

Development and Optimization of a Noncircular Pulley for Motion Decoupling in Cable-Driven Serial Robots

Jinsai Cheng, first author

College of Aeronautics and Engineering

Kent State University Kent, Ohio

Email: jcheng9@kent.edu

Tao Shen, second author¹

College of Aeronautics and Engineering

Kent State University Kent, Ohio

Email: tshen3@kent.edu

ABSTRACT

Cable-driven serial robots have emerged with high potential for widely applications due to their compact size and low inertia properties. However, developing this type of robots encounters a motion coupling issue that the movement of one joint leads to motion of other joints, resulting in complex control. In this paper, we proposed a novel approach for motion decoupling based on a noncircular pulley. The length change of the driving cable caused by the motion coupling problem is resolved by using the noncircular pulley. The calculation process of the profile for the noncircular pulley is illustrated in detail. An optimization process based on brute force method is presented to identify the optimal parameters to minimize the compensation error. A cable-driven serial robot based on the decoupling method is prototyped for assessments. Experiments are conducted to evaluate the performance of the proposed motion decoupling method. The results reveal that the proposed method can effectively resolve the motion coupling issue by maintaining almost constant cable length with a maximum accumulative error only as 0.086mm, demonstrating the effectiveness of the method.

Keywords: Cable-driven serial robots, motion decoupling, noncircular pulley, optimization

¹ Corresponding author.

1. INTRODUCTION

Serial robots have been widely used in different applications such as assembly [1], surgery [2] and mine rescue [3]. They are characterized by a single open-loop kinematic chain, which consists of a series of links that are interconnected through revolute or prismatic joints [4]. The serial robot usually has a compact size and can provide a larger workspace when compared to a parallel robot operating under similar conditions [5]. Furthermore, serial robots can also provide high adaptability, programmability, high dexterity, and good maneuverability [6].

Traditional serial robots are commonly designed with electric motors and gearboxes mounted at the joints, which can result in high weight and a high level of inertia in the robotic arm [7]. This can pose challenges for improving operational speed [8,9] and increase the energy consumption of the system [10]. Various solutions have been used to solve the problem such as using counterweight mechanism [11] or spring mechanism [12]. The counterweight mechanism usually attaches a counter mass to fix the center of robot mass and compensate gravity [10]. However, the counterweight method can result in an increase of the overall mass and inertia of the system [13] and sacrifice the dexterity of the robot [14]. For the spring mechanism, it utilizes the stored spring potential energy to remove the fluctuations in the gravity potential energy [11]. However, it can potentially cause vibrations in the spring and transmit the spring moment to the corresponding joints [15]. Traditional serial robots also encounter application challenges. For example, in surgical application, the serial robots face challenges including space limitation and potential environmental hazards. First, in order to access the human abdominal cavity,

the size of the surgical robot must be smaller than the incisions or natural orifices. However, it can be challenging to maintain both a compact size and sufficient actuation force when employing directly motor-driven actuation that includes electronic components on the joints [16]. Second, as surgical robots require close interaction with human skin or internal organs, it is essential to sterilize the end-effectors of the medical robots to prevent the spread of infections or contamination. However, this process can pose a significant risk of damaging the electronic components of the robot when these components are directly attached to the robotic joints [17,18].

Cable-driven actuation method represents a viable solution to solve the challenges. The cable-driven actuation method allows electrical motors and gearboxes to be situated away from the joints, and the motion and force are transmitted by cables. Thus, the weight and inertia of the serial robot can be reduced [19], and this method also permits the minimization of robot size, while still ensuring sufficient actuation power, as remote motors are not constrained by size. Furthermore, relocating the electronic components of the serial robot ensures that they are not damaged during the sterilization process.

One type of cable-driven serial robots has a snake-like or continuum configuration, wherein all the links are driven together by wires, instead of being driven with each individual robotic joint [20]. Despite being compact and resistant to hazards, continuum/snake-like serial robots suffer from under-actuation, meaning that they lack the actuation capacity to drive each joint individually. Consequently, the desired manipulator trajectories may be deflected by the load presented at the end link or any

external disruptive forces acting on the robotic body [16]. Another type of cable-driven serial robot is a fully actuated serial robot with each joint driven by a pair of wires individually. However, one challenge in creating a fully actuated rigid-link serial robotic manipulator with the cable-driven method is related to motion coupling. Specifically, cables driving the upper joints must pass through the lower joints. As a result, driving the lower joints will not only rotate themselves but will also lead to the movement of the upper joints, thus decreasing the robot's controllability. Therefore, decoupling the motion in the cable driving serial robots represents a crucial area of research focus.

Different methods have been proposed by researchers to address the motion coupling challenges in cable-driven serial robots. These methods can be broadly categorized into software compensation and mechanism compensation methods [21]. Software compensation techniques involve developing algorithms to compensate for the length change in driving cables caused by motion coupling. Quigley et al. [22] utilized a feedforward term in the algorithm to decouple motion joints in a four-joint manipulator. Chen et al. [23,24] used kinematics analysis to compensate for the length change. Sang et al. [25] developed a surgical instrument and decoupled the joint motion by deriving the relationship between rotor space and joint space. However, the software compensation methods are computationally demanding, require precise motion information and are challenging to simultaneously compensate for all the joints [26]. Furthermore, a lack of real-time synchronization of all the joints can cause slack or significant internal force in the cables, which can impede the decoupling process [27]. In contrast, mechanism compensation methods aim to compensate for the cable length change by developing

98 novel mechanism structures. Glachet et al. [28] designed a mechanism that synchronizes
99 the rotational movement of the forearm and drives motors to maintain a constant cable
100 length, thereby eliminating the motion coupling. However, the size and mass of the
101 mechanism increases significantly because of the bulky bar linkage structure and the
102 moving motors, leading to a degraded dynamic performance [26]. Lee et al. [29]
103 developed a decoupling mechanism by adding a moving pulley, which moves linearly
104 using a decoupling link to compensate for the cable length change when the forearm joint
105 rotates. This method may increase the complexity of cable routing [30]. Feng et al. [21]
106 decoupled the coupled motion by using a differential planetary gear set, which consists
107 of one sun gear, three planet wheels, one gear shaft and one under wire wheel. During
108 motion, the sun gear drives the three planet wheels, causing the under-wire wheel to
109 rotate in the same direction, thus decoupling the coupled motion. The structures of the
110 decoupling mechanisms are usually complex, and the complexity of the structure can also
111 complicate the cable routing. Thus, the purpose of this research is to design a compact
112 mechanism to decouple the motion in cable driven serial robots. Mechanism
113 compensation methods usually use pulleys to guide the cable routing. In [31], the authors
114 used noncircular pulleys to guide the trajectory of the end-effectors by controlling the
115 cable winding/unwinding on the pulleys. It is inspired that a noncircular pulley can not
116 only guide the cable routing but also can impact the cable winding/unwinding with its
117 noncircular profile. Thus, it has the potential to compensate for the length change for
118 motion decoupling in cable driven serial robots.

In this paper, we propose a novel mechanism compensation method that utilizes a noncircular pulley to route the cable and, and decouple the motion in cable-driven serial robots. By incorporating this noncircular pulley into the mechanism, the control system is simplified, and the robotic link can be kept compact. Compared with our previous study in a conference publication [32] new calculation process of the noncircular pulley profile is proposed to reduce the compensation error. And more constraints are taken into consideration to reflect practical scenarios during the calibration process. In addition, a novel optimization process based on the brute force method is introduced to find the best parameters set that can minimize the compensation error during motion decoupling, and at the same time, make the robotic structure compact. Furthermore, a prototype with 2 degree-of-freedom is developed to demonstrate the performance of the motion decoupling method. The structure design and methodology for the proposed mechanism are detailed in the paper. Experiments are conducted to validate the effectiveness of the motion decoupling method.

The paper is structured as follows: Section II describes the mechanical design and the process of calculating the profile of the noncircular pulley. Section III introduces the optimization process for the noncircular pulley. In Section IV, the performance verification of the noncircular pulley in compensating for cable length change and motion decoupling is presented.

2. Motion Decoupling Design

The mechanism of the circular pulley plays a pivotal role in achieving motion decoupling, preventing undesired movement of other joints. Thus, the details of the mechanism and the calculation process of the noncircular pulley are illustrated in the following section.

