A Rapid and Robust Tendon-Driven Robotic Hand for Human-Robot Interactions Playing Rock-Paper-Scissors

Accepted at IEEE International Conference on Robot and Human Interactive Communication (ROMAN) 2024. Final version published at: `https://doi.org/10.1109/RO-MAN60168.2024.10731344`

Xiang Deng†, Stefan Weirich‡, Robert K. Katzschmann‡, and Tobi Delbruck†

***Abstract*— Rapid human-robot interactions require fast hardware platforms with minimal latency and high reliability. In response, we present a cost-effective, electrically actuated, tendon-driven robotic hand. This hand features a unique spool-free actuation mechanism that achieves a limit-to-limit flexion movement in less than 60 ms, matching human speed. To our knowledge, it is among the fastest electric motor tendon-driven robotic hands available today. The high speed of the robotic hand was successfully demonstrated in public by playing Rock Paper Scissors at a science fair. This research work outlines the design methodology and introduces a simulation-optimization framework that allows users to preview the motion of the hand, quantify the actuation performance, and customize the design parameters prior to fabrication. The proposed actuation mechanism, along with the simulation and optimization tools, illustrates design principles and computational methods applicable to other dynamic human-robot applications that require fast reaction times. The Dextra hand design is available at `https://sensorsini.github.io/dextra-robot-hand`.**

I. INTRODUCTION

The game Rock Paper Scissors (**RPS**) originated in China during the Han Dynasty about 2200 years ago and has since maintained popularity. Due to its simplicity and accessibility, people of all ages, speaking different languages, can easily understand its rules and play it anywhere with no tools needed. It is a game that gives both players an equal chance of winning, making people wonder whether they can increase the chance of victory by improving observation and prediction.

We have used the game RPS since 2016 [1], [2] as a public educational demonstrator robot called *Dextra* to show the concepts of activity-driven perception and sparse Deep Neural Network (**DNN**) inference with our event cameras and hardware neural accelerators. Here, we present our latest developments in making a highly reactive, low-cost, tendon-driven robotic hand to play RPS. We call it the *Dextra hand*.

Fig. 1 shows the Dextra robot with the hand during a public science fair. The supplementary video (Footnote 1) shows that the Dextra hand matches the speed of an adult human to show the RPS symbols. The hand achieves an average limit-to-limit flexion time of 54 ms and an average limit-to-limit extension time of 97 ms. It costs less than 200 USD in materials, including 3D printing materials and actuators.

†Sensors Group, Inst. of Neuroinformatics, UZH-ETH Zurich, CH; `https://sensors.ini.ch`; `{dxiang,tobi}@ini.ethz.ch`.

‡Soft Robotics Lab, ETH Zurich, CH; `https://srl.ethz.ch`; `{sweirich, rkk}@ethz.ch`.

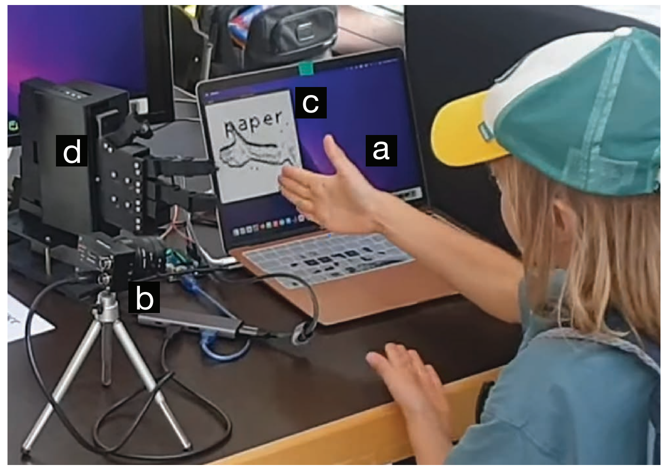

Fig. 1: Dextra playing rock-scissors-paper with public audience at the Swiss Scientifica 2023. **a**) Human throws 'paper' gesture, **b**) event camera observes the human hand, **c**) 'paper' gesture is detected, and **d**) Dextra throws 'scissors' gesture.

Fig. 2 shows the Dextra perception-actuation pipeline. In combination with the symbol inference latency of about 20 ms, which is much quicker than the human perceptual latency of about 200 ms, we create the convincing illusion that the robot can outguess humans in nearly 100% of the throws.

Our contributions include three main aspects:

1) Design principles and methodologies that enable this low-cost, fast and durable robotic hand.
2) A simulation and optimization framework that allows users to preview tendon-driven motion, estimate actuation efficiency, or optimize design parameters (*e.g.*, for desired speeds) before hardware fabrication and deployment.
3) An open source project[1] provides CAD models, robot code, and 3D printing and assembly instructions to make this hand more accessible and allow makers from any background to be able to build and use this hand for research and educational projects.

II. RELATED WORK

Although speed is one of the key results in this work, simplicity, cost, robustness, and compliance were also priorities, inspired by lessons from previous exhibitions. A 5-DOF robotic hand (Aliexpress Bionic Hand) that we

[1] `https://sensorsini.github.io/dextra-robot-hand/`

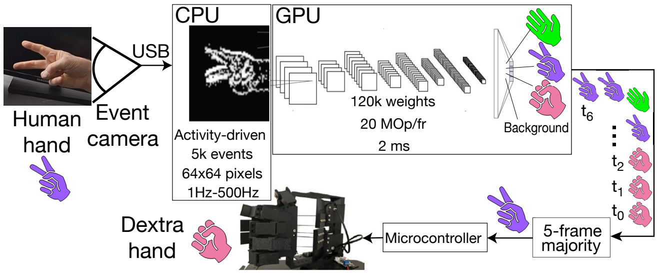

Fig. 2: System pipeline of the Dextra hand playing RPS with people. It is driven by variable frame-rate constant event-count input frames. A small convolutional neural network (CNN) classifies each frame. The robot decision is based on a majority vote over the past five predictions, whereupon the correct symbol is shown by the hand [1]. The game is configured always to tie or beat the human participant. Here, scissors are beaten by rock. The code and CNN are included with the hand design (Footnote 1).

