Shaft Compliance as a Soft Sensor to Eliminate Stiction in Hybrid Haptic Devices

Jacob Horne[1], Milan Djordjevic[1], Samuel Lovett[1], Antoine Weill-Duflos[2], Colin Gallacher[2], and Carlos Rossa[1]
[1]*Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada*
[2]*Haply Robotics, Montréal, QC, Canada*

***Abstract*—Haptic devices that rely on electric motors to generate force feedback suffer from an inherent trade-off between performance and stability. A newer class of devices use hybrid actuation with brakes and motors to improve force rendering. However, while brakes are intrinsically stable, they can only oppose an input torque. When the brake and motor are activated at the same time, the brake blocks both the motor's and the user's input torque and can create stiction - an unwanted resistance to motion that negatively affects simulation realism. In this paper, we propose the use of shaft compliance as a soft torque sensor in a hybrid actuator integrating a DC motor and a brake. The angular displacement of the compliant element is used in a controller to infer the user's input torque from readings of a single position encoder. The controller shares a desired torque between each actuator while preventing stiction. Experimental evaluation of the proposed sensor confirms the sensor's ability to detect torque and the controller's ability to simulate virtual walls, despite sensor accuracy limitations and nonlinearities. The proposed sensor concept offers a reliable and low-cost alternative for torque sensors in portable haptic devices.**

I. Introduction

As the prevalence of virtual environments (VE) increases in society, there comes a need to improve the connection between humans and virtual objects. These VEs encompass a wide range of industries, including virtual reality games, remote robotic control, and medical training simulators [1]. While these applications vary widely, they all directly benefit from haptic feedback devices. Haptic feedback devices bridge the gap between virtual and physical environments by applying forces to an operator to emulate the sensation of interacting with virtual objects. These forces can be as simple as vibrations, or as complex as movement in 6 degrees-of-freedom. An ideal haptic device is transparent, i.e., it can simulate an infinite range of impedance, allowing the user to move the device freely without experiencing inertia or friction, and to interact with VEs simulating rigid inert objects [2].

The majority of haptic feedback devices use electric motors to generate force feedback. Traditionally, the VE calculates the required torque/force based on the motor's position and speed. This method works well for simulating soft objects but motors struggle with stiff environments, where the desired force changes drastically with a minimal positional change, such as a virtual wall approximated as a stiff spring. Due to the finite simulation and position measurement sample rate, a motor produces more energy when the virtual spring is decompressed than it dissipates when the spring is compressed [3]. If gained energy is greater than the energy lost through friction, the haptic device becomes unstable, generating unwanted vibrations. The maximum stiffness the motor can render before instability can be increased by adding viscous damping to the simulation model, however, this negatively impacts transparency [4].

A solution to the above is to replace motors with (or to add) a passive actuator, such as a brake. Although passive actuators are intrinsically stable, enabling them to emulate higher simulation stiffness than motors, they have two major limitations in haptic devices. Firstly, they resist movement in all directions. If the brake activates to prevent a user from moving into a wall, it will also stop the user from moving away from the wall, a phenomenon often referred to as "striction" [5]. Secondly, they have slow response times, often in the range of 40 to 80 ms, and are subjected to magnetic hysteresis, adding additional control challenges [6].

Hybrid haptic feedback devices combining motors and brakes seek to solve the issues inherent to each actuation modality alone. The simplest approach is to connect a brake and a motor in parallel, as in Fig. 1 (a) [7], [8], [9]. In this configuration the total torque is the sum of the brake and motor's torque, but the brake can block both the motor and the user. In Fig. 1 (b), a force sensor is added to measure the user input torque and deactivate the brake when the user applies a force/torque in the direction of the VE's desired torque [10], [11]. The series elastic actuators (SEA) in Fig. 1 (c) place a compliant element between the user and either a brake or a motor [12], [13]. Since the compliant element naturally pushes the user away from a virtual wall, the actuator eliminates striction. However, with a brake, the maximum stiffness the actuator can display is limited to that of the compliant element. The arrangement in Fig. 1 (d) uses a non-backdrivable motor connected to a spring through a differential clutch that regulates the output torque via the brake torque [14]. While it can provide high torque, the actuator is bulky and not suited for portable devices.