2.1. Motion coupling analysis

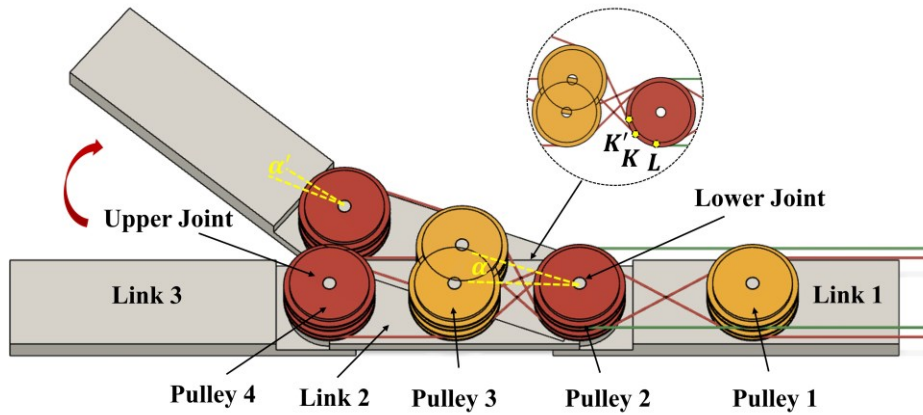


Fig. 1: Motion coupling issue in cable-driven serial robots

Cable-driven serial robots are prone to motion coupling issues due to the routing of cables. The cables driving the upper joints need to pass through the lower joints. As the lower joint rotates, the motion will cause the length change of the driving cable for the upper joint and thus lead to the rotation of the upper link.

As shown in Fig. 1, three robotic links are joined by a lower joint and an upper joint. The lower joint is driven by the green cables, while the upper joint is driven by the red cables, which pass through the lower joint. Four pulleys are fixed on the links to route the passing cables. And two of them, Pulley 2 and 4, are installed co-axially with the lower joint and upper joint, respectively. *Arc KL* is the passing-by cable wrapped on Pulley 2 before rotation. As the lower joint rotates clockwise by an angle α , *Arc KL* changes to

Arc K'L. Thus, the change of the length of the passing-by cable can be quantitatively expressed by the following equation:

$$\Delta l = r * \alpha \quad (1)$$

where r is the radius of the joint pulley (Pulley 2), and Δl represents the length change of the driving cable for the upper joint during the rotation of the lower joint.

Since the passing-by cable drives the upper joint, any length change can cause a α' degree movement of the upper joint, resulting in motion coupling issues. This coupling issue can significantly affect the performance and accuracy of cable-driven robots, particularly in applications that require high precision, such as medical procedures and manufacturing processes. Therefore, it is essential to develop effective mechanisms to compensate for length changes in cable-driven systems and mitigate the impact of motion coupling issues.

2.2. Mechanical design

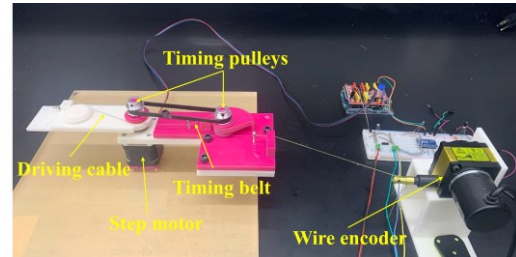
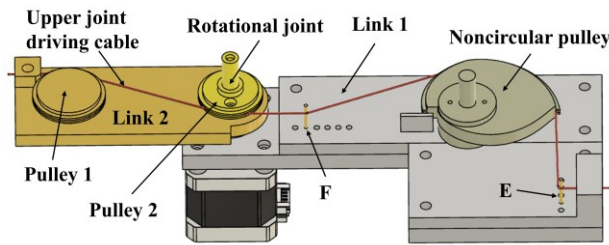


Fig. 2: (a) Concept design of the robotic link; (b) The prototype of the robotic link.

To address the motion coupling issue in cable-driven serial robots, we propose a method that uses a rotatable noncircular pulley in place of Pulley 3 to compensate for the length change. The objective of using the noncircular pulley is to ensure that the length of the passing-by cables remain constant during the movement of the lower joints.

The overview of the conceptual design is illustrated in Fig. 2(a). The robot design has two links and one rotational joint to demonstrate the idea of how the noncircular pulley compensates for the length change during motion. Link 2 has two fixed guided pulleys, Pulley 1 and Pulley 2, and Link 1 has a rotatable noncircular pulley. Pulley 2 is installed co-axially with the rotational joint. A cable, used to simulate driving an upper joint in the cable driven robot, is guided by the three pulleys, and passes through the rotational joint. One terminal of the cable is attached on the left end side of Link 2 and another terminal of the cable is attached to a wire encoder as shown in Fig. 2(b), which is used to monitor the length change during motion. If only the lower joint rotates, and the rotation does not change the length of the cable driving upper joint, it demonstrates a decoupling function. Two guide pins, *Pin E* and *Pin F*, are designed to guide the cable's route. Fig. 2 (b) shows the prototype of the cable-driven robot. The method proposed in this research requires the noncircular pulley to rotate the same degree as Pulley 2. Therefore, as shown in Fig. 2 (b), two identical timing pulleys connected by a timing belt are used to synchronize the motion of Pulley 2 and the noncircular pulley. Since our research is primarily concerned with the motion coupling issue in the robotic link, the structure of the robotic link is simplified by using a step motor to directly drive the rotational joint instead of the cable-driven method. And the research problem is centered around evaluating whether the length of the cable can be maintained constant by using the proposed noncircular pulley during joint rotation.

2.3. Noncircular pulley profile calculation

As the cable length compensation is related to the profile of the noncircular pulley, this section focuses on the calculation process and design of the noncircular pulley profile. Compared with our previous study [32], a new connecting points calculation process of the driving cable is introduced, and a more accurate arc length calculation method is used to enable more precise control of the length compensation. In addition, more constraints are being considered in the calculation process to make it suitable for practical application.

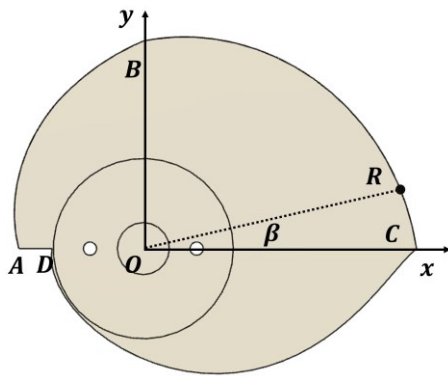


Fig. 3: The profile of the noncircular pulley.

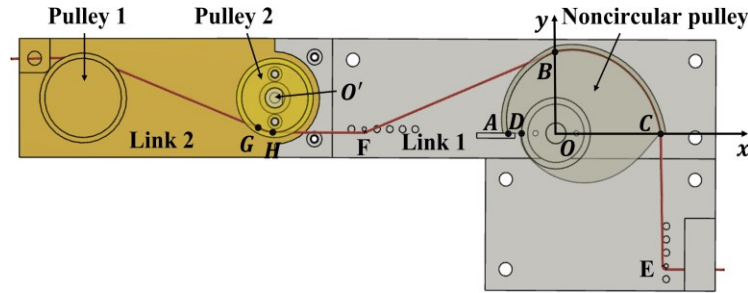
The profile of the noncircular pulley is depicted in Fig. 3, which comprises of four parts *Arc AB*, *Arc BC*, *Arc CD*, and *Line DA*, with *Point O* serving as the rotation origin. The cable is in contact only with *Arc AB* and *BC*. The shape of *Arc BC* is predefined. *Point R*(x_R, y_R) on *Arc BC* can be calculated by using the length of *Line OC* and the decreasing coefficient d_c , which can be represented by the following equations:

$$x_R = (l_{OC} - d_c * \beta) * \cos(\beta) \quad (2)$$

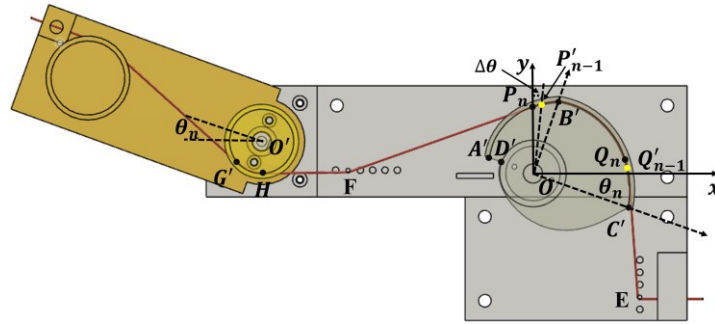
$$y_R = (l_{OC} - d_c * \beta) * \sin(\beta) \quad (3)$$

where l_{OC} is the length of *Line OC*, β is the angle between *OR* and *OC*, and d_c is the decreasing coefficient to adjust the curvature of *Arc BC*. When the coefficient is set to 0,

then *Arc BC* becomes a quarter circle. *Arc AB* is the noncircular part that needs to be calculated based on the length change during rotation.