used for previous demonstrations closely resembles a recent bionic hand [3]. It uses a laser-cut plastic linkage bar-based mechanism for actuation, with one servo and linkage bar per finger. The linkage-based mechanism, which includes multiple interconnected rigid parts and joints that form a kinematic chain, is inherently complex in structure and is prone to wear. It did not last a single day of exhibition without adjustment or repair. Tolerance degrades over a single day, where small variances between moving parts and joints accumulate, leading to jamming and failure. On the other hand, bar-linkage-based implementations can offer benefits in enabling bidirectional control of joints [4], in particular by a single actuator for both flexion and extension. In connection to our work, we draw inspiration from the underactuated mechanism, which uses fewer motors than joints to enable finger flexion and extension; yet, we wanted to reduce the complexity in both actuation and structure for higher robustness and ease of maintenance: The simpler the design, the fewer parts that can fail.

Other underactuated robotic hands leverage tendon-driven mechanisms [5], [6], [7], passive components [8], [9], pneumatic actuation [10], [6], [11], [12], or by Shape Memory Alloy (**SMA**) [13]. Among the underactuated mechanisms, we identify tendon-driven and passive components as the most affordable and practical options to integrate into robotic hands for public interaction, particularly when considering the complexity and cost of the fabrication compared to pneumatic or SMA alternatives. In addition to reducing the number of motors in one certain direction of actuation, at least one tendon-driven robotic hand has demonstrated the possibility of using a single actuator (electric motor) for bidirectional movement (flexion and extension) per finger [14].

Among the many known robotic tendon-driven electric motor hands, a spool or wheel that attaches directly to a motor is often used to wind or unwind the tendon [15], [16]. These mechanisms demonstrate strengths in actuation precision and efficiency, where the tendon is tangentially aligned with the spool for optimized transmission. However,

	Shadow [6]	Janken [17][18]	EPT [19]	Aliexpress Bionic Hand[f]	**Dextra** (this work)
Actuation	Pneum. Tendon	DC mot. per joint	DC mot. Tendon	Servo Linkage	Servo Tendon
L-L Flex time (ms)[a]	70	20/30	450	160-220	54
L-L Extend time (ms)[a]	70	20/30	NA	160-220	97
Range[b]	100%	~60 %	~60 %	100%	100%
Cost	$> \$5k$[c]	$> \$20k$[d]	$< \$500$	\$150	\$200[e]

[a] Time from limit to limit.
[b] Movement range from limit to limit for the reported time.
[c] Est. from the parts cost table and 3D printing (not the \$4k CNC cost in [6]).
[d] Est. from CNC and custom actuators.
[e] Table II [f] "UNO Open Source AI Bionic Robot Hand"

TABLE I: Comparisons with related work.

these benefits come with disadvantages in space constraints and the limited scalability in speed—to increase the speed via increased spool radius requires a quadratic increase in spool area. The space and speed scalability constraints would hinder scaling up the speed during iterative design phases, thus motivating us to seek alternatives to the spool-based actuation methods in the hand design.

For high-speed robotic hands, Janken [17] is a well-known high-speed RPS robot. The reported speed of hand movements that are sufficient (*i.e.*, partial hand closing/opening) to indicate a hand symbol is 30 ms. From further tuning of the actuation parameter, [18] shows that the sign formation time can be improved to only 20 ms. The action is performed with a custom high-power mini DC motor in each CNC machined aluminum joint that delivers a high current flow in a short time (0.1 s, allowing the joints to open or close at a rate of 180°/0.1s [20]). For the tendon-driven counterparts, [6] presents a pneumatic actuation that achieves a 70 ms limit-to-limit movement. [8] reports a gripper that combines tendon-driven actuation and spring elements; the tendon pulls the fingers to extend and then releases to allow the passive element to grasp quickly. [8] reports passive grasping that takes 96 ms. [19] provides a speed reference using the spool-based tendon-driven mechanism actuated by DC motors: they show that with a spool radius of 10 mm and no load speed of 100 RPM, it takes about 450 ms to complete half of a full finger flexion.

Table I compares these high-speed hands with the Dextra hand. Dextra is as quick as the pneumatic Shadow but costs 20X less. It is compliant, unlike the Aliexpress Bionic and Janken hands.

We are aware of several related works in the development of simulation methods for tendon-driven robotic hands, including multibody modeling of soft continuum robots under tendon-driven actuation [21], physics-based simulation of joints and bodies [22] or more specifically in the context of gripper grasping [23]. The simulation framework of this paper distinguishes from the examples in that it concerns quantifying pulling actuation efficiency; the movement of the tendon during actuation and the changing contact between the tendon and other parts of the actuation mechanism are explicitly modeled, which play key roles in simulation analysis.

III. Methodology

A. Design choices

1) Actuation Mechanism: One main objective in designing our robotic hand is optimizing the opening and closing movement speed. We considered two possible actuation mechanisms, spool-based actuation and servo-driven lever (extended arm) actuation. An extended arm and a more giant spool are beneficial in delivering higher speed at the tendon's pulling point. In our design, we decided to employ a servo-driven lever over a spool-based mechanism. The decision was based on considering the following potential advantages: 1) compactness; 2) simplicity in the overall components; and 3) eliminating uncertainty factors, including inter-layer friction and slippage. These advantages are particularly effective when the complete opening and closing movements are achieved with a small change in servo angle. Once this condition is met, a full-size spool would occupy much more space than a lever. Fewer components limit the number of failure sources and allow easier maintenance and repair. These practical considerations motivated us to choose a motor-driven lever actuation mechanism over spool-based actuation.

A naive implementation of the selected actuation mechanism may compromise the speed and mechanical transmission efficiency because only the projection of the applied force at the tip of the lever on the direction of the tendon contributes to the movement of the tendon. To better illustrate, the angle ϕ between the force and the direction of pulling is shown in Fig. 4B and will be discussed in detail in Section III-B.

