Unwarping Encoder Ripple in Low Cost Haptic Interfaces[*]

Dale A. Lawrence[†]
Aerospace Engineering
University of Colorado
Boulder, CO 80309–0429

Lucy Y. Pao[‡]
Electrical Engineering
University of Colorado
Boulder, CO 80309–0425

Sutha Aphanuphong[§]
Aerospace Engineering
University of Colorado
Boulder, CO 80309–0429

Abstract

This paper describes a technique for unwarping encoder ripple in a low cost haptic interface prototype. We previously demonstrated how using a low-resolution optical encoder to electrically commutate a stepper motor can produce high torque levels with low ripple. The optical encoder scheme produces a pair of quadrature voltages that are approximately sinusoidal functions of rotor position. However, because of variations in the optical return from the low-precision encoder disk, quadrature phasing and signal amplitude vary over a mechanical turn. In this paper, we discuss an effective self-calibration technique for unwarping the encoder ripple. The technique is used in a one degree-of-freedom haptic interface prototype, and experiment results quantifying the improved performance due to encoder unwarping are presented.

CR Categories: H.5.2 [User Interfaces]: Haptic I/O

Keywords: self-calibration, high-resolution, damping rendering

1 Introduction

Haptic interfaces have been in use for over four decades in a range of force-feedback applications, ranging from teleoperation systems to scientific data visualization to virtual training systems. Haptic interface designs span a wide range for these disparate applications, from power-grip interfaces to more dextrous finger-grip haptic interface designs [1, 2, 4, 5, 11]. The primary technical difficulty in all these designs is providing accurate, high frequency force information to the user, in multiple degrees of freedom, and with large range of motion. This poses unique challenges in mechanism and control design [1, 2, 3, 4, 5, 7, 10, 11, 12, 17]. To achieve hard "crisp" surfaces needed in many haptic interface applications [3, 6], smooth velocity measurements with small computational delay are required [8]. This is often provided via position sensing, resulting in requirements for position sensor resolution that far exceed what users can kinesthetically distinguish. High bandwidth force transmission to the fingers requires a lightweight, stiff transmission, and a low mass actuator with high force capability and low friction and ripple forces.

We recently described a motor and encoder approach [8] for haptic interfaces aimed at achieving these design objectives while keeping hardware costs to a minimum. We have also shown how the motor and encoder can be combined with a new bow spring and tendon design that provides a compact, large range of motion prismatic actuator [9], as shown in Figure 1. The motor and encoder design uses a low-cost stepper motor and a low line-count optical encoder, together with high speed digital signal processing to achieve high torque levels with low ripple, as well as high-resolution position and velocity sensing. While this approach requires more involved electronics and signal processing, these components see wide use in other high-volume products, resulting in advantageous cost reductions compared to motors and sensors. We believe this approach will lead to high performance haptic interfaces which are significantly lower in cost.

High resolution position and velocity sensing is achieved using a low-line count encoder disk by placing optical sensors such that the voltages produced by the sensors are in quadrature. However, the low-cost (low-precision) encoder disk results in variation in optical sensor voltages from one optical period to another, resulting in sensed signals that vary in amplitude and in quadrature phasing, in turn producing errors (warping) in the measured position [15].

In this paper, we discuss a technique for unwarping the encoder ripple. Section 2 briefly reviews the haptic interface setup [8, 9] and the control approach used. Section 3 discusses the approach to data collection and fitting to achieve a self-calibrating unwarping method. Experimental results using this method are presented in Section 4, quantifying the performance improvement of encoder unwarping in a simple damping rendering task.

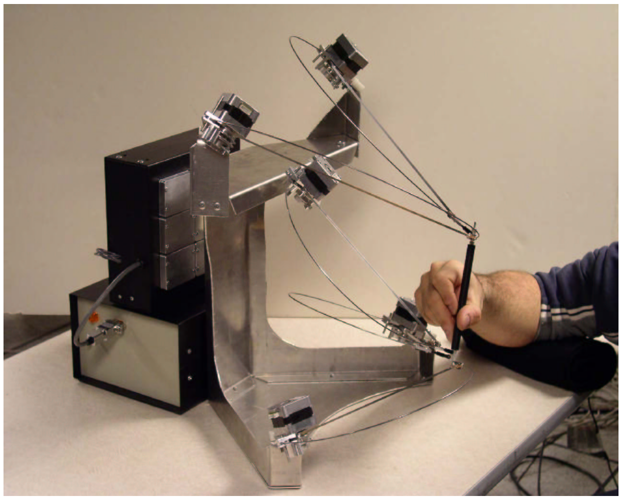


Figure 2: Prototype of a 5 degree-of-freedom haptic interface using bow spring and tendon actuators and integrated stepper motor/optical encoder units.