(a)



(b)

Fig. 4: (a) Cable routing at the initial position; (b) Cable routing after rotation.

Fig. 4a shows the top view and the cable routing of the robotic link. The joint has a 90-degree range of motion. As the lower joint rotates, the noncircular pulley will rotate synchronously with the lower joint through the timing belt, maintaining the same angle as shown in Fig 4b. After 90° rotation, *Arc AB* will completely replace the original *Arc BC* and become the new contact profile with the cable. The profile of *Arc AB* needs to be calculated to compensate the length change during motion.

To calculate *Arc AB*, the motion is divided into N steps. For each step, the increased length on Pulley 2 can be obtained, which is then used to calculate the point of

Arc AB on the noncircular pulley. This calculation process ensures that the length change is countered during rotation. And the final profile of *Arc AB* can be obtained by connecting all the calculated points.

As illustrated in Fig. 4a, *Arc GH* and *Arc BC* correspond to the original cable that connects with Pulley 2 and the noncircular pulley, respectively. While Link 2 rotates around *Point O'*, the cable length increases on Pulley 2. For each step, the increased length and rotation angle can be obtained by the following equations:

$$\Delta l_{i1} = r_1 * \Delta\theta \quad (4)$$

$$\Delta\theta = \frac{90*\pi}{N*180} \quad (5)$$

where Δl_{i1} is the increased length on Pulley 2, r_1 is the radius of Pulley 2 and $\Delta\theta$ is the rotation angle for each step. To compensate for the length change at each step, the noncircular pulley will rotate around *Point O* by the same degree. As shown in Fig. 4, *Point B* and *Point C* are the initial connecting points of the driving cable on the noncircular pulley. As the pulley rotate rotates n steps, *Point P_n* on *Arc AB* and *Point Q_n* on *Arc BC* are two connecting points of the driving cable on the noncircular pulley. For each step, the two points replace the connecting points *Point P_{n-1}* and *Point Q_{n-1}* in the last step and become the new connecting points in the current step. *Point P'_{n-1}* is the *Point P_{n-1}* after rotation in this step, which can be calculated by the following equation:

$$P'_{n-1} = \begin{bmatrix} \cos(\Delta\theta) & \sin(\Delta\theta) \\ -\sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} P_{n-1} \quad (6)$$

Compared to the $(n - 1)$ th step, the length change around *Point* P_n can be described as follows:

$$\Delta l_{i2} = l_{FP_n} + l_{P_n P'_{n-1}} - l_{FP_{n-1}} \quad (7)$$

where Δl_{i2} represents the increased length on the noncircular pulley in the current step, l_{FP_n} is the length between *Point* F and *Point* P_n , $l_{P_n P'_{n-1}}$ is the length of *Line* $P_n P'_{n-1}$, and $l_{FP_{n-1}}$ is the length of the *Line* FP_{n-1} in the last step.

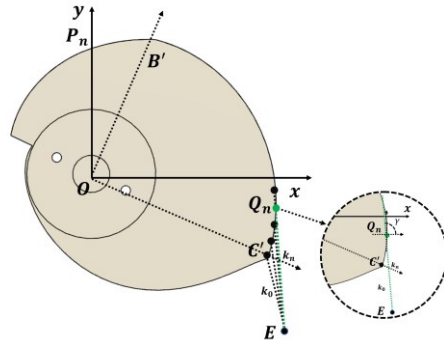


Fig. 5: Connecting *Point* Q_n calculation.

As for *Point* Q_n , it is the connecting point of *Arc* $B'C'$ and *Line* Q_nE shown in Fig. 5. To calculate the connecting point, *Arc* $B'C'$ is the rotated *Arc* BC and is divided into N points, each point on the arc will be connected with *Point* E to form a line, and the corresponding slope of the line, which can be calculated by $\tan(\gamma)$, will be calculated. The line with the smallest slope is the *Line* Q_nE in this step and the corresponding *Point* Q_n can be obtained. After getting the actual connecting *Point* Q_n , the length change around *Point* Q_n can be calculated by comparing to the last step.

The length change around *Point* Q_n comes from two part: the length change of *Arc* $C'Q_n$ compared with the length of *Arc* $C'Q'_{n-1}$, and the length change of *Line* Q_nE

compared to the length of *Line* $Q_{n-1}E$, where Q_{n-1} is the connecting point of the last step and Q'_{n-1} is the rotated Q_{n-1} in this step. For the length change of *Arc* $C'Q_n$, it can be obtained by comparing the connecting point of the last step and this step. The length change is the arc length between the *Point* Q'_{n-1} and the current connecting *Point* Q_n , which can be represented as follows:

$$\Delta l_d = l_{Q'_{n-1}Q_n} \quad (8)$$

where Δl_d is the decreased length in this step.

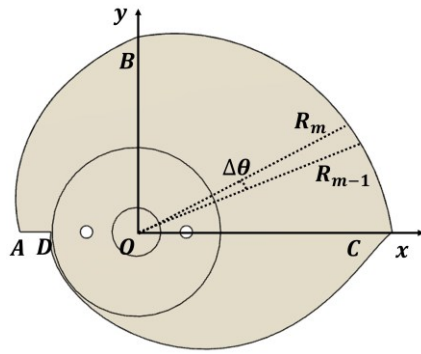


Fig. 6: Arc length calculation.

To calculate $l_{Q'_{n-1}Q_n}$, the arc length of all the N segments on the *Arc* BC are calculated. As shown in Fig. 6, since *Arc* BC is not a circle when the dc is not 0, the length of OR_{m-1} and OR_m is different. Thus, the Law of Cosines is used to calculate the arc length of each segment with the following equation:

$$l_{R_{m-1}R_m} = \sqrt{l_{OR_{m-1}}^2 + l_{OR_m}^2 - 2l_{OR_{m-1}}l_{OR_m}\cos(\Delta\theta)} \quad (9)$$

where $l_{R_{m-1}R_m}$ is the arc length of $R_{m-1}R_m$, $l_{OR_{m-1}}$ is the length of *Line* OR_{m-1} and l_{OR_m} is the length of *Line* OR_m . After getting the length of each segment on *Arc* BC , the length change can be obtained by adding the segments between *Point* Q'_{n-1} and

288 *Point* Q_n . For example, assuming the index of *Point* Q'_{n-1} in *Arc* BC is i , the index of
 289 *Point* Q_n is j and $i \leq j$, then the decreased length l_d can be described as follows:

$$290 \quad \Delta l_d = l_{R_i R_{i+1}} + l_{R_{i+1} R_{i+2}} + \cdots + l_{R_{j-1} R_j} \quad (10)$$

291 For the length change l_{i3} of the *Line* $Q_n E$, it can be obtained by using the
 292 following equation:

$$293 \quad \Delta l_{i3} = l_{Q_n E} - l_{Q_{n-1} E} \quad (11)$$

294 where $l_{Q_n E}$ is the length of *Line* $Q_n E$ and $l_{Q_{n-1} E}$ is the length of *Line* $Q_{n-1} E$ in the last
 295 step.

296 Thus, to make the cable length constant, the increased length and decreased
 297 length should be the same, which can be described as follows:

$$298 \quad \Delta l_{i1} + \Delta l_{i2} + \Delta l_{i3} = \Delta l_d \quad (12)$$

299 Δl_{i1} is a constant value since r_1 and $\Delta\theta$ are constant. And since *Point* E , *Point* F ,
 300 l_{OC} and dc are predefined values, Δl_{i3} and Δl_d can be calculated after getting the
 301 connecting point Q_n . Δl_{i2} is related to *Point* P_n , which needs to be calculated to form the
 302 *Arc* AB . Here *Point* P_n is constrained on the y axis, and it can be described as:

$$303 \quad P_n = (0, y_n) \quad (13)$$

304 where y_n is the coordinate value of *Point* P_n on the axis y .

