The Team

Registration Number: Phy-102

Name of team member: FONG Su
School: Pui Ching Middle School Macau
City, Country: Macau

Name of supervising teacher: MIN Hao Nian
Job Title: Teacher
School/Institution: Pui Ching Middel School Macau
City, Country: Macau

Title of Research Report

An Analysis and Optimization of Double Parallelogram Lifting Mechanism

Date

31 August 2021

Commitments on Academic Honesty and Integrity

We hereby declare that we

1. are fully committed to the principle of honesty, integrity and fair play throughout the competition.
2. actually perform the research work ourselves and thus truly understand the content of the work.
3. observe the common standard of academic integrity adopted by most journals and degree theses.
4. have declared all the assistance and contribution we have received from any personnel, agency, institution, etc. for the research work.
5. undertake to avoid getting in touch with assessment panel members in a way that may lead to direct or indirect conflict of interest.
6. undertake to avoid any interaction with assessment panel members that would undermine the neutrality of the panel member and fairness of the assessment process.
7. observe the safety regulations of the laboratory(ies) where the we conduct the experiment(s), if applicable.
8. observe all rules and regulations of the competition.
9. agree that the decision of YHSA(Asia) is final in all matters related to the competition.

We understand and agree that failure to honour the above commitments may lead to disqualification from the competition and/or removal of reward, if applicable; that any unethical deeds, if found, will be disclosed to the school principal of team member(s) and relevant parties if deemed necessary; and that the decision of YHSA(Asia) is final and no appeal will be accepted.

(Signatures of full team below)

X 方

Name of team member: FONG Su

X 闵浩年

Name of supervising teacher: MIN Hao Nian

Noted and endorsed by

X 高錦輝

(signature)

Name of school principal: KOU KAM FAI,
Pui Ching Middle School Macau

An Analysis and Optimization of Double Parallelogram Lifting Mechanism

Author: FONG Su

Abstract

Double Parallelogram Lifting Mechanism (DPLM) is a compact and stable lifting mechanism with a large extension range widely adopted in robot designs. Rubber bands and springs are often installed on the DPLM to lighten the motors' load and maintain its height, yet the installation positions are often obtained through trial and error. This project aims at finding the optimal rubber band installation positions for DPLM using modeling and optimization techniques.

A mathematical model which describes the forces and moments acting on all the linkages of DPLM was derived based on the conditions for the static equilibrium and verified with a 3D simulation software. A genetic algorithm (GA) was implemented to optimize rubber band installation positions, which managed to find solutions with the overall root-mean-square-error (RMSE) of the net moment less than 2 for 2 to 6 rubber bands. A further statistical analysis of 50000 random rubber band samples showed that installing rubber bands in triangles is the best solution with the overall lowest RMSE. A test was conducted with a prototype of the DPLM and the results were consistent with our model and optimization.

This project derived and verified a mathematical model for the DPLM, and found the optimal way and positions to install rubber bands. The results of this project provides a theoretical basis for controlling DPLM with rubber bands, allowing it to be further adopted in industrial robots that require repetitive lifting and lowering such as inspection robots and aerial work platforms.

Keywords:
Linkage, Parallelogram mechanism, Double parallelogram, Lifting mechanism, Optimization, Elastic material

Table of Contents

1. Introduction

Double Parallelogram Mechanism (DPM) is a compact and stable multi-linkage mechanism. This mechanism is widely adopted in both heavy-load and light-load situations because of its light weight and its large range of extension. Heavy-load applications of DPM mainly include lifting mechanisms and aerial work platforms [1], in which each parallelogram is driven independently by a hydraulic cylinder in the lifting process. DPM is also used in forklifts[2] in which it is driven by only one hydraulic cylinder, moving both parallelograms simultaneously. This mechanism is also used in the design of stack parking systems, in which the whole mechanism can be driven by a single motor to realize three motion periods, including lifting, translation, and fillet transition of vehicles[3].

Applications of DPM in light-load situations include Remote Center of Motion (RCM) Mechanisms for surgical operations[4], exoskeletons[5][6], and force-reflective robots[7]. Some novel studies also investigated the kinetic energy transmitted through small deformations of DPM.[8][9]

In this paper, a robot is designed with DPM as its lifting mechanism (Figure 1). Therefore, this mechanism will be called a Double Parallelogram Lifting Mechanism (DPLM) in the rest of this paper. The DPLM enables the robot to elevate along a substantially vertical path at a high speed only with a single motor installed on the mechanism, while having a great working height without increasing the retracted height. The load platform of the DPLM is maintained horizontal throughout the whole lifting process, making it desirable to be used as a platform for inspection purposes. For example, when the lifting apparatus is used in inspecting large vehicles, such as aircraft and water vessels, inspection devices can be installed on the load platform.[10]

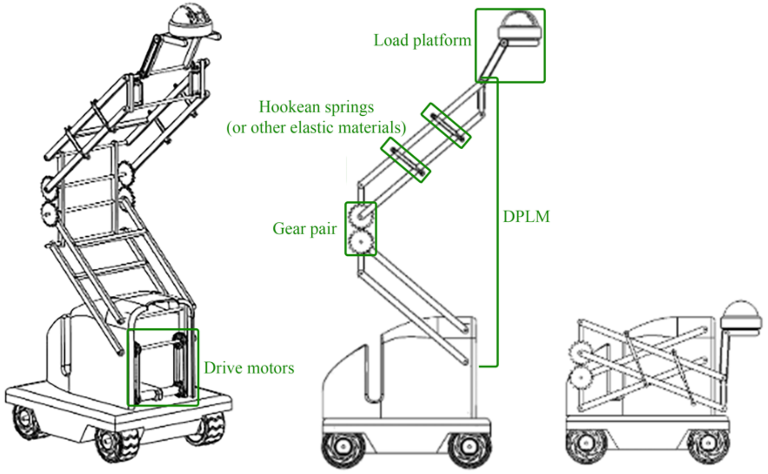


Figure 1. The designed robot with a DPLM as its lifting mechanism.

Double Parallelogram Lifting Mechanism (DPLM) is composed of two four-bar linkages connected with a pair of meshing gears, which enable the two linkages to elevate simultaneously at high speed with only one drive motor. However, if the motion of the whole DPLM depends on only one motor, a motor with very high power will be needed, which will highly increase the weight and the cost of the DPLM. In addition, the motor is also more prone to overloading and fatigue in such a situation. Therefore, in order to reduce the load of the drive motor, Rubber bands, and other elastic materials, are often installed on the parallelogram pair to resist the gravitational moment due to the weight of the DPLM.

However, the optimal installation positions and the parameters of the Rubber bands are currently obtained through testings. Therefore, in this paper, the method of using the elastic force of Rubber band to maintain DPLM in static equilibrium at any height is studied. Numerical simulation is used in optimizing the parameters of Rubber band as well as their installation positions.

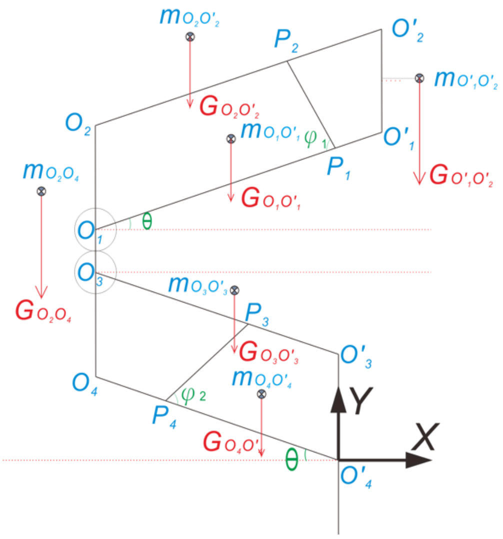


Figure 2. DPLM with a positive rotation angle(θ)

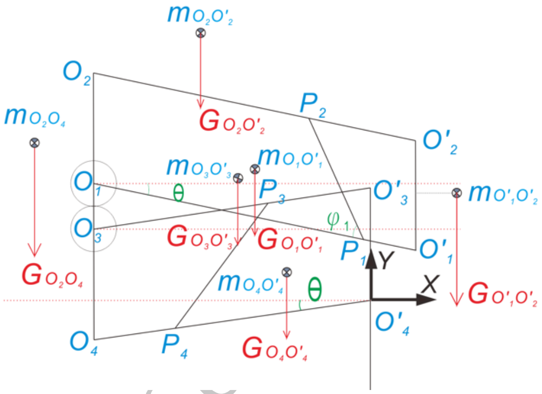


Figure 3. DPLM with a negative rotation angle(θ)

2. Objectives

The objectives of this section are as following:

1. Model DPLM with math equations,
2. Verify that the model is correct,
3. Optimize the installation positions and the parameters of the elastic material with given apparatus parameters (length and mass of linkages)
4. Find a way to minimize parameter changes when the load on the apparatus changes.

3. Description of DPLM

DPLM, as shown in Figure 2 and Figure 3, is composed of the lower four-bar linkage $O_3O'_3O'_4O_4$ and the upper four-bar linkage $O_1\ O_2\ O'_2O'_1$. The lower four-bar linkage is fixed to the base through the hinge at O'_4, while the upper four-bar linkage is connected to the lower four-bar linkage through a pair of meshing gears. Since the upper and lower parallelograms can move simultaneously, the drive actuator can be installed at any hinge of O_1 , O_2 , O_3 , O_4 , O'_3, O'_4.

$P_1\ P_2$ and $P_3\ P_4$ are Rubber bands installed on the upper or lower four-bar linkage to resist the gravitational moment. The Rubber bands can be ordinary springs, rubber bands, or gas springs.

4. Calculation model

A coordinate system is established with O'_4 as the origin as shown in Figure 2 and Figure 3. The angle between the linkage $O_4O'_4$ and the negative X-axis is θ.

4.1. Naming

- Centroids:
  - m (e.g. $m_{O_1O_2}$ represents the mass of the linkage O_1O_2)
- Forces:
  - H (e.g. H_{O_1} represents the horizontal force at pivot O_1)

  - ○ V (e.g. V_{O_1} represents the vertical force at pivot O_1)
  - ○ f (e.g. f_1 represents the tension exerted by a spring)
- Length:
  - ○ L (e.g. $L_{O_1O_2}$ represents the length of linkage O_1O_2)
  - ○ X (e.g. $X_{O_1O_1'}$ represents the horizontal distance between the centroid of linkage O_1O_1' and its rotational axis)

4.2. Force analysis

Each part of DPLM is analyzed separately, and an equation describing all the factors involved in moment balancing is established at the end according to the principles of theoretical mechanics.[11]

The free body diagrams(FBD) of all linkages in the DPLM are shown below in Figure 4 to 9:

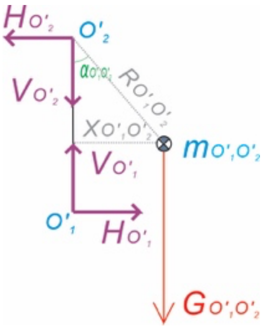


Figure 4. FBD of linkage $O_1^{'}O_2^{'}$.

