On Simple Planar Variable Geometry Trusses

Robert L. Read *

Founder, Public Invention, an educational non-profit.

January 27, 2018

DRAFT

1 Introduction

A *variable geometry truss* is a truss in which changes shape by means of the change of the length of members, in contrast to many robot arms which change joint angles. We call a member which can change length an *actuator*. A truss constructed purely by starting with a triangle and repeatedly adding two members and a new joint to one side of an existing triangle to form a new triangle is a *simple truss*. This paper is concerned only with simple planar trusses. In particular, we focus on trusses isomorphic to the Warren truss (e.g., an unbranching configuration of triangles.) Furthermore, although the world *truss* connotes forces and structural analysis, we are concerned purely with geometry. Although motivated by robotics, we assume that our trusses and actuators are strong enough that we need not consider the forces acting upon the truss—in other words, we are treating it as mechanism and not a machine.

If we imagine a joint, usually at one end of our truss, to be an end effector, the fundamental goal of this paper is to answer the question:

> How does a change in length of an actuator change the position of an end effector?

2 Formulation

A *truss* structure is a graph and a set of fixed nodes $T = (G, F)$. $G = (V, E)$. V is a set of nodes or joints. E is a set of lines or members which are 2-element subsets of V. At least two nodes are considered to be fixed in space via a set of fixed nodes $F = (i, x, y) \| i \in V x, y \in \mathbb{R}$.

*read.robert@gmail.com

We number joints from zero and designate them with a subscript. Members are designated by two subscripts, naming the joints they connect.

A *configuration* is a function mapping each member of a truss structure to a non-negative length. $C : V \times V \mapsto \mathbb{R}$.

A *geometry* is a placement of each joint in the Cartesian plane. $G : V \mapsto \mathbb{R} \times \mathbb{R}$.

A *simple truss* is a truss constructed from a triangle by adding a single joint and two members to a side repetitively. A *Warren truss* is an unbranched simple truss, which is isomorphic to a truss that is a single chain of equilateral triangles. Furthermore, we insist that each even node n occurs as a anti-clockwise turn (in the anti-clockwise semiplane) from the vector $\overrightarrow{n-2, n-1}$, and each odd node occurs as a clockwise turn (in the clockwise semiplane) from the previous to nodes.

For a Warren truss, there is a simple algorithm W for computing a geometry from a configuration: $W : C \mapsto G$.

A goal joint is a joint j such that $j \in V$ and a desired position given by a goal function $d : V \mapsto \mathbb{R}^2$. In robotics, a goal joint is often called an *end effector*.

A *scoring function* is a function that that takes a geometry and returns a real, non-negative value based on the goal node positions given by by a goal function and the actual position given by a Geometry. A score of zero is considered a perfect score and the a higher score represents a less sought-after result.

A *linear distance scoring function* is a special scoring function that that takes a geometry and returns a real, non-negative value based solely on a summation of distances between the goal node positions given by by a goal function and the actual position given by a Geometry.

A truss *problem* $P = (T, d, s)$ is a truss with a scoring function and a desired position function. A solution to a problem is a configuration and a geometry. An optimal solution is a solution which minimizes the score.

Our fundamental goal is to develop a formula for the partial derivative of the end effector with respect to the change in length of a member.

A further goal is to be able to render a diagram illustrating the impact on the end-effector of a change of each member, as drawn by rendering a vector from the center point of each member.

A further goal is to have algorithms to determine:

- What is the minimum overall change in length to a group of actuators to solve a Problem?
- Can a Problem be solved with a single change in length?
- Assuming bounds on the lengths of actuators, can we solve Problems?
- What is the workspace of a truss?

DRAFT

In the remainder of this paper we will consider only Warren trusses, linear distance scoring functions, and desired position functions that map only one node, called an end effector, to a desired position. Furthermore, we will assume that the first two nodes are fixed and that the furthest node (by path length) from the first node is the end effector.

3 Moving an External Member

We seek a formula for the change in position of the end effector e with respect to change in the length of a member given a geometry. In the case of a Warren truss, all members can be divided between external members and internal members. A change to the length of an external member is particularly simple.

The change in position is a centered on the goal node representing the place the goal node would move to with a unit change in length. However, the derivative is only valid as an infinitessimal, but as an infinitessimal its magnitude and direction may be usefully added to or compared to other such vectors. We can thus usefully tell which member would change the position of the goal node most rapidly in response to a minute change in different members.

The fundamental observation is that for an external member (C, A), a change in length generates a rotation θ about joint B whose position is given by (B_x, B_y). θ is the angle of the vector from the pivot point B to the end-effector E with the x-axis. This rotation applies to the triangle defined by three joints: $\triangle B, C, E$. Not that this triangle doeS not exist as a physical structure in the truss.