305 By using Eq. 12, the value of y_n can be calculated. After calculating *Point* P_n , a
 306 rotation matrix is applied to rotate *Point* P_n counterclockwise to find the original
 307 *Point* $P_n^{(o)}$ on the *Arc* AB :

$$308 \quad P_n^{(o)} = \begin{bmatrix} \cos(n * \Delta\theta) & -\sin(n * \Delta\theta) \\ \sin(n * \Delta\theta) & \cos(n * \Delta\theta) \end{bmatrix} \quad (14)$$

2.4. Constraints

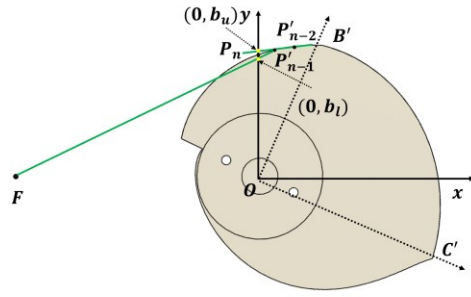


Fig. 7: Upper and lower limits.

When calculating *Point* P_n , it is important to consider several constraints. Only when all the constraints are satisfied can the length compensation process operate accurately and properly.

First, to make sure the cable wraps on the pulley, the profile of the pulley must be convex, meaning that the outer edges of the pulley are curved outward. To make the pulley a convex profile, an upper-limit value is applied when calculating *Point* P_n . As shown in Fig. 7, *Point* P_n is the new connecting point that needs to be calculated in this step. *Point* P'_{n-1} is rotated connecting *Point* P_{n-1} in the last step and *Point* P'_{n-2} is the rotated connecting *Point* P_{n-2} in the step before last. To ensure the profile is convex, the y_n value of *Point* P_n must be less than b_u :

$$y_n \leq b_u \quad (15)$$

where b_u is the intercept of *Line* $P'_{n-1}P'_{n-2}$. The value of b_u can be obtained by calculating the line of *Line* $P'_{n-1}P'_{n-2}$.

Second, in the calculation process, *Point* P_n is set as the new connecting point at each step, thus, it is important to make sure that *Point* P_n is connectable in the rotation process. As shown in Fig. 7, when *Point* P_n is below the point $(0, b_l)$, the cable will

directly connect with *Point* P'_{n-1} instead of *Point* P_n , which will fail the length compensation. Thus, the value of y_n of *Point* P_n must follow the constraints below:

$$y_n \geq b_l \quad (16)$$

where b_l is the intercept of *Line* FP'_{n-1} .

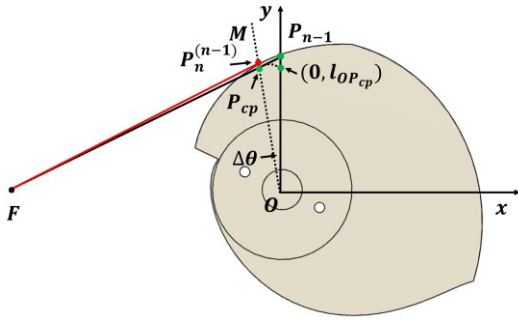


Fig. 8: Upper constraint of *Point* P_n .

The last constraint is that the current *Point* P_n must not affect *Point* P_{n-1} in the last step. As shown in Fig. 8, *Point* $P_n^{(n-1)}$ is the *Point* P_n in the $(n - 1)$ s step. If the position of *Point* $P_n^{(n-1)}$ is above *Line* FP_{n-1} , then the cable will directly connect with *Point* $P_n^{(n-1)}$ instead of *Point* P_{n-1} (shown as the red line). Thus, when calculating *Point* P_n in this step, it is important to ensure that *Point* $P_n^{(n-1)}$ is below *Line* FP_{n-1} .

To maintain *Point* $P_n^{(n-1)}$ under *Line* FP_{n-1} , the cross point P_{cp} of *Line* FP_{n-1} and *Line* OM is calculated as shown in Fig. 8. And the length of OP_{cp} is used as the maximum y value of *Point* P_n on the y axis. The *Line* FP_{n-1} can be obtained by using *Point* F and *Point* P_{n-1} in the last step. For *Line* OM , it can be obtained by using the slope-intercept form. Since *Point* O is the original point, the intercept is 0. The slope depends on $\Delta\theta$, which can be represented by the following equation:

$$k_{OM} = \frac{-1}{\tan(\Delta\theta)} \quad (17)$$

where $\Delta\theta$ is the rotation angle for each step. And the cross point $P_{cp}(x_{cp}, y_{cp})$ can be obtained by combining the two lines. And the length of $LineOP_{cp}$ is set as the maximum y value of the *Point* P_n on the y axis, which can be described as follows:

$$y_n \leq l_{OP_{cp}} \quad (18)$$

where $l_{OP_{cp}}$ is the length of *Line* OP_{cp} . It is equal to $\sqrt{x_{cp}^2 + y_{cp}^2}$. And the final y_n will be constrained at:

$$b_l \leq y_n \leq \min(b_u, l_{OP_{cp}}) \quad (19)$$

2.5. Error analysis

When calculating the profile, the error of length compensation may arise. The error can be categorized as error at each step e_n and the accumulative error e_t . And they can be described as follows:

$$e_n = \Delta l_{i1} + \Delta l_{i2} + \Delta l_{i3} - \Delta l_d \quad (20)$$

$$e_t = e_1 + e_2 + \dots + e_n + \dots + e_N \quad (21)$$

where $e_1, e_2 \dots e_n$ are the errors at each step. The error may come from different sources.

Firstly, for each step, *Point* P_n must satisfy all the constraints. When using Eq. 12 to calculate the y_n value, the solution could be outside of the boundary of the constraints. In this case, the boundary of the constraints (b_l, b_u or $l_{OB_{cp}}$) will be used as the y_n value.

However, Eq. 20 will not be 0 when using this y_n value and small error will be generated.

Second, the actual error is also related to the value of angle for each step. When Pulley 2 rotates, the length of the driving cable routed on Pulley 2 is continuously increasing. However, when *Arc* AB is being calculated, the points on *Arc* AB is

intermittent and is divided into N steps, which means the points during the rotation angle inside each step will not be taken into consideration. The smaller step interval will have smaller errors but may also increase the computational load.

3. Parameter optimization

When calculating *Arc AB*, the position of *Point F* (F_x, F_y), *Point E* (E_x, E_y), the length of *OC* and the decreasing coefficient dc will affect the error during each step, which can directly affect the performance of length compensation during motion. Thus, it is necessary to find an optimal parameter set that can minimize the error. In addition to the error, it is also important to keep the robotic link compact. Therefore, the goal of this optimization process is to find the optimal parameter set that can minimize the error and keep the robotic link as compact as possible.

3.1. Optimization process

Since the relationship between the parameters and the error remains unknown, an optimization process based on brute force method is applied to find the optimal parameter set.

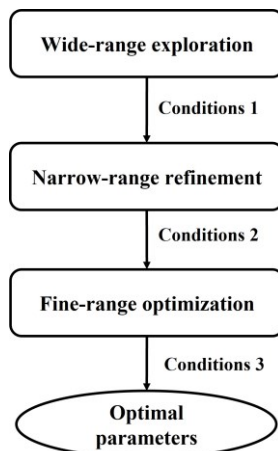


Fig. 9: Optimization process.

Brute force method provides the advantage of exploring all possible combinations of the parameters [33]. To utilize the brute force method for parameters optimization, it is essential to first establish the parameter space and the range of values for each parameter. Once this is accomplished, each combination of parameter values will be evaluated, and the outcomes are assessed to determine the optimal parameter set. As shown in Fig. 9, the optimization process contains three steps: wide-range exploration, narrow-range refinement and fine-range optimization. For each step, F_x , F_y , E_y , l_{OC} and dc are the parameters to be evaluated, where l_{OC} is the length of OC . Ex is set equal to l_{OC} . The searching ranges of the parameter F_x , F_y , E_y , l_{OC} and dc in each step are divided into multiple small steps, and then combined into different parameter sets to find the optimal one.