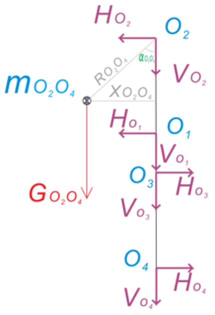


Figure 5. FBD of linkage O_2O_4.

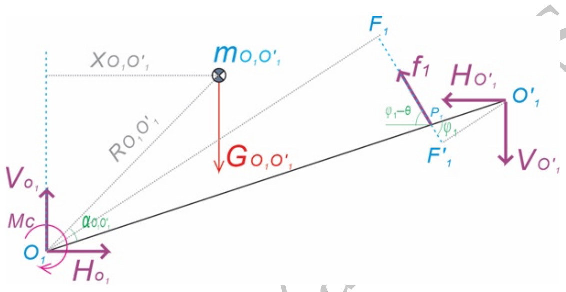


Figure 6. FBD of linkage O_1O_1'.

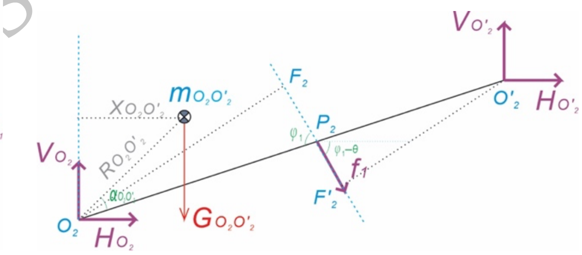


Figure 7. FBD of linkage O_2O_2'.

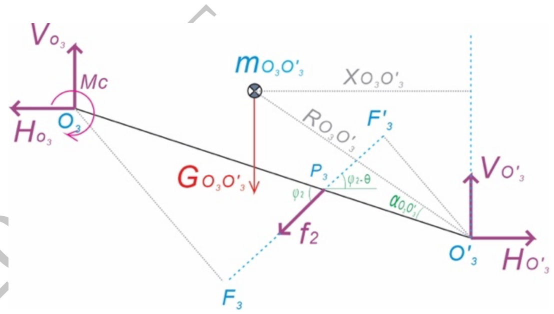


Figure 8. FBD of linkage O_3O_3'.

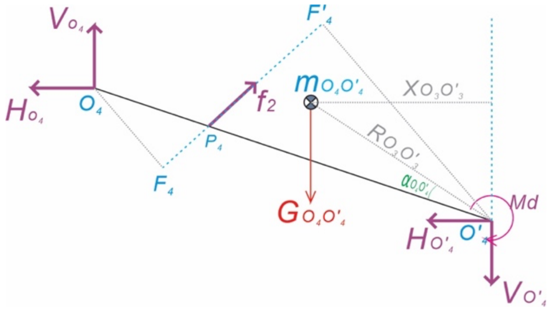


Figure 9. FBD of linkage O_4O_4'.

4.2.1. Linkage $O_1^{'}O_2^{'}$

By the conditions for equilibrium of rigid bodies, as shown in Figure 4, three balance equations (1)(2)(3) are established.

$$\sum H_x = H_{O_1^{'}} - H_{O_2^{'}} = 0 \tag{1}$$

$$\sum V_y = V_{O_1'} - G_{O_1'O_2'} - V_{O_2'} = 0 \tag{2}$$

$$\sum M_{O_2'} = H_{O_1'} \cdot L_{O_1'O_2'} - G_{O_1'O_2'} \cdot X_{O_1'O_2'} \tag{3}$$

The expressions of $H_{O_1'}$ and $H_{O_2'}$ are derived as in formula (4) by substituting formula (3) into formula (1).

$$\frac{G_{O_1'O_2'} \cdot X_{O_1'O_2'}}{L_{O_1'O_2'}} = H_{O_1'} = H_{O_2'} \tag{4}$$

4.2.2. Linkage O_2O_2'

By the conditions for equilibrium of rigid bodies, as shown in Figure 7, three balance equations (5)(6)(7) are derived.

$$\sum V_y = V_{O_2} + V_{O_2'} - f_1 \cdot sin(\phi - \theta) - G_{O_2O_2'} = 0 \tag{5}$$

$$\sum H_{O_2} = -f_1 \cdot cos(\varphi - \theta) - H_{O_2'} \tag{6}$$

$$\sum M_{O_2} = -f_1 \cdot L_{O_2F_2} - G_{O_2O_2'} \cdot X_{O_2O_2'} \cdot cos\,\theta - H_{O_2'} \cdot L_{O_2O_2'} \cdot sin\,\theta + V_{O_2'} \cdot L_{O_2O_2'} \cdot cos\,\theta = 0 \tag{7}$$

4.2.3. Linkage $O_1\ O_1'$

By the conditions for equilibrium of rigid bodies, as shown in Figure 6, three balance equations (8)(9)(10) are derived.

$$\sum V_y = V_{O_1} - V_{O_1'} + f_1 \cdot sin(\phi - \theta) - G_{O_1O_1'} = 0 \tag{8}$$

$$\sum H_{O_1} = f_1 \cdot cos(\phi - \theta) + H_{O_1'} \tag{9}$$

$$\sum M_{O_1} = f_1 \cdot L_{O_1F_1} - G_{O_1O_1'} \cdot X_{O_1O_1'} \cdot cos\,\theta + H_{O_1'} \cdot L_{O_1O_1'} \cdot sin\,\theta - V_{O_1'} \cdot L_{O_1O_1'} \cdot cos\,\theta - M_c = 0 \tag{10}$$

4.2.4. Linkage $O_2\ O_4$

By the conditions for equilibrium of rigid bodies, as shown in Figure 5, two balance equations (11)(12) are derived.

$$\sum H_x = -H_{O_1} - H_{O_2} + H_{O_3} + H_{O_4} = 0 \tag{11}$$

$$\sum V_y = -V_{O_1} - V_{O_2} - G_{O_2O_4} - V_{O_3} - V_{O_4} = 0 \tag{12}$$

4.2.5. Linkage $O_3\ O_3'$

By the conditions for equilibrium of rigid bodies, as shown in Figure 8, a balance equation (13) is derived.

$$\sum M_{O_3'} = f_2 \cdot L_{O_3'F_3'} + G_{O_3'O_3} \cdot X_{O_3O_3'} \cdot cos\,\theta + H_{O_3} \cdot L_{O_3'O_3} \cdot sin\,\theta - V_{O_3} \cdot L_{O_3'O_3} \cdot cos\,\theta - M_c = 0 \tag{13}$$

4.2.6. Linkage $O_4\ O_4'$

By the conditions for equilibrium of rigid bodies, as shown in Figure 9, a balance equation (14) is derived.

$$\sum M_{O_4'} = -f_2 \cdot L_{O_4'F_4'} + G_{O_4'O_4} \cdot X_{O_4O_4'} \cdot cos\,\theta + H_{O_4} \cdot L_{O_4'O_4} \cdot sin\,\theta - V_{O_4} \cdot L_{O_4'O_4} \cdot cos\,\theta + M_d = 0 \tag{14}$$

The M_d in formula (14) is the moment exerted by the actuator, which is zero when the actuator is not in use.

4.2.7. Finding tensions

In order to obtain the relation between tension f_1 and f_2 , by adding formula (7) and formula (13), then substituting formula (4) into the equation, we further derive formula (15):

$$f_1 \cdot (L_{O_1F_1} - L_{O_2F_2}) - G_{O_2O_2'} \cdot X_{O_2O_2'} - G_{O_1O_1'} \cdot X_{O_1O_1'} + \left(H_{O_1'} - H_{O_2'}\right) \cdot L_{O_2O_2'} \cdot sin\theta + \left(V_{O_2'} - V_{O_1'}\right) \cdot L_{O_1O_1'} \cdot cos\,\theta$$
$$- M_c = 0$$

$$f_1\cdot\left(L_{O_1F_1}-L_{O_2F_2}\right)-G_{O_2O_2'}\cdot X_{O_2O_2'}-G_{O_1O_1'}\cdot X_{O_1O_1'}-G_{O_1'O_2'}\cdot L_{O_1O_1'}\cdot\cos\theta-M_c=0 \tag{15}$$

By adding formula (10) and formula (14) we obtain the temporary equation below：

$$f_2\cdot L_{O_3'F_3'}-f_2\cdot L_{O_4'F_4'}+G_{O_3'O_3}\cdot X_{O_3O_3'}\cdot\cos\theta+G_{O_4'O_4}\cdot X_{O_4O_4'}\cdot\cos\theta+(H_{O_3}+H_{O_4})\cdot L_{O_3'O_3}\cdot\sin\theta$$
$$-\left(V_{O_3}+V_{O_4}\right)\cdot L_{O_3'O_3}\cdot\cos\theta-M_c+M_d=0$$

Then by substituting formulae (5)(6)(8)(9)(11) into the temporary equation, after further simplifications we obtain formula (16)：

$$f_2\cdot\left(L_{O_3'F_3'}-L_{O_4'F_4'}\right)+G_{O_3'O_3}\cdot X_{O_3O_3'}+G_{O_4'O_4}\cdot X_{O_4O_4'}+\left(G_{O_1'O_2'}+G_{O_2O_2'}+G_{O_1O_1'}+G_{O_2O_4}\right)\cdot L_{O_3'O_3}\cdot\cos\theta$$
$$-M_c+M_d=0 \tag{16}$$

4.2.8. Establishing the relationship between f_1 and f_2

Subtracting (15) from (16), M_C will be eliminated and we obtain (17):

$$f_1\cdot\left(L_{O_1F_1}-L_{O_2F_2}\right)+f_2\cdot\left(L_{O_4'F_4'}-L_{O_3'F_3'}\right)-G_{O_2O_2'}\cdot X_{O_2O_2'}-G_{O_1O_1'}\cdot X_{O_1O_1'}-G_{O_1'O_2'}\cdot L_{O_1O_1'}\cdot\cos\theta-G_{O_3'O_3}\cdot X_{O_3O_3'}$$
$$-G_{O_4'O_4}\cdot X_{O_4O_4'}-\left(G_{O_1'O_2'}+G_{O_2O_2'}+G_{O_1O_1'}+G_{O_2O_4}\right)\cdot L_{O_3'O_3}\cdot\cos\theta+M_d=0 \tag{17}$$