Section III-B presents a method that simulates the tendon movement trajectory, evaluates the costs in terms of misalignment and range of motion, and is optimized for transmission efficiency and speed.

2) Hand Components: Fig. 3 shows the proposed robotic hand extended and flexed with the two servo-driven lever actuators (also see Fig. 4B). Each motor pulls on the tendons that run through its fingers. One motor connects to the index and middle fingers for the scissor symbol, while the other motor connects to the remaining fingers and thumb. When both pull, the hand shows *rock*, and when both release, the hand shows *paper*.

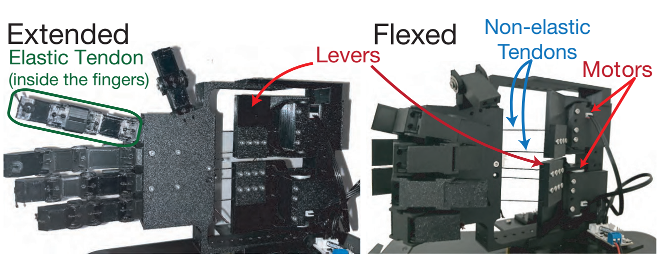

Fig. 3: Photos of the fabricated robotic hand prototype.

Fig. 4A shows the combined flexion and extension mechanism. Each finger combines active flexion and extension based on a passive spring return. The finger integrates a spring return mechanism inspired by [9]. Flexion is by a non-elastic tendon (blue), and passive extension is by an elastic tendon (green). The bottom ends of the elastic tendon are tightened by a screw-clamp (black) instead of knotting, allowing length adjustment for springiness control.

The motors pull the tendons through an external fixed pulley guide (red in Fig. 4B). The main purpose of this circular guide is to provide a smooth redirection of the pulling force from the point of application at the motor arm to the tendon line at the bottom of each finger (blue in Fig. 4A). The relative offsets H and L between the axis of the tendon guide and the motor axis play a key role in both the kinematics of the tendon and the actuation efficiency; the optimization framework in Section III-B explicitly takes these parameters into account in terms of motion optimization and design optimization.

The pulling tendons are attached to the motor arms by clamping screws (see Fig. 3). Inside the clamping structure, each tendon is placed inside a channel, allowing one to adjust each tendon length and ensure full tension before the tendon is tightly attached to an arm. The tendons have a diameter of 0.75 mm. Each channel has variations in width from 0.6–1.2 mm over different segments of the channel, to prevent tendon slippage during operation. This binding method offers advantages over a naive implementation of knotting, including ease of length adjustment, avoiding potential slippage, overly concentrated force, and reduced wear by splitting the force more evenly inside the clamping structure.

Table II summarizes the costs of the hardware components of the Dextra hand. The assembly time for the robotic hand is typically less than five hours. 3D fabrication on a hobby printer (maximum printing speed of 100 mm/s) of all parts takes about 2 days. The non-elastic tendons are fishing lines (WFT CAT 100kg 0.75 mm) and the elastic tendons are generic 1.5 mm TPU lines. All joint bearings (except the metal ball bearing in the thumb, mainly for aesthetic reasons) are fabricated with 3D printing (see Fig. 4A). The 3D printing materials we use are PLA and TPU, with TPU-95 for the tips and middle parts of the fingers (to reduce impact forces) and PLA(+) for the hard parts.

Components	**Costs**
Actuators[2]	2 × \$80
3D printing materials (PLA and TPU)	\$15
Motor Driver	\$8
Bolts and tendons	\$5
Total	**\$188**

TABLE II: Hardware costs of the Dextra hand.

[2]Waveshare ST3025 servo, 0.12 sec/60°, 40 kg.cm at 12 V, no-load.

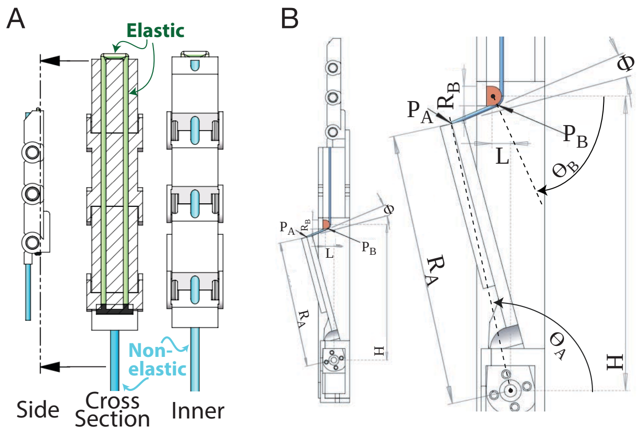

Fig. 4: **A**: Finger design showing elastic (green) and non-elastic (blue) tendons. **B**: Side view of the hand design with a zoom-in showing key parameters for simulation and optimization.

B. *Optimization of flexion*

Optimizing hardware through repetitive builds is costly. The main aim of developing a simulation optimization tool is to provide a realistic kinematic preview of motion trajectories and the estimation of the desirable range of motion before fabrication. A simulation-based preview can be performed given user-defined design parameters such as the length of the lever arm R_A and the relative positions between the fixed pulley and the motors. Users can also leverage the tool to find better kinematic design parameters by varying these parameters and evaluating the performance of simulated motion trajectories.

Our method optimizes the power transmission efficiency and speed of the robot hand. A key component of the approach is the kinematic modeling of the tendon motion as the motor angle changes. It allows us to simulate how the tendon is pulled and to determine when the pulling distance is sufficient to close the finger. The cost J considers the inefficiency factor of the power transmission and the required displacement at the angle of the motor $\Delta\theta_A$.

The pull speed of the tendon is the projection of the linear speed of the lever arm at tip P_A by $\cos\phi$, where ϕ is the angle between the tangent to the arc described by the lever tip and the direction of the tendon $\overrightarrow{P_B P_A}$ at P_A. We define the *pulling efficiency factor* as $\cos\phi$ to express this important quantity. To optimize power transmission efficiency, the inefficiency factor $1-\cos\phi$ must be reduced as much as possible during the entire range of motor rotation. Minimizing $1-\cos\phi$ partially improves speed, but not completely. The range of motor angles $\Delta\theta_A$ must also be minimized and forms another part of the cost. In practice, we observe in Section III-B.2 that optimizing one objective does not prevent improving the other objective because the two objectives converge and the choice of the relative cost weighting factor γ is not critical.