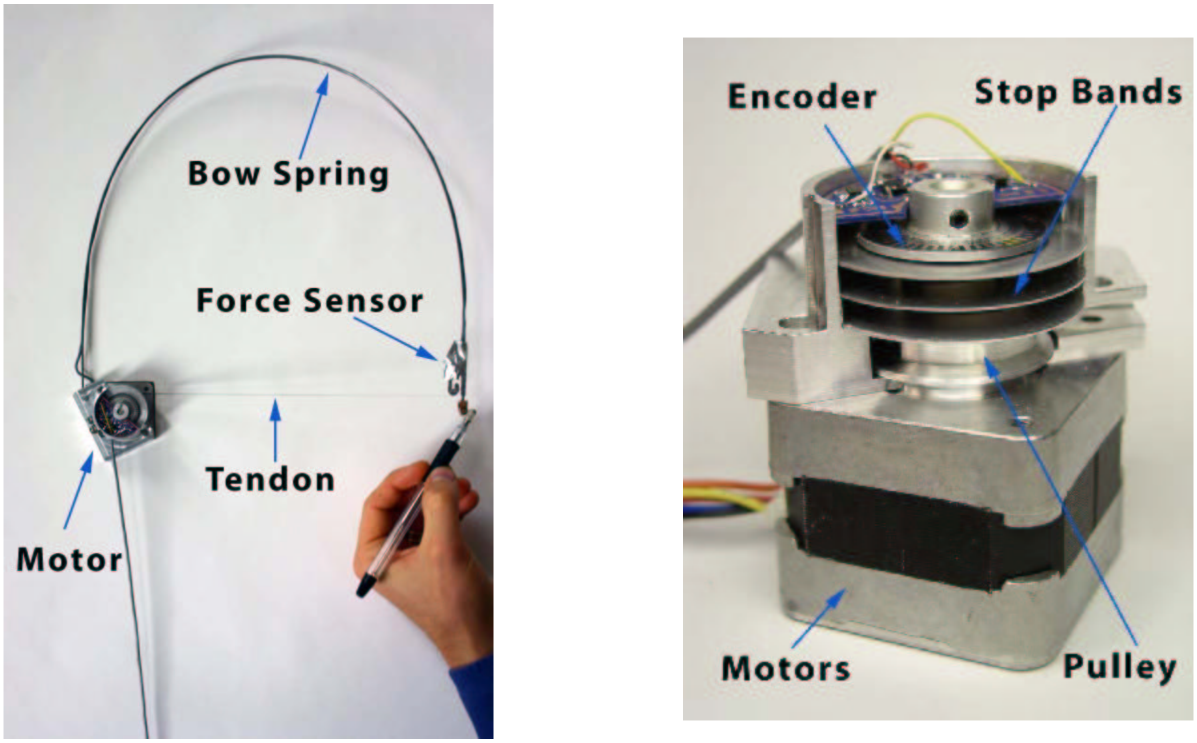
[*]This work was supported in part by the National Science Foundation (HRD-0095944).

[†]dale.lawrence@colorado.edu

[‡]pao@colorado.edu

[§]sutha.aphanuphong@colorado.edu

Symposium on Haptic Interfaces for
Virtual Environment and Teleoperator Systems 2006
March 25 - 26, Alexandria, Virginia, USA
1-4244-0226-3/06/$20.00 ©2006 IEEE



(a)

(b)

Figure 1: (a) Bow spring/tendon actuator shown in mid extension, (b) close-up of stepper motor and low line-count optical encoder.

2 Low Cost Haptic Interface Prototype

Figure 1 shows the prototype one degree-of-freedom low cost haptic interface we recently developed [8, 9], showing the bow spring/tendon actuator, the motor/encoder, as well as a low-cost force sensor located at the distal tip of the actuator near the stylus attachment. The current prototype has a prismatic motion range of 41 cm, an angular motion range at the base attachment of 180 degrees about the pulley axis, and unlimited rotation about the pivot. Forces are transmitted to the stylus by varying tension forces in the thin, stainless steel tendon (of 76 μm by 2.5 mm cross-sectional area) connecting the motor to the stylus. The tendon is lightweight yet very stiff in tension. The bow spring is a length of 1.25 mm diameter music wire that provides a preload of approximately 5 N in the tendon, enabling bi-directional forces up to ± 5 N to be transmitted to the user. The tendon is simply wrapped on a 22.8 mm diameter pulley to convert motor rotation to prismatic motion. As discussed in [8], the stepper motor is capable of relatively large torques due to its many, closely spaced salient poles (50 per mechanical turn), so gear reduction is not required. This "electromechanical gearing" enables a stiff, low friction, zero backlash transmission from electromechanical torque to the forces on the user's fingers.

Several of these one degree of freedom (DOF) actuators can be attached to a haptic interface grip (e.g., a stylus) to produce a parallel, compact multi-DOF interface with a relatively large range of motion. Figure 2 shows a prototype 5-DOF configuration.