Notice that when tension f_1 and f_2 in formula (17) are substituted with arrays of tensions of multiple Rubber bands, the derivation of (17) isn't affected. By this we obtain (18) :

$$\sum_i f_{1i}\cdot\sin\phi_i\cdot\left(L_{O_{1i}P_{1i}}-L_{O_{2i}P_{2i}}\right)+\sum_j f_{2j}\cdot\sin\phi_{2j}\cdot\left(L_{O_{4j}'P_{4j}}-L_{O_{3j}'P_{3j}}\right)-G_{O_2O_2'}\cdot X_{O_2O_2'}G_{O_1O_1'}\cdot X_{O_1O_1'}$$
$$-G_{O_1'O_2'}\cdot L_{O_1O_1'}\cdot\cos\theta-G_{O_3'O_3}\cdot X_{O_3O_3'}-G_{O_4'O_4}\cdot X_{O_4O_4'}$$
$$-\left(G_{O_1'O_2'}+G_{O_2O_2'}+G_{O_1O_1'}+G_{O_2O_4}\right)\cdot L_{O_3'O_3}\cdot\cos\theta+M_d=0 \tag{18}$$

The $L_{O_{1i}P_{1i}}-L_{O_{2i}P_{2i}}$ and $L_{O_{4j}'P_{4j}}-L_{O_{3j}'P_{3j}}$ in formula (18), which are the differences in the distances between the two installation points of a spring and their respective rotation axes, are named respectively D_i and D_j.

Formula (18) also suggests that the magnitude of the moment of the tension of a spring can be determined not only by its exact installation position but also by its D value. The absence of $X_{O_1'O_2'}$ and $X_{O_2O_4'}$ in formula (18) also suggests that the gravitational moment of the whole DPLM is independent of the position of the center of mass of linkages $O_1'O_2'$ and O_2O_4.

5. Model Verification

Here the validity of the proposed model in (18) is checked by comparing

1. the gravitational moment at O_4' at any angle without installation of any Rubber band on the mechanism, and
2. the tension required for equilibrium with different installation positions of spring at multiple angles

obtained in the MATLAB model (Figure 11) with those obtained in experiments conducted with a virtual prototype in SOLIDWORKS (SW).

A virtual prototype of DPLM, as shown in Figure 10, is set up in SW and the position of linkage $O_3'O_4'$ is fixed. The parameters of the prototype are shown in Table 1.[12]

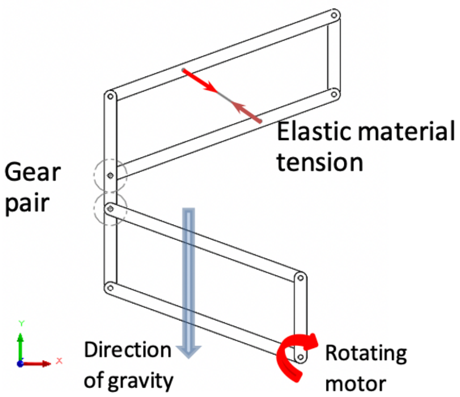


Figure 10. A virtual prototype of DPLM in SW

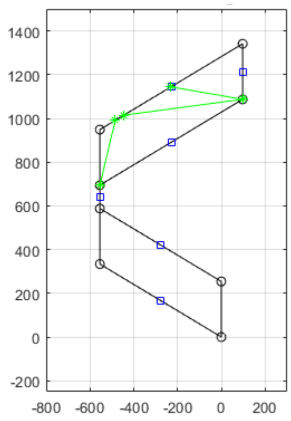


Figure 11. A MATLAB model of DPLM

Table 1.Parameters of the SW DPLM prototype.

Linkage	Length(mm)	Mass(g)
O_1O_1', O_2O_2'	762	847.34661
O'_1O_2'	254	298.70661
O_2O_4	615	679.04403
O_3O_3', O_4O_4'	648	724.22661

5.1. Validating the moment at O_4'

An actuator is added at O_4' in the SW virtual prototype and it is made to rotate at a constant angular velocity with θ within the range [-20,60], causing the whole mechanism to rise. Then, the moment exerted by the actuator with respect to θ, which by formula (18) is equal to the sum of gravitational moment of all linkages, is exported from SW. This exported moment is compared with the gravitational moment calculated in MATLAB.

The results of the SW simulation and MATLAB calculation are plotted in Figure 12, with the root-mean-square error(RMSE) between the two results being only 8.7051. This has shown that the MATLAB model (18) is completely in accordance with the SW simulation results

Note that no Rubber band is installed on the DPLM in this simulation.

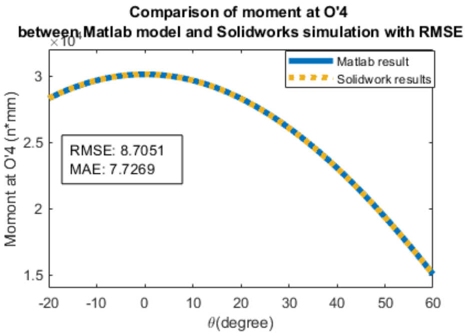


Figure 12. A comparison between the results of MATLAB and SW on the moment at O_4'.

5.2. Validating the tension required for the DPLM prototype to obtain equilibrium with different installation positions of spring.

To show that the tension part of the model (18) is in accordance with the SW simulation, multiple pairs of (D, θ) are selected randomly and the respective tensions required for equilibrium are calculated in MATLAB. These forces are then used as anchor points in finding the force required for equilibrium in the SW simulation. The tested equilibrium point(at which the mechanism stays in equilibrium), the tested falling point(at which the mechanism falls), and the respective ranges of error between the MATLAB model and the SW simulation results are presented in Table 2.

Table 2. Results from SW simulation

D	θ	Required force calculated in MATLAB model(N)	Tested equilibrium point from SW(N)	Tested falling point from SW(N)	Range of error between MATLAB SW results (N)
260	60	60.7905	60.70	60.68	[+0.0905,+0.1105]
260	40	99.2001	99.05	99.02	[+0.1501,+0.1801]
260	20	134.6107	134.37	134.35	[+0.2407,+0.2607]
260	0	165.9367	165.70	165.60	[+0.2367,+0.3367]
260	-20	192.2237	191.90	191.85	[+0.3237, +0.2237]
100	60	207.3691	207.37	207.36	[-0.0009,+0.0091]
100	40	242.8567	242.86	242.85	[-0.0033,+0.0067]
100	20	283.7354	283.74	283.73	[-0.0046,+0.0054]
100	0	324.0134	324.02	324.01	[-0.0066,+0.0034]
100	-20	359.8104	359.82	359.81	[-0.0096,+0.0004]

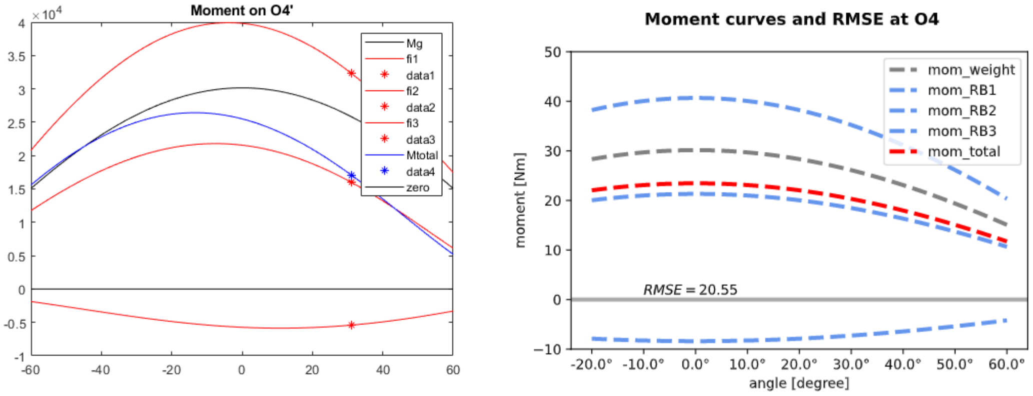
The table shows that the error between our MATLAB model and the SOLIDWORKS simulation is less than +0.4N under all tested (D, θ) pairs.

The two experiments show that our model is in accordance with the SolidWorks simulation and is a legitimate approximation to the physical world.

6. Optimization

6.1. Definition of fitness

Before proceeding to any optimization processes, the fitness of our model must be defined. If we visualize the function we aim to minimize, the proper definition of fitness would become very intuitive:

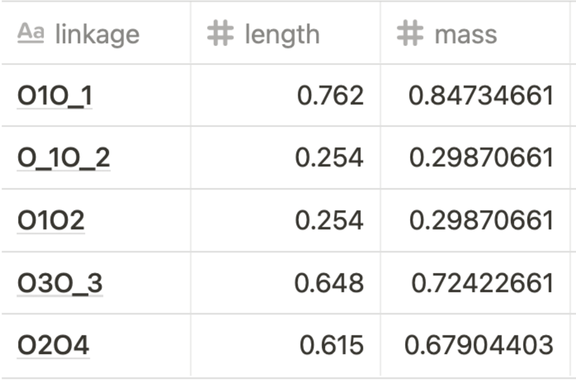


As in the figure above, the three red curves represent the moment produced by the three springs installed on the DPLM and the black curve represents the moment produced by its own weight. Subtracting the black curve from the sum of the red ones gives the blue curve, which is the net moment, the net tendency for the DPLM to rotate. Our aim, to achieve equilibrium across the whole angle range, means making the blue curve as close to the horizontal zero as possible, so that the moment of springs completely cancels out the moment of weight. To quantify this goal, the root-mean-square error (RMSE) between the net moment and zero across the range of angles is adopted to be the estimate of how well the DPLM is balancing. That is, the fitness function of this problem is defined to be the RMSE. The smaller it is, the better:

$$fitness = RMSE = \frac{\sum(net\ moment - 0)^2}{size\ of\ the\ range\ of\ angles}$$

6.2. DPLM Model setup

The parameters of the DPLM model used in the following optimization were obtained from a former FRC robot our school team had. (PHOTO) The parameters are as follow:

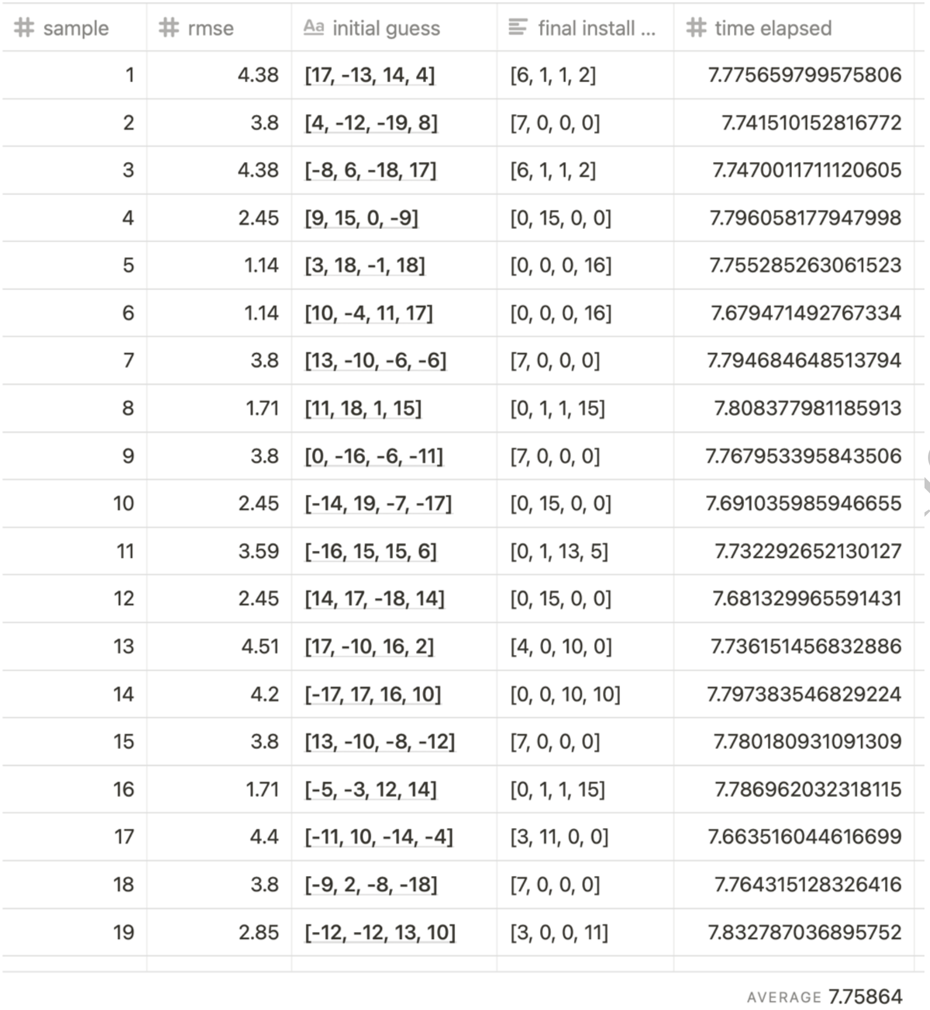


Aa linkage	# length	# mass
O1O_1	0.762	0.84734661
O_1O_2	0.254	0.29870661
O1O2	0.254	0.29870661
O3O_3	0.648	0.72422661
O2O4	0.615	0.67904403

The range of angle the optimization would apply to is $\theta \in [-20^\circ, 60^\circ]$.

6.3. Initial method: greedy

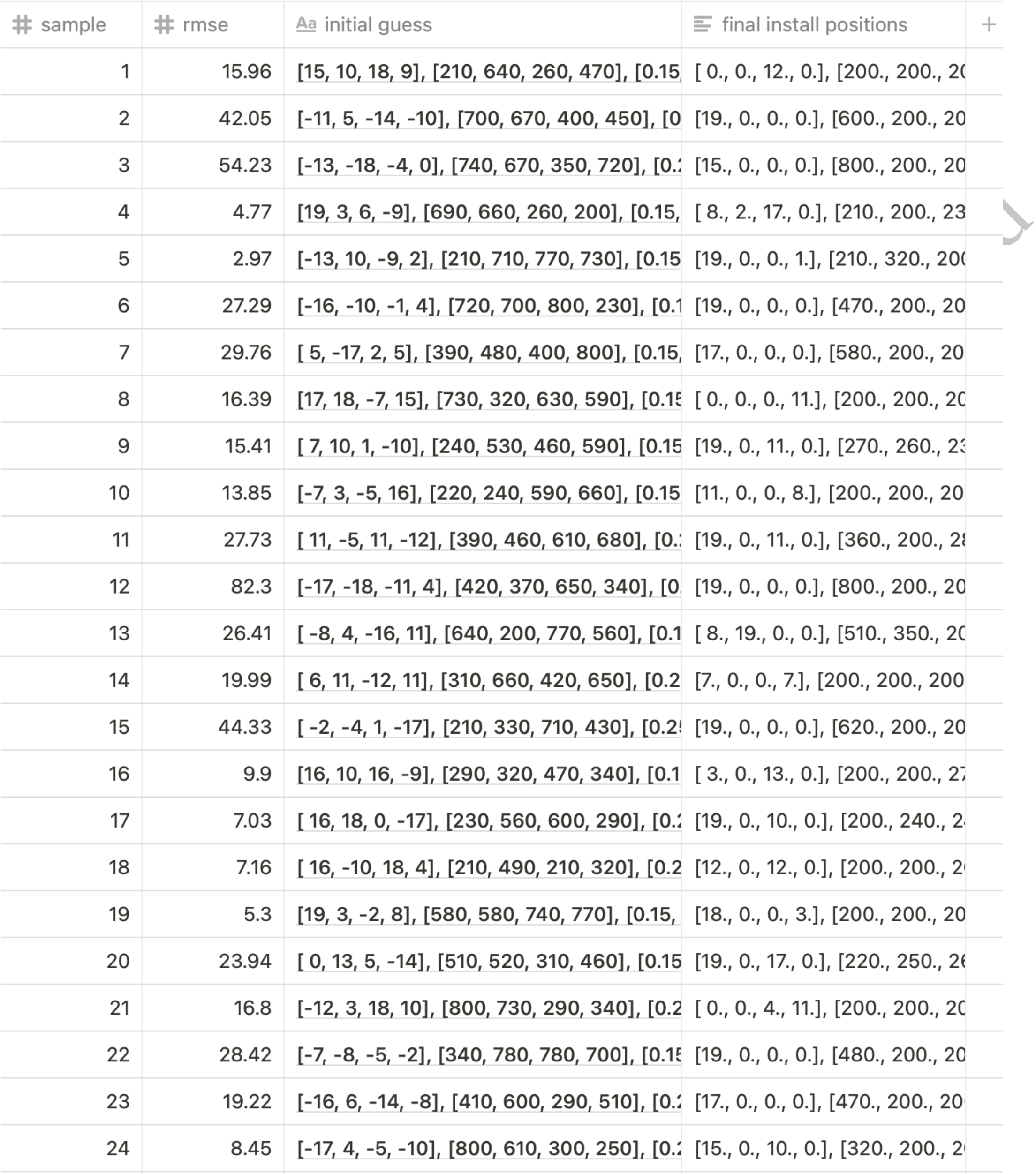
Our first approach (guess) to solve this optimization problem is using the greedy algorithm. For a given set of spring parameters, we ran the algorithm 100 times, 4 loops each time for four springs with random initial rubber band install positions to search for the optimal rubber band install positions. The results data and a random selection of the produced moment graphs are shown below:



# sample	# rmse	initial guess	final install ...	# time elapsed
1	4.38	[17, -13, 14, 4]	[6, 1, 1, 2]	7.775659799575806
2	3.8	[4, -12, -19, 8]	[7, 0, 0, 0]	7.741510152816772
3	4.38	[-8, 6, -18, 17]	[6, 1, 1, 2]	7.7470011711120605
4	2.45	[9, 15, 0, -9]	[0, 15, 0, 0]	7.796058177947998
5	1.14	[3, 18, -1, 18]	[0, 0, 0, 16]	7.755285263061523
6	1.14	[10, -4, 11, 17]	[0, 0, 0, 16]	7.679471492767334
7	3.8	[13, -10, -6, -6]	[7, 0, 0, 0]	7.794684648513794
8	1.71	[11, 18, 1, 15]	[0, 1, 1, 15]	7.808377981185913
9	3.8	[0, -16, -6, -11]	[7, 0, 0, 0]	7.767953395843506
10	2.45	[-14, 19, -7, -17]	[0, 15, 0, 0]	7.691035985946655
11	3.59	[-16, 15, 15, 6]	[0, 1, 13, 5]	7.732292652130127
12	2.45	[14, 17, -18, 14]	[0, 15, 0, 0]	7.681329965591431
13	4.51	[17, -10, 16, 2]	[4, 0, 10, 0]	7.736151456832886
14	4.2	[-17, 17, 16, 10]	[0, 0, 10, 10]	7.797383546829224
15	3.8	[13, -10, -8, -12]	[7, 0, 0, 0]	7.780180931091309
16	1.71	[-5, -3, 12, 14]	[0, 1, 1, 15]	7.786962032318115
17	4.4	[-11, 10, -14, -4]	[3, 11, 0, 0]	7.663516044616699
18	3.8	[-9, 2, -8, -18]	[7, 0, 0, 0]	7.764315128326416
19	2.85	[-12, -12, 13, 10]	[3, 0, 0, 11]	7.832787036895752
				AVERAGE 7.75864

The graphs and the data show that the greedy algorithm converges well to many different reasonable solutions with acceptable RMSE when only optimizing the install position of the rubber bands. Yet the algorithm fails to converge to one single optimal solution. The average time of each optimization is 7.76s.