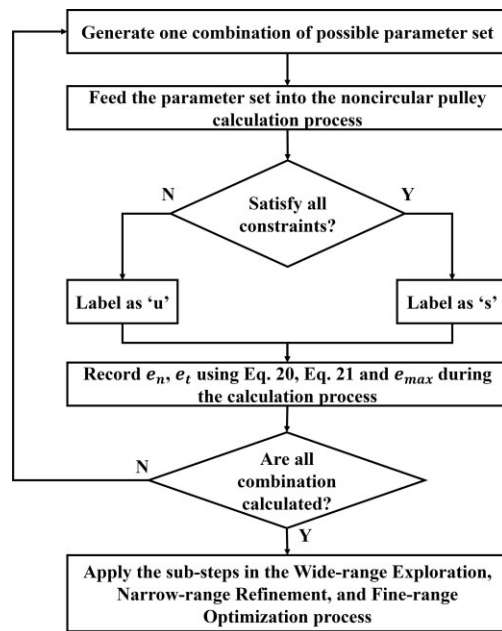


Fig. 10 The flowchart of each optimization step based on brute force method.

Fig. 10 shows the flowchart of the process in each optimization step based on the brute force method. First, one possible parameter set is generated and then fed into the

calculation process to calculate the profile of the noncircular pulley. Constraints are applied to the calculation process to check whether the profile is satisfied with all the constraints. If yes, the parameter set will be labeled as 's', otherwise, it will be labeled as 'u'. The errors are recorded including the error in each step e_n , the accumulative error e_t , and the error e_{max} , which is the one with the maximum absolute value among e_n and e_t . After evaluating all possible parameter combinations, sub-steps in either Wide-range Exploration, Narrow-range refinement, or Fine-range Optimization will be applied to determine the optimized range for next step or the final optimal parameters.

3.1.1. Wide-range Exploration

During the first step, a rough search is performed in a relatively wide range with big intervals of the parameters to estimate the potential optimal range.

After searching in the wide range, three sub-steps are used to narrow down the searching area.

1. Remove unsatisfied parameter sets.
2. Set the maximum allowable range for error e_{max} .
3. Choose small Ey from the remained available values.

First, the solution of Eq. 12 must be valid. Some parameter sets may cause the value of b_l bigger than b_u or $l_O B_{cp}$ and some may cause Eq. 12 to be unsolvable. These parameter sets are labeled as 'u' in the calculation process and will be discarded.

Second, the error needs to be as close to 0 as possible, no matter the e_n at each step or the accumulative error e_t . Thus, the maximum signed error e_{max} for each

parameter set is set in an acceptable range to identify the parameter range that can produce a small error.

Third, to create a compact robotic link, it is essential to keep E_y and l_{OC} as small as possible. And it is important to note that E_y plays a more significant role in determining the size of the link. Therefore, to achieve a compact design, minimizing the value of E_y should be given first priority. Once an appropriate range for E_y is established, the focus can then be shifted towards minimizing the value of l_{OC} . After the first step, a small range with potential optimal parameter set is generated.

3.1.2. Narrow-range Refinement

The narrow-range refinement involves a more focused search within a smaller range obtained from the first step to further narrow down the range of the parameters. The narrow-range refinement contains three sub-steps, which includes:

1. Remove unsatisfied parameter sets.
2. Sort the absolute e_{max} in ascending order, and find the parameter ranges of the first 50th parameter set.

First, the unsatisfied parameter sets are eliminated as mentioned in the wide-range exploration step. And then the data will be sorted in ascending order based on the absolute e_t error. This process will help to identify the most suitable parameter sets that can be used for further analysis.

3.1.3. Fine-range Optimization

The fine-range optimization is focused on finding the optimal set of parameter values that can minimize the error. It also consists of two sub-steps:

1. Remove unsatisfied parameter sets.

2. Sort the absolute value of error e_{max} from smallest to largest, and find the 1st parameter set.

First, the unsatisfied parameter sets are removed. The rest parameter sets are then sorted according to the absolute value of e_{max} in ascending order. And the first parameter set with the smallest absolute value of error e_{max} is set as the optimal parameter set.

3.2. Simulation

This section demonstrates how the optimal parameter set is found based on the above-mentioned process. The radius of the Pulley 2 is 13mm. As the joint rotates 90° , the cable length increases on the Pulley 2 and the total length change is 20.41mm after 90° rotation, which can be obtained by using Eq. 4. To determine the best parameters for the noncircular pulley, *Point E* and *Point F*, the optimization process is applied.

First, during the wide-range exploration step, the range of each parameter is set to a relatively large range to estimate the potential range of parameter values that could lead to the optimal parameter set. To initiate the optimization process, the ranges of each parameter are defined along with a corresponding step size and the step N is set to 90. The parameter ranges and step sizes in Table 1 are carefully selected to allow for a comprehensive evaluation of the parameter space, while also minimizing the computational burden of the optimization process:

Table 1. Parameters of wide range exploration

Parameters	Range	Step Size	Optimized range
------------	-------	-----------	-----------------

F_y	[0, 10]	2	[0, 8]
F_x	[-100, -45]	5	[-65, -45]
E_y	[-40,10]	5	[-30, -10]
l_{OC}	[20,40]	2	[34,40]
d_c	[-0.2,0]	0.05	[-0.15, -0.05]

465

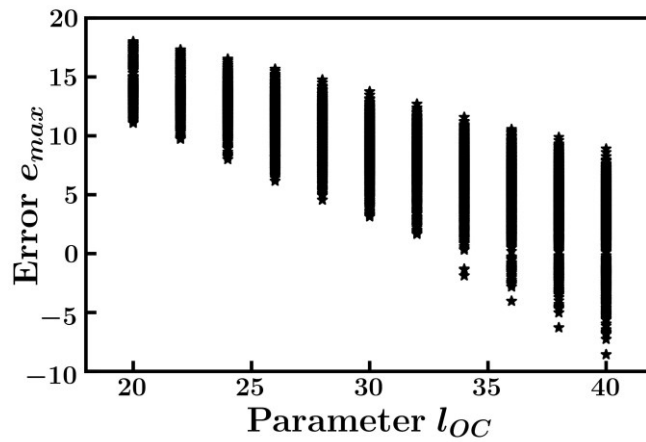
466

467 The parameter ranges and corresponding step sizes defined above result in a total

468 of 27720 possible parameter combinations. The error e_n , e_t , and e_{max} are recorded and

469 the unsatisfied parameter sets are labeled as 'u' during calculation process. After

470 collecting the data, the unsatisfied parameter sets are filtered and 13874 parameter sets

471 are left. Then, the e_{max} is set to (-0.1,0.1) to get the parameter sets that has small length472 compensation error. In addition, to make the robotic link compact, the maximum E_y is

473

474 Fig. 11: The relationship between l_{OC} and e_{max} .

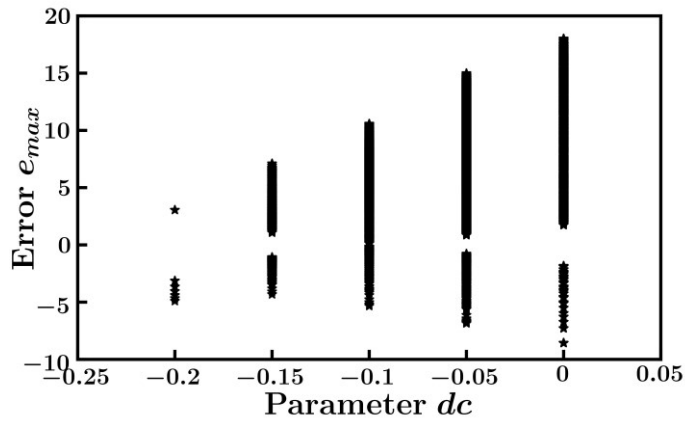


Fig. 12: The relationship between dc and e_{max} .