Conti and Khatib [15] proposed the solution in Fig. 1 (e) where the brake and motor are arranged in parallel with a spring placed between the user and brake. The spring stores energy and reduces power consumption. The torque in the spring is calculated from two rotary encoders and is limited by the maximum brake torque. The motor handles the differences between the VE torque and spring torque. A soft spring limits the upper impedance bound of the actuator. An alternative solution to eliminate stiction is to use two brakes, each with

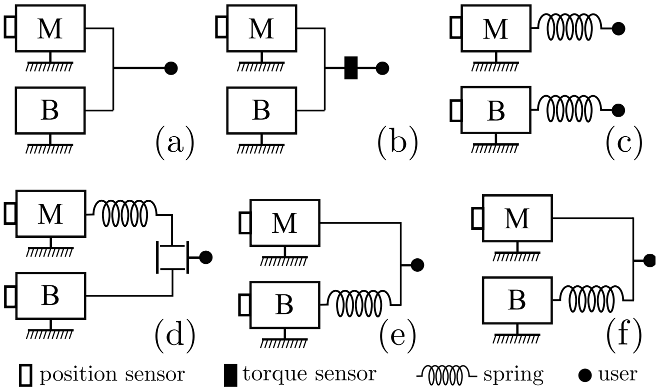

Fig. 1: Hybrid actuators. (a) has a brake (B) and motor (M) in parallel, (b) adds a torque sensor, (c) shows a series elastic actuator with a motor (top) and brake (bottom), in (d) the output is connected to a differential clutch, (e) has a soft spring between the brake and motor, and (f) shows the proposed design with shaft compliance and only one position sensor.

a one-way bearing [7]. Since each brake can only resist a torque in one direction, the user is free to move away from a virtual wall when the brake is active. This solution requires two brakes, increasing complexity.

In such hybrid actuators, a control law must partition a desired VE torque into a brake and motor torque. When the user applies a torque opposing the direction of the VE torque, both the motor and brake can be used to dissipate energy. However, when the user applies a torque in the same direction as that of the VE torque, the brake must turn off as it can only oppose the user's torque. This results in a torque discontinuity referred to here as a ”torque gap”. A solution proposed by Antolini et al. [16] uses a transfer function to first activate the brake and then gradually increase the motor torque while reducing the brake's until the desired torque is reached and only the motor remains active. The controller in [7] estimates the stiffness of the VE and limits the motor's so as not to exceed a predefined upper bound and uses the brake to compensate for the difference. However, the controller cannot eliminate stiction. A control algorithm that addresses stiction without a torque sensor, while concurrently limiting the motor-simulated stiffness to avoid instability has not been reported.

In this paper we propose the use of a compliant element between the brake and motor as a soft sensor to infer the user's torque, as in Fig. 1(f). We describe how the sensor integrated in the actuator estimates the input torque, allowing the configuration shown in Fig. 1(e) to operate without the second encoder. The use of a stiff compliant element instead of a soft spring adds minimal compliance and does not compromise transparency. The sensor works concurrently with a novel decision making algorithm that uses the estimated torque to partition a desired torque between the brake and motor while avoiding stiction. To be best of our knowledge, shaft compliance as a soft sensor has not been proposed before.

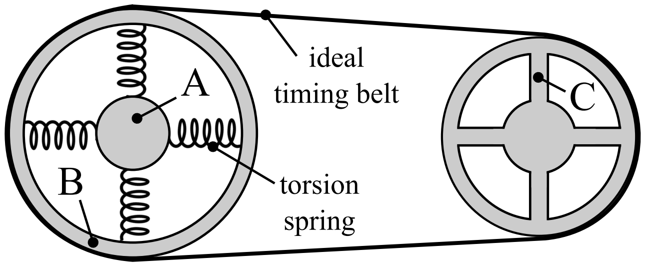

Fig. 2: Proposed actuator concept. The user applies a torque T_u to A, rigidly connected to a motor applying torque T_m. A rotary encoder measures the position θ of A. The user/motor shaft A is linked to pulley B through a built-in, stiff torsion spring of stiffness K. The brake is attached to shaft/pulley C and applies torque T_b. A belt connects C to B.

The rest of the paper is structured as follows: Sec. II explains the proposed concept along with its control algorithm. Sec. III describes the prototype and the experimental setup built to validate the system. The experimental results in Sec. IV from 3 sets of tests confirm that the actuator can effectively simulate virtual walls of varying stiffness while avoiding stiction. The results are followed by a discussion and concluding remarks.

