Modeling a Strandbeest Bike

Scarlett Bonner and Nebi Samuel

4 December 2024

1 Physical System and Problem Statement

Theo Jansen was the original creator of "strandbeesten" which translates to "beach animals" in Dutch. He makes walking creatures out of multiple "Jansen linkages" as seen in Figure 1. Some are even powered by the wind. We, however, plan to use the Jansen linkages to replace the rear wheel on a bike to produce a walking bike as the final physical product, to be completed by the end of the spring semester. For the scope of this project, we built and modeled one of the six legs to be used for the final product, to scale, and calculated parameters relating to the physical system of each leg (i.e. footpath, kinetic energy, and power).

Figure 1: Strandbeest Bike [1]

The Jansen linkage is central to the bike's design, as each leg is made of a single Jansen linkage (Figure 2). It is a single degree of freedom system where a crankshaft (Figure 5, point 2) turns the crank arm (Figure 5, segment c) and causes the lowermost point on the linkage to move in a walking motion. The axle and the crankshaft are fixed with respect to the rest of the bike. Segment c is the crank arm and its angle determines where the rest of the system is. Most of the angles within the linkage do vary over time but it is worth noting that there are two rigid bodies at the “shoulder” and the “foot” of the linkage made of two fixed triangles. It is also worth noting that the speed of most of the points in the linkage is not constant, speeding up the most as the “foot” lifts to take another step. Although this is not a concern in simulation, this will cause an imbalance in power required throughout a single cycle when pedaling, assuming there is a constant angular velocity. However, this will be mitigated by having multiple legs connected out of phase to balance this power distribution throughout a cycle.

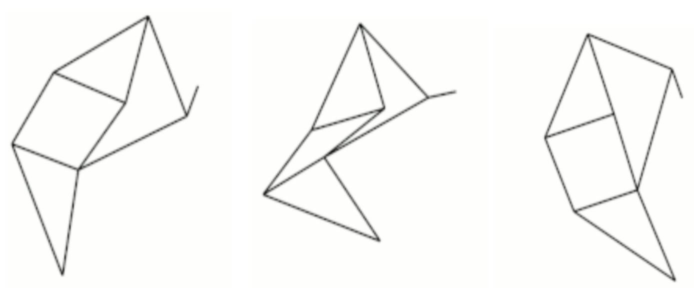

Figure 2: Jansen linkage at different moments of a single rotation [2]

There are a few considerations that we have to make in order to replace a bike wheel with Jansen linkages. The first is that a single bike “tire” is always in contact with the ground throughout its rotation. To accomplish this, we will have 6 legs built from the linkages, 3 on each side of the bike positioned out of phase with one another by 120°. The goal is that there are at least 4 legs (two on each side) in contact with the ground through the entire cycle. By examining the path of the foot, we know that the foot will remain in contact with the ground through at least 180° of the rotation of the crankshaft, which confirms that having 3 legs out of phase with one another by 120° will ensure that at least one foot is on the ground per side at all times. Our goal for this project is to build a working strandbeest bike and to be able to model the entire system given a crank arm angle.

1.1 Assumptions

We will be using the predetermined ratios between links/struts known as Jansen's Linkage, which will allow us to scale the Strandbeest legs to meet our design goals [3]. This linkage consists of 11 links in the configuration below, in which each link's movement is dependent on the rotational movement of the crankshaft (segment c), of which the range of motion is depicted by the blue circle (Figure 3). For the purpose of this project, we will be operating based on the following assumptions.

First, we will assume that the Strandbeest bike is assumed to operate on a flat, rigid surface without significant variations in terrain. This assumption simplifies the analysis of the bike's kinematics and dynamics. We will assume that loads from the rider from pedaling will remain within the design limits of the frame and linkage mechanisms, and any fatigue of the materials and joints will not be included in this analysis. For modeling purposes, we will also assume that the force applied from the rider to turn the crankshaft will produce a constant angular velocity of segment c. The impact of wind resistance is assumed to be minimal and therefore excluded from this model.

Furthermore, the linkage joints are assumed to be perfectly aligned without initial misalignments. However, friction in the joints is not assumed to be negligible. For power calculations, a friction coefficient will be assigned to each joint to account for energy losses due to rotational resistance. This coefficient will be estimated to be 0.15 based on the study of friction between lubricated nylon and steel from Lube-Tech [5].

To model the power calculations, the mass of each component in the linkage was calculated using the volume of the part in our CAD model and the density of the wood we are using. Furthermore, the $\widehat{b_y}$ component of the inertia dyadic is assumed to be negligible for the simulation.

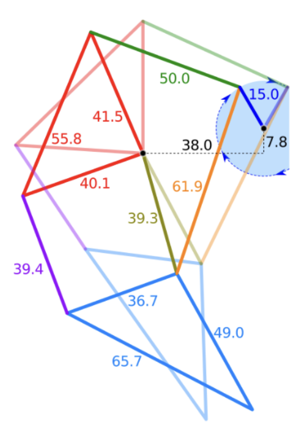

Figure 3: Jansen's Linkage and Crankshaft [2]

2 Defining the System

2.1 Reference Frames

We created reference frames for each of the rigid bodies/struts in order to define their orientation through simple angular rotations relative to the Newtonian reference frame **N**, as shown in Figure 4. Each rotation can be characterized by a simple angular velocity about the $\widehat{n_z}$ axis.