A new stepper motor drive was presented in [8] that uses an inexpensive low-resolution optical encoder to electrically commutate the motor, producing high torque levels with low ripple. Compared to a number of motor options [8], the hybrid stepper motor used provides significantly reduced moving mass for the same delivered torque and motor temperature. Figure 1(b) shows the prototype stepper motor and encoder, where the motor body is 42mm × 42mm × 34mm. The encoder disk is 25mm in diameter, and the low cost reflective optical sensor electronics are visible in the figure.

Our particular design uses a 50 pole stepper motor that has a step size of 1.8 deg: 4 steps per pole × 50 poles per mechanical turn. Due to mechanical imperfections in manufacturing the teeth of the rotor and stator poles, the torque τ developed in the motor is only approximately sinusoidal. Thus, the actual torque produced by the motor consists of sinusoidal components, plus a ripple torque τ_R which is a function of both rotor-stator angle and coil current.

Our approach to commanding the motor is to use sine wave commutation. However, successful sine wave commutation requires that the angular position of the rotor be known to a small fraction of an electrical turn so that the resulting motor current can be a smooth sinusoid. Further, we also want to cancel the ripple torques τ_R, which are the differences between the actual torques and the assumed sinusoidal torques. This requires that additional feedback torque be provided that is a function of rotor position [9]. In order to significantly reduce the residual torque ripple, it is necessary to measure these ripple disturbances sufficiently accurately. Data is taken in both directions because Coulomb friction in the motor results in a torque offset in the data that differs in the distal and proximal motion directions. We have shown that a combination of control system motion and force feedback compensation (as shown in Figure 3) provides significant reduction of undesirable forces due to bow spring bias, Coulomb friction, and stepper motor torque ripple [9].

While optical encoders with sufficient lines per turn are available to provide the position and velocity sensing resolution required in haptic interface applications, these higher line count encoders are large and expensive. In our recent design [8], the optical encoder scheme developed produces an analog output instead of a digital quadrature output. By choosing the optical and electrical parameters appropriately, it is possible to produce a pair of quadrature voltages that are approximately sinusoidal functions of rotor position, periodic on the angular spacing of the optical stripes on the encoder

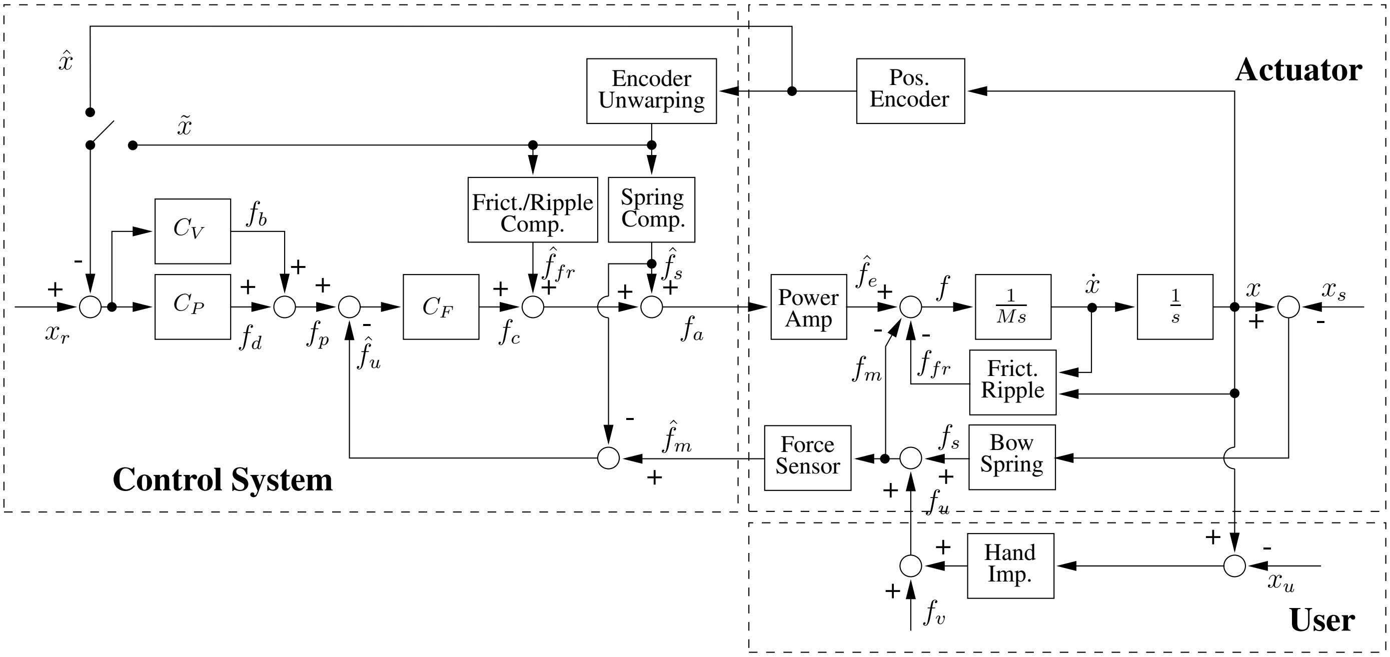


Figure 3: Overall block diagram of the tendon bow actuator control system, indicating feedback of both motions and forces to compensate for bow spring bias, step motor friction, and step motor ripple torques. This paper compares the effects of using warped position data $\hat{x}$ versus unwarped data $\tilde{x}$.

disk. Quadrature phasing of the two optical sensors can be provided by locating the two optical sensors at an appropriate spacing around the circumference of the encoder disk, requiring only one track of lines on the disk. The angular position of the rotor within one optical period can then be computed by taking the arctangent of the two quadrature signals. Total angle can be computed by accumulating the changes of this optical angle, as in a conventional incremental encoder.