1) Problem Formulation: We first present an optimization formulation formally, including the formation of cost J and constraints as follows. Section III-B.2 explains the optimization based on the following formulation. Fig. 4B illustrates key parameters. The minimization problem is Eq. (1):

$$\min_{\theta_0,\Delta\theta} \quad J = \int_{\theta_{A0}}^{\theta_{A0}+\Delta\theta_A} (1-\cos\phi(\theta))\, d\theta_A + \gamma|\Delta\theta_A| \tag{1}$$

$$\text{s.t.} \quad \|\overrightarrow{P_A P_B} + \delta\overrightarrow{P_A P_B}\| + |\theta_B + \delta\theta_B| * R_B - \|\overrightarrow{P_A P_B}\| + |\theta_B| * R_B = \delta\theta_A * R_A * \cos\phi \tag{2}$$

$$\Delta\mathcal{L} = R_A \int_{\theta_{A0}}^{\theta_{A0}+\Delta\theta_A} \delta\theta_A * \cos\phi \tag{3}$$

$$\overrightarrow{O_B P_B} \cdot \overrightarrow{P_A P_B} \leq 0 \tag{4}$$

$$\cos\phi(\theta) = \frac{\overrightarrow{P_A P_B} \cdot \overrightarrow{O_A P_A}^{\perp}}{\|\overrightarrow{P_A P_B}\| R_A} \tag{5}$$

$$\theta_{min} \leq \theta_0 + \Delta\theta \leq \theta_{max} \tag{6}$$

$$\theta_{min} \leq \theta_0 \leq \theta_{max} \tag{7}$$

where J is the cost, $\theta = [\theta_A, \theta_B]$, $\theta_A \in [\pi/2, \theta_{Amax}]$, $\theta_B \in [-\pi/2, 0]$ (horizontally referenced). O_A, O_B are the centers of motor rotation and of the fixed pulley, with relative horizontal and vertical distances L and H. θ_A is the servo angle, θ_B is the polar angle of the exit point on the pulley guide. P_A is the point of attachment of the tendon in the extended arm, while P_B is the exit point of the tendon, with corresponding polar coordinates (θ_A, R_A) and (θ_B, R_B). Consequently, $\overline{P_A P_B}$ is the segment of the tendon that is directly pulled by the extended arm. ϕ is the angle between $\overrightarrow{P_A P_B}$ and the linear velocity of the lever arm at P_A. $\Delta\theta = [\Delta\theta_A, \Delta\theta_B]$ is the pair of angle changes in the servo angle and the polar angle of the exit point during the full range of movement. The part of the tendon that wraps around the fixed pulley is explicitly considered, and the changed arc length will be calculated from the changes in the polar angle of the exit point. The required tendon pulling distance for full flexion is indicated as $\Delta\mathcal{L}$ (Eq. (3)).

Costs and constraints explanation

The cost function J is a weighted sum of two objectives: the cumulative pulling inefficiency factor, which we quantify as $1-\cos\phi$, integrated over the range of motion, plus a weighted $\Delta\theta_A$ required to pull the tendon to a target length $\Delta\mathcal{L}$.

Eq. (2) specifies the differential constraint of the delta change in the tendon pulling distance per delta change in the motor angle. Eq. (3) constrains the total pulling length to be equivalent to the target pulling length. The inequality constraint Eq. (4) ensures that the tendon segment $\overline{P_A P_B}$ does not cross the pulley. In Eq. (5), the pulling efficiency factor $\cos\phi$ is the projection factor of the lever's linear velocity at the point of application P_A on the tendon direction. The unit velocity vector is calculated by rotating $\frac{1}{R_A}\overrightarrow{O_A P_A}$ 90 degrees, denoted by $\frac{1}{R_A}\overrightarrow{O_A P_A}^{\perp}$.

In summary, this section provides the formulation of performance costs and constraints, including kinematic, geometric, and pulling length constraints that are required to simulate, evaluate, and optimize the pulling motion.

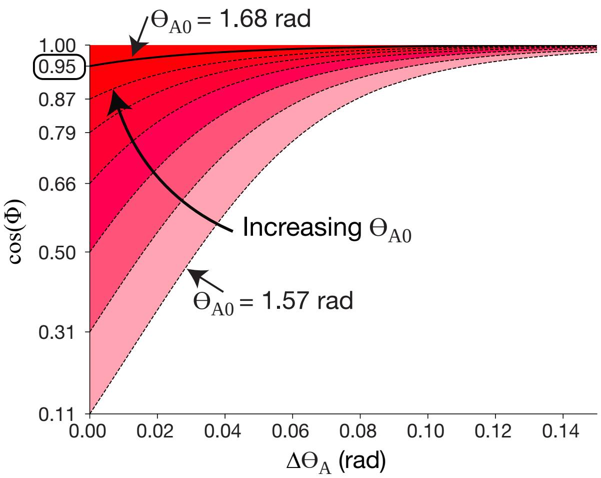

Fig. 5: Comparison of tendon pulling efficiency factor $\cos\phi$, over the range of motor angular displacement $\Delta\theta_A$. Each curve corresponds to a simulated trajectory with a different motor startup angle θ_{A0}. The area between each curve and the flat reference line at 1 (ideal) shows a cumulative inefficiency factor $1-\cos\phi$. Among the trajectories shown in this example, the efficiency strictly increases as θ_{A0} increases. The simulation process allows finding a trajectory above a threshold $\cos\theta$, *e.g.*, 95 percent in bold black. Table III shows the parameters used here and are the hand parameters used for the experimental results.