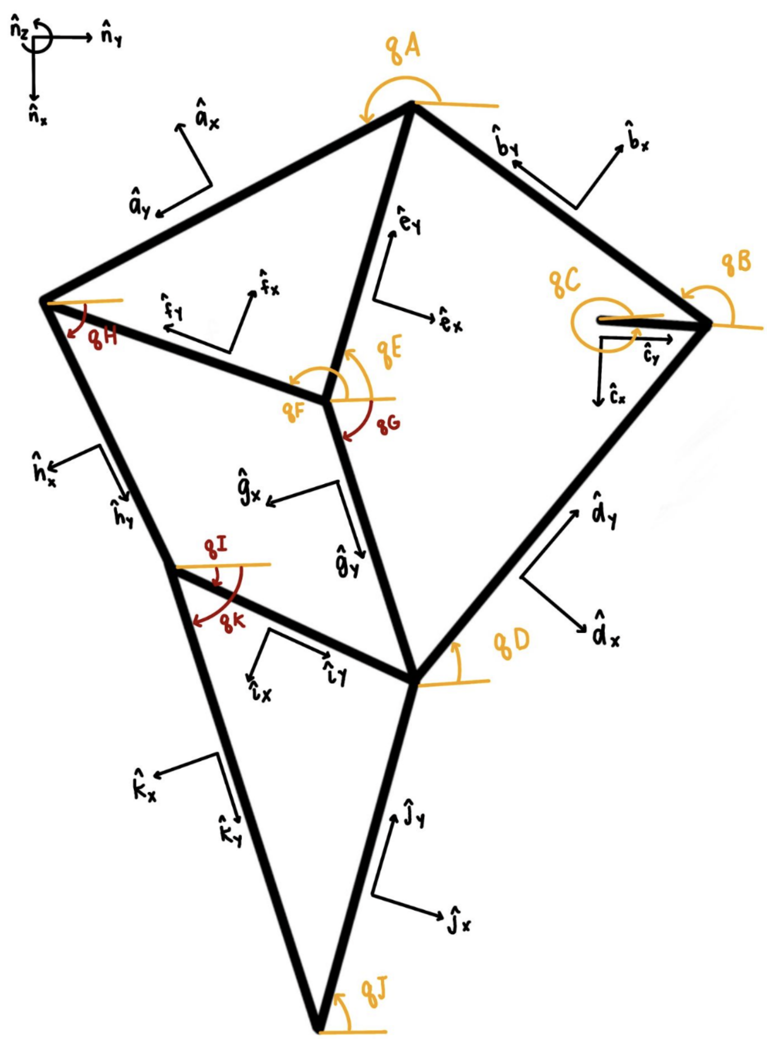

Figure 4: Diagram of reference frames and defined angles

The rotations of the reference frames are defined below:

> Frames **A**, **B**, **C**, **D**, **E**, **F**, and **J** are rotated by angles q_A, q_B, q_C, q_D, q_E, q_F, q_J respectively from **N** in the $+\widehat{n_z}$ direction.
>
> Frames **G**, **H**, **I**, and **K** are rotated by angles q_G, q_H, q_I, and q_K respectively from **N** in the $-\widehat{n_z}$ direction.

The rotation matrices for frames **A**, **B**, **C**, **D**, **E**, **F**, and **J** follow the structure of the rotation matrix in Table 1. The rotation matrices for frames **G**, **H**, **I**, and **K** follow the structure of the rotation matrix in Table 2.

${}^nR^a$	$\widehat{a_x}$	$\widehat{a_y}$	$\widehat{a_z}$
$\widehat{n_x}$	$cos(q_a)$	$-sin(q_a)$	0
$\widehat{n_y}$	$sin(q_a)$	$cos(q_a)$	0
$\widehat{n_z}$	0	0	1

Table 1: Rotation Matrix from **N** to **A**

${}^nR^g$	$\widehat{g_x}$	$\widehat{g_y}$	$\widehat{g_z}$
$\widehat{n_x}$	$cos(q_g)$	$-sin(q_g)$	0
$\widehat{n_y}$	$sin(q_g)$	$cos(q_g)$	0
$\widehat{n_z}$	0	0	1

Table 2: Rotation Matrix from **N** to **G**

2.2 Constants and Variables

We defined a number of constants and variables in order to better describe the system, outlined in the figure and table below (Figure 5) (Tables 3 and 4). The table gives a description of each variable, the symbol used, the type of variable, the units, and the value that was assigned to each variable in the simulation. The initial values are based on the Jansen's linkage ratios, the resulting mass approximations discussed in Section 2.