Afterward, we try to optimize the install positions, length, and spring rate of the rubber bands together with greedy. The results data and a random selection of the produced moment graphs are shown below:



# sample	# rmse	initial guess	final install positions
1	15.96	[15, 10, 18, 9], [210, 640, 260, 470], [0.15	[0., 0., 12., 0.], [200., 200., 2
2	42.05	[-11, 5, -14, -10], [700, 670, 400, 450], [0	[19., 0., 0., 0.], [600., 200., 20
3	54.23	[-13, -18, -4, 0], [740, 670, 350, 720], [0.	[15., 0., 0., 0.], [800., 200., 20
4	4.77	[19, 3, 6, -9], [690, 660, 260, 200], [0.15,	[8., 2., 17., 0.], [210., 200., 23
5	2.97	[-13, 10, -9, 2], [210, 710, 770, 730], [0.15	[19., 0., 0., 1.], [210., 320., 20
6	27.29	[-16, -10, -1, 4], [720, 700, 800, 230], [0.1	[19., 0., 0., 0.], [470., 200., 20
7	29.76	[5, -17, 2, 5], [390, 480, 400, 800], [0.15,	[17., 0., 0., 0.], [580., 200., 20
8	16.39	[17, 18, -7, 15], [730, 320, 630, 590], [0.1	[0., 0., 0., 11.], [200., 200., 2
9	15.41	[7, 10, 1, -10], [240, 530, 460, 590], [0.15	[19., 0., 11., 0.], [270., 260., 2
10	13.85	[-7, 3, -5, 16], [220, 240, 590, 660], [0.15	[11., 0., 0., 8.], [200., 200., 20
11	27.73	[11, -5, 11, -12], [390, 460, 610, 680], [0.	[19., 0., 11., 0.], [360., 200., 2
12	82.3	[-17, -18, -11, 4], [420, 370, 650, 340], [0	[19., 0., 0., 0.], [800., 200., 20
13	26.41	[-8, 4, -16, 11], [640, 200, 770, 560], [0.1	[8., 19., 0., 0.], [510., 350., 2
14	19.99	[6, 11, -12, 11], [310, 660, 420, 650], [0.2	[7., 0., 0., 7.], [200., 200., 200
15	44.33	[-2, -4, 1, -17], [210, 330, 710, 430], [0.2	[19., 0., 0., 0.], [620., 200., 20
16	9.9	[16, 10, 16, -9], [290, 320, 470, 340], [0.1	[3., 0., 13., 0.], [200., 200., 2
17	7.03	[16, 18, 0, -17], [230, 560, 600, 290], [0.	[19., 0., 10., 0.], [200., 240., 2
18	7.16	[16, -10, 18, 4], [210, 490, 210, 320], [0.2	[12., 0., 12., 0.], [200., 200., 2
19	5.3	[19, 3, -2, 8], [580, 580, 740, 770], [0.15,	[18., 0., 0., 3.], [200., 200., 20
20	23.94	[0, 13, 5, -14], [510, 520, 310, 460], [0.15	[19., 0., 17., 0.], [220., 250., 2
21	16.8	[-12, 3, 18, 10], [800, 730, 290, 340], [0.2	[0., 0., 4., 11.], [200., 200., 2
22	28.42	[-7, -8, -5, -2], [340, 780, 780, 700], [0.1	[19., 0., 0., 0.], [480., 200., 20
23	19.22	[-16, 6, -14, -8], [410, 600, 290, 510], [0.	[17., 0., 0., 0.], [470., 200., 20
24	8.45	[-17, 4, -5, -10], [800, 610, 300, 250], [0.	[15., 0., 10., 0.], [320., 200., 2

As shown in the graphs, the greedy algorithm responded with dramatically different results for different inputs. This showed that the equation we were trying to optimize was highly susceptible to local minima and therefore any traditional gradient-based optimization methods might not be suitable.

The two above results showed that although the greedy algorithm could handle the problem when the number of parameters to be optimized was small, it failed to produce meaningful output when more parameters were to be optimized. Therefore, we would have to find other methods for the optimization problem to search for the optimal rubber band parameters.

6.3.1. After greedy

Realizing derivative-based optimization is not very likely to work on our problem due to the multi-modal nature of our problem, we started to search for some derivative-free optimization method. After doing some research, we decided to try using reinforcement learning and genetic algorithm to optimize our equation. After discussing with our teammates, we decided to do reinforcement learning first since if both of them worked, reinforcement learning would produce a model that encompassed a large range of rubber band parameters that could be later reused while genetic algorithm could only find a solution (a set of solutions) for a particular rubber band setting.

6.4. Reinforcement learning

To use reinforcement learning properly, we first learned the basic concepts of RL online. We learned about the Agent-Environment model of RL, what action space and observation space were. Then we further read about some more technical topics of RL such as policies (π), the State-Value function, the Action-Value function, the Bellman Optimality Equation, and the idea of Exploration versus Exploitation. We also wrote a simple Q-learning implementation for the FrozenLake problem.

After learning the basics, we had to build an environment for our apparatus to operate in so that the algorithms could take action based on the reward and observation. We decided to use build our environment with the OpenAI Gym interface due to its high portability and easiness to use.

First, to test whether RL worked on our problem, we tried to design en environment that only optimize the installation position the rubber bands. The setting of the environment is as follow:

- Observation space:

  The installation positions of the k rubber bands are used as the observation space. We set the space type to multi-discrete to simulate the fact that there are usually premade holes on the linkages so rubber bands can only be installed in certain positions. For a mechanism with n positions for installation, there would be 2n-1 ways a rubber band could be installed considering the fact that two identical rubber bands installed in parallel would produce the same moment regardless the difference in position. In our code, the observation space is defined as:

  ```
  self.observation_space = spaces.MultiDiscrete(np.array([2n-1]*k)
  ```
- Action space:

  The action the agent is allowed to take is to move any of the k rubber bands to another position. In our environment, in each action, the agent can move one of the rubber bands to either left or right, 1, 3, 5, or 7 positions. So for any of the rubber bands there are 8 possible discrete actions +-1, +-3, +-5, and +-7. For k rubber bands 8k actions need to be defined. In our code, the action space is defined as:

  ```
  self.action_per_spring_num = 8
  self.action_per_spring = [-1,1, -3, 3, -5, 5, -7, 7]
  self.action_space = \\
  spaces.Discrete(spring_num * self.action_per_spring_num)
  ```
- Reward:

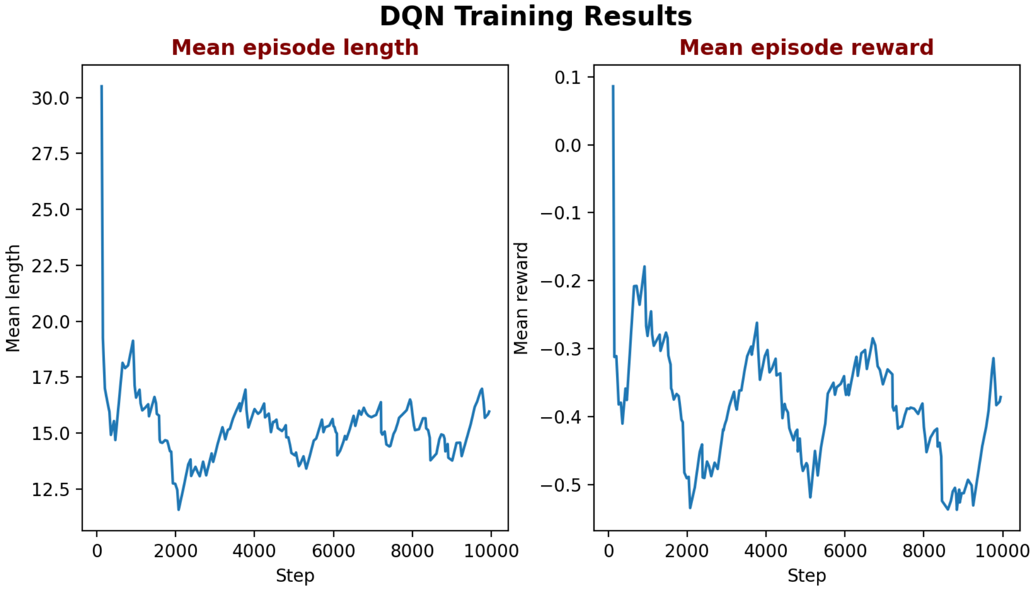
  Instead of giving rewards when certain goals were achieved, we decided to use our fitness function as the reward. Since the smaller the RMSE is the better, we used the reciprocal of the RMSE as the reward and -1 if the action is illegal (agent trying to install spring at a position outside the mechanism)
- Algorithms:

  We used the python library Stable Baseline3 for RL algorithm implementation mainly for its close integration with OpenAI Gym, clear syntax and simplicity of use. Two algorithms, DQN and A2C were used to train the agent.

6.4.1. Results

- DQN

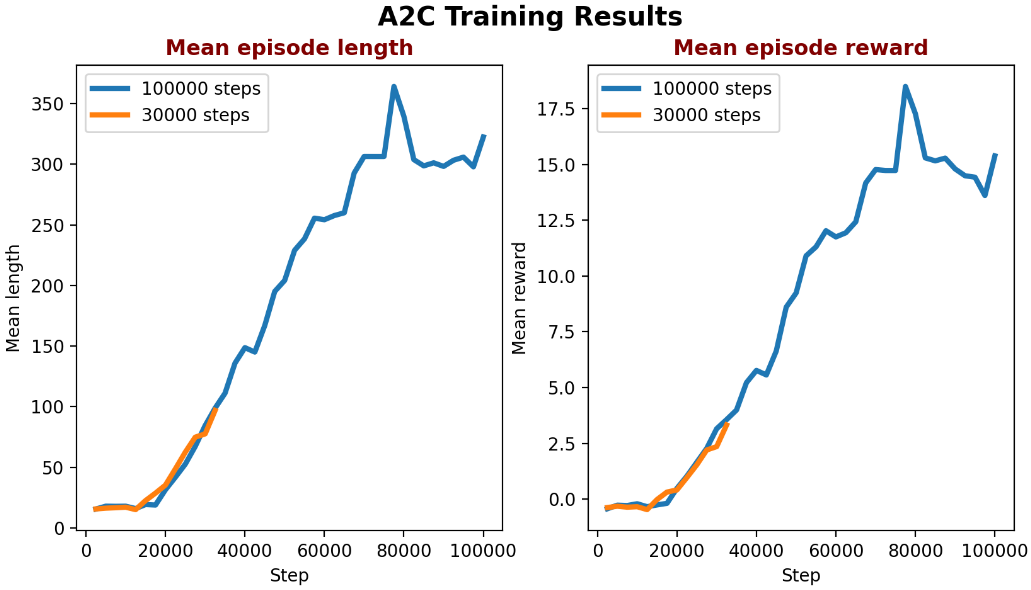
The agent learned for a total of 10000 steps with the DQN algorithm and these are its learning curves:



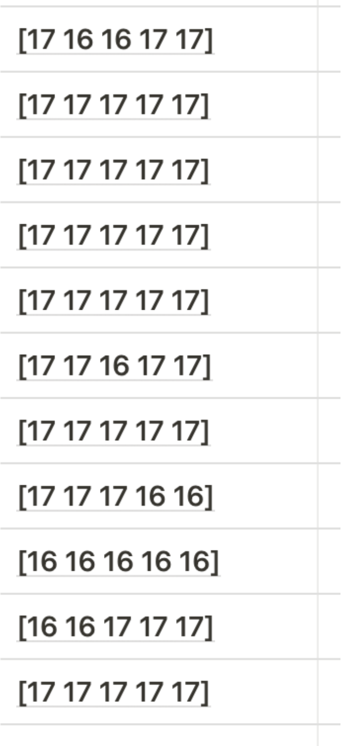
The mean episode length and mean episode reward graphs didn't show any increase or trend during the learning process. More importantly, the mean episode reward graph showed that the reward the agent was able to get was never greater than zero. These two graphs showed that the agent didn't make any practical improvement after training and DQN was ineffective in our problem.