2) Simulation and Optimization Algorithm: To solve the optimization of Section III-B.1, we first generate a collection of candidates for the initial conditions. Then, for each initial condition, we roll out the trajectory under the constraints. Each trajectory simulation ends when the target distance is reached. For each simulated trajectory rollout, we evaluate the cost with the function J and select the optimal trajectory to identify the θ and $\Delta\theta$ pair associated with the optimal trajectory as the solution.

Fig. 5 illustrates how the optimization obtains a solution of Section III-B.1 and provides insight in how changing the initial θ can quantitatively affect J. Fig. 6 shows trajectory simulation snapshots of two simulated trajectories with different initial conditions highlighted in Fig. 5 .

Next, we describe the process that generates the set of initial conditions in preparation for trajectory simulation. The set of initial conditions is denoted as $\boldsymbol{\Theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$, which is a set of initial pairs θ_A and θ_B defined by $\{[\theta_{A_1}, \theta_{B_1}], [\theta_{A_2}, \theta_{B_2}], \ldots, [\theta_{A_k}, \theta_{B_k}]\}$. It should be noted that the initial position of the exit point (thus its polar angle θ_B) can be calculated based on the value of θ_A, given the assumption that the tendon is fully tensioned in the hardware

Parameters	R_A	R_B	H	L
Value (m)	0.1126	0.008	0.1233	0.0069

TABLE III: Parameters used in the simulation experiment in Fig. 5 and in the hand reported in Section IV.

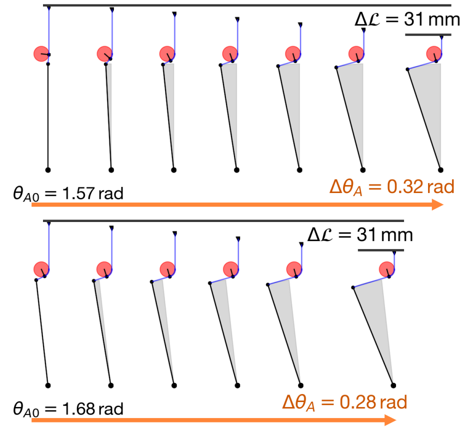

Fig. 6: Snapshots of tendon pulling simulation of Fig. 5. Top and bottom correspond to the two different initial motor angles θ_{A0} and different levels of pulling efficiency factor $\cos\phi$ labeled in Fig. 5. The motor angle increases (with its value represented by the length of the orange arrow) until the same target pulling length $\Delta\mathcal{L}$ is reached. Gray indicates the area swept by the motor arm. The more efficient setup (bottom) requires less motor angular displacement $\Delta\theta_A$ to achieve the same pulling distance.

setup. Therefore, there is no need to generate θ_B. Essentially, we only generate a set of initial motor angle candidates $\boldsymbol{\Theta_A} = \{\theta_{A_1}, \theta_{A_2}, \ldots, \theta_{A_k}\}$, evenly spaced in its min and max range. Then we solve for the polar angle of each exit point and put them in the set $\boldsymbol{\Theta_B} = \{\theta_{B_1}, \theta_{B_2}, \ldots, \theta_{B_k}\}$. Finally, we combine the corresponding elements from $\boldsymbol{\Theta_A}$ and $\boldsymbol{\Theta_B}$ into the set $\boldsymbol{\Theta}$.

Because the tendon is pulled tight without slack, to solve for θ_B given θ_A, we can solve for the shortest path length between point P_A and point P_B, where the constraint Eq. (8) imposes that $\overline{P_A P_B}$ is aligned with the tangent line at P_B:

$$\begin{aligned} \underset{\theta_B}{\text{minimize}} \quad & \|\overrightarrow{P_A P_B}\| \\ \text{subject to} \quad & \overrightarrow{O_B P_B} \cdot \overrightarrow{P_A P_B} = 0 \\ & \theta_{Bmin} \leq \theta_B \leq \theta_{Bmax} \\ & \theta_A = \theta_{A_i} \in \boldsymbol{\Theta_A} = \{\theta_{A_1}, \theta_{A_2}, \ldots, \theta_{A_k}\} \end{aligned} \tag{8}$$

With this constraint, the solution will be unique. After solving θ_B for each $\theta_A \in \Theta_A$, we combine all corresponding pairs into Θ, the set of initial conditions, after which we will perform the trajectory roll-out under each initial condition.

The trajectory rollouts integrate kinematics over discrete step intervals, *i.e.*, by integrating the delta change $\delta\theta$ given the delta input $\delta\theta_A$ per step. When the differential equation is given explicitly in the form of $\delta\theta(k) = f(\theta(k), \delta\theta_A(k))$, $\theta(1) = \theta \in \Theta$, the roll-out of the trajectory is constructed through numerical integration of the differential equation—the forward propagation of the state in discrete steps.

However, in our case, the differential equation is not explicit and the system kinematics is under complex constraints, as shown in Eqs. (2) and (5). As such, we make use of optimization techniques to solve for the differential change. We cast the optimization similar to the shortest path problem presented before, but with constraints modified.

Given the input $\delta\theta_A$, our optimization solution of the differential kinematics becomes:

$$\begin{aligned} \underset{\delta\theta_B}{\text{minimize}} \quad & \|\overrightarrow{P_A P_B} + \delta\overrightarrow{P_A P_B}\| \\ \text{subject to} \quad & \|\overrightarrow{P_A P_B} + \delta\overrightarrow{P_A P_B}\| + |\theta_B + \delta\theta_B| * R_B - \\ & \|\overrightarrow{P_A P_B}\| + |\theta_B| * R_B = \delta\theta_A R_A \cos\phi \qquad (9) \\ & \overrightarrow{O_B P_B} \cdot \overrightarrow{P_A P_B} \leq 0 \qquad (10) \\ & \theta_{min} \leq \theta + \Delta\theta \leq \theta_{max} \end{aligned}$$

Remark 1: In practice, we found that replacing the tangent constraint Eq. (8) with the differential constraint Eq. (2) provides higher numerical accuracy. Although it seems reasonable to claim that the shortest path condition should remain valid and can instead use the constraint 8 as before.