Description	Symbol	Type	Units	Value/IC
Length of segment a	a	Constant	m	Value
Length of segment b	b	Constant	m	Value
Length of segment c	c	Constant	m	Value
Length of segment d	d	Constant	m	Value
Length of segment e	e	Constant	m	Value
Length of segment f	f	Constant	m	Value
Length of segment g	g	Constant	m	Value
Length of segment h	h	Constant	m	Value
Length of segment i	i	Constant	m	Value
Length of segment j	j	Constant	m	Value
Length of segment k	k	Constant	m	Value
Mass of segment a	Ma	Constant	kg	0.156
Mass of segment b	Mb	Constant	kg	0.140
Mass of segment c	Mc	Constant	kg	0.042
Mass of segment d	Md	Constant	kg	0.173
Mass of segment e	Me	Constant	kg	0.116
Mass of segment f	Mf	Constant	kg	0.112
Mass of segment g	Mg	Constant	kg	0.110
Mass of segment h	Mh	Constant	kg	0.110
Mass of segment i	Mi	Constant	kg	0.103
Mass of segment j	Mj	Constant	kg	0.137
Mass of segment k	Mk	Constant	kg	0.184
Angle between segment a and segment e	L	Constant	degrees	45.82
Angle between segment e and segment f	N	Constant	degrees	86.23
Angle between segment k and segment i	O	Constant	degrees	47.42
Angle between segment k and segment j	R	Constant	degrees	33.47
Distance in $\widehat{n_x}$ between axle and crankshaft	x	Constant	m	value
Distance in $\widehat{n_y}$ between axle and crankshaft	y	Constant	m	value

Table 3: Constants

Description	Symbol	Type	Units	Value/IC
Angle from $\widehat{n_y}$ to $\widehat{a_y}$ with $+\widehat{n_z}$ sense	q_a	Variable	degrees	252
Angle from $\widehat{n_y}$ to $\widehat{b_y}$ with $+\widehat{n_z}$ sense	q_b	Variable	degrees	160
Angle from $\widehat{n_y}$ to $\widehat{c_y}$ with $+\widehat{n_z}$ sense	q_c	Variable	degrees	121
Angle from $\widehat{n_y}$ to $\widehat{a_d}$ with $+\widehat{n_z}$ sense	q_d	Variable	degrees	72
Angle from $\widehat{n_y}$ to $\widehat{e_y}$ with $+\widehat{n_z}$ sense	q_e	Variable	degrees	112
Angle from $\widehat{n_y}$ to $\widehat{f_y}$ with $+\widehat{n_z}$ sense	q_f	Variable	degrees	158
Angle from $\widehat{n_y}$ to $\widehat{g_y}$ with $-\widehat{n_z}$ sense	q_g	Variable	degrees	75
Angle from $\widehat{n_y}$ to $\widehat{h_y}$ with $-\widehat{n_z}$ sense	q_h	Variable	degrees	66
Angle from $\widehat{n_y}$ to $\widehat{i_y}$ with $-\widehat{n_z}$ sense	q_i	Variable	degrees	20
Angle from $\widehat{n_y}$ to $\widehat{j_y}$ with $+\widehat{n_z}$ sense	q_j	Variable	degrees	120
Angle from $\widehat{n_y}$ to $\widehat{k_y}$ with $-\widehat{n_z}$ sense	q_k	Variable	degrees	28

Table 4: Variables

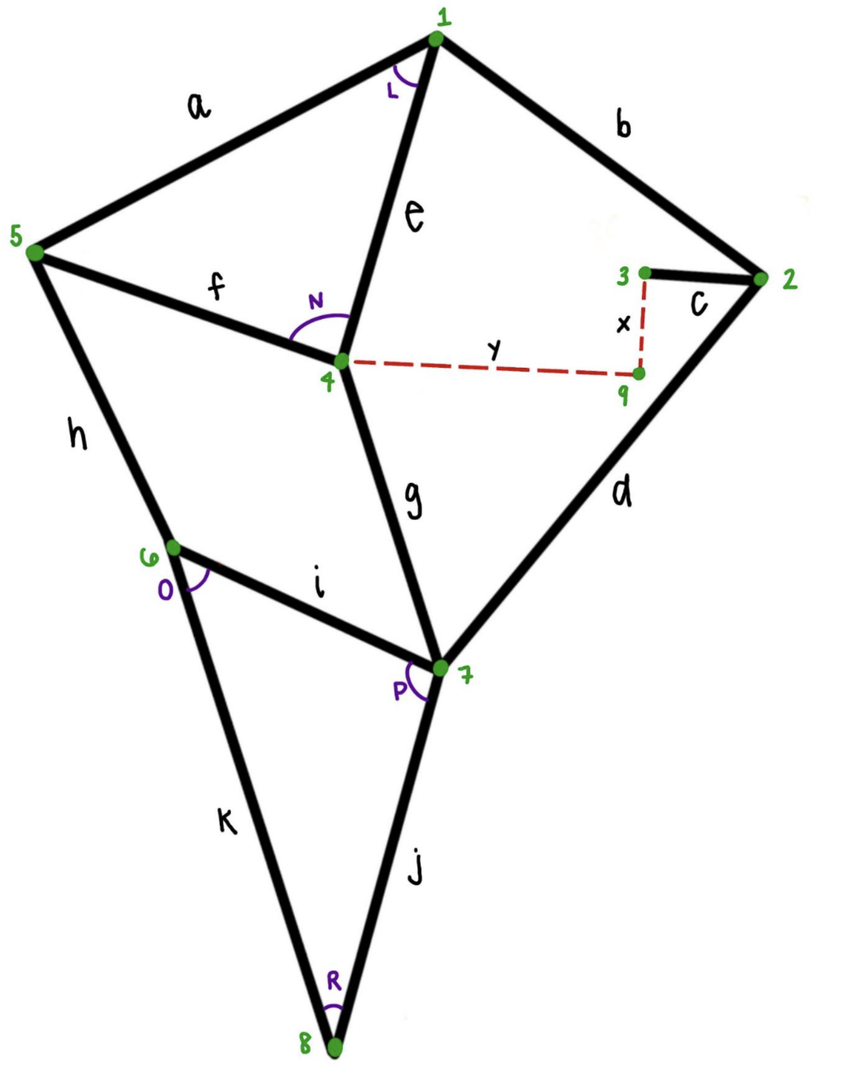

Figure 5: Diagram of our Jansen linkage with segments, points, and angles

3 Kinematics

3.1 Loop Equations

We can define the following loop equations in order to obtain 10 equations to solve for our 10 unknown angles. We know from class that each loop equation will equal a zero vector. First, we can define the loop from point $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 4 \rightarrow 1$ and set it equal to the zero vector as follows:

$$\vec{0} = b * \widehat{b_y} - e * \widehat{e_y} + y * \widehat{n_y} - x * \widehat{n_x} + c * \widehat{c_y} \tag{1}$$

We can write this in terms of the **N** basis vectors using our rotation matrices and dotting by both $\widehat{n_x}$ and $\widehat{n_y}$ to decompose into two separate scalar equations:

$$0 = -bsin(q_b) + esin(q_e) - x - csin(q_c) \tag{2}$$

$$0 = bcos(q_b) - ecos(q_e) + y + ccos(q_c) \tag{3}$$

Next, we can write the loop from point $4 \rightarrow 9 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 4$ and set it equal to the zero vector as follows:

$$\vec{0} = y * \widehat{n_y} - x * \widehat{n_x} + c * \widehat{c_y} - d * \widehat{d_y} - g * \widehat{g_y} \tag{4}$$

Again, we can write this in terms of the **N** basis vectors using our rotation matrices and dotting by both $\widehat{n_x}$ and $\widehat{n_y}$ to decompose into two separate scalar equations:

$$0 = -x - csin(q_c) + dsin(q_d) + gsin(-q_g) \tag{5}$$

$$0 = y + ccos(q_c) - dcos(q_d) - gcos(-q_g) \tag{6}$$

Finally, we can write the loop from point $4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 4$ and set it equal to the zero vector as follows:

$$\vec{0} = f * \widehat{f_y} + h * \widehat{h_y} + i * \widehat{i_y} - g * \widehat{g_y} \tag{7}$$

Writing this equation in terms of the **N** basis vectors and dotting by $\widehat{n_x}$ and $\widehat{n_y}$ gives us the following:

$$0 = fsin(q_f) - hsin(-q_h) - isin(-q_i) + gsin(-q_g) \tag{8}$$

$$0 = fcos(q_f) + hcos(-q_h) + icos(-q_i) - gcos(-q_g) \tag{9}$$

In order to develop 10 equations for our 10 unknowns, we were able to write four redundant equations to characterize the angles of the two fixed triangles in terms of known angles. We followed the same process, writing loop equations for the top and bottom triangles from point $6 \rightarrow 7 \rightarrow 8 \rightarrow 6$ and from point $1 \rightarrow 5 \rightarrow 4 \rightarrow 1$ as follows:

$$\vec{0} = i * \widehat{i_y} - j * \widehat{j_y} - k * \widehat{k_y} \tag{10}$$

$$\vec{0} = a * \widehat{a_y} - f * \widehat{f_y} + e * \widehat{e_y} \tag{11}$$

We then dotted by $\widehat{n_x}$ and $\widehat{n_y}$ after writing in terms of the **N** basis vectors to find four independent equations:

$$0 = -isin(-q_i) + jsin(-q_j) + ksin(-q_k) \tag{12}$$

$$0 = icos(-q_i) - jcos(q_j) - kcos(-q_k) \tag{13}$$

$$0 = -asin(q_a) + fsin(q_f) - esin(q_e) \tag{14}$$

$$0 = acos(q_a) - fcos(q_f) + ecos(q_e) \tag{15}$$

Solving this system of 10 equations (Equations 2, 3, 5, 6, 8, 9, 12, 13, 14, and 15) when given a starting value for q_c, the angle of the crankshaft, will then allow us to find all other resulting angles.

4 Kinetic Energy

We calculated the rotational and translational kinetic energy of each rigid body in order to obtain the total kinetic energy of the system. For each body **B**, the rotational kinetic energy was calculated as follows:

$$^{n}K^{b}_{rot} = \frac{1}{2} *^{n} \omega^{b} \cdot \overrightarrow{\overrightarrow{I}}^{B/B_{cm}} \cdot^{n} \omega^{b} \tag{16}$$

We know that for any body **B** in our system, the angular velocity ${}^{n}\vec{\omega}^{b}$ can be written as $q_b' \widehat{b_z}$. Furthermore, we know that the inertia dyadic for any body in our system will be as follows:

$$\overrightarrow{\overrightarrow{I}}^{B/B_{cm}} = I_{xx}\widehat{b_x}\widehat{b_x}I_{zz}\widehat{b_z}\widehat{b_z} \tag{17}$$

We know there will be no $I_{yy}\widehat{b_y}\widehat{b_y}$ element as there will be no component of inertia about the by vector based on our assumptions. While there will theoretically be a $\widehat{b_y}$ component of inertia in our physical system, we are operating using the assumption that this value will be negligible for our simulation. When we dot this with the angular velocity vector on both sides, we find that the kinetic energy of any body **B** in our system can be expressed as:

$$^{n}K^{b}_{rot} = \frac{1}{2} * I_{zz} * q_{B}^{'\,2} \tag{18}$$

In each of these cases, the inertia of a rod rotated about its center, or I_{zz}, is known as $I_{zz} = \frac{1}{12} * m^b * b^2$, where b is the length of body **B**. Similarly, for each body **B**, the translational kinetic energy was calculated as follows:

$$^{n}K^{b}_{trans} = \frac{1}{2} * m^b *^n v^{b_{cm}} \cdot^n v^{b_{cm}} \tag{19}$$