While many different optical periods could be used, it is most convenient for commutating the motor to match the optical period to the electrical period, so the encoder signals have the same period as the motor quadrature torques. This way, the position within an electrical turn can be known absolutely, with no issues of initialization or slip that occur with incremental measurements. In our design, since our stepper motor has 50 electrical turns, this requires only a 50 line optical encoder disk, which can be small and inexpensive to produce.

Variations in optical reflectance and eccentricity in mounting the optical encoder disk occur, primarily because precision manufacturing and alignment were avoided in this design to lower costs. This results in a non-circular and variable (warped) relationship between the quadrature signals, which leads to smooth, but inaccurate position measurement via the arc tangent function [13, 15]. The next section discusses a method for unwarping these measurements.

3 Unwarping Encoder Ripple

If an accurate measurement of position x were available, it would be straightforward to unwarp the encoder-derived positions $\hat{x}$, since deviations

$$\Delta x = \hat{x} - x = f(x) \tag{1}$$

from the true position could be modeled, e.g., with the approximate function $\tilde{f}(x)$ and subsequently subtracted from the encoder measurements, i.e., $x \approx \tilde{x} = \hat{x} - \tilde{f}(x)$. However, this would require a high-accuracy measurement instrument to be connected to the motor to provide x for unwarping calibration. Instead, we investigate a self-calibration method which requires no external equipment. In the approach of [15], multiple read stations are averaged to obtain estimates of the true position. The approach of [13] assumes a fixed mechanical load on the motor, which is modeled by a Kalman filter to predict actual position x. In haptic interface applications, the mechanical load is varied by the user's grip impedance and exerted voluntary forces, which are difficult to model. Here we describe a method that is independent of mechanical load. It utilizes a spline fit of encoder data taken in a single off-line, high-speed motion experiment. A similar approach [16] utilizes a neural network to produce parameterized data fits.

Suppose the motor could be made to move at an approximately constant velocity over an encoder ripple period. Then an accurate estimate of the motor position versus time could be determined by fitting a constant velocity $\bar{x}$ function to the measured $\hat{x}$ data. An obvious approach to prescribing velocity is to use a velocity control loop to force measured velocity to follow a desired reference. Unfortunately, the only measurements available here are derived from the encoder position, which has embedded warping errors. This would result in ripple in the closed-loop motor velocity $\dot{x}$, even though the measured velocity would appear to be equal to the constant reference velocity. No estimate of the true velocity would be available. But suppose a mechanical means could be devised to regulate the motor velocity, then deviations caused by encoder warping could be estimated. In fact, a constant velocity (as assumed in [14]) is not really needed, only a velocity that is approximately constant over the period of the encoder warping. Due to the large number (fifty) of encoder periods in one mechanical turn, all that is needed is a motor velocity that varies smoothly enough from turn to turn. Since the bow spring in the actuator provides a self-generated torque on the motor, and this torque is slowly and smoothly varying as the bow spring extends, it was thought that this could produce a motor speed that was sufficiently smooth over a mechanical turn.

The idea is to use motor torque to wind up the tendon from its distal stop by several turns, compressing the bow spring. Then release the spring to accelerate the motor through several turns. Taking data over a full turn during this acceleration (and before the distal stop is encountered) would provide a position trajectory that is at an approximately constant velocity over an encoder warping period. By fitting a low-order polynomial to this data over a full mechanical turn, the polynomial approximation provides an estimate $\bar{x}$ of the actual motor position x, forming the reference from which to calculate encoder warping deviations $f(\bar{x})$.

There are two problems with this idea. The first problem is that the stepper motor has significant cogging torques, which are *not* smooth over an encoder period. In fact, the ripple torques in the motor are quasi-periodic on this same period by design, so that the encoder provides an absolute measure of motor position within this period for reliable winding commutation (see [9] for more details). These torque ripples can cause velocity variations within an encoder period which will corrupt the estimate of true velocity. That is, variations in measured velocity can no longer be attributed solely to encoder warping, and the resulting deviations from the polynomial fit will overcompensate for encoder warping.