- A2C

The agent was trained with A2C for 2 times with 30000 and 100000 total step respectively and these are the learning curves produced in each training.



The above graphs showed that both the mean episode length and reward increased steadily, indicating that the agent was learning something. Therefore, we saved the model and tested it with random initial rubber band positions. These are the actions the agent took when tested with 10 different samples:



[17 16 16 17 17]
[17 17 17 17 17]
[17 17 17 17 17]
[17 17 17 17 17]
[17 17 17 17 17]
[17 17 16 17 17]
[17 17 17 17 17]
[17 17 17 16 16]
[16 16 16 16 16]
[16 16 17 17 17]
[17 17 17 17 17]

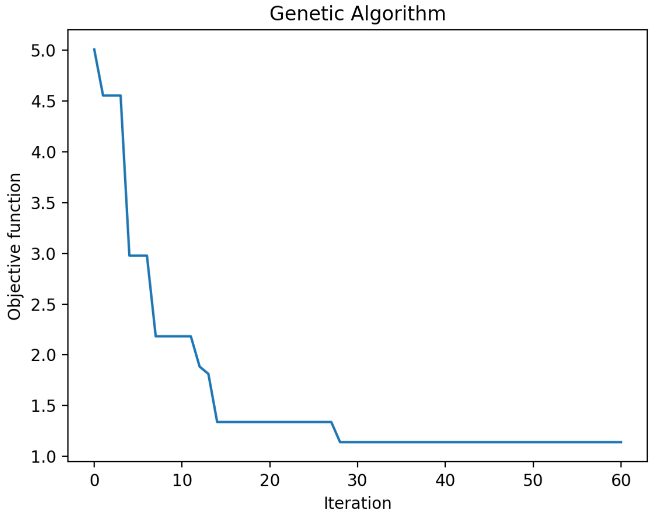
This showed that the only two actions the agent had learned to take were actions 16 and 17, which were the actions of moving the third rubber band 1 position to the left and right respectively. In other words, what the agent learned was to move to the oppositive direction when it was close to an illegal position, in which it would be punished by a -1 reward, to "survive" as long as possible, instead of finding the optimal install positions for the rubber bands. This "survival mindset" of the agent caused it to constantly receive very similar rewards and led to the almost identical shape of the mean episode length and reward curves.

Our work above showed that reinforcement learning might not be an effective method for our problem.

6.5. Genetic algorithm

After our setback in reinforcement learning, we continued to carry out our second plan, genetic algorithm (GA). We chose GA because of its effectiveness in tackling different problems and its ability to produce high-quality solutions consistently. An elitist genetic algorithm was used in the following optimization processes. The RMSE/fitness function was used as the objective function for the GA to minimize. The parameters of the GA are:

```
aps={'max_num_iteration': None,\
              'population_size':3000,\
              'mutation_probability':0.1,
              'elit_ratio': 0.05,\
              'crossover_probability': 0.5,\
              'parents_portion': 0.3,\
              'crossover_type':'uniform',\
              'max_iteration_without_improv':200}
```



At first, we tried to only optimize the install position of the rubber bands as we did in the first greedy algorithm optimization. The genetic algorithm found the solution to be [0. 0. 0. 16.] with RMSE = 1.14, which is identical to the best result the greedy algorithm found. The time elapsed in one search is 280. This showed that although a longer time might be needed, the genetic algorithm was very effective in finding a decent, if not the best, solution to our problem. The followings are the original output result of the GA algorithm and the objective function curve.

```
Timer starts at 1618597776.441518
 The best solution found:
 [ 0.  0.  0. 16.]

 Objective function:
 1.1398206821796109
```

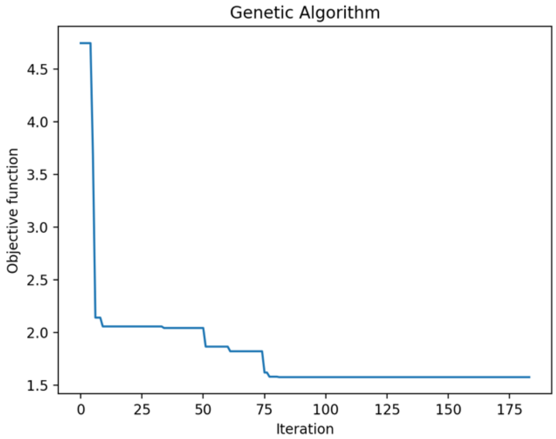
```
Warning: GA is terminated due to the maximum number of iterations without improvement was met!
timer ends at 1618598056.639311
time elapsed: 280.197793006897
```

After successfully optimizing the install positions of rubber bands, we decided to optimize the other two parameters of the rubber bands, the length and the elastic constant. Three rubber bands were optimized and the searchable range of the parameters are:

1. elastic constant: [200,400]
2. length of rubber band: [0.2,0.4]
3. Install position [-19, 19]

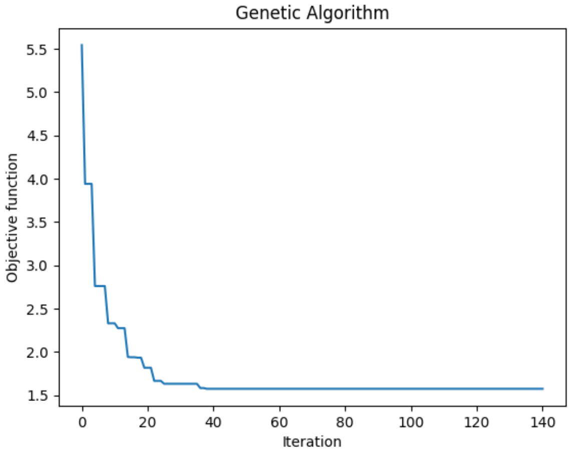
The optimal solution is found to be the install position = [18, -2, -1], the length of the rubber bands = [0.2,0.36,0.32], and the elastic constant = [230,380,270] with RMSE = 1.58. The time elapsed in one search is 316 seconds. The followings are the original output result of the GA algorithm and the objective function curve.

```
The best solution found:
[18. -2. -1. 20. 36. 32. 23. 38. 27.]

Objective function:
1.5756093705311682
time elapsed: 316.34238505363464
```



We then tried to optimize all the parameters (18 in total) when 6 rubber bands installed with the same ranges of parameters. Results are:

```
The best solution found:
[18. -2.  0. -1. -2.  0. 20. 40. 40. 36. 39. 26. 23. 24. 29. 28. 25.   35.]

Objective function:
1.5756093705311682
time elapsed: 5870.809265851974
```

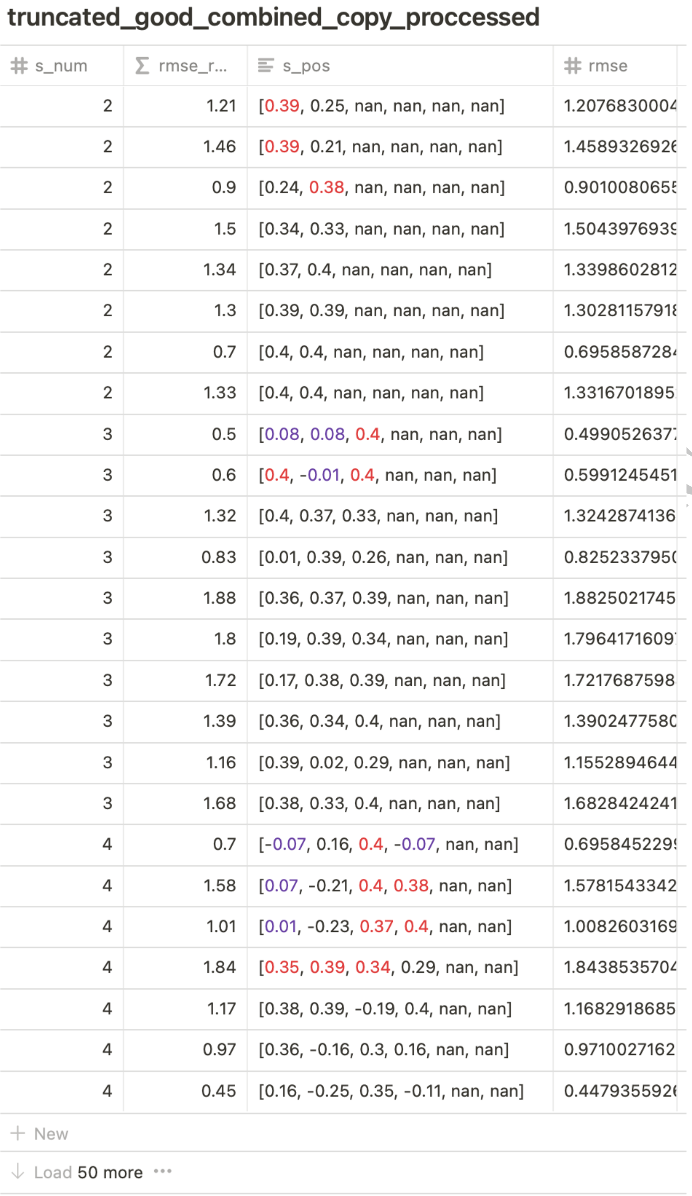


The result had showed that genetic algorithm continued to be effective even when the number of parameter increased and the time required for each calculation did not increase dramatically. Although the genetic algorithm managed to find reasonable solutions, it didn't produce a model that can predict optimal parameters for other situations. Therefore, in order to create such a model, we decided to use the solutions from the GA as the training set to train a neural network, so that the it could tell the optimal install position of rubber bands immediately when we gave it the number of rubber bands and the parameters of it.