By using the law of cosines, where $a = \|\overrightarrow{A, B}\|, b = \|\overrightarrow{A, C}\|, c = \|\overrightarrow{B, C}\|$, where b is the member opposite the pivot joint B which changes the angle $\phi_{i-1} = \angle ABC$. In other words, ϕ is a signed angle measure BA moved into BC, with positive representing anti-clockwise.

$$\cos\phi = \frac{a^2 - b^2 + c^2}{2ac}$$

or

$$\phi = \arccos\frac{a^2 - b^2 + c^2}{2ac}$$

Using Woflram Alpha to differentiate this, we obtain:

$$\frac{\partial\phi}{\partial b} = \frac{b}{ac\sqrt{1 - \frac{(a^2+c^2-b^2)^2}{4a^2c^2}}}$$

This derivation loses the sign information, which we recover by considering whether $S = \operatorname{sign}\phi$.

The change in the end effector is always perpendicular to the drawn from the pivot joint to the effector and proportional to its length. An alternative way of looking at this

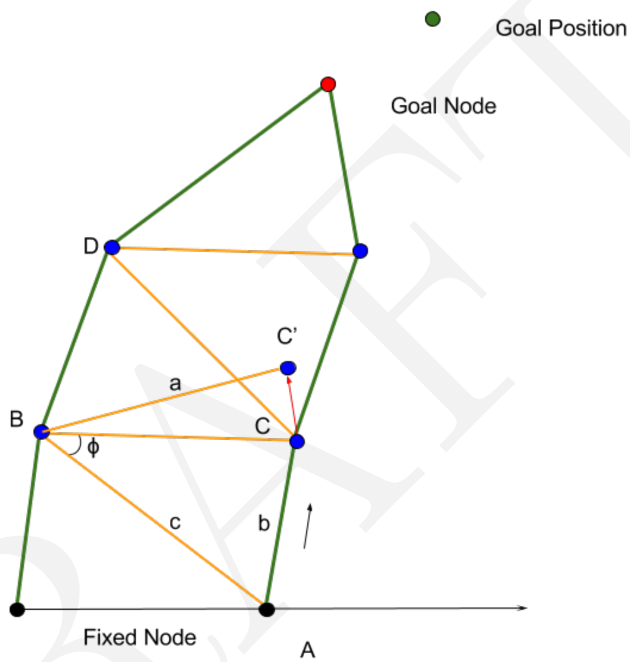

Figure 1: A Trusss Problem with Change to an External Member Length

DRAFT

is that the result should be orthogonal to the vector

$$\begin{bmatrix}(e_x - x)\\(e_y - y)\end{bmatrix}$$

In other words:

$$S \cdot \begin{bmatrix}-(e_y - y)\\(e_x - x)\end{bmatrix}$$

With the sign S is negative if ϕ is clockwise. ϕ cannot be zero or equal or exceed π in a physical machine is excluded for that reason.

Since the change in ϕ is equal to the change in θ, we can use the chain rule:

$$\frac{\partial e}{\partial b} = \begin{bmatrix}\frac{\partial e_x}{\partial b}\\\frac{\partial e_y}{\partial b}\end{bmatrix} = \frac{Sb}{ac\sqrt{1 - \frac{(a^2+c^2-b^2)^2}{4a^2c^2}}}\begin{bmatrix}-(e_y - y)\\(e_x - x)\end{bmatrix}$$

So this is a closed-form expression for the change in the position of an end effector for any external member $i, i-2$ we choose. If we then know how the scoring functions changes as the end effector moves, we can compute the change in the scoring functions as we change $b = \|AC\|$, which is what we need for numeric optimization.

4 Moving an Internal Member

In the Internal Member Length change diagram, $\angle ABC = \beta$, $\angle CBD = \psi$, $\angle BCD = \chi$, $\angle ACB = \gamma$, $\angle BDC = \delta$, and $\angle BAC = \alpha$. The lengths are marked a, b, c, f, g, with a being opposite node A and a diagonal of the quadrilateral $ABDC$.

Furthermore, $\angle ABD = \beta + \psi$, and $\angle ACD = \gamma + \chi$.

The absolute angle between the line between the fixed nodes A and B and the x-axis is ρ (counting positive as anticlockwise from the x-axis.) Both A and B are considered fixed, but both C and D move as a changes. D rotates about B and C rotatest about A.

Because the member DC does not change its length, we conceptualize the impact of a change to length a as translating between C and D and rotating BC about M. By knowing the change in the absolute rotation and the change in the position of M as a changes infintessimally, we know how the end effector E changes.