After the first step, the ranges of parameters F_x , l_{OC} and dc are significantly reduced. By analyzing the relationship between the individual parameter and the error, it is also observed that l_{OC} and dc has a more significant impact on the error as compared to other parameters. As shown in Fig. 11, the error e_{max} decreases as the parameter l_{OC} increase. And Fig. 12 shows the relationship between dc and e_{max} for all the satisfied parameter sets. The satisfied parameter sets have a positive relationship with dc . When dc is -0.1, there are parameter sets that can make the error smaller. Thus, the parameter dc has a significant impact on the error and in the next step, the parameter dc will be set to a smaller range of [-0.15, -0.05]. Both parameters will be divided into small steps in the next two steps. The range of each parameter for the narrow down refinement is described in Table 2:

Table 2. Parameters of narrow-down refinement process

Parameters	Range	Step Size	Optimized range
F_y	[0, 8]	1	[0, 8]
F_x	[-65, -45]	5	[-65, -45]

E_y	[-30, -10]	5	[-30, -15]
l_{oc}	[34, 40]	0.5	[34.5, 40]
dc	[-0.15, -0.05]	0.01	[-0.11, -0.08]

489

490 Second, the narrow-down refinement process is conducted based on the
 491 parameter ranges obtained from the first step with a total of 32175 parameter set
 492 combinations. After obtaining the data, the unsatisfied parameter sets are removed and
 493 the first 50 parameter sets sorted by the absolute e_{max} are selected. The parameter
 494 ranges after this step can be represented in Table 3:

495 Table 3. Parameters of fine-range optimization process

Parameters	Range	Step Size	Optimal parameters
F_y	[0, 8]	1	2
F_x	[-65, -45]	5	-45
E_y	[-30, -15]	5	-20
l_{oc}	[34.5, 40]	0.5	36
dc	[-0.11, -0.08]	0.01	-0.09

496

497 Finally, the fine-range optimization process is applied based on the remaining
 498 parameter range in Table 3 and with a large step $N(450)$ to find the optimal parameter
 499 set. The total number of the parameter combination is 8640. After running through all
 500 the combinations and removing the unsatisfied parameter sets, the data is sorted
 501 according to the error e_{max} in ascending order. And the 1st parameter set is selected as

the optimal parameter set with a value of $F_y = 2$, $F_x = -45$, $E_y = -20$, $l_{OC} = 36$, and $dc = -0.09$.

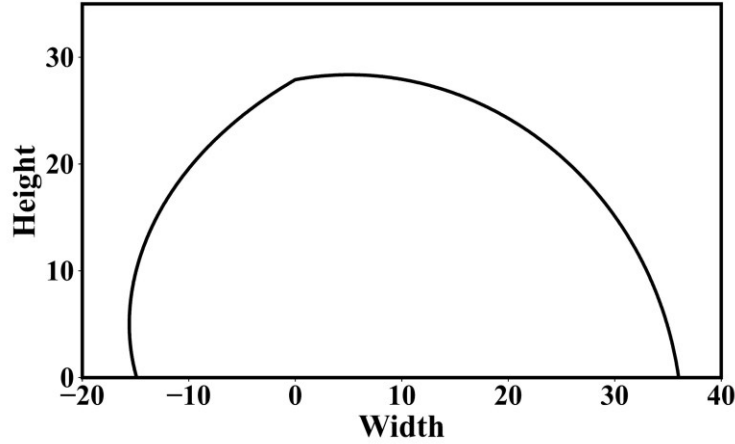


Fig. 13: The profile of the noncircular pulley with the optimal parameters.

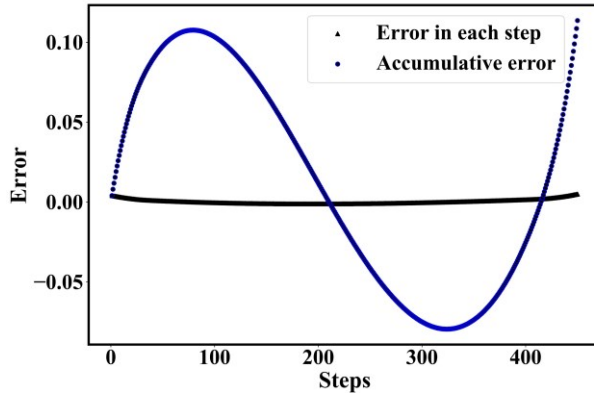


Fig. 14: Error during motion.

After obtaining the optimal parameter set, the profile of the noncircular pulley in the first quadrant can be obtained using Eq. 2 and Eq. 3. And also the points $P_n^{(o)}$ obtained in the calculation process can form the *Arc AB* in the second quadrant. Fig. 13 shows the profile of the noncircular pulley in the first and second quadrant.

For each step, the compensation error of the noncircular pulley is shown in Fig. 14. The black line shows the error of each step, while the blue line shows the accumulative

error during motion. The maximum error e_{max} during rotation is 0.114mm. Compared to the accumulative length increase on Pulley 2 without the noncircular pulley, which is 20.41mm, the length change is within a very small range and has negligible effect on the motion of the upper joint. In addition, the e_{max} is much smaller than that with non-optimized parameters in our previous research [32], which has 0.61 mm maximum error.

4. Experiments

To evaluate the performance of the noncircular pulley with the optimal parameter set, a prototype is designed and experiments are conducted to detect the length compensation error and evaluate the performance of motion decoupling by using the noncircular pulley during rotation.

4.1. Performance of Length compensation

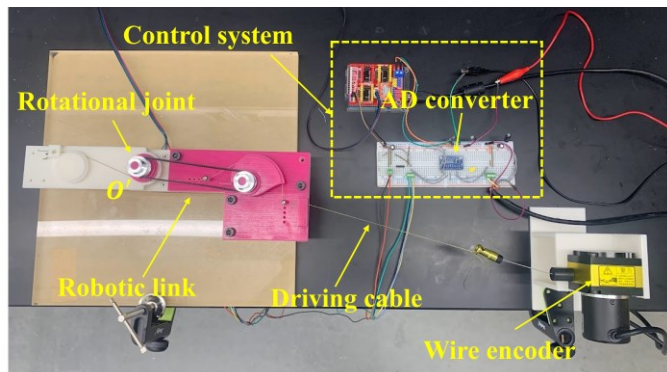


Fig. 15: Length compensation experiment setup.

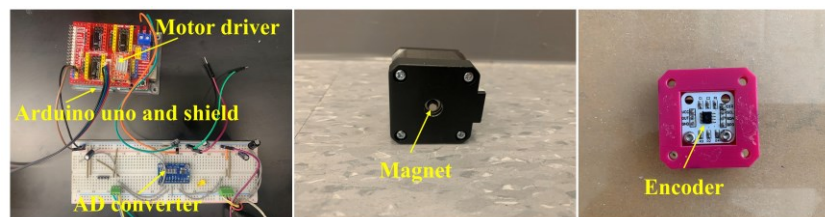


Fig. 16: Control system of the robotic link.

The first experiment is to evaluate the performance of the noncircular pulley in keeping a constant cable length. The experiment setup is shown in Fig. 15. The prototype of the robotic link and a wire encoder is fixed on the table. The wire encoder is utilized to detect the actual length change of the cable-driven robot. An ADC converter (ADS1115, manufactured in Shenzhen, China) with 16-bit resolution is employed to capture voltage fluctuations of the wire encoder during motion and convert them to length changes. A calibration process is applied to establish the relationship between the voltage and the length. The details of the calibration process are described in [32]. A cable is attached to the robotic link on one end, routes through the joint, and then connects to the wire encoder on the other end. The lower joint is driven by a stepper motor that receives commands from an Arduino Uno. As shown in Fig. 16, a magnet with two poles is coaxially mounted with the stepper motor. To control the stepper motor, a motor driver (A4988, Motor driver, Shenzhen, China) is used, and it is configured to generate 800 pulses to rotate the motor one revolution. The motor driver connects with the Arduino uno through an Arduino Uno shield. A magnetic encoder is attached to the stepper motor, facing the magnet, to detect its actual position with high precision of 12 bits. This encoder information serves as feedback that compares the desired position with the actual position to adjust the motion of the stepper motor for accurate positioning.

During the experiment, the robotic link rotates 90° around axis of the rotational joint (*Point O' if top viewed*) as shown in Fig. 15. This rotation is discretized into 90 steps. For each step, the actual rotation of Link 2 and the cable length change are recorded by the encoder and wire encoder, respectively.

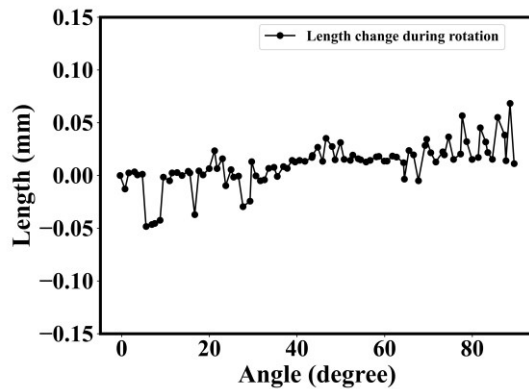


Fig. 17: Length changes during motion.