To train a network, the size of the training set had to be enormous. Therefore I started generating good solutions with the computers in our lab. The GA was instructed to generate solutions for random settings with the ranges below:

```
# spring constants [150, 500] (N/m)
s_c_range = np.array([150, 500])
# spring initial length [0.15, 0.4] (m)
s_l_range = np.array([0.15, 0.4])
# extremity load [0, 10] (kg)
e_l_ran     ge = np.array([0, 10])
# number of springs (2 to six)
s_num_range = np.array([2, 6])
```

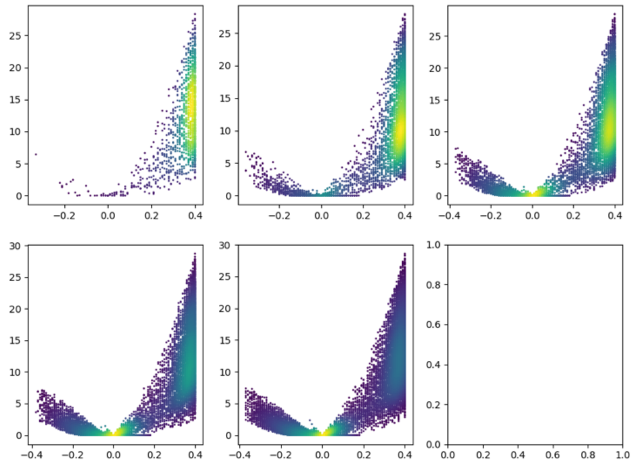
After generating solutions on 3 computers for 24 hours straight, I extracted the data and started examining. Solutions that didn't successfully optimize the install position of the rubber bands, that is, reduce the RMSE below a certain value, which was chosen to be 2 in our case, were discarded. The following table is a the first 100 successful solutions:



truncated_good_combined_copy_proccessed

s_num	rmse_r...	s_pos	rmse
2	1.21	[0.39, 0.25, nan, nan, nan, nan]	1.2076830004
2	1.46	[0.39, 0.21, nan, nan, nan, nan]	1.4589326926
2	0.9	[0.24, 0.38, nan, nan, nan, nan]	0.9010080655
2	1.5	[0.34, 0.33, nan, nan, nan, nan]	1.5043976939
2	1.34	[0.37, 0.4, nan, nan, nan, nan]	1.3398602812
2	1.3	[0.39, 0.39, nan, nan, nan, nan]	1.3028115791
2	0.7	[0.4, 0.4, nan, nan, nan, nan]	0.695858728
2	1.33	[0.4, 0.4, nan, nan, nan, nan]	1.3316701895
3	0.5	[0.08, 0.08, 0.4, nan, nan, nan]	0.499052637
3	0.6	[0.4, -0.01, 0.4, nan, nan, nan]	0.5991245451
3	1.32	[0.4, 0.37, 0.33, nan, nan, nan]	1.3242874136
3	0.83	[0.01, 0.39, 0.26, nan, nan, nan]	0.825233795
3	1.88	[0.36, 0.37, 0.39, nan, nan, nan]	1.8825021745
3	1.8	[0.19, 0.39, 0.34, nan, nan, nan]	1.796417160
3	1.72	[0.17, 0.38, 0.39, nan, nan, nan]	1.7217687598
3	1.39	[0.36, 0.34, 0.4, nan, nan, nan]	1.390247758
3	1.16	[0.39, 0.02, 0.29, nan, nan, nan]	1.1552894644
3	1.68	[0.38, 0.33, 0.4, nan, nan, nan]	1.6828424241
4	0.7	[-0.07, 0.16, 0.4, -0.07, nan, nan]	0.695845229
4	1.58	[0.07, -0.21, 0.4, 0.38, nan, nan]	1.5781543342
4	1.01	[0.01, -0.23, 0.37, 0.4, nan, nan]	1.0082603169
4	1.84	[0.35, 0.39, 0.34, 0.29, nan, nan]	1.843853570
4	1.17	[0.38, 0.39, -0.19, 0.4, nan, nan]	1.1682918685
4	0.97	[0.36, -0.16, 0.3, 0.16, nan, nan]	0.9710027162
4	0.45	[0.16, -0.25, 0.35, -0.11, nan, nan]	0.447935592

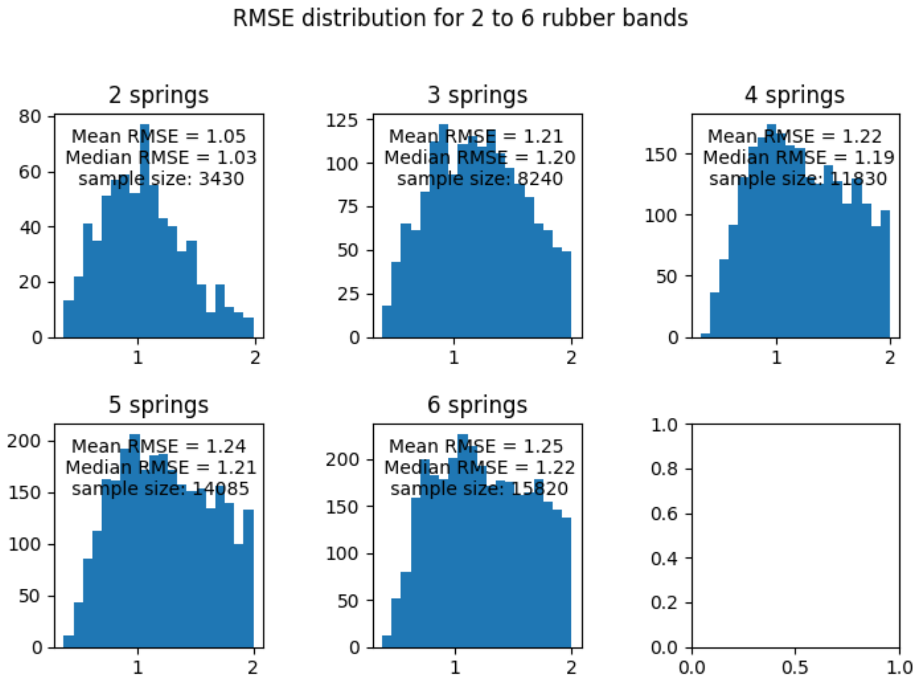
From the above table, we observed that many rubber bands were installed very close to the upper limit 0.4 (red) and the center 0 (purple). This phenomenon is universal across different numbers of springs and spring parameters. To dig deeper into this phenomenon, we plotted the distribution of spring positions for different numbers of springs and further investigate.



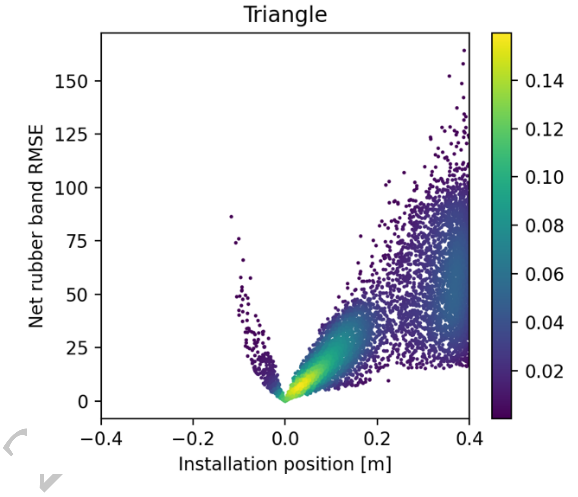
The graphs above showed the distribution of rubber bands install positions from 2 to 6 rubber bands. Each dot on the graph represents one spring that was installed on The x-axes of the graphs were the install position, which ranged from -0.4 to 0.4. The y-axes were the net RMSE of that moment of the rubber band without considering the moment of the weight. In other words, the y-axes represents the "rotational effect" the rubber band had on the DPLM. The color on the graphs represents the density of rubber bands according the the color bar in the bottom.

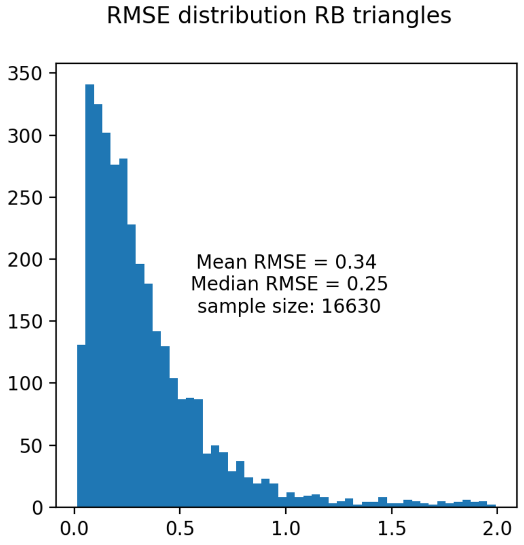
When only two rubber bands were installed, a huge amount of rubber bands with high rotational effects (large y values) accumulated at the upper limit of install position, 0.4. This corresponds to rubber band 1 in the figure below. As the number of rubber bands increased, more rubber bands with a low negative rotational effect appeared on the left of the zero position, which corresponds to rubber band 2 in the figure below. When the number of rubber bands reached four and above, many rubber bands accumulated at the zero points, which corresponded to a vertically-installed rubber band such as rubber band 3. These rubber bands were completely redundant as they didn't produce any moment on the DPLM. In addition, the shapes of the figures were almost identical. This similarity between the shapes of the figures and the redundancy of rubber bands in the graphs prompted the possibility that any number of rubber bands on the DPLM could be replaced by only 2 or 3 of them while preserving almost the same effect on the DPLM if minimizing the overall RMSE was the objective.

Then we plotted the distribution of RMSE values for 2 to 6 rubber bands. The RMSE is the lowest when 2 rubber bands were installed and there was generally no difference between the distribution of 3 to 6 rubber bands. This observation not only provided evidence for our claim that the effect of 4 or more rubber bands could be achieved with just 2 or 3 of them but also suggested that installing 2 rubber bands could produce the best solutions.



We continued to test installing rubber bands in triangles.





The mean RMSE of rubber bands triangles is far lower than that of 2 rubber bands. This shows that a triangle is the optimal way to install rubber bands on DPLM.

We went on to conduct an experiment with a phsical demo of the mechanism and the results were consistent with the opmization.