The velocity $^{n}\overrightarrow{v}^{b_{cm}}$ was evaluated using position vectors and the velocity definition; because point 3, the axle of the crankshaft is fixed in **N**, we were able to calculate the velocity of b_{cm} in **N**, $^{n}\overrightarrow{v}^{b_{cm}}$, by taking the derivative of the position vector from point 3 to b_{cm}, $^{3}\overrightarrow{r}^{B_{cm}}$ (Figure 5). We found this position vector by defining loop equations from point 3 to the center of mass of each body **B**, as follows:

$$^{3}\overrightarrow{r}^{B_{cm}} = c * \widehat{c_y} + \frac{1}{2} b * \widehat{b_y} \tag{20}$$

These values were then used in Equation 19 to find the translational kinetic energy of each body.

Finally, in order to trace the path that the foot of the linkage takes as the crank is rotated, found by the position of point 8 relative to point 3, we dotted the position vector $^{3}\overrightarrow{r}^{B_{cm}}$ with $\widehat{n_x}$ and $\widehat{n_y}$ to get the x and y components of the position in the **N** frame (Figure 6). From this path, we can see the expected shape that point 8 will trace during a cycle.

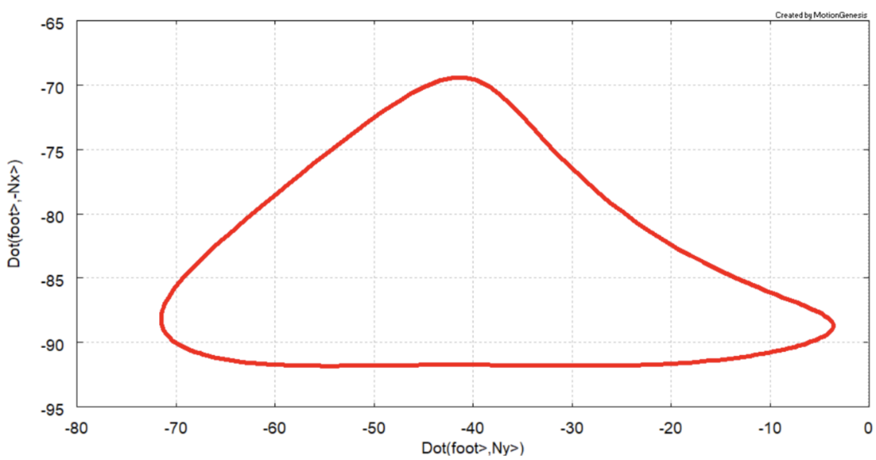

Figure 6: Path of the foot drawn by MotionGenesis

5 Power

Using the power/kinetic energy rate principle, we can calculate the amount of power needed to turn the crankshaft of the bike for a single leg (this can be scaled up to all six legs). We will first calculate the power needed assuming no friction in any of the joints, so the power needed will only include the inertia of the bodies and the force of gravity. We will then experimentally find the power needed to turn the crankshaft and the difference in power should give us the power needed to overcome friction and therefore the average coefficient of friction in the joints (b). The power on **A** in **N** will be calculated using the following equation:

$$ {}^{N}P^{A} = F^{A_{cm}} \cdot {}^{N}\vec{v}^{A_{cm}} + (T^{A/B} \cdot {}^{B}\vec{\omega}^{A}) \tag{21} $$

In this equation, the force on **A** acting at the center of mass is defined as follows:

$$ F^{A_{cm}} = m^{A} * g \tag{22} $$

We can write T^{A} to be the torque on body **A** due to a friction coefficient n between bodies **A** and **B**, which is proportional to the angular velocity between the two bodies:

$$ T^{A/B} = -n * b \cdot {}^{B}\vec{\omega}^{A} \tag{23} $$

The power-kinetic energy principle tells us that the total power of the system is equal to the derivative of the kinetic energy or the system, so we can create a zero equation using Equation 20 to solve for the required input power to turn the crank:

$$ InputPower = \frac{d}{dt}\left[{}^{n}K_{rot}^{syst} +^{n} K_{trans}^{syst} -^{n} P^{syst}\right] \tag{24} $$

6 Results

We graphed the expected input power over time assuming a constant angular velocity of rad/sec of the crankshaft, which rendered the following results shown in Figure 7:

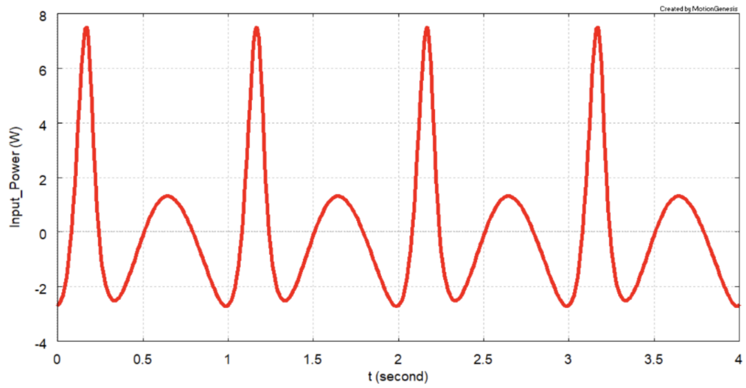

Figure 7: Simulated Input Power

7 Build and Testing

7.1 Build Process

For the build process, we first had to get our materials, which are listed in the budget in Section 9. Once acquiring the materials, we began the build process. We referenced the build documentation by Brandon Green [1]. Using his documentation and outlines for the struts, we CNC'd the individual wood parts for one leg with 0.75" and 0.5" plywood. Then, we used the lathe to make the bushings for each hole. This was necessary as our materials had not arrived, so we had made bushings from 1" diameter nylon stock, from which we bored 0.5" diameter holes through the center and cut them into 0.5" and 0.75" segments. Next, we cut axles from $\frac{1}{2}$" diameter steel stock with varying lengths. Three of these axles were tapped in order to accommodate a XX bolt. Finally, a thin metal piece was constructed of appropriate length to act as the crankshaft (strut c). The leg was assembled with the bushings press-fit into the holes in the plywood leg struts, with the steel axles threaded through the 0.5" holes in the bushings.

To allow the single leg to rotate, we also constructed a mount for the leg to hold it upright, above the ground. Two upright beams were connected to single piece of wood to be used as a base plate and mounting beams. These upright mounts were reinforced with smaller 45 degree wooden braces. A hole was drilled in each upright mount, corresponding to two holes on a larger aluminum sheet on which the leg will be mounted. This aluminum sheet has holes in the appropriate places to fix the axle 3 and 4 by a bolt threaded through the sheet and into the tapped axle.

7.2 Testing

For the testing process, we examined the force necessary to rotate the leg assembly. To translate the rotational force into a linear pulling force such that we could measure with a digital force gauge, we fixed one end of a string to a 3D printed disk that connected point 3 and point 4 and another to the end of the measurement device. The string was then wrapped around the disk. The force was measured by pulling the device steadily away from the axle, allowing the crankshaft to be turned.

Because this force varies with time, we found the best way to capture meaningful data is to find the value of the two peaks that occur in each cycle as shown in Figure 8. Using a digital force gauge, we found that at one rotation per second, the two peaks were at 19.5N and 4.2N. After turning the force into a torque and the torque into power, we found that the two peaks in the power vs time graph should have a value of 24.5W and 5.3W as opposed to the 7.5W and 1.3W peaks we see on the simulated frictionless power graph (Figure 7).

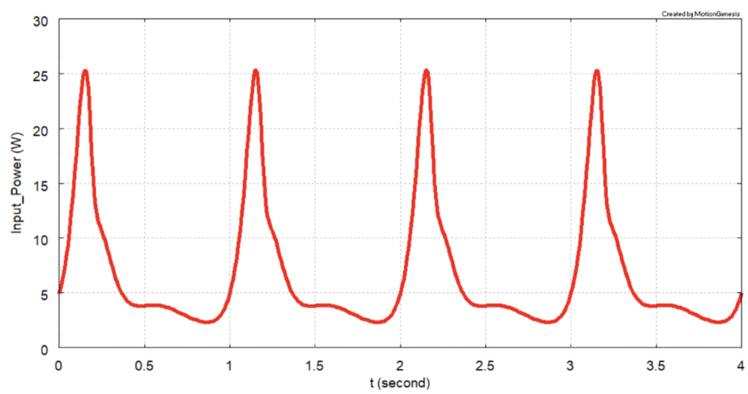

Figure 8: Measured Input Power

In order to solve for b, the "average friction coefficient", we figured the easiest method would be to try a value of b and check the graph until the peak matched our measured value. Figure 8 shows this graph.

8 Discussion

Based on our calculations and research, we expected to see an average friction coefficient of around 0.15. However, using our approach to see what value would correspond with the power peaks, we observed a value of about 0.03. This was not what we expected, but may be as a result of a nonlinear relationship between friction and angular velocity, varying of friction based on the normal force in the axle throughout a period, or inconsistencies in the construction of our system such as the loose fit of our bushings/axles.

Furthermore, we noticed that the value of input power in the simulated graph goes below zero for some points in the cycle. This makes sense because at certain times during the rotation cycle, the crank without external force. However, we did not see this in our experimental power results - the input force remained above 0 for the duration of the cycle. It may be possible that this is a result of experimental error due to the increased friction in our physical system.