II. Proposed Sensing Approach

The proposed actuator architecture follows the concept depicted in Fig. 1(f), and is implemented using the arrangement detailed in Fig. 2. The user applies a torque T_{in} to shaft A, which is rigidly connected to a motor whose torque at A is T_m. The position θ of shaft A (i.e., the motor and output shaft position) is the only measured position. Shaft A is attached to pulley B through a torsion spring of stiffness K. The brake applies a torque T_b to pulley C, which is linked to B through a timing belt. The following assumptions will made: 1) B and C have the same diameter, 2) the spring is linear, and 3) the timing belt has no friction or elasticity.

In the next subsections we show how the compliant element can be used as soft sensor to infer the user's torque and how this information can be leveraged in a controller.

A. Shaft Compliance as a Soft Sensor

If a torque is applied to shaft A while shaft C is locked without moving, the torque T_k in the shaft's spring is equivalent to the applied torque T_{in}, that is:

$$T_{in} = T_k = \theta K \tag{1}$$

where θ is the angular displacement of A. Now suppose that shaft C is only held in place by friction. If the applied torque at A is large enough, shafts C and B begin to rotate and the torque in the spring cannot exceed the brake torque at C. Additionally, when both A and B are moving, the torque in the spring is equivalent to the frictional torque at C. When C is actuated with a variable source of friction (e.g., a magnetic particle brake), it is possible to estimate the torque in the spring. The torque in the spring $T_k(t)$ at time index t can be calculated as the sum of discrete changes torque $\Delta T_k(t)$:

$$\Delta T_k(t) = K\left[\theta(t) - \theta(t-1)\right]. \tag{2}$$

The accuracy of the estimation above is limited by the resolution of the position sensor and the sampling rate. Quantization error increases with K potentially compromising the control logic. A high-resolution encoder and high-frequency sampling can help mitigate these issues. The maximum torque T_{km} that can be developed in the spring cannot exceed the brake torque $T_b(t)$, therefore:

$$T_{km}(t) = \min\left[|T_k(t-1) + \Delta T_k|, T_b(t)\right], \tag{3}$$

giving the torque in the spring as is:

$$T_k(t) = T_{km}\text{sign}\left[T_k(t-1) + \Delta T_k(t)\right] \tag{4}$$

Hereafter, the dependency on t will be dropped for simplicity.

B. Control Law

The control law must make no assumptions about the VE model, and partition a desired VE toque T_d into a suitable brake and motor torque. The direction of the torque T_k in the spring is used to infer the direction of the user's input torque. When the user's applied torque opposes the direction of the VE torque, the controller must deactivate the brake and use the motor only to avoid stiction. To define a control law, first the motor torque T_m is set to the difference between T_d and the spring torque T_k, limited to the maximum torque the motor can output, T_{mm}. Also, when the spring torque T_k opposes the desired torque T_d, T_m is set to T_d so that motor does not overcompensate for the spring torque being applied in the opposite direction. Therefore:

$$T_m = \begin{cases} \min(|T_d - T_k|, T_{mm})\text{sign}(T_d - T_k) & \text{if} \quad T_s T_k \geq 0 \\ T_d & \text{if} \quad T_s T_k < 0 \end{cases}$$

Once the motor torque is calculated, to estimate the brake torque the previous commanded brake torque T_{bc} is passed through a first order transfer function that accounts for the brake's transient response:

$$T_b(s) = \frac{1}{\tau_b s + 1} T_{bc}(s), \tag{5}$$

where τ_b is the brake's torque time constant, and s is the Laplace variable. Note that the brake torque never exceeds the spring torque, i.e., $T_b \leq T_k$. When the spring torque acts in the same direction as the desired torque, the commanded brake torque T_{bc} must be set to a value large enough to let at torque T_k develop in the spring, else, it should be set to zero:

$$T_{bc} = \begin{cases} 0 & \text{for} \quad T_d T_k < 0 \\ \beta T_d & \text{for} \quad T_d T_k \geq 0 \end{cases} \tag{6}$$

where $0 < \beta < 1$ is a constant that determines how much of the desired torque is taken up by the brake/spring, that is, the maximum torque in the spring is limited to βT_d. The controller must set a minimum bound on the brake torque T_b in (5) to allow change in the estimated T_k. Otherwise, the condition $T_d T_k < 0$ triggers the brake repeatedly when the user's motion is assisted by the motor.