Fig. 17 illustrates the length change during the 90° motion. The figure shows the actual length change during the rotation achieved by the proposed noncircular pulley. As demonstrated in the figure, the length changes throughout the entire motion are small when using the noncircular pulley, as evidenced by a maximum length change of 0.086 mm and an average length change of 0.028 mm. Compared with the decoupling mechanism in [34], which has a compensation error of 1.26mm, the proposed research has better performance in length compensation. In [26], the mechanism with decoupling links and a moving pulley can control the length change within 0.01 mm. However, the structure of the proposed method in this research is much more compact, and routing of the cable is easier compared with the decoupling link mechanism. The results indicate that the noncircular pulley effectively compensates for length changes during motion.

4.2. Performance of Motion decoupling

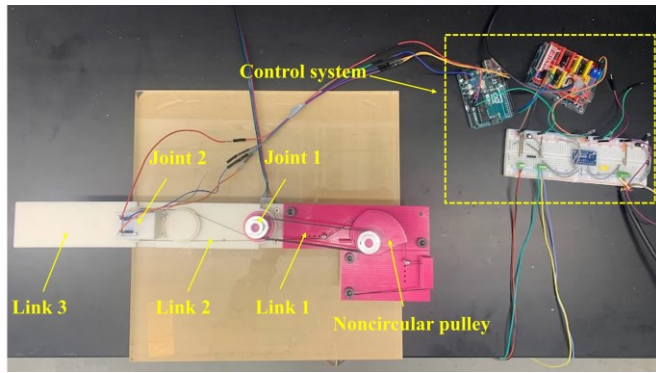


Fig. 18: Motion decoupling experiment setup

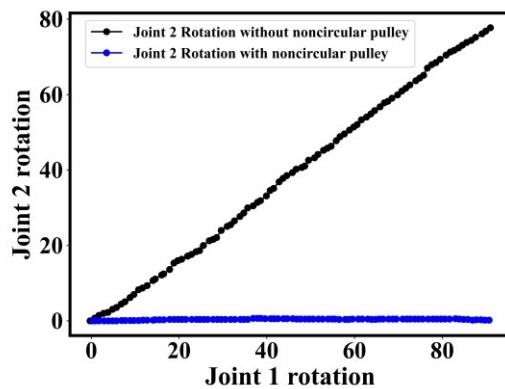


Fig. 19: Results of motion decoupling experiment.

The Last experiment is to verify the performance of motion decoupling when using the noncircular pulley. A third link is added to the original prototype and joined with Link 2 to evaluate the effectiveness. Fig. 18 shows the setup of the experiment. The prototype consists of 3 links and 2 joints. Joint 1 enables the rotation of Link 2, while Joint 2 facilitates the rotation of Link 3. Two cables are routed around the joints to drive Link 3. One end of the cables is fixed on Joint 2, while the other end is fixed on one pin on Link 1. An encoder is installed at Joint 2 to read the real-time position of Joint 2. To read the data from the encoder, another Arduino Uno is introduced and communicates with the main Arduino Uno via UART. During the experiment, Link 2 rotates around Joint 2 for 90° with

1° for each step. The motion of Link 3 is recorded during the experiment under the condition of the circular pulley and the noncircular pulley. Fig. 19 shows the results of the motion of Joint 2 during the experiment. The black line shows the Joint 2 rotation when using the circular pulley with a radius of 15cm. As Joint 1 rotates 90°, Joint 2 has coupled motion with a maximum coupling angle of 77.74°. The blue line shows Joint 2 rotation when using the noncircular pulley. During the 90° rotation of Joint 1, Joint 2 can stay in the original position with a maximum rotation angle of 0.67°. The performance of the noncircular pulley is much better than the mechanism in [34], which has a coupling rotation from -2° to +2° and has similar performance with that in [35], which has a maximum offset error about 0.29°. The results demonstrate that the implementation of the proposed noncircular pulley can effectively decouple the motion and is a viable solution to the coupling issue encountered in cable-driven robots. A video link² is also attached to show the performance of the noncircular pulley in motion decoupling.

5. Conclusion

In this paper, we propose a novel approach for motion decoupling of cable-driven serial robot by utilizing a noncircular pulley. With the noncircular pulley, the cable length increased on the lower joint pulley can be compensated by the decreased length on the noncircular pulley. The details of the calculation process for the noncircular pulley profile are introduced. In comparison to our previous study [32], we have improved the calculation process for designing the noncircular pulley profile by incorporating additional constraints and calculating the actual connecting point of the cable and the noncircular

² https://youtu.be/WbRvkwiHI_M

pulley. These improvements are aimed at enhancing the accuracy of the compensation provided by the noncircular pulley. An optimization process is described to find the optimal parameter set to minimize errors. A prototype is designed and experiments are conducted to evaluate the performance of the noncircular pulley in length compensation and motion decoupling, and the results indicate that the noncircular pulley successfully compensated for length changes with a maximum error of 0.086 mm and decoupled the motion with a maximum 0.67° rotation of Joint 2 during the 90-degree rotation of the lower joint. These findings demonstrate the effectiveness of the noncircular pulley method to address coupling problems in cable-driven serial robots. Compared with the mechanism in [26], which has a four-bar linkage and a moving pulley, the structure designed in this research is more compact and the routing is simpler by using the noncircular pulley. Besides, compared with the mechanisms in [34,35], which uses a fixed wheel, a following wheel, a driving wheel and/or gears, using the noncircular pulley proposed in our research can achieve more simplicity in routing as well as assembling.

To improve the current prototype, the future works would focus on increasing the motion range of the joints and designing a fully cable driven robot to further evaluating the motion decoupling performance of the noncircular pulley method. In addition, a more efficient optimization method will be explored to improve the optimization process.

ACKNOWLEDGMENT

This material is based upon work supported by the US National Science Foundation under Grant No. 2138903.

REFERENCES

- 625 [1] Tan, N., Clevy, C., Laurent, G. J., Sandoz, P., and Chaillet, N., 2015, "Accuracy
626 Quantification and Improvement of Serial Micropositioning Robots for In-Plane
627 Motions," *IEEE Transactions on Robotics*, 31(6), pp. 1497–1507.
- 628 [2] Jiang, Z., Zhou, W., Li, H., Mo, Y., Ni, W., and Huang, Q., 2018, "A New Kind of
629 Accurate Calibration Method for Robotic Kinematic Parameters Based on the
630 Extended Kalman and Particle Filter Algorithm," *IEEE Transactions on Industrial
631 Electronics*, 65(4), pp. 3337–3345.
- 632 [3] Li, Y., and Zhu, H., 2018, "A Simple Optimization Method for the Design of a
633 Lightweight, Explosion-Proof Housing for a Coal Mine Rescue Robot," *Journal of
634 the Brazilian Society of Mechanical Sciences and Engineering*, 40(7), pp. 1–10.
- 635 [4] Tsai, L.-W., 1999, *Robot Analysis: The Mechanics of Serial and Parallel
636 Manipulators*, John Wiley & Sons.
- 637 [5] Pandilov, Z., and Dukovski, V., 2014, "COMPARISON OF THE CHARACTERISTICS
638 BETWEEN SERIAL AND PARALLEL ROBOTS.," *Acta Technica Corviniensis-Bulletin
639 of Engineering*, 7(1).
- 640 [6] Slamani, M., and Chatelain, J.-F., 2015, "Issues and Challenges in Robotic
641 Trimming of CFRP," 2015 12th International Conference on Informatics in
642 Control, Automation and Robotics (ICINCO), IEEE, pp. 400–405.
- 643 [7] Sanan, S., Lynn, P. S., and Griffith, S. T., 2014, "Pneumatic Torsional Actuators for
644 Inflatable Robots," *J Mech Robot*, 6(3).
- 645 [8] Wu, G., and Shen, H., 2021, "Parallel PnP Robots," *Res Intell Manuf*, 7, pp. 87–
646 120.
- 647 [9] Gupta, V., Saha, S. K., and Chaudhary, H., 2019, "Optimum Design of Serial
648 Robots," *Journal of Mechanical Design*, 141(8), p. 082303.
- 649 [10] Lee, D., Park, J. W., and Seo, T., 2016, "Low-Inertia Serial Manipulator with
650 Counterbalance Mechanism," 2016 13th International Conference on Ubiquitous
651 Robots and Ambient Intelligence, URAI 2016, pp. 369–370.
- 652 [11] Agrawal, S. K., and Fattah, A., 2004, "Gravity-Balancing of Spatial Robotic
653 Manipulators," *Mech Mach Theory*, 39(12), pp. 1331–1344.
- 654 [12] Nguyen, V. L., 2022, "A Design Approach for Gravity Compensators Using Planar
655 Four-Bar Mechanisms and a Linear Spring," *Mech Mach Theory*, 172.
- 656 [13] Arakelian, V., and Briot, S., 2015, *Balancing of Linkages and Robot Manipulators:
657 Advanced Methods with Illustrative Examples*, Springer.
- 658 [14] Jamshidifar, H., Khajepour, A., Sun, T., Schmitz, N., Jalali, S., Topor-Gosman, R.,
659 and Dong, J., 2021, "A Novel Mechanism for Gravity-Balancing of Serial Robots
660 with High-Dexterity Applications," *IEEE Trans Med Robot Bionics*, 3(3), pp. 750–
661 761.