Consider the effect of unreeling the motor with the bow at various speeds. The motor ripple torques are fixed in amplitude (since they are due to variation in motor reluctance with angle), and occur at higher frequencies as the motor velocity increases. These higher frequency disturbance torques cause less change in motor velocity due to the smoothing effects of the motor moment of inertia: the torque to velocity frequency response for mass decreases at 20 dB/decade as frequency is increased. This suggests that if higher velocity can be obtained during the position data recording interval, then the effects of motor torque ripple can be reduced. This effect can be studied by winding up the tendon to varying distances, providing larger initial bow spring torque and longer acceleration times before the distal stop.

Figure 4 shows data from an experiment taking encoder position $\hat{x}$ data over the same range of positions, but at various speeds during passive bow extension from a wound-up position. The data shown is the residual deviation $\Delta x = \hat{x} - \bar{x}$ from 6th order polynomial fits $\bar{x}$ over one mechanical turn. The top plot corresponds to slow average velocity (about 0.5 m/s), with increasing average speeds in the lower plots (0.65 m/s and 1.0 m/s, respectively). Note that the encoder-period ripple in these plots is not varying with average motor speed, but there are longer-period variations that are reduced in amplitude as speed in increased. These latter components of the observed ripple are due to torque ripple in the motor. Only the geometric encoder warping is independent of velocity, hence independent of the frequency-dependent filtering of the motor dynamics. Clearly, the high velocity experiment is preferred for estimating $\bar{x}$.

The second problem is that the estimates $\bar{x}$ of actual position are only available in post-processing of data over a whole turn, and are not available during normal operation, hence the function approximation $\tilde{f}(\bar{x})$ cannot be obtained for real-time unwarping. This can be addressed by writing the measured position $\hat{x}$ in (1) as an explcit function of the estimated motion $\bar{x}$

$$\hat{x} = x + f(x) = g(x) \quad \longrightarrow \quad \hat{x} \approx g(\bar{x}) \tag{2}$$

Now an estimate $\bar{x}$ for the true position can be calculated from warped measurements $\hat{x}$ if the function g^{-1} can be modeled off line, and interpolated during real-time operation. We actually use a slightly modified form of this, where the deviations Δx are expressed as an explicit function of measured position $\hat{x}$ for spline fitting

$$\Delta x = \hat{x} - \bar{x} = \hat{x} - g^{-1}(\hat{x}) = h(\hat{x}) \tag{3}$$

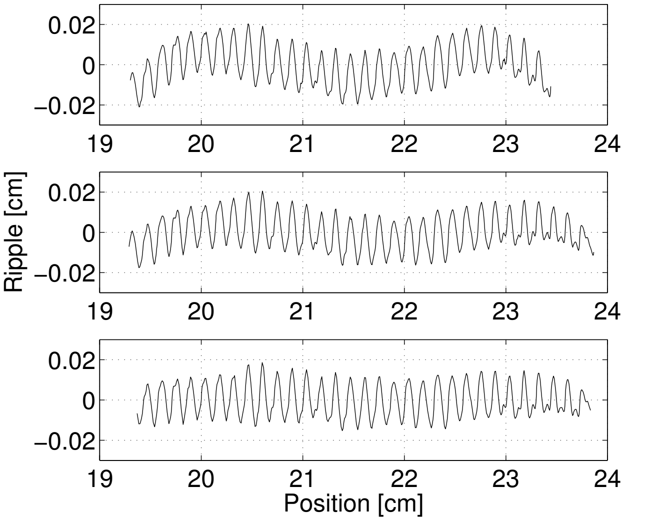


Figure 4: Encoder deviation Δx versus fit position $\bar{x}$, for average velocities of 0.5 m/s (top), 0.65 m/s (center), and 1.0 m/s (bottom).

and the unwarped estimate $\tilde{x}$ of position is recovered by differencing the interpolation $\tilde{h}$ of h with the measured position:

$$\tilde{x} = \hat{x} - \Delta x = \hat{x} - \tilde{h}(\hat{x}) \tag{4}$$

This approach begins with $\hat{x}(t_i)$ data from a high-speed bow extension experiment, fitting this data with a 6th-order polynomial approximant $\bar{x}(t)$ over one mechanical turn. Interpolating this function at times t_i provides a $\bar{x}(t_i)$ corresponding to every value of $\hat{x}(t_i)$ measured. The function $\Delta x(t_i) = \hat{x}(t_i) - \bar{x}(t_i) = h(\hat{x}(t_i))$ is then fit with a cubic spline, using 500 control points t_i per mechanical turn (10 control points per encoder period), with 10 points of additional data before and after the one-turn interval to reduce fitting errors on the one-turn boundary. The spline fit approximates the $h(\hat{x})$ data by a smooth function

$$\tilde{h}(\hat{x}) = A\Delta x_j + B\Delta x_{j+1} + C\Delta x''_j + D\Delta x''_{j+1} \tag{5}$$

where the $''$ indicates second-derivative estimates from sampled data, and the interpolation coefficients are given by [18]:

$$A \equiv \frac{\hat{x}_{j+1} - \hat{x}}{\hat{x}_{j+1} - \hat{x}_j}\ , \quad B \equiv 1 - A = \frac{\hat{x} - \hat{x}_j}{\hat{x}_{j+1} - \hat{x}_j} \tag{6}$$

$$C \equiv \frac{1}{6}(A^3 - A)(\hat{x}_{j+1} - \hat{x}_j)^2\ , \quad D \equiv \frac{1}{6}(B^3 - B)(\hat{x}_{j+1} - \hat{x}_j)^2 \tag{7}$$