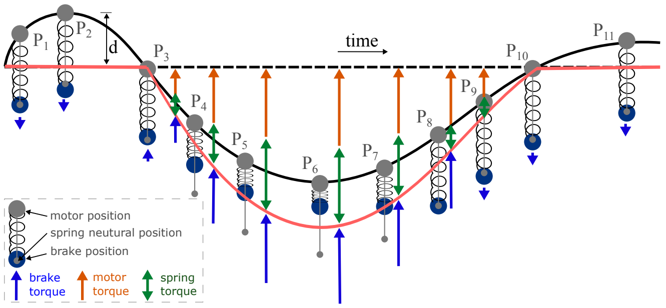

Fig. 3: Simplified diagram visualizing the position and torque of the actuator undergoing a virtual wall simulation.

C. Normal Operation

Fig. 3 shows the algorithm simulating a virtual wall of stiffness K_{ve}, whose surface is represented by the dashed line. The grey dot is the position of the end effector. When above the vertical line, the desired torque is $T_d = 0$. When below the line, the end-effector penetrates the wall and the desired torque T_d is proportional to the penetration into the wall. Between P_1 and P_2, the end effector moves away from the wall. The brake's off-state torque allows the spring to deform but since $T_d = 0$ the actuator is not activated. At P_3 the end-effector reaches the wall. Since $\text{sign}(T_d) = \text{sign}(T_k)$, the brake is powered proportionally to T_d. Since $\beta = 0.5$, $T_{bc} = 0.5T_d$, the brake shares an equal load with the motor. Due to the brake's slow response time, the motor initially accounts for the torque. Between P_4 to P_6, the brake torque continues to increase to allow $0.5T_d$ to develop in the spring. When the user begins to move back towards the surface of the wall at P_7, the spring decompresses returning energy to the user. In cases where the wall stiffness is much smaller than the spring's, the spring fully decompresses while the user is still inside the wall (P_8). At this point, the estimated spring torque reaches zero, triggering the internal condition to turn off the brake and preventing the spring torque from opposing the input torque. At P_9 and P_{10}, the motor pushes the user out of the wall.

D. Torque and Stiffness Ranges

The maximum desired torque the actuator can accurately describe, henceforth referred to as the critical torque T_c, depends on the VE stiffness. The specific conditions that limit T_c are different for low and high simulation stiffness. The top-left of Fig. 4 shows how the actuator torques evolve during the simulation of the virtual wall of stiffness K_{ve}. The slope of the desired, spring, and motor torques are, respectively: $\alpha_1 = \arctan(K_{ve})$, $\alpha_2 = \arctan(K)$, and $\alpha_3 = \arctan(K_{ve} - K)$. When the motor torque T_m increases faster than the spring torque T_k, ($\alpha_3 > \alpha_2$) the maximum torque is limited by the motor's saturation torque T_{mm}. This always occurs when $\beta < 0.5$, and otherwise when $\beta K_{ve} < K$, since the rate of change of T_k cannot exceed K. If T_{mm} is reached at position θ_c, then:

$$T_{mm} = \min\left[(K_{ve} - K)\theta_c, (1 - \beta)K_{ve}\theta_c\right]. \tag{7}$$

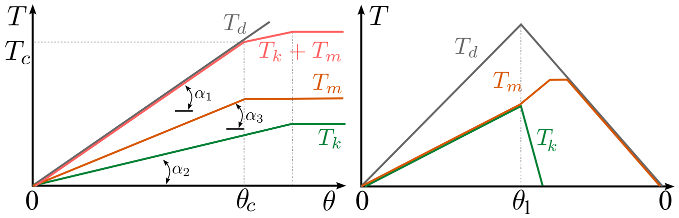

Fig. 4: Virtual wall simulation with stiff (left) and soft (right) stiffness. On the left, the torques increase and decrease along the same slopes. On the right, the shaft stiffness is softer than the wall stiffness. When motion reverses, the motor must compensate for the torque difference.

By replacing $T_c = K_{ve}\theta_c$ in the above, the critical torque as function of K_{ve}, K, and β, is:

$$T_c = \frac{K_{ve}}{K_{ve} - \min(K, \beta K_{ve})} T_{mm}. \tag{8}$$

For a low VE stiffness, the actuator only fails to follow the desired torque when moving away from the virtual wall and if $\beta K_{ve} > K$, see the top-right of Fig. 4. The user pushes the actuator into the wall until $\theta = \theta_1$ before reversing motion towards $\theta = 0$. As the user moves into the wall, the motor and spring torque increase at the same rate ($\alpha_2 = \alpha_3$). When motion reverses, the compliant element decompresses at the spring rate K, and the motor torque must increase to compensate for the decrease in spring torque. T_c occurs when the motor force fails to compensate for the required torque:

$$(1 - \beta)K_{ve}\theta_c + (K - K_{ve})\theta_r < T_{mm}, \tag{9}$$

where the first term is the required motor torque at θ_c and the second term is the change in torque due to the decompression of the VE and compliant element. In the above, θ_r is the angular distance it takes for the shaft spring to decompress:

$$\theta_r = \beta \frac{K_{ve}}{K} \theta_c. \tag{10}$$

Solving for critical torque yields:

$$T_c = \frac{T_{mm}}{1 - \beta \frac{K_{ve}}{K}} \tag{11}$$

The equation above shows that the critical torque depends on the ratio of the VE stiffness to that of the shaft stiffness.