6.6. Summary of methodology

In this project, a mathematical model that theoretically described all the forces and moments involved in the Double Parallelogram Lifting Mechanism was derived, implemented in both MATLAB and Python, and verified by comparing its result with a SolidWorks simulation. To optimize the spring install positions and spring parameters, we defined the fitness of our model to be the root-mean-square-error (RMSE) between the net moment at the base hinge across the range of allowed angles. Then multiple methods of optimization were applied to the model. The greedy algorithm managed to produce reasonable solutions when the number of parameters was small, yet produced dramatic results when the number of parameters increased. DQN and A2C learning was used later but both of them failed to learn to find optimal solutions. The genetic Algorithm was then used and proved to be able to produce consistently high-quality results as good as that produced by the greedy algorithm when the number of parameters was few and maintained its effectiveness even when the number of springs increased to 6 with 18 parameters in total. Genetic Algorithm was then used to produce a training set for deep learning. Yet, after examining those data carefully, we discovered a universal distribution of the install positions and the rotational effect (net RMSE) of rubber bands on the DPLM regardless of the number of them installed. The distributions were similar, almost identical when different numbers of rubber bands were used. It also showed that a large number of redundant rubber bands when the number reached 4 and therefore the effect of any number of rubber bands should be able to be closely approximated by using just 2 or 3 of them. Installing 2 rubber bands is optimal among all numbers of rubber bands. Rubber band triangles produce an overall much lower RMSE than that of 2 rubber bands, therefore, concludes that triangle is the overall optimal way to install rubber bands on a DPLM.

7. Conclusion

This project provides a theoretical basis for controlling DPLM in real-time with rubber bands by:

- Deriving and verifying a mathematical model of DPLM.
- Optimizing RB installation positions and parameters with GA.
- Showing that a triangle is the overall optimal way to install rubber bands for the balance of DPLM.
- verifying the optimization results with a prototype.

8. Applications

As we are doing researches on our double parallelogram lifting mechanism, we are always considering where we can apply this mechanism to our daily life. This project was inspired by the design of our VEX & FRC robots. From the result of our robot competition, we can prove that the double parallelogram lifting mechanism is a great success and it really helped us in competitions. If we can combine this mechanism with something else, it may be a good innovation. In the following essay, we are going to list some of the usages of the double parallelogram lifting mechanism and what we and done and designed.

8.1. Standing Wheelchair

A standing wheelchair must be a savior for wheelchair users. It allows users to stand up from a seating position by raising up the seat. With the wheelchair, users can interact with others at eye level, or grab things at a high position. Using a standing wheelchair is also a benefit for health like assisting blood circulation and prevent from osteoporosis.

The existing standing wheelchairs are divided into electrical and manual control. The electrical type uses a motor to lift the lever underneath the seat and supporting the users in a standing position. This kind of standing wheelchairs is convenient, precise, and user-friendly because users are allowed to control their position with their fingers. But it cost much more than a usual wheelchair, creates a higher weight, and the components take up plenty of space.

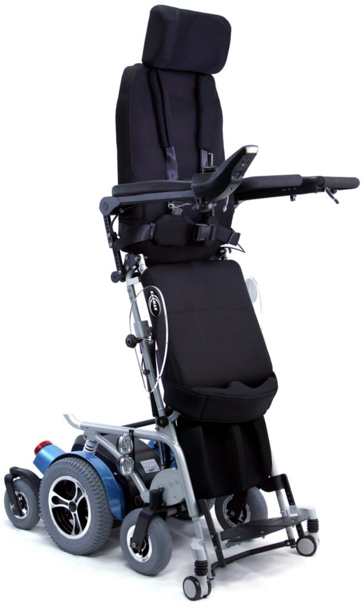


Figure 18: Electric-powered standing wheelchair

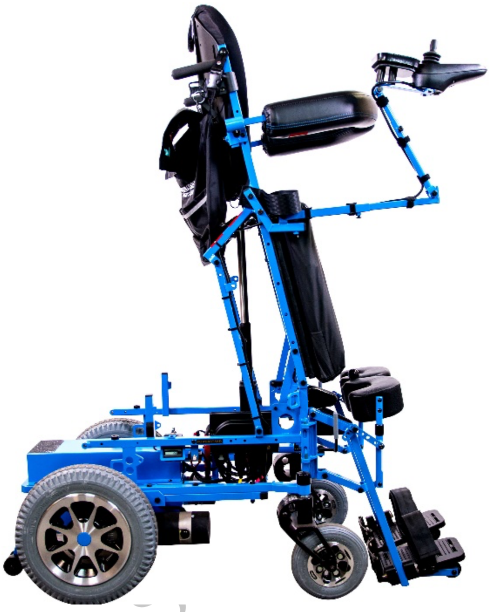


Figure 19: Electric-powered standing wheelchair

A manual standing wheelchair is also a good choice. It's much cheaper than an electronic one, and the mechanism is much more simple so it takes up less space. There are few types of manual standing wheelchairs in the market. Some use hydraulic rods, some use a belt transmission system to make the seat lifted up easily. Users are allowed to lift the seat up and stand up with just a little force of their hands.

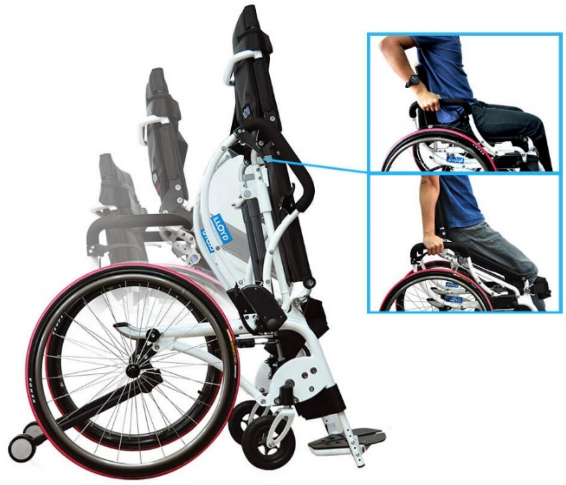


Figure 20: Manual standing wheelchair

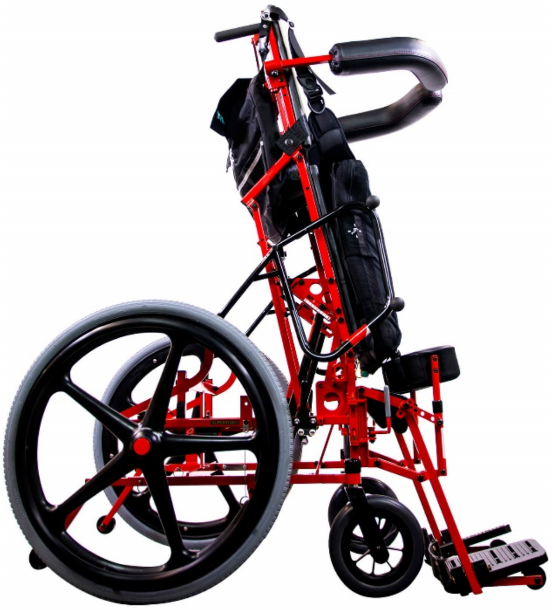


Figure 21: Manual standing wheelchair

The special thing about our parallelogram lifting mechanism is that we added some elastic material to it. With the elastic material, the whole mechanism can be lifted easily and stay at any height which makes it suitable to make up a standing wheelchair. Compare to the existing hydraulic-powered standing wheelchair, the one equipped with our mechanism can be lighter and take up less space than the hydraulic one dose. Furthermore, our mechanism can be lifted and stay at any angle without any force. This allows users to stand at any angle and stay there.

9. Future

In the future, the effects of the changes in centers of mass of linkages on the moment balance of DPLM, the moment balance of DPLM when it is tilted at a certain angle, and the optimization method for DPLM with multiple Rubber bands installed will be further discussed and investigated based on the model and method proposed in this paper.

Rehabilitation is very important to people after injured or people who suffer from an illness like Muscular dystrophy. The main purpose of rehabilitation is to help patients move by themselves persistently so that their muscles will not lose their function. Common exercises for rehabilitation are walking practice and sit-to-stand exercises, and we found out that our parallelogram lifting mechanism can be used in a sit-to-stand assisting machine.

The top parallelogram can be used as a lever to lift the whole mechanism; the bottom parallelogram can be mounted on the legs of the user. Users can lift their body with just a little bit of force from their hands. There are some similar products in the market now, for example 'Sit2Stand' from Biodex[13]. It is also a machine that help patients to do stand up practice, but the differences is that the users can only use it on the seat of the machine and the size of the whole machine is huge. With the parallelogram lifting mechanism, our sit to stand assist machine can be mounted on the patient's body. Campare to the 'Sit2Stand' type of sit to stand assisting machine, our design is more flexible.

10. Reference

[1] Stucky J L and Stucky R K 1973 DOUBLE PARALLELOGRAM LINKAGE LIFT *U. S. Patent* US3768591A

[2] York L E, Peoria and Ill 1980 APPARATUS FOR LIFTING A MEMBER USING PARALLELOGRAM MOUNTED LINKS *U. S. Patent* US4431083A

[3] Zhang C Y, Zhang X M, Ye H Y, Wei M and Ning X 2019 An Efficient Parking Solution: A Cam-Linkage Double-Parallelogram Mechanism Based 1-Degrees of Freedom Stack Parking System *J. Mechanisms Robotics* 11(2):045001

[4] Aksungur S and Koca T 2015 Remote Center of Motion (RCM) Mechanisms for Surgical Operations *Int. J. Applied Mathematics Electronics Computers* 3(2):119

[5] Christensen S and Bai S P 2016 A Novel Shoulder Mechanism with a Double Parallelogram Linkage for Upper-Body Exoskeletons, Wearable Robotics: Challenges and Trends: Proceedings of the 2nd International Symposium on Wearable Robotics *WeRob2016 (Segovia, Spain)* pp 51-56.

[6] Bouffard C 2019 An Upper Extremity Exoskeleton Utilizing a Modified Double Parallelogram Linkage Mechanism with Proximally Located Actuators *Honors College* 488

[7] Hadavand M, Mirbagheri A, Salarieh H and Farahmand F 2011 Design of a Force-Reflective Master Robot for Haptic Telesurgery Applications: RoboMaster1 *33rd Annual Int. Conf. of the IEEE EMBS Boston (Massachusetts USA)*

[8] Gandhi P S, Sonawale K and Soni V 2011 Development of Double Parallelogram Flexure Mechanism via Assembly Route *15th National Conference on Machines and Mechanisms.*

[9] Hao G B, He X Y and Awtar S 2019 Design and Analytical Model of a Compact Flexure Mechanism for Translational Motion *Article in Mechanism and Machine Theory* (Elsevier Ltd.)

[10] Min H N 2020 A KIND OF INSPECTION ROBOT *CHN Patent* CN202011014422

[11] Williams M S 1999 *Structures: Theory and Analysis* (Red Globe Press, 1999th Edition)

[12] Lombard M 2018 *Mastering SolidWorks* (Sybex vol 1)