. In order to validate our controller and to determine its performance, we made a prototype of the actuation unit (Fig. 8) that is similar to the one to be used for the snake. The unit consists of a motor connected to a lead screw and a nut assembly. The lead screw is  $0.25in$  ( $6.25mm$ ) in diameter with a 40 threads per inch ( $0.635mm$  pitch) and a total travel of  $6in$  ( $152.4mm$ ). The motor used is a Maxon RE-10 ( $10mm$  diameter) gearmotor with a 64:1 planetary gear reducer and a 12 count encoder (48 with quadrature decoding). The motor develops about  $64.64mNm$  torque and its no-load speed at nominal  $12V$  is  $178.125rpm$ . Provision is made on the nut to attach different loads to simulate different loading conditions of the backbones.

#### B. Measurement of Force Applied to Motor

Our first test was to verify that we could use the motor current feedback to estimate the motor load and if so, to use the collected data for calibration. Because our system is non-backdriveable, we did not know a priori how well this would work. We performed the test by attaching several known weights to the pulley and moving the motor at different speeds. We recorded the motor current feedback over 1000 samples ( $5s$ ). Figure 9 shows a plot of motor feedback current vs. applied load at several different motor speeds. Note that the relationship is linear and fairly independent of motor speed. The mean variation in the measured motor current across all speeds and loads is about  $1.5mA$ . After performing the calibration of load with respect to motor current, this variation corresponds to an uncertainty of  $0.8N$  in the estimated force. This is less than 3% of the full scale load applied to the system.

#### C. Control of Force Applied by Motor

Our next aim was to control the maximum force applied by the motor. Given that the current feedback provides a good measurement of motor load, one technique is to use the current feedback to close the loop in software. We chose, however, to use the current limiting capability of the linear amplifier, which is provided by a motor current feedback loop internal to the OPA547 op amp. We performed the test by attaching a known weight to the pulley and moving the motor at a given speed. We then slowly reduced the specified current limit ( $I_{set}$  in Figure 4) until the motor stalled. We repeated this test for different weights and at different speeds. Figure 10 plots the motor load vs. the specified current limit (in mA after

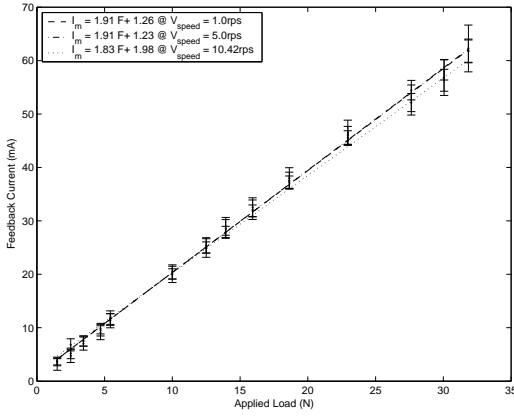


Fig. 9. Current Feedback Performance

calibration) at the stall condition for the different commanded speeds. The error bars represent the difference between the specified current and the measured current, and the mean variation is about  $1.8\text{mA}$  across all speeds and loads. This corresponds to a force of  $0.9\text{N}$ , less than 3% of the full scale force.

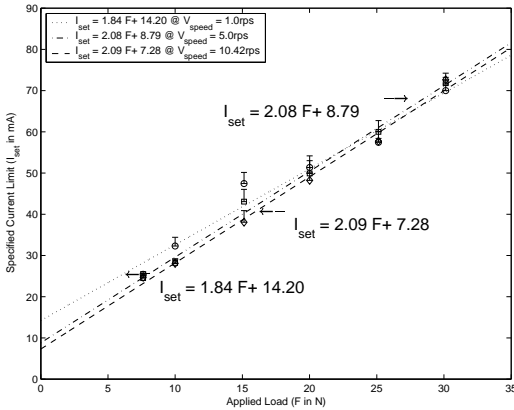


Fig. 10. Current Limit Performance

#### D. Discussion and Future Work

The results of these tests are encouraging because they indicate that the motor current feedback allows us to estimate motor loads (forces) with a resolution of less than  $1\text{N}$ . The power amplifier also enables us to limit the motor load with about the same resolution. We believe that these resolutions are sufficient to control the actuation redundancy in the snake and to prevent buckling of the backbones. In our future work we will evaluate whether we can obtain adequate measurement of forces that are externally applied to the three-axis snake robot. This is extremely challenging because we must first model the backbone forces and then subtract them from our measurements to resolve the external forces.

Because the system is non-backdriveable, the force estimation only works if the controller is actively trying to move the motor. It is interesting to note, however, that the force estimate

is accurate even when the motor is moving slowly (Figure 9) or not at all (stall case, Figure 10). This suggests that it would be possible to obtain force feedback from motor currents in a non-backdriveable system with an appropriate control law. We plan to investigate this in our future work.

#### IV. CONCLUSION

This paper presented the Low Power Motor Controller (LoPoMoCo) that we designed to control the small DC motors of a 34-axis robot system with multiple snake-like arms that is intended for MIS throat surgery. Although this robot system is not yet constructed, we believe that we have two important features that are required for its control: 1) the ability to estimate motor load by measuring motor current and 2) the ability to control the motor load by specifying the motor current limit. The first feature allows us to develop redundancy resolution control laws based on the estimated backbone tensions. The second feature ensures that the snake unit does not fail due to backbone buckling.

An interesting feature of the LoPoMoCo is that the power amplifier contains two control inputs — one to set the desired motor speed and another to set the motor current limit. The software can therefore dynamically switch the amplifier control mode (e.g., fix the current limit while varying the desired speed and vice-versa). The board also provides precise speed measurement via hardware processing of the incremental encoder feedback. The LoPoMoCo design enables us to explore numerous complex control laws in software (on the PC) or even in firmware (on the FPGA). Although initially designed for controlling the snake robot, it is applicable to other low-power robot systems.

#### ACKNOWLEDGMENT

We thank Hamid Wasti of Regan Designs, Inc. (Coeur d'Alene, Idaho) for his extensive help with the design and layout of the controller board. This work was partially funded by the National Science Foundation (NSF) under Engineering Research Center grant #EEC9731478, NSF grant #IIS9801684, and by the Johns Hopkins University internal funds.

#### REFERENCES

- [1] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery - multi-backbone bending snake-like slaves for teleoperated dexterous surgical tool manipulation," in *Proc. IEEE International Conference on Robotics and Automation*, 2004, pp. 351 – 357.
- [2] N. Simaan, R. Taylor, and P. Flint, "High dexterity snake-like robotic slaves for minimally invasive telesurgery of the upper airway," in *Proc. Intl. Conf. MICCAI*, Sept. 2004, accepted for presentation.
- [3] B. Trump, "Dc motor speed controller: Control a dc motor without tachometer feedback," Burr-Brown, Tucson, Arizona, Application Note, Oct. 1999. [Online]. Available: <http://www-s.ti.com/sc/psheets/sboa043/sboa043.pdf>
- [4] E. E. Wallingford and J. D. Wilson, "High resolution shaft speed measurements using a microcomputer," *IEEE Trans. Instrum. Meas.*, vol. IM-26, pp. 113–116, 1977.
- [5] A. Dunworth, "Digital instrumentation for angular velocity and acceleration," *IEEE Trans. Instrum. Meas.*, vol. IM-18, pp. 132–138, 1969.
- [6] P. Bhatti and B. Hannaford, "Single chip velocity measurement system for incremental optical encoders," *IEEE Trans. Contr. Syst. Technol.*, vol. 5, no. 6, pp. 654–661, June 1997.