III. Experimental Setup

To validate the proposed sensing approach, the prototype shown in Fig. 5 was built. A DC motor (TT TRS-775PM-2470) with maximum torque of T_{mm} = 85 mNm and a torque constant of 15 mNm/A was used. The motor's torque is controlled via motor's current with a PID controller. A magnetic particle brake (Placid B1-24-2FM) with a maximum torque of T_{bm} = 110 mNm is also used. The brake torque is assumed to be proportional to its input voltage. A 8192 pulses per revolution encoder is attached to the motor shaft. Both the brake and motor are powered by H-bridge amplifiers and the system was controlled using Simulink Real-time through a Humusoft MF634 card running at 1 kHz.

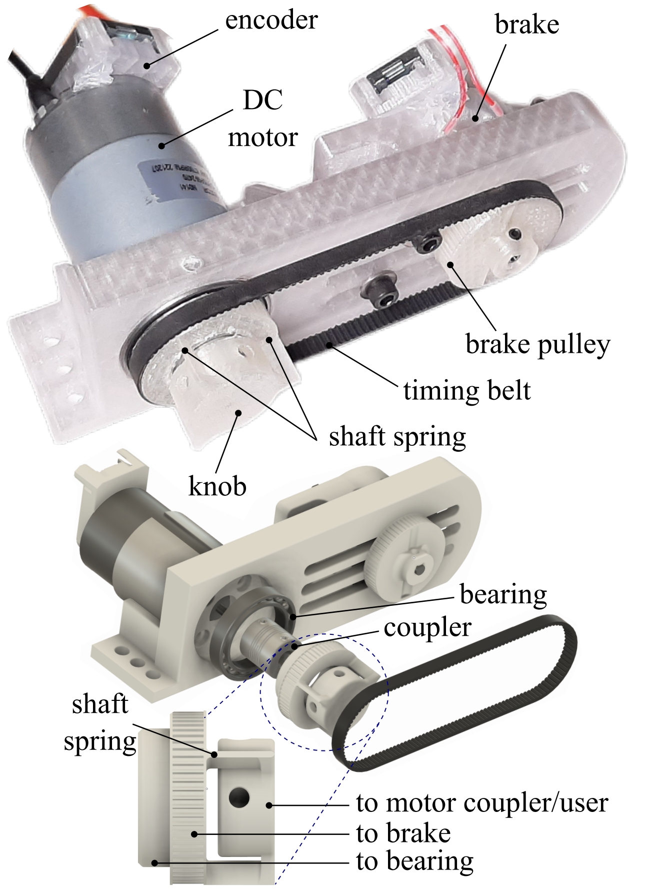

Fig. 5: Actuator prototype. The shaft spring is a semi-rigid 3D printed part that connects the knob to the brake, adding a small compliance between the user/motor and the brake. A particle brake is connected to the knob with a timing belt. An encoder measures the position of the motor's shaft.

The compliant element was built within the knob the user holds following the arrangement shown in Fig. 2. It consists of two concentric 3D printed drums connected though a semi-rigid 3D printed part that acts as a torsion spring. The inner drum is connected rigidly to the motor and knob, while the outer drum serves as a pulley, connecting to the brake via a timing belt. The stiffness of the compliant element was chosen to be sufficiently large to minimize the compliance, but soft enough to allow the controller to operate without oscillations. Its stiffness was determined experimentally by setting the brake to maximum torque and providing a ramping input torque to the motor, while measuring the motor's shaft displacement as shown in Fig. 6. In the linear range, the identified stiffness was K = 2.873 Nm/rad, or 35 mNm/deg.

IV. Experimental Results

To evaluate the proposed actuator and controller, a series of experiments were conducted in three different scenarios.