The real-time unwarping code implements this by determining which interval $[\hat{x}_j, \hat{x}_{j+1}]$ the measured data $\hat{x}$ resides, then computes the interpolation coefficients A, B, C, D, and then looks up the values $\Delta x''_j, \Delta x''_{j+1}$ in a previously stored data table. The interpolated value $\tilde{x} = \hat{x} - \tilde{h}(\hat{x})$ from (5) is then used as the unwarped angular position estimate.

4 Experimental Results

The above method for encoder unwarping is applied in the block diagram of Figure 3 in the feedback portion of the position control loop. The motor ripple torque compensation can be based on either the warped or unwarped encoder measurements, since all that is necessary there is that the proper torques as a function of measured angle are applied. However for the position loop, the warped position data may cause perceptible vibrations. In the case of a damping control law $C_V(s) = Bs$, smooth mechanical motion produced by voluntary user motions produce measured velocities that contain velocity variations. In turn, these produce force commands f_b which contain ripple. If these are at an amplitude and frequency that can be perceived at the stylus grip, a rough (rather than smooth) damper results. In the case of virtual surface rendering via the control block C_P, sliding along a smooth virtual surface, say in three degrees of freedom, should produce smooth forces normal to the virtual surface. However, if the encoder signals are warped, smooth mechanical motion produces measured motion that appears to undulate. When compared to the smooth virtual surface reference location, periodic surface penetration is calculated by the control loop, resulting in ripple in the applied reaction force f_d by the virtual surface impedance. This could result in a rough surface perceived by the user.

Since a single degree of freedom is used in this study (only one tendon/bow actuator), the virtual surface effects of encoder ripple cannot be evaluated. Instead, we examine the case of virtual damping rendering. We wish to measure the effects of using the raw encoder data $\hat{x}$ versus the unwarped data $\tilde{x}$ on the forces f_b applied at the actuator (C_F is unity in Figure , on the forces $\hat{f}_m$ measured by the force sensor near the stylus grip, and on the forces f_u perceived by the user. This provides a first step in quantifying the benefits of unwarping on the quality of haptic rendering.

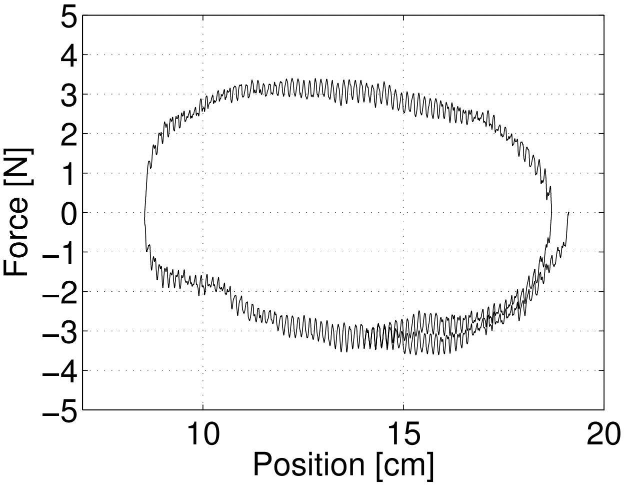


Figure 5: Force commands f_b resulting from using warped encoder measurements $\hat{x}$ in rendering a 13 Ns/m damping field.

A simple experiment was conducted where the user moves the stylus back and forth in an approximately sinusoidal displacement covering a distance of about 10 cm in a period of approximately 1 sec, producing a maximum velocity of about 30 cm/sec. A virtual damper with damping constant $B = 13$ Ns/m was rendered by the position control loop, using finite differences of the measured position (either $\hat{x}$ or $\tilde{x}$ in Figure 3) to obtain velocity. A low pass filter within C_V was applied to these velocity estimates with a corner frequency of 30 Hz to reduce noise accentuation. This corner frequency is well above the motion frequency, but lower than the main encoder ripple frequency for this motion (210 Hz), hence this filtering attenuates the encoder-period ripple in the force command. However, this filtering is the same for both warped and un-warped cases.

Figure 5 shows the resulting force command f_b from the damping rendering block in Figure 3 when the raw (warped) position measurements $\hat{x}$ are used. Peak-to-peak ripple of approximately 0.5 N can be seen in this motion. If the unwarped positions $\tilde{x}$ are used instead, the force commands shown in Figure 6 are obtained. The ripple at the encoder warping period is indistinguishable from random noise, showing that the unwarping method discussed above is quite effective.