9 MotionGenesis Modeling

We came up with two different ways of modeling the Jansen linkage. They both define rigid frames that correspond with each segment of the linkage. The frames are all defined as a simple rotation from the N frame, which is fixed to the floor. The first method uses 5 loops to define the entire Jansen linkage, including the two rigid triangles with unchanging angles. Another possible method uses the first 3 loops to define the varying angles of the linkage and the fact that those triangles have unchanging angles to define the remaining 4 needed equations. The MG code for the triangle method along with its output is below:

```
% Defining all of the RigidFrames fixed to all of the leg segments
RigidBody A, B, C, D, E, F, G, H, I, J, K, N

% We define every leg segment's angle with respect to the ground (frame N)
Variable qa'', qb'', qd'', qe'', qf'', qg'', qh'', qi'', qj'', qk''
Specified qc''

% Constants a-k are the lengths of leg segments
% Constants x and y are the distance in nx> and ny> between the axle and the crankshaft
% Constants L, N, O, R are the fixed angles of the two rigid bodies in the leg
% (the two triangles whose angles never change)
% Constant z is the scalar for the ratio of the length of the leg segments
Constant a, b, c, d, e, f, g, h, i, j, k, x, y, L, N, O, R, Z, grav, fric
```

```
% Defining all of the rotation matrices. Some of the angles were
% more convinent to define negatively.
% Be careful to define negatives correctly
A.RotateZ(N,qa)
B.RotateZ(N,qb)
C.RotateZ(N,qc)
D.RotateZ(N,qd)
E.RotateZ(N,qe)
F.RotateZ(N,qf)
G.RotateZ(N,-qg)
H.RotateZ(N,-qh)
I.RotateZ(N,-qi)
J.RotateZ(N,qj)
K.RotateZ(N,-qk)

% Defining centers of mass for kinetic energy calculation
% We are defining these vector from a point stationary in N, (end of the crankshaft)

Acm> = c*cy> + b*by> + 0.5*a*ay>
Bcm> = c*cy> + 0.5*b*by>
Ccm> = 0.5*c*cy>
Dcm> = c*cy> - 0.5*d*dy>
Ecm> = c*cy> + b*by> -0.5*e*ey>
Fcm> = c*cy> + b*by> - e*ey> + 0.5*f*fy>
Gcm> = c*cy> - d*dy> - 0.5*g*gy>
Hcm> = c*cy> + b*by> - e*ey> + f*fy> + 0.5*h*hy>
Icm> = c*cy> - d*dy> -0.5*i*iy>
Jcm> = c*cy> - d*dy> - 0.5*j*jy>
Kcm> = c*cy> - d*dy> - j*jy> - 0.5*k*ky>
foot> = c*cy> - d*dy> - j*jy>

v_Acm> = Dt(Acm>, N)
v_Bcm> = Dt(Bcm>, N)
v_Ccm> = Dt(Ccm>, N)
v_Dcm> = Dt(Dcm>, N)
v_Ecm> = Dt(Ecm>, N)
v_Fcm> = Dt(Fcm>, N)
v_Gcm> = Dt(Gcm>, N)
v_Hcm> = Dt(Hcm>, N)
v_Icm> = Dt(Icm>, N)
v_Jcm> = Dt(Jcm>, N)
v_Kcm> = Dt(Kcm>, N)
v_foot> = Dt(foot>, N)
```

```
% Define mass properties
A.SetMass(mA)
B.SetMass(mB)
C.SetMass(mC)
D.SetMass(mD)
E.SetMass(mE)
F.SetMass(mF)
G.SetMass(mG)
H.SetMass(mH)
I.SetMass(mI)
J.SetMass(mJ)
K.SetMass(mK)

A_Izz = (1/12)*mA*(a^2)
B_Izz = (1/12)*mB*(b^2)
C_Izz = (1/12)*mC*(c^2)
D_Izz = (1/12)*mD*(d^2)
E_Izz = (1/12)*mE*(e^2)
F_Izz = (1/12)*mF*(f^2)
G_Izz = (1/12)*mG*(g^2)
H_Izz = (1/12)*mH*(h^2)
I_Izz = (1/12)*mI*(i^2)
J_Izz = (1/12)*mJ*(j^2)
K_Izz = (1/12)*mK*(k^2)

% Total (translational and rotational) kinetic energies
A_kin = 0.5*mA*Dot(Acm>, Acm>) + 0.5*A_Izz*qa'
B_kin = 0.5*mB*Dot(Bcm>, Bcm>) + 0.5*B_Izz*qb'
C_kin = 0.5*mC*Dot(Ccm>, Ccm>) + 0.5*C_Izz*qc'
D_kin = 0.5*mD*Dot(Dcm>, Dcm>) + 0.5*D_Izz*qd'
E_kin = 0.5*mE*Dot(Ecm>, Ecm>) + 0.5*E_Izz*qe'
F_kin = 0.5*mF*Dot(Fcm>, Fcm>) + 0.5*F_Izz*qf'
G_kin = 0.5*mG*Dot(Gcm>, Gcm>) + 0.5*G_Izz*qg'
H_kin = 0.5*mH*Dot(Hcm>, Hcm>) + 0.5*H_Izz*qh'
I_kin = 0.5*mI*Dot(Icm>, Icm>) + 0.5*I_Izz*qi'
J_kin = 0.5*mJ*Dot(Jcm>, Jcm>) + 0.5*J_Izz*qj'
K_kin = 0.5*mK*Dot(Kcm>, Kcm>) + 0.5*K_Izz*qk'