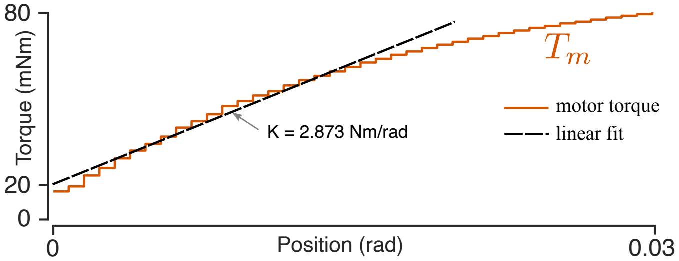

Fig. 6: Shaft stiffness characterization. With the brake powered to its maximum torque the motor applied a ramp torque input, while the motor's displacement was measured. The linear range is limited to 0.015 rad. Outside this range, nonlinear behaviour occur due to material compliance.

Scenario 1 - To determine the maximum stiffness the actuator can emulate before instability, with the brake turned off, the motor was spun until it reached a desired speed of 18 rad/s as to acquire a momentum. Upon reaching the desired speed, the motor was powered off. The VE then simulates a virtual spring connected to the actuator end-effector, starting at the position where the motor reached the desired speed. The end-effector then oscillates as it compresses and stretches the virtual spring. If the system is stable, these oscillations decay to zero. The number of times the motor crosses the centre point of the virtual spring is then recorded for a range of simulated stiffness throughout the 1 second test period. The experimental results show that the number of bounces remains relatively constant so long as $K_{ve} < 10$ Nm/rad (174 mNm/deg), after which the actuator becomes unstable. Another point of instability occurred when $K_{ve} = K$.

Scenario 2: The VE emulates a wall with a stiffness ranging from 10 to 100 mNm/deg, with $\beta = 0.5$. The user moves the actuator knob towards the virtual wall to a maximum position, and then reverses motion. The results are shown in Fig. 8. When the stiffness of the virtual wall is greater than that of the shaft's, the total actuator torque ($T_b + T_m$) follows the desired torque closely. Since part of the torque is provided by the shaft spring, stiction does not occur. When the shaft spring torque opposes the desired torque, the brake is turned off, allowing the spring to decompress.

Conversely, when the virtual wall stiffness is softer than the shaft spring, the torque gap illustrated in Sec. IIA is observed when reversing motion. The torque gap creates a sharp decrease in torque at approximately 1.7 rad and 3.4 rad in the 30 and 10 mNm/deg tests, respectively. When $T_k \rightarrow 0$, the motor overcompensates for the decrease in torque and causes a short increase in the output torque. This error can be partially attributed to the nonlinearity of the shaft spring stiffness, and undesired compliance between the motor and actuator knob, both of which are not accounted for in the controller. When the stiffness is underestimated, the spring decompresses faster than the controller expects it to. The torque that the actuator can provide against the VE torque ($T_k + T_m$) is greater than in the opposite direction (T_m). When $T_d > T_{mm}$, a torque gap is inevitable when exiting a virtual wall. This discontinuity, inherent to any hybrid actuator, does not affect the user's perception of stiff VEs [9].

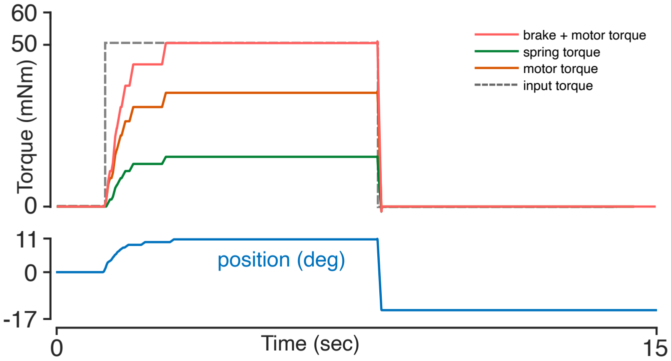

Fig. 7: Experimental results in Scenario 3. An external torque is applied to the actuator shaft while it simulates a virtual wall. When the torque is removed, the actuator exits the wall without external excitation, showing that there is no stiction.

Scenario 3: To show that the controller can disengage the brake and prevent stiction before changes in position occur, an external torque was applied to the actuator as it simulated a virtual wall with $K_{ve} = 25$ mNm/rad. A torque of 50 mNm was applied to the knob by hanging a weight to the actuator knob with a string. The string was then cut, removing the torque. The result of this test, shown in Fig. 7, demonstrates that as the external torque is removed, T_k pushes the end-effector out of the wall. Due to inertia, when T_k changes sign, $T_{bs} = 0$ and the end-effector moves freely without stiction.