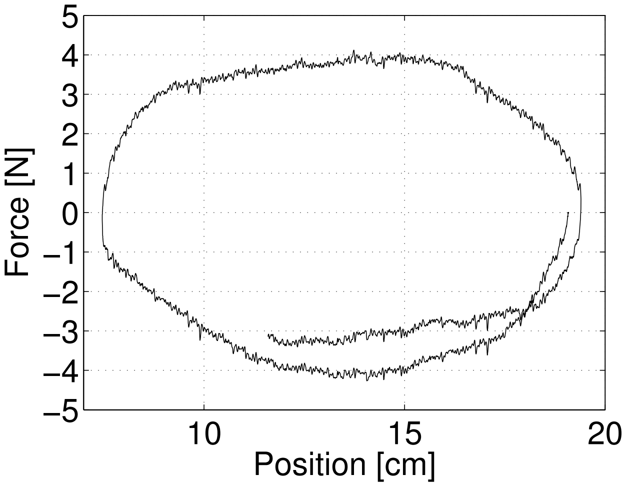


Figure 6: Force commands f_b resulting from using unwarped encoder measurements $\tilde{x}$ in rendering a 13 Ns/m damping field.

Next, we address the question of the difference in forces $\hat{f}_m$ produced at the distal tip of the tendon/bow actuator, near the stylus attachment. Figure 7 shows the force measured by the force sensor, which connects the tendon to the bow spring and stylus, for the case where the warped encoder measurements $\hat{x}$ are used. The flat portion in the plot is due to slack in the tendon during part of the motion. Figure 8 shows the corresponding tip force when the unwarping function is used. The difference in tip force between the two cases is less pronounced than for the commanded forces (Figures 5 and 6). Still, force ripple at most positions is noticeably smaller in

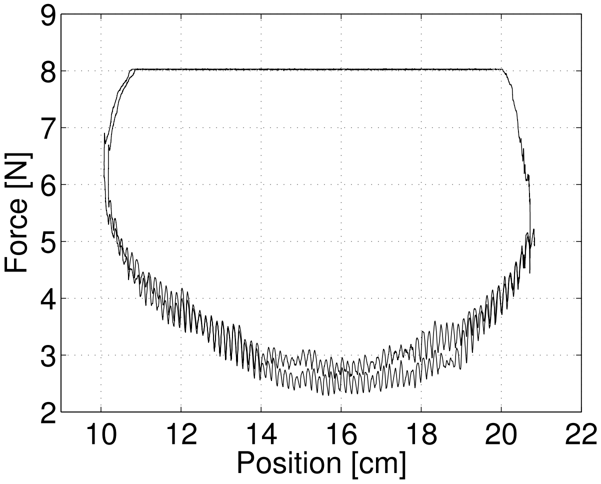


Figure 7: Measured tip force $\hat{f}_m$ when warped encoder measurements $\hat{x}$ are used to render a 13 Ns/m damping field.

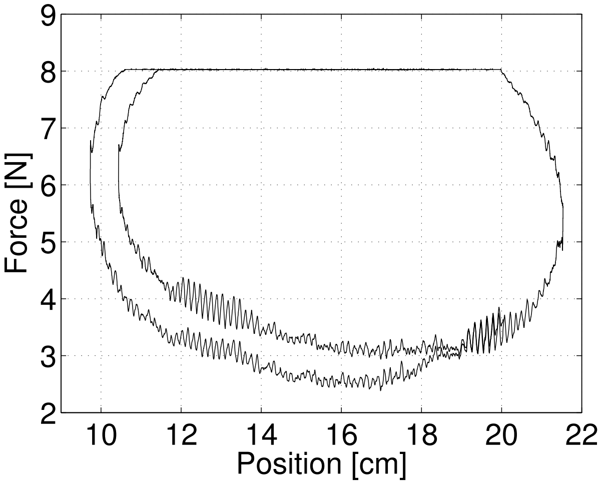


Figure 8: Measured tip force $\hat{f}_m$ when unwarped encoder measurements $\tilde{x}$ are used to render a 13 Ns/m damping field.

this plot than in the warped case. Evidently, there are imperfectly canceled torque ripples in the actuator, which produce measurable tip forces despite the absence of significant force commands in the unwarped case. In the warped case, tip forces are increased, but relative to the baseline torque ripple, by a smaller factor than in the commands. More significant, however, is that no difference between these two cases could be felt by the authors. Apparently, the difference in force produced at the motor using warped and unwarped position measurements is below our discrimination threshold, at least when the masking effects of residual motor torque ripple are present. If motor torque ripple could be reduced further, the advantage of encoder unwarping may become more apparent. Further, other rendering modes (e.g., virtual surfaces) may be more sensitive to encoder warping.