Sys_Kin = A_kin + B_kin + C_kin + D_kin + E_kin +
F_kin + G_kin + H_kin + I_kin + J_kin + K_kin


% Power
P_A = Dot(mA*grav*nx>, v_Acm>) + Dot(-1*fric*w_A_B>,w_A_B>) + Dot(-1*fric*w_A_H>, w_A_H>)
P_B = Dot(mB*grav*nx>, v_Bcm>) + Dot(-2*fric*w_B_C>, w_B_C>)
P_C = Dot(mC*grav*nx>, v_Ccm>)
```

```
P_D = Dot(mD*grav*nx>, v_Dcm>) + Dot(-1*fric*w_D_C>, w_D_C>)
P_E = Dot(mE*grav*nx>, v_Ecm>) + Dot(-1*fric*w_E_N>, w_E_N>)
P_F = Dot(mF*grav*nx>, v_Fcm>)
P_G = Dot(mG*grav*nx>, v_Gcm>) + Dot(-2*fric*w_G_N>, w_G_N>) + Dot(-2*fric*w_G_D>, w_G_N>)
P_H = Dot(mH*grav*nx>, v_Hcm>)
P_I = Dot(mI*grav*nx>, v_Icm>) + Dot(-1*fric*w_I_D>, w_I_D>) + Dot(-1*fric*w_I_H>, w_I_H>)
P_J = Dot(mJ*grav*nx>, v_Jcm>)
P_K = Dot(mK*grav*nx>, v_Kcm>)

Sys_Power = P_A + P_B + P_C + P_D + P_E+ P_F+ P_G+ P_H+ P_I + P_J + P_K

Input_Power = Dt(Sys_Kin,N) - Sys_Power

% Defining the loop equations
ZeroVec1> = b*by> - e*ey> + y*ny> -x*nx> + c*cy>
ZeroLoop[1] = Dot(ZeroVec1>, nx>)
ZeroLoop[2] = Dot(ZeroVec1>, ny>)

ZeroVec2> = y*ny> - x*nx> + c*cy> -d*dy> - g*gy>
ZeroLoop[3] = Dot(ZeroVec2>, nx>)
ZeroLoop[4] = Dot(ZeroVec2>, ny>)

ZeroVec3> = f*fy> + h*hy> +i*iy> - g*gy>
ZeroLoop[5] = Dot(ZeroVec3>, nx>)
ZeroLoop[6] = Dot(ZeroVec3>, ny>)

ZeroLoop[7] = qf-qe-N
ZeroLoop[8] = qe-L-qa
ZeroLoop[9] = qk-qi-O
ZeroLoop[10] = pi-qi-qj-O-R

% Define Inputs

deg2rad=units(deg,rad)
rad2deg=units(rad,deg)

% Z has the inches to meters conversion built in
% (should give our actual lengths in meters)
% Z = 1
Z = 0.00563
% lengths in meters
a = 55.8*Z
b = 50.0*Z
c = 15.0*Z
d = 61.9*Z
e = 41.5*Z
```

```
f = 40.1*Z
g = 39.3*Z
h = 39.4*Z
i = 36.7*Z
j = 49.0*Z
k = 65.7*Z

x = 7.8*Z
y = 38.0*Z

L = acos((a^2+e^2-f^2)/(2*a*e))
N = acos((e^2+f^2-a^2)/(2*e*f))
O = acos((i^2+k^2-j^2)/(2*i*k))
R = acos((j^2+k^2-i^2)/(2*j*k))

% Define IC for crankshaft angle and angular velocity

Input mA = 0.156 kg, mB = 0.140 kg, mC = 0.042 kg, mD = 0.173 kg, mE = 0.116 kg,
mF = 0.112 kg, mG = 0.110 kg, mH = 0.110 kg, mI = 0.103 kg, mJ = 0.137 kg, mK = 0.184 kg
Input fric = 0.03, grav = 9.8 m/s^2


Input qc = 121*deg2rad
Input qc' = 2*pi

% SolveDt to get angles, omegas and alphas of every angle

SolveSetInputDt(ZeroLoop=0, qa=252*deg2rad, qb=160*deg2rad, qd=72*deg2rad,
qe=112*deg2rad, qf=158*deg2rad, qg=75*deg2rad, qh=66*deg2rad, qi=20*deg2rad,
qj=120*deg2rad, qk=28*deg2rad)

qc'' = 0

% Setup ODE to analyze variables change over time (footpath, KE etc)

Input tFinal = 4 sec, tStep = 0.01 sec
% Output t, qa, qb, qd, qe, qf, qg, qh, qi, qj, qk
% Output t, qa', qb', qd', qe', qf', qg', qh', qi', qj', qk'
% Plot footpath and power graph
OutputPlot Dot(foot>,Ny>), Dot(foot>,-Nx>)
OutPutPlot t, qc, Input_Power
ODE
```

10 References

[1] Brandon “AGr”, “Strandbeest Bike Mk2,” YouTube, May 04, 2019. https://www.youtube.com/watch?v=TYhyN1sDbz4 (accessed Oct. 04, 2024).

[2] Wikipedia Contributors, “Jansen’s linkage,” Wikipedia, May 28, 2019. https://en.wikipedia.org/wiki/Jansen

[3] K. Rudy, L. Fawzy, S. Bianco, T. Santelle, A. Ton, and Van Den Bogert, “The Jansen Linkage.” Available: https://www.csuohio.edu/sites/default/files/47A-2016.pdf

[4] Green, Brandon, “Rogue Strandbeest Mk2 Build [COMPLETE],” Dallas Makerspace Talk, Apr. 12, 2019. https://talk.dallasmakerspace.org/t/rogue-strandbeest-mk2-build-complete/49518/27 (accessed Oct. 05, 2024).

[5] Fox, Malcolm, “Polymer Tribology,” Lube Magazine No.135, Oct. 2016. https://www.lube-media.com/wp-content/uploads/2017/11/Lube-Tech106-PolymerTribology.pdf (accessed Nov. 01, 2024).