The actuator's performance is influenced by the resolution of the position encoder, the stiffness of the compliant element, and the maximum brake and motor torques. With 8192 pulses per revolution and a shaft stiffness of 2.873 Nm/rad, the minimum detectable torque change is 2.2 mNm. In addition, the critical torque in (8) and (11) limits the operating range of the prototype, yet it can reliably simulate virtual walls with stiffness up to approximately 100–150 mNm/deg. The stiffness is further constrained by stability limits of the motor. With such a narrow linear range, the actuator is best suited for short-range interactions requiring moderate fidelity.

V. Conclusion

In this paper, we propose the use of shaft compliance as a soft sensor to estimate the user's input torque. The estimated torque can effectively eliminate stiction when leveraged in the proposed controller. Unlike other approaches where a soft spring is used to store energy and estimate the input torque from the readings of two encoders, we show that a small amount of compliance achieves better results. Limited compliance maintains a high simulation stiffness and transparency, while forgoing the use of the second encoder.

At low simulation stiffness a discontinuity is observed when moving away from a virtual wall. While in part this behaviour is expected, some of it is due to the nonlinear stiffness of the compliant element, which introduces errors in the controller's

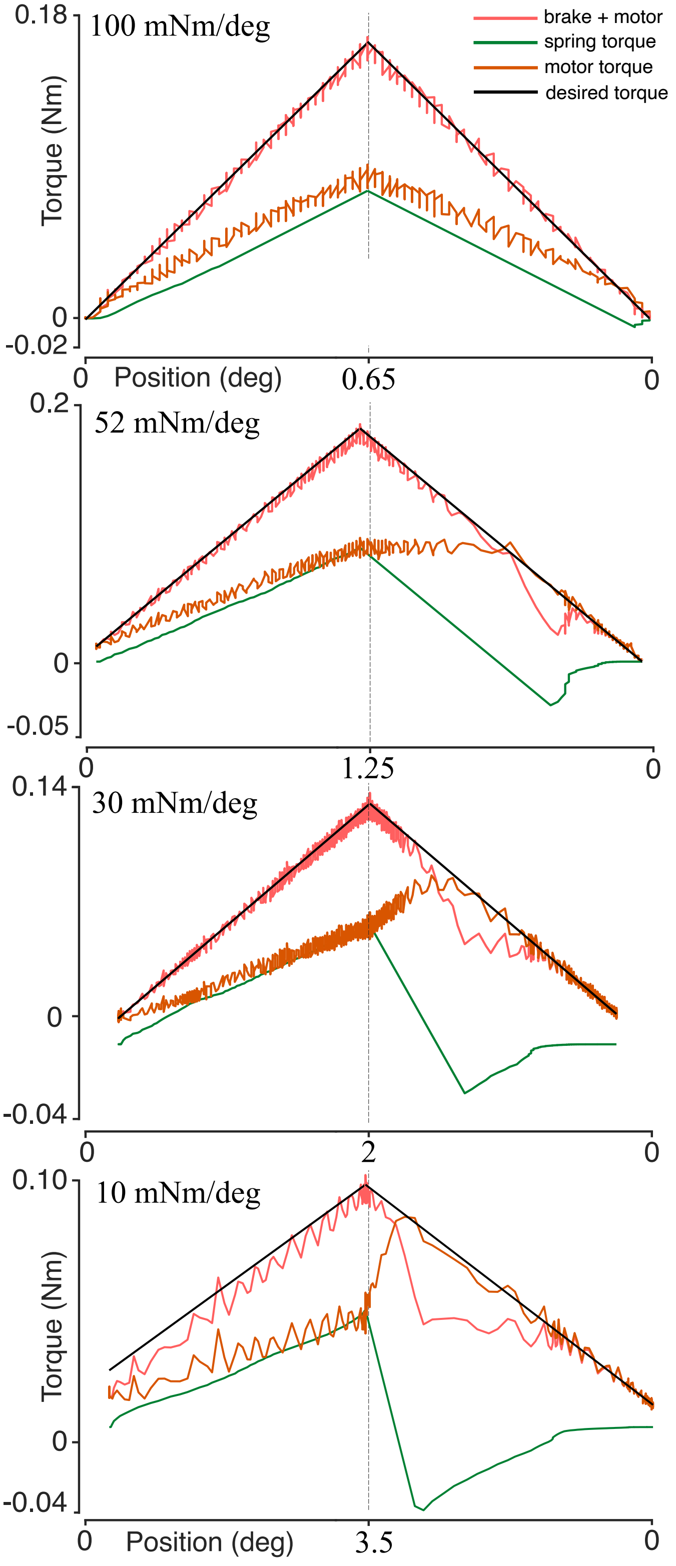

Fig. 8: Experimental results simulating a virtual wall. In the first half of the horizontal axis the user moves into the wall, and away from it during the second half. The horizontal axis is mirrored about the maximum displacement.