We can compute $\frac{\partial \beta}{\partial a}$ directly (using Wolfram Alpha) from ϕ:

$$\beta = \arccos\frac{a^2 - b^2 + c^2}{2ac}$$

$$\frac{\partial \beta}{\partial a} = -\frac{\frac{1}{c} - \frac{a^2-b^2+c^2}{2a^2c}}{\sqrt{1 - \frac{(a^2-b^2+c^2)^2}{4a^2c^2}}}$$

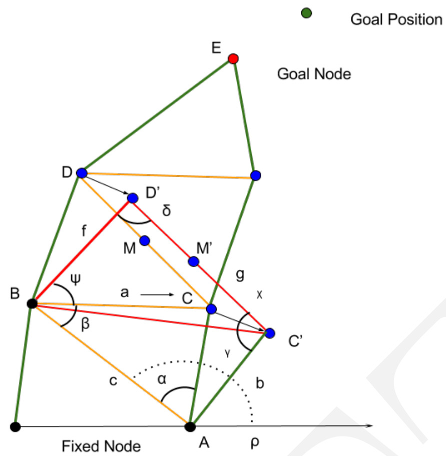

Figure 2: A Trusss Problem with Change to an Internal Member Length

Calculating the same way for α in order to have C

$$\alpha = \arccos \frac{-a^2 + b^2 + c^2}{2bc}$$

$$\frac{\partial \alpha}{\partial a} = \frac{a}{bc\sqrt{1 - \frac{(-a^2+b^2+c^2)^2}{4b^2c^2}}}$$

$$\gamma = \pi - (\alpha + \beta)$$

Now we compute χ from the law of cosines:

$$\chi = \arccos \frac{a^2 - f^2 + g^2}{2ag}$$

$$\frac{\partial \chi}{\partial a} = -\frac{\frac{1}{g} - \frac{a^2-f^2+g^2}{2a^2g}}{\sqrt{1 - \frac{(a^2-f^2+g^2)^2}{4a^2g^2}}}$$

$$\begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} A_x + b\cos(\rho - \alpha) \\ A_y + b\sin(\rho - \alpha) \end{bmatrix}$$

Taking the derivative of this with respect to a,

$$\begin{bmatrix} \frac{\partial C_x}{\partial a} \\ \frac{\partial C_y}{\partial a} \end{bmatrix} = \begin{bmatrix} 0 + \frac{\partial b \cos(\rho-\alpha)}{\partial a} \\ 0 + \frac{\partial b \sin(\rho-\alpha)}{\partial a} \end{bmatrix} = \begin{bmatrix} b \frac{a \sin(\rho-\alpha)}{bc\sqrt{1-\frac{(-a^2+b^2+c^2)^2}{4b^2c^2}}} \\ -b \frac{a \cos(\rho-\alpha)}{bc\sqrt{1-\frac{(-a^2+b^2+c^2)^2}{4b^2c^2}}} \end{bmatrix} = \frac{a}{c\sqrt{1-\frac{(-a^2+b^2+c^2)^2}{4b^2c^2}}} \begin{bmatrix} \sin(\rho-\alpha) \\ -\cos(\rho-\alpha) \end{bmatrix}$$

The angle θ between CD and the x-axis is therefore given by:

$$\begin{aligned} \theta &= \\ (\rho-\alpha) + (\pi - (\chi+\gamma)) &= \\ \rho + \pi - (\alpha + \chi + (\pi - (\alpha+\beta)) &= \\ \rho + \pi - (\chi + (\pi - \beta)) &= \\ \rho + -\chi + \beta & \end{aligned}$$

We seek $\frac{\partial \theta}{\partial a}$. However, χ and β, computed from arccos, are "interanl angles". We need to seek absolute orientation.

In computing all of this, one must be very careful to keep the signs straight.

In fact, the cleanest way to keep angles straight is to convert angles quickly from "internal angles" between two vectors and compute the "absolute orientation" or vectors (that is, the angle measure anticlockwise against the x-axis.

The computation of $\frac{\partial \theta}{\partial a}$ above should be considered the computation of an "absolute orientation".

We can think of the computation of $\theta = \rho - \chi + \beta$ is a zig zag compuation. To find the change in the rotation of θ, we find the change in χ and then the change in β.