- 662 [15] Kolarski, M., Vukobratović, M., and Borovac, B., 1994, "Dynamic Analysis of
663 Balanced Robot Mechanisms," *Mech Mach Theory*, 29(3), pp. 427–454.
- 664 [16] Chowdhury, A. M. M. B., Cheng, J., Yu, D., and Shen, T., 2022, "Development of a
665 Self-Decoupled Wire-Driven Robotic Universal Joint Toward Medical
666 Application," *Proceedings of the 2022 Design of Medical Devices Conference*,
667 DMD 2022.
- 668 [17] Tzemanaki, A., Fracczak, L., Gillatt, D., Koupparis, A., Melhuish, C., Persad, R.,
669 Rowe, E., Pipe, A. G., and Dogramadzi, S., 2016, "Design of a Multi-DOF Cable-
670 Driven Mechanism of a Miniature Serial Manipulator for Robot-Assisted
671 Minimally Invasive Surgery," *Proceedings of the IEEE RAS and EMBS International
672 Conference on Biomedical Robotics and Biomechatronics*, 2016-July, pp. 55–60.
- 673 [18] Wang, Y., Cao, Q., Zhu, X., and Wang, P., 2021, "A Cable-Driven Distal End-Effector
674 Mechanism for Single-Port Robotic Surgery," *Int J Comput Assist Radiol Surg*,
675 16(2), pp. 301–309.
- 676 [19] Bryson, J. T., Jin, X., and Agrawal, S. K., 2016, "Optimal Design of Cable-Driven
677 Manipulators Using Particle Swarm Optimization," *J Mech Robot*, 8(4).
- 678 [20] Shen, T., Nelson, C. A., and Bradley, J., 2018, "Design of a Model-Free Cross-
679 Coupled Controller with Application to Robotic NOTES," *Journal of Intelligent &
680 Robotic Systems* 2018 95:2, 95(2), pp. 473–489.
- 681 [21] Mei, F., Yili, F., Bo, P., and Xudong, Z., 2012, "An Improved Surgical Instrument
682 without Coupled Motions That Can Be Used in Robotic-Assisted Minimally
683 Invasive Surgery," <http://dx.doi.org/10.1177/0954411912447145>, 226(8), pp.
684 623–630.
- 685 [22] Quigley, M., Asbeck, A., and Ng, A., 2011, "A Low-Cost Compliant 7-DOF Robotic
686 Manipulator," *Proc IEEE Int Conf Robot Autom*, pp. 6051–6058.
- 687 [23] Chen, W., Dong, L., Zhang, J., and Wu, X., 2007, "Tension Analysis for a Cable-
688 Driven 7-DOF Manipulator," *ICIEA 2007: 2007 Second IEEE Conference on
689 Industrial Electronics and Applications*, pp. 862–867.
- 690 [24] Chen, W., Chen, Q., Liu, R., and Zhang, J., 2012, "Homing Algorithm Analysis for a
691 Cable-Driven 3-DOF Shoulder Joint," *IEEE International Conference on Industrial
692 Informatics (INDIN)*, pp. 1170–1175.
- 693 [25] Sang, H., Meng, J., and Yun, J., 2011, "Kinematic Analysis of a Class of Multi-DOF
694 Tendon-Driven Minimally Invasive Surgical Instruments," *Proceedings of 2011
695 International Conference on Computer Science and Network Technology, ICCSNT
696 2011*, 2, pp. 607–612.
- 697 [26] Lee, J. K., Choi, C. H., Yoon, H., Lee, H. J., Park, S., and Yoon, J. S., Design and
698 Evaluation of Cable-Driven Manipulator with Motion-Decoupled Joints.

- [27] Liang, Y., Du, Z., Wang, W., Yan, Z., and Sun, L., 2019, "An Improved Scheme for Eliminating the Coupled Motion of Surgical Instruments Used in Laparoscopic Surgical Robots," *Rob Auton Syst*, 112, pp. 49–59.
- [28] Glachet C, G. J.-P., Pontchartrain, J., and Vertut, J., 1973, "Remote Manipulator," 817(3), p. 403.
- [29] Lee, J. K., Choi, C. H., Yoon, K. H., Park, B. S., and Yoon, J. S., 2008, "Design of a Servomanipulator with Tendon Transmission," 2008 International Conference on Control, Automation and Systems, ICCAS 2008, pp. 1653–1656.
- [30] Rodriguez-Cianca, D., Verstraten, T., Rodriguez-Guerrero, C., Jimenez-Fabian, R., Naef, M. B., Vanderborght, B., and Lefeber, D., 2020, "The Two-Degree-of-Freedom Cable Pulley (2DCP) Transmission System: An under-Actuated and Motion Decoupled Transmission for Robotic Applications," *Mech Mach Theory*, 148, p. 103765.
- [31] Fedorov, D., and Birglen, L., 2018, "Differential Noncircular Pulleys for Cable Robots and Static Balancing," *J Mech Robot*, 10(6), p. 061001.
- [32] Cheng, J., and Shen, T., 2023, "Motion Decoupling for Cable-Driven Serial Robots Based on a Noncircular Pulley," 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, pp. 739–745.
- [33] Robinson, A. C., and Quinn, S. D., 2018, "A Brute Force Method for Spatially-Enhanced Multivariate Facet Analysis," *Comput Environ Urban Syst*, 69, pp. 28–38.
- [34] Jiang, S., Hua, D., Wang, Y., Ju, F., Yin, L., and Chen, B., 2018, "Design and Modeling of Motion-Decoupling Mechanism for Cable-Driven Joints," <https://doi.org/10.1177/1687814018777428>, 10(5), pp. 1–10.
- [35] Jiang, S., Wang, Y., Li, B., Chen, B., and Hua, D., 2019, "Design and Verification of a Novel Motion-Decoupled Cable-Driven Manipulator," <https://doi.org/10.1177/0959651819885205>, 234(6), pp. 690–700.

729
730

Figure Captions List

- Fig. 1 Motion coupling issue in cable-driven serial robots
- Fig. 2 Fig. 2: (a) Concept design of the robotic link (b) The prototype of the robotic link
- Fig. 3 The profile of the noncircular pulley
- Fig. 4 Fig. 4: (a) Cable routing at the initial position (b) Cable routing after rotation
- Fig. 5 *Point Q_n* calculation
- Fig. 6 Arc length calculation
- Fig. 7 Upper and lower limits
- Fig. 8 Upper constraint of *Point P_n*
- Fig. 9 Optimization process
- Fig. 10 The flowchart of each optimization step based on brute force method
- Fig. 11 The relationship between l_{oc} and e_{max}
- Fig. 12 The relationship between dc and e_{max}
- Fig. 13 The profile of the noncircular pulley with the optimal parameters
- Fig. 14 Error during motion
- Fig. 15 Length compensation experiment setup
- Fig. 16 Control system of the robotic link
- Fig. 17 Length changes during motion

Fig. 18 Motion decoupling experiment setup

Fig. 19 Results of motion decoupling experiment

731

732

733
734

Table Caption List

Table 1	Parameters of wide range exploration
Table 2	Parameters of narrow-down refinement process
Table 3	Parameters of fine-range optimization process

735
736