estimated spring torque. Further refinement of the control algorithm and a more precise modelling of the stiffness fo the shaft spring may be necessary to improve performance. Furthermore, a more thorough analysis should be performed for extended ranges of β and T_{mm} and T_{bm} to customize the actuator characteristics to specific applications. The concept of shaft compliance as a soft sensor offers a reliable, low cost, and simple alternative to the use of torque sensors in compact, low-cost haptic devices.

Future work will focus on designing a compact integrated actuator and expand the controller to multiple degrees of freedom. In addition, the design of the compliant element will be optimized to achieve a wider linear range, and a more thorough characterization and evaluation of the soft sensor approach will be conducted.

References

[1] A. Lelevé, T. McDaniel, and C. Rossa, "Haptic training simulation," *Frontiers in virtual reality*, vol. 1, p. 3, 2020.

[2] M. Łacki and C. Rossa, "Design and control of a 3 degree-of-freedom parallel passive haptic device," *IEEE Transactions on Haptics*, vol. 13, no. 4, pp. 720–732, 2020.

[3] J. Colgate, P. Grafing, M. Stanley, and G. Schenkel, "Implementation of stiff virtual walls in force-reflecting interfaces," in *IEEE Virtual Reality Annual International Symposium*, 1993, pp. 202–208.

[4] C. Rossa, J. Lozada, and A. Micaelli, "Stable haptic interaction using passive and active actuators," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2386–2392.

[5] C. Cho, M. Kim, and J.-B. Song, "Performance analysis of a 2-link haptic device with electric brakes," in *Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 47–53.

[6] B. Liu, W. Li, P. B. Kosasih, and X. Zhang, "Development of an mr-brake-based haptic device," *Smart materials and structures*, vol. 15, no. 6, p. 1960, 2006.

[7] C. Rossa, J. Lozada, and A. Micaelli, "Design and control of a dual unidirectional brake hybrid actuation system for haptic devices," *IEEE Transactions on Haptics*, vol. 7, no. 4, pp. 442–453, 2014.

[8] J. An and D.-S. Kwon, "Five-bar linkage haptic device with dc motors and mr brakes," *Journal of Intelligent Material Systems and Structures*, vol. 20, no. 1, pp. 97–107, 2009.

[9] C. Rossa, M. Anastassova, A. Micaelli, and J. Lozada, "Perceptual evaluation of the passive/active torque and stiffness asymmetry of a hybrid haptic device," in *Haptics: Neuroscience, Devices, Modeling, and Applications: 9th Inter. Conference, EuroHaptics 2014, Versailles, France, June 24-26, 2014*. Springer, 2014, pp. 55–60.

[10] T.-B. Kwon and J.-B. Song, "Force display using a hybrid haptic device composed of motors and brakes," *Mechatronics*, vol. 16, no. 5, pp. 249–257, 2006.

[11] C. Rossa, J. Lozada, and A. Micaelli, "Interaction power flow based control of a 1-DOF hybrid haptic interface," in *Haptics: Perception, Devices, Mobility, and Communication: International Conference, EuroHaptics 2012, Tampere, Finland*. Springer, 2012, pp. 151–156.

[12] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *International conference on intelligent robots and systems.*, vol. 1. IEEE, 1995, pp. 399–406.

[13] B. Chen *et al.*, "Development of robotic ankle–foot orthosis with series elastic actuator and magneto-rheological brake," *Journal of Mechanisms and Robotics*, vol. 13, no. 1, p. 011002, 2021.

[14] B. DeBoon, S. Nokleby, N. La Delfa, and C. Rossa, "Differentially-clutched series elastic actuator for robot-aided musculoskeletal rehabilitation," in *International Conference on Robotics and Automation*. IEEE, 2019, pp. 1507–1513.

[15] F. Conti and O. Khatib, "A new actuation approach for haptic interface design," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 834–848, 2009.

[16] M. Antolini, O. Köse, and H. Gurocak, "A first order transfer function to balance the workload in brake-motor hybrid actuators," in *IEEE Haptics Symposium (HAPTICS)*, 2014, pp. 509–514.