We then roll out a trajectory of each $\theta_i^1 \in \Theta$ into a trajectory $\Omega_i^M = [\theta_i^1, \theta_i^2, \ldots, \theta_i^M]$ through the numerical integration of the differential kinematic changes calculated per step, integrating from the initial condition $\theta_i^1 \in \Theta$ per trajectory. Each trajectory roll-out ends when the target pull length is reached (the latest pulling displacement is an accumulated value by integrating $\delta\mathcal{L} = \delta\theta_A R_A \cos\phi$ until the current simulation step M, see Eq. (3)).

Finally, we evaluate the cost J of all trajectories Ω_i, for $i = 1, 2 \ldots, k$ and select the trajectory Ω_* with the optimal cost J^*. The selected trajectory encompassing all state changes is used for motor control reference. The most useful information from the selected trajectory is the initial condition $\theta_*^1 \in \Theta$ and the accumulated change $\Delta\theta_*$, indicating the starting angle of the motor and the range of angular displacement.

3) Performance preview under parameter changes: The design parameters $\{R_A, R_B, L, H\}$ (Fig. 4B) all affect performance. The simulation-optimization method allows the user to preview the kinematic motion under different design parameters and select the one with the desired performance. Fig. 7 is an illustrative example showing that using the simulation method, one can visualize the quantitative change of the required motor angle range for the same pulling length before and after changing the lever length R_A. Users can configure simulated trajectories with an almost identical initial pull efficiency factor $\cos(\phi_0) \geq 0.95$ for comparisons. Table IV lists the change in lever length used for Fig. 7 and simulated quantitative comparisons of $\Delta\theta_A$.

IV. Experimental results

The Dextra hand hardware parameters used in the experiments are consistent with the previous section and are listed in Table III. The supplementary video (Footnote 1) shows that the Dextra hand matches human speed. This section shows quantitative measurements of the speed of the Dextra hand. The section also includes basic grasping and holding demonstrations.

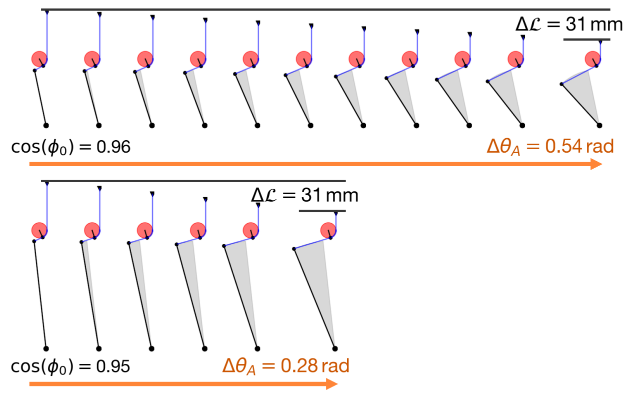

Fig. 7: Preview of changes in kinematic motion required for the same pulling length and almost identical pull efficiency $\cos\phi_0$ factor under variation of design parameter lever arm length R_A.

R_A	$\cos(\phi_0)$	$\Delta\theta_A$	R_A Scale	$\Delta\theta_A$ Scale
0.0586 (m)	0.96	0.54 (rad)	1×	1×
0.1126 (m)	0.95	0.28 (rad)	1.92×	0.52×

TABLE IV: Change in the motor arm length and simulated quantitative comparisons of $\Delta\theta_A$, corresponding to Fig. 7. Other parameters are the same as in Table III.

A. Flexion and extension speed measurements

Fig. 8 shows the bend sensor[3] used to measure the flexion of the finger over time attached to the index finger. It passes through all the axes of the finger joints and bends in alignment with the finger. Its resistance increases with bending. We measured the resistance with a voltage divider and read out the voltage using an analog-to-digital converter.

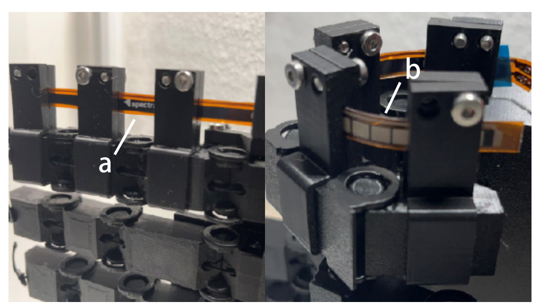

Fig. 8: Flex sensor setup for measuring Dextra's finger flexion. Two views: **a**) extended flex sensor, and **b**) curled flex sensor.

Fig. 9 show flexion and extension measurements. We collect flex sensor readings along with encoder readings of the motor from the moment the motor command is sent. To make sure we measured limit-to-limit movements, we intentionally flexed the finger so it always hit the palm. Data were collected from 20 trials, each time in the order of flexion and then extension. In the plots, we show the mean ± 1 standard deviation of both encoder readings (steps,

[3] Spectra Symbol's original (legacy) flex sensor SEN-08606

$\in [0, 4096]$) and voltage readings (mV) of the flex sensors over time. We can observe very little variation; the motions are highly repeatable and precise.

Time zero is when the motor movement command is triggered on the microcontroller. Then comes the response latency of the motor, which is typically 18 ms in our servos and is independent of load. The span corresponding to the limit-to-limit movement is shown in shaded red, while the shaded blue region indicates the backlash latency[4]. For the flexion experimental data, there is an overshoot in flex sensor readings at the red cut-off because the finger was commanded to overflex and hit the palm frame, which complies with the impact and allows overshoot to a certain extent. The backlash latency of extension is mainly due to intentional motor angle overshoot during flexion. The endpoint of the limit-to-limit movement region of extension is set to where the flex sensor value reaches the steady-state value at the end of the experiment (at 476 ms). There is an offset in flex sensor reading from the end of flexion to the start of extension due to flex sensor (Footnote 3) drift between the measurements.

These measurements show that the Dextra hand achieves a limit-to-limit latency of 54 ms for finger flexion and 97 ms for finger extension. The total latency from the microcontroller to full flexion/extension (including motor response latency) is 72/125 ms.