We will say the "sense" of an angle $\angle XYZ$ is clockwise or anticlockwise depending on if the smallest turn from X to Z through Y is clockwise or anitclockwise.

$$\text{sense}\,\angle XYZ = \begin{cases} 0 & \angle XYZ = \pi \\ 1 & \text{ if } \angle XYZ \text{ is anticlockwise moving } \overrightarrow{Y,X} \text{ into } \overrightarrow{Y,Z} \\ -1 & \text{otherwise} \end{cases}$$

The orientation $\tau = \overrightarrow{B,C}$ is thus $\rho + -(\pi - \|\beta\|)$ if $\angle ABC$ is clockwise, and $\rho + (\pi - \|\beta\|)$ if $\angle ABC$ is anticlockwsie.

The orientation $\theta = \overrightarrow{C,D}$ is thus $\tau + -(\pi - \|\chi\|)$ if $\angle BCD$ is clockwise, and $\tau + (\pi - \|\chi\|)$ if $\angle BCD$ is anticlockwise.

$$\begin{aligned} \theta &= \rho + (\text{sense}\,\angle ABC)(\pi - \|\beta\|) + (\text{sense}\,\angle BCD)(\pi - \|\chi\|) \\ &= \rho + (\text{sense}\,\angle ABC)\pi + (\text{sense}\,\angle BCD)\pi + -(\text{sense}\,\angle ABC)\|\beta\| + -(\text{sense}\,\angle BCD)\|\chi\| \end{aligned}$$

DRAFT

The quantities π and sense $\angle ABC$ and sense $\angle BCD$ don't depend on a.

$$\begin{aligned}
\frac{\partial\theta}{\partial a} &= 0 + -(\text{sense}\,\angle ABC)\frac{\partial\beta}{\partial a} + -(\text{sense}\,\angle BCD)\frac{\partial\chi}{\partial a} \\
&= -(\text{sense}\,\angle ABC)\frac{\frac{1}{c} - \frac{a^2-b^2+c^2}{2a^2c}}{\sqrt{1-\frac{(a^2-b^2+c^2)^2}{4a^2c^2}}} + -(\text{sense}\,\angle BCD)\frac{\frac{1}{g} - \frac{a^2-f^2+g^2}{2a^2g}}{\sqrt{1-\frac{(a^2-f^2+g^2)^2}{4a^2g^2}}}
\end{aligned}$$

Now E is simply a translation by $\frac{\partial C}{\partial a}$ by a rotation about C, which as we have previously shown

$$\frac{\partial e}{\partial a} = \begin{bmatrix} \frac{\partial C_x}{\partial a} \\ \frac{\partial C_y}{\partial a} \end{bmatrix} + \frac{\partial\theta}{\partial a}\begin{bmatrix} -(e_y - C_y) \\ (e_x - C_x) \end{bmatrix}$$

$$\frac{\partial e}{\partial a} = \frac{a}{c\sqrt{1-\frac{(-a^2+b^2+c^2)^2}{4b^2c^2}}}\begin{bmatrix} \sin(\rho-\alpha) \\ -\cos(\rho-\alpha) \end{bmatrix} + \frac{\partial\theta}{\partial a}\begin{bmatrix} -(e_y - C_y) \\ (e_x - C_x) \end{bmatrix}$$

5 Obstacles

Obstacles may be added to the universe by successfully modifying both the derivative and the objective consistently.

For example, an obstacle may be modelled as a sphere, which provides a positive value for each node which is in the sphere, and zero for all those not in the sphere. However, it is better to make this smooth, by for example a polynomial based on the distance from the center of the sphere. This allows a derivative to be computed as a direction. The derivative for a given node will accurately reflect the change in the value of the objective function as moving toward the the center.

In order to use a gradient-based method with obstacles, we must have a practical way of composing obstacles into the space of the configuration of the VGT into the objective function, and we must be able to compute an accurate derivative of the objective function. A naive approach would be to use a flat space which provides a heavy penalty for a node inside the obstacle. However, it makes more sense to have an objective function which provides directional guidance to "push" a node out of the obstacle. I therefore propose to model an obstacle as disc around a center point and a radius, with the an increase in the objective function:

$$\mathcal{O}(\overrightarrow{c}, r, \overrightarrow{x}) = \big(\frac{1}{1+d^2} + 1\big) \cdot \mathcal{H}(r-d)$$

where $d = \| \overrightarrow{c} - \overrightarrow{x} rVert$, and $\mathcal{H}(a,b)$ is the Heaviside step function:

DRAFT

$$\mathcal{H}(n) = \begin{cases} 1, & \text{if } n < 0 \\ 0, & \text{if } n \geq 0 \end{cases}$$