5 Conclusions and Future Work

A low cost encoder that provides high resolution, but low accuracy, was studied for use in haptic interfaces. This produces warped position measurements which can cause unwanted vibration and roughness in rendered haptic effects. A method to calibrate-out (unwarp) these measurements was presented which does not require a high-accuracy reference sensor. This approach allows the encoder to be self-calibrated using the actuator motor and integral bow spring. Although the unwarping was shown to be effective in reducing ripple forces in a damping rendering example, it was also shown that these ripple forces are at a frequency and amplitude, in this case, such that they are filtered by the tendon/bow to a level below the user's discrimination threshold. That is, the unwarping did not result in perceptible improvement in forces at the user's stylus grip in this rendering mode. Further testing is needed to determine if unwarping is necessary in other modes. If unwarping is not needed, then the low cost encoder is in itself sufficient. If unwarping is necessary in some rendering scenarios, the unwarping method presented here provides a simple, practical, and effective technique for haptic interfaces.

References

[1] G. Burdea and J. Zhuang. "Dextrous Telerobotics with Force Feedback–an Overview," *Robotica*, Vol. 9, 1991.

[2] P. Buttolo and B. Hannaford. "Pen-Based Force Display for Precision Manipulation in Virtual Environments," *Proc. IEEE Virtual Reality Symp.*, pp. 217–224, 1995.

[3] J. Edward Colgate and J. Michael Brown. "Factors Affecting the Z-Width of a Haptic Display," *Proc. IEEE Conf. Robotics Automat.*, pp. 3205–3210, San Diego, CA, May 1994.

[4] V. Hayward. "Toward a Seven Axis Haptic Device," *Proc. Conf. Intelligent Robots & Systems*, 1995.

[5] H. Iwata. "Artificial Reality with Force Feedback: Development of Desktop Virtual Space with Compact Master Manipulator," *Computer Graphics*, 24(4), 1990.

[6] D. A. Lawrence, L. Y. Pao, A. M. Dougherty, M. A. Salada, and Y. Pavlou. "Rate-Hardness: A New Performance Metric for Haptic Interfaces," *IEEE Trans. Robotics and Automation*, 16(4): 357–371, Aug. 2000.

[7] C. D. Lee, D. A. Lawrence, and L. Y. Pao. "A High-Bandwidth Force-Controlled Haptic Interface," *Proc. ASME Dyn. Sys. Control Div.*, DSC–Vol. 69–2, Orlando, FL, pp. 1299–1308, Nov. 2000.

[8] D. A. Lawrence, L. Y. Pao, A. C. White, and W. Xu. "Low Cost Actuator and Sensor for High-Fidelity Haptic Interfaces," *Proc. 12th Int. Symp. Haptic Interfaces*, Chicago, IL, pp. 74–81, Mar. 2004.

[9] D. A. Lawrence, L. Y. Pao, S. Aphanuphong. "Bow Spring/Tendon Actuation for Low Cost Haptic Interfaces," *Proc. World Haptics Conf.*, Pisa, Italy, Mar. 2005.

[10] G. Luecke and Y.-H. Chai. "An Exoskeleton Manipulator for Application of Electro-Magnetic Virtual Forces", *Proc. ASME Dyn. Sys. Control Div.*, Atlanta, GA, Nov. 1996.

[11] T. H. Massie and J. K. Salisbury. "The Phantom Haptic Interface: A Device for Probing Virtual Objects," *Proc. ASME Dyn. Sys. Control*, Vol. 1, pp. 295–301, 1994.

[12] S. E. Salcudean, R. Six, R. Barman, C. Jennings, S. Kingdon, D. Murray, and V. Tucakov. "Design Tradeoffs in the Development of a 6-DOF Desktop Haptic Interface," *Proc. ASME Dyn. Sys. Control Div.*, Anaheim, CA, Nov. 1998.

[13] S. Venema and B. Hannaford. "Kalman Filter Based Calibration of Precision Motion Control," *Proc. Intelligent Robots and Systems*, Vol. 2, pp. 224–229, Aug. 1995.

[14] H. Hagiwara, Y. Suzuki, and M. Murase. "A Method of Improving the Resolution and Accuracy of Rotary Encoders Using a Code Compensation Technique," *IEEE Trans. Instr. and Meas.*, Vol. 41, No. 1, pp. 98–101, Feb., 1992.

[15] A. M. Madni, M. Jumper, and T. Malcolm, "An Absolute High Performance, Self Calibrating Optical Rotary Positioning System", *Proc IEEE Aerospace Conf.*, Vol 5, March, 2001, pp. 2363 - 2373.

[16] K. K. Tan and K. Z. Tang, "Adaptive Online Correction and Interpolation of Quadrature Encoder Signals Using Radial Basis Functions", *IEEE Trans. Control System Tech.*, Vol. 13, No. 3, May, 2005, pp. 370–377.

[17] R. L. Williams, II, "Cable Suspended Haptic Interface," *Proc. ASME Dyn. Sys. Control*, DSC-Vol. 64, pp. 207–212, 1998.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edition, Cambridge University Press, 2002.