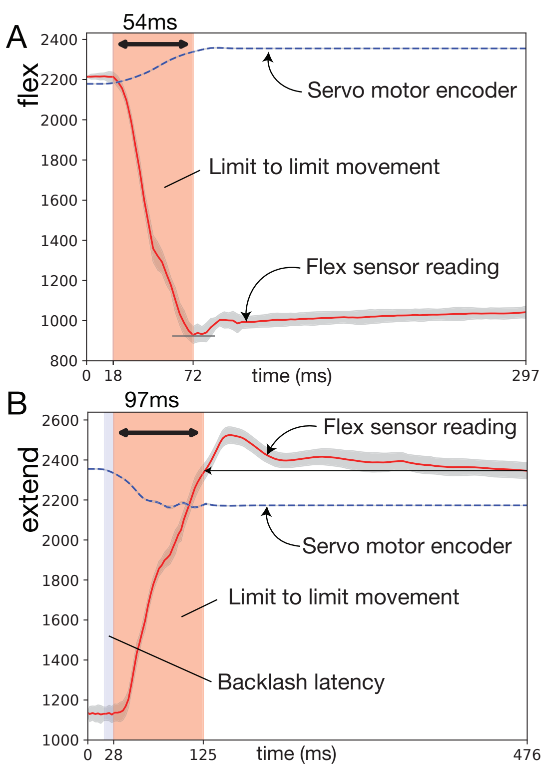

Fig. 9: Flex sensor + encoder measurements during flexion (**A**) and extension (**B**) movement.

To validate these measurements, we also used an event camera (our DAVIS346 [24]) to record the hand movements.

[4]The time interval between the motor starting to move and the finger starting extension.

	Screwdriver	Scrub pad	E-toothbrush	Mug
Size (cm)	1.7×1.7×13.6	1.8×8.5×11.5	2.3×2.8×20	10×10×9.5
Mass (g)	26	7	123	334
Success	✓	✓	✓	×

TABLE V: Grasping results.

We record the movement of the index finger from the top-down view. All other fingers are disabled to avoid noisy data. The event camera data are collected by the software jAER [25], which streams the events to the PC.

Fig. 10 shows 5 ms snapshots of accumulated ON and OFF brightness change events from finger flexion and extension. Since the hand is black and the background is white, OFF events (dark) show the leading edge and ON events (white) show the trailing edge. The event camera observations confirm the flex sensor measurements in Fig. 9.

The above benchmark experiments are performed under the condition that one motor actuates one finger to align with the benchmark methods used in the related work in Table I. When the two motors simultaneously actuate multiple fingers (three shorter fingers and the two longer (index and middle) fingers), we observed an average increase of 4 ms (3 shorter fingers) and 13 ms (2 longer fingers) in the limit-to-limit flexion time due to increased load.

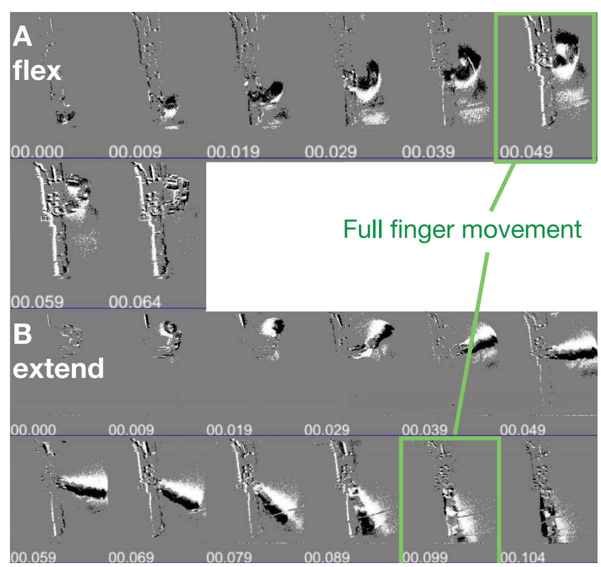

Fig. 10: Event camera motion capture of 65 ms finger flexion (**A**) and 105 ms extension (**B**); each event frame shown has a 5ms accumulation time (*i.e.*, 5ms time resolution).

B. Grasping and Holding Tests

Although grasping is beyond the scope of this study, we demonstrate its functionality and utility for future work. Table V summarizes success and failure in grasping objects of different sizes and weights shown in Fig. 11. With only two motors, the robotic hand can still grasp and hold a set of objects with different dimensions and weights.

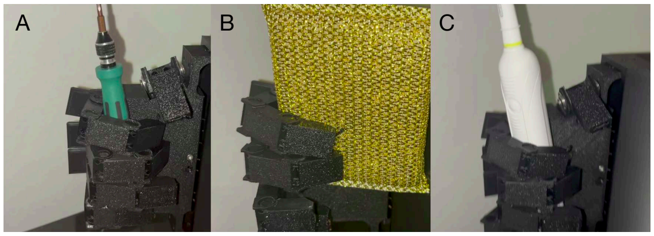

Fig. 11: Grasping and holding demonstrations with a screwdriver (**A**), sponge (**B**), and electric toothbrush (**C**).

V. CONCLUSIONS

This work stemmed from the motivation to build a quick, robust, low-cost, and compliant tendon-driven electric robotic hand that, with the integration of perception, can interact with people. The Dextra hand reported in this paper is inexpensive and simple to fabricate and assemble. It matches human speed. We hope that it can be useful on its own to others or serve as a starting point for future improvements.

The proposed actuation mechanism, simulation, and optimization tools are rooted in the pursuit of design principles and computation methodologies, which we hope are transferable to others' applications.