The contribution of the obstacle $\mathcal{O}$ to the objective function is:

$$g(\mathcal{O}(\overrightarrow{c}, r, \overrightarrow{x})) = \sum_{n \in G} \mathcal{O}(c, r, \overrightarrow{n})$$

which allows us to compute the derivative with respect to the change of any member a:

$$\begin{aligned} \frac{\partial}{\partial a} g(\mathcal{O}(\overrightarrow{c}, r, \overrightarrow{x})) &= \frac{\partial}{\partial a} \sum_{n \in G} \mathcal{O}(c, r, \overrightarrow{n}) \\ &= \sum_{n \in G} \frac{\partial}{\partial a} \mathcal{O}(c, r, \overrightarrow{n}) \end{aligned}$$

Considering the contribution of a single node n:

$$\frac{\partial}{\partial a} \mathcal{O}(c, r, \overrightarrow{n}) = \frac{\partial \mathcal{O}(c, r, \overrightarrow{n})}{\partial d(n, c)} \cdot \frac{\partial d(n, c)}{\partial a}$$

where:

$$\frac{\partial d(n, c)}{\partial a} = \frac{\partial \overrightarrow{n}}{\partial a} \cdot (\overrightarrow{n} - \overrightarrow{c})$$

(where $\cdot$ is the inner product.) Note that these two factors are very easy to compute; the previous section gives the formulae for the change in position of a node with respect to the change in the length of a member.

$$\frac{\partial \mathcal{O}(c, r, \overrightarrow{n})}{\partial d} = \frac{\partial}{\partial d} \Big(\frac{1}{(1 + d)^2} + 1 \Big) \cdot \mathcal{H}(r - d)$$

The partical derivative of the Heaviside step function will in fact be discontinous when $r - d = 0$. However, this is largely irrelevant from a numerical optimization point of view, so long as we can actually compute the correct derivative. At this (technically unlikely to occur) position we could simply allow the derivative to jump. If we discover that discontinuity of the derivative is causing a problem, we could use one of several approximations mentioned by the Wikipedia article `https://en.wikipedia.org/wiki/Heaviside_step_function`.

Thanks to Wolfram Alpha, we know this derivative is:

$$\frac{\partial \mathcal{O}(c, r, \overrightarrow{n})}{\partial d} = -\Big(\frac{1}{(d + 1)^2} + 1 \Big) \delta(r - d) - \frac{2\mathcal{H}(r - d)}{(d + 1)^3}$$

where δ is the dirac delta function, which has a non-zero value only when $r = d$. It is realitively easy to test for this condition in computer code, as is the computation of the Heaviside step function, so the partial derivative is easily computed.

DRAFT

6 Testing

The best way to test this is to build a diagram that draws the vector $\frac{\partial e}{\partial l_{BC}}$ at at the center of each edge $\vec{BC}$. This should allow both the direction and magnintude to be visualized in a reasonable way. We could call this a *delta diagram*.

Such a "Delta Diagram" system has been built. We should, however, build a system, such that we can compare the contributions from various components in the derivative. That is, we wish to render the total derivative, and its components (head to tail.)

7 Open Questions And ToDos

What is the actual time it takes to do an optimization?

Answer: Less than 50 millisconds for 40 notes, about 100 milliseconds for 100.

Is it so fast that we could do motion planning and robot avoidance?

Probably yes!!!

Can we add other weights into the Derivatives and Objective (such as all joints close to median, or as large as possible) and compute somehting nicely?

- Decide if it is better to use three.js and javascript as a front end.
- Feature: Add obstacles — can handle with our objective/derivative system?
- Can we make a system of combining objective/derivative systems?
- Can we draw a line, and quickly compute regular motion along this line?
- Can we thread through obstacles via motion?

8 Unvetted References

A starting point is: [1].

This paper use gradient computation in a hybrid approach:

`http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=478427`

It very explicitly uses "backbone curve" and then fits manipulators to it. This says : "In fact, the need to derive expressions for these derivatives for the parallelherial structures often used in real hyper-redundant systems is a major drawback of the Jacobian based methods."

However, in the case of this statically determinate VGTs it doesn't seem to bad.

This article is interesting:

Kinematic transformations for remotely-actuated planar continuum robots

I really need to read this:

"Tetrahedral Robotics for Space Exploration"

DRAFT

Need to understand this:

Cite this chapter as: Gilbert H.B., Rucker D.C., Webster III R.J. (2016) Concentric Tube Robots: The State of the Art and Future Directions. In: Inaba M., Corke P. (eds) Robotics Research. Springer Tracts in Advanced Robotics, vol 114. Springer, Cham

References

[1] Kazuyuki Hanahara and Yukio Tada. Variable geometry truss with sma wire actuators (basic consideration on kinematical and mechanical characteristics). 01 2008.

DRAFT