REFERENCES

[1] I.-A. Lungu, F. Corradi, and T. Delbruck, "Live demonstration: Convolutional neural network driven by dynamic vision sensor playing RoShamBo," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. Baltimore, MD, USA: IEEE, May 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8050403/

[2] A. Aimar, H. Mostafa, E. Calabrese, A. Rios-Navarro, R. Tapiador-Morales, I.-A. Lungu, M. B. Milde, F. Corradi, A. Linares-Barranco, S.-C. Liu, and T. Delbruck, "NullHop: A flexible convolutional neural network accelerator based on sparse representations of feature maps," *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 3, pp. 644–656, Mar. 2019. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2018.2852335

[3] J. Fu, M. Poletti, Q. Liu, E. Iovene, H. Su, G. Ferrigno, and E. De Momi, "Teleoperation control of an underactuated bionic hand: Comparison between wearable and vision-tracking-based methods," *Robotics*, vol. 11, no. 3, p. 61, 2022. [Online]. Available: https://doi.org/10.3390/robotics11030061

[4] U. Kim, D. Jung, H. Jeong, J. Park, H.-M. Jung, J. Cheong, H. R. Choi, H. Do, and C. Park, "Integrated linkage-driven dexterous anthropomorphic robotic hand," *Nature Communications*, vol. 12, no. 1, pp. 1–13, 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-27261-0

[5] Y. Toshimitsu, B. Forrai, B. G. Cangan, U. Steger, M. Knecht, S. Weirich, and R. K. Katzschmann, "Getting the ball rolling: Learning a dexterous policy for a biomimetic tendon-driven hand with rolling contact joints," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–7.

[6] V. Kumar, Z. Xu, and E. Todorov, "Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 1512–1519. [Online]. Available: https://doi.org/10.1109/ICRA.2013.6630771

[7] H. Li, C. J. Ford, M. Bianchi, M. G. Catalano, E. Psomopoulou, and N. F. Lepora, "BRL/Pisa/IIT SoftHand: a low-cost, 3D-printed, underactuated, tendon-driven hand with soft and adaptive synergies," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8745–8751, 2022. [Online]. Available: https://doi.org/10.1109/LRA.2022.3187876

[8] A. McLaren, Z. Fitzgerald, G. Gao, and M. Liarokapis, "A passive closing, tendon driven, adaptive robot hand for ultra-fast, aerial grasping and perching," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5602–5607. [Online]. Available: https://doi.org/10.1109/IROS40897.2019.8968076

[9] N. Nikafrooz and A. Leonessa, "A single-actuated, cable-driven, and self-contained robotic hand designed for adaptive grasps," *Robotics*, vol. 10, no. 4, p. 109, 2021. [Online]. Available: https://doi.org/10.3390/robotics10040109

[10] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Robust proprioceptive grasping with a soft robot hand," *Autonomous Robots*, vol. 43, no. 3, pp. 681–696, Mar. 2019. [Online]. Available: https://doi.org/10.1007/s10514-018-9754-1

[11] R. Deimel and O. Brock, "A compliant hand based on a novel pneumatic actuator," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2047–2053.

[12] T. H. Hong, S.-H. Park, J.-H. Park, N.-J. Paik, and Y.-L. Park, "Design of pneumatic origami muscle actuators (POMAs) for a soft robotic hand orthosis for grasping assistance," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2020, pp. 627–632.

[13] E. Deng and Y. Tadesse, "A soft 3D-printed robotic hand actuated by coiled SMA," in *Actuators*, vol. 10, no. 1. MDPI, 2020, p. 6.

[14] T. Chen, X. Zhao, G. Ma, B. Tao, and Z. Yin, "Design of 3D-printed cable driven humanoid hand based on bidirectional elastomeric passive transmission," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, p. 76, 2021. [Online]. Available: https://doi.org/10.1186/s10033-021-00595-y

[15] A. Toedtheide, J. Kühn, E. P. Fortunić, and S. Haddadin, "An integrated, force-sensitive, impedance controlled, tendon-driven wrist: Design, modeling, and control," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 25–32.

[16] W.-H. Chen, S. Misra, J. D. Caporale, D. E. Koditschek, S. Yang, and C. R. Sung, "A tendon-driven origami hopper triggered by proprioceptive contact detection," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2020, pp. 373–380.

[17] Y. Katsuki, Y. Yamakawa, and M. Ishikawa, "High-speed human/robot hand interaction system," in *Proc. of the Tenth Annual ACM/IEEE Intl Conference on Human-Robot Interaction Extended Abstracts*, 2015, pp. 117–118. [Online]. Available: https://doi.org/10.1145/2701973.2701984

[18] K. Ito, T. Sueishi, Y. Yamakawa, and M. Ishikawa, "Tracking and recognition of a human hand in dynamic motion for Janken (rock-paper-scissors) robot," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 891–896. [Online]. Available: https://doi.org/10.1109/COASE.2016.7743496

[19] K. W. O'Brien, A. Xu, D. J. Levine, C. A. Aubin, H.-J. Yang, M. F. Xiao, L. W. Wiesner, and R. F. Shepherd, "Elastomeric passive transmission for autonomous force-velocity adaptation applied to 3D-printed prosthetics," *Science Robotics*, vol. 3, no. 23, p. eaau5543, 2018. [Online]. Available: https://doi.org/10.1126/scirobotics.aau5543

[20] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko, "Development of a high-speed multifingered hand system and its application to catching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2666–2671. [Online]. Available: https://doi.org/10.1109/IROS.2003.1249273

[21] W. R. Wockenfuß, V. Brandt, L. Weisheit, and W.-G. Drossel, "Design, modeling and validation of a tendon-driven soft continuum robot for planar motion based on variable stiffness structures," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3985–3991, 2022.

[22] S. Wittmeier, M. Jäntsch, K. Dalamagkidis, M. Rickert, H. G. Marques, and A. Knoll, "CALIPER: A universal robot simulation framework for tendon-driven robots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1063–1068.

[23] T. Ko, "A tendon-driven robot gripper with passively switchable underactuated surface and its physics simulation based parameter optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5002–5009, 2020.

[24] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A $240\times$ 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014. [Online]. Available: http://dx.doi.org/10.1109/JSSC.2014.2342715

[25] T. Delbruck, "Frame-free dynamic digital vision," in *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, vol. 1. Citeseer, 2008, pp. 21–26. [Online]. Available: http://dx.doi.org/10.5167/uzh-17620