Exploring Solvability of the String Link Concordance Group Using Milnor's Invariants

Madison Ford[1], Benjamin Pagano[2], Sarah Pritchard[3], and Erin Wilkerson[4]

[1]Wayne State University, [2]Occidental College, [3]Georgia Institute of Technology, [4]Clark Atlanta University

Summer 2020

Abstract

Linking number is a tool used to measure how "linked" two components of a link or string link are or, rather, how hard it is to separate them. We use Milnor's invariants, which are higher order versions of linking number, to explore how close $\mathcal{C}(2)/\mathcal{P}(2)$ is to being abelian (in other words, the solvability of this group). This project uses two methods to calculate the Milnor's invariants of string link commutators within this group. The first method involves generating surfaces bounded by each component of the string link, then taking their intersections in an iterative process. The second method involves deriving Milnor's invariants using group theoretic techniques using the fundamental group of the link complement. There are a few cases where the Milnor's invariants of a string link are always zero, meaning the components are trivially linked and concordant to the unlink. For more complex links, there may be non-zero Milnor's invariants. Our research focuses on the Milnor's invariants of commutators in the string link group and when they are zero or non-zero. We proved that there is a class of string links for which the Milnor's invariants are always zero. We also developed several tools for more quickly calculating Milnor's invariants in non-trivial examples. In further research, we hope to prove that $\mathcal{C}(2)/\mathcal{P}(2)$ is not solvable by showing that there are non-trivial commutators of arbitrary length in $\mathcal{C}(2)/\mathcal{P}(2)$.

1 Introduction

As far back as the 17th century, mathematicians have worked to comprehend the "shape" and features of spaces. They developed ideas which lead to the concept of topology: the study of the shape of spaces. This paper focuses on low-dimensional topology (dimension less than or equal to 4). In particular, we discuss knots and links and the spaces in which they exist.

Definition 1.1. *A knot is a smooth embedding* $\sigma : S^1 \to S^3$.

An n-component link is a set of multiple knots, often linked together but never intersecting. Formally,

Definition 1.2. *An n-component link is a smooth embedding* $\sigma : \bigsqcup_n S^1 \to S^3$.

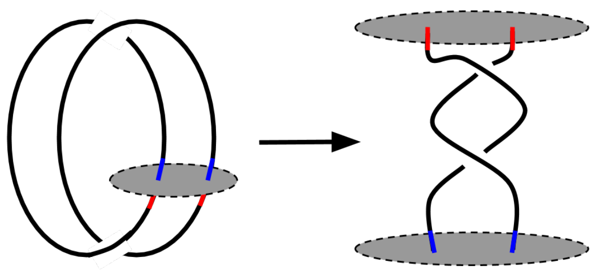
There is no n-component link group. Because the connected sum operation for knots is not well defined on links, more work is needed to define a group operation. We overcome this by presenting links as string links. String links are a generalization of pure braids; they are very similar to braids but are allowed to loop back on themselves in ways that pure braids cannot.

Definition 1.3. *[1] Let D be the unit disk, I the unit interval, and $\{p_1, p_2, \ldots, p_k\}$ be n points in the interior of D. An n-component string link is a smooth, proper embedding $\sigma : \bigsqcup_n I \to D \times I$ such that*

$$\sigma|_{I_i(0)} = \{p_i\} \times \{0\}$$
$$\sigma|_{I_i(1)} = \{p_i\} \times \{1\}$$

We can turn a link into a string link by first adding an oriented disk to a link intersecting each component exactly once with all intersections positive and then slicing the disk to pull the link into a braid-like presentation.

Example 1.1. *Turning the Hopf link into a string link.*

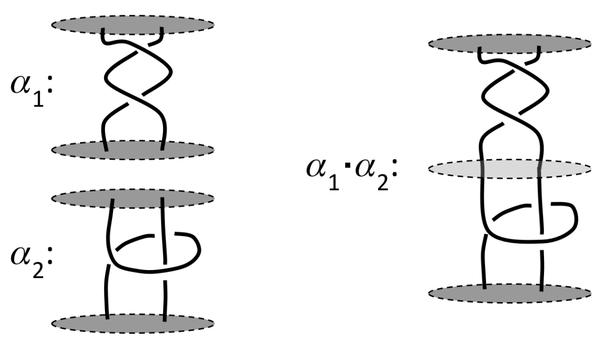


Similarly, the closure of a string link is a link.

Definition 1.4. *Let c be a string link. Then let $\hat{c}$ be the closure of c, formed by attaching arcs to the top and bottom of the string link.*

String links can be formed by *stacking*. The following example is an intuitive description of the composition process:

Example 1.2. *Let α_1, α_2 be string links. Then, denote the composition of α_1 and α_2 by $\alpha_1 \cdot \alpha_2$, formed by stacking α_1 on top of α_2 and connecting the two string links, as shown below.*



Formally,

Definition 1.5. *Let $\sigma : \bigsqcup_n I \to D \times I$ and $\gamma : \bigsqcup_n I \to D \times I$ be string links. We define $\sigma \cdot \gamma$ as the embedding $\lambda : \bigsqcup_n I \to D \times I$ where*

$$\lambda|_{I_i(x)} = \sigma|_{I_i(2x)} \qquad 0 \leq x \leq \frac{1}{2}$$
$$\lambda|_{I_i(x)} = \gamma|_{I_i(2x-1)} \qquad \frac{1}{2} \leq x \leq 1$$

The last stipulation for forming a string link group is that two string links are considered equal in the group if and only if those string links are concordant; that is, concordance is an equivalence relation on the group.

Definition 1.6. *[1] Two n-component string links σ_1, σ_2 are concordant if there is a smooth embedding $H : \bigsqcup_n (I \times I) \to B^3 \times I$ such that:*

$$H|_{(\bigsqcup_n I \times \{0\})} = \sigma_1$$
$$H|_{(\bigsqcup_n I \times \{1\})} = \sigma_2$$
$$H|_{(\bigsqcup_n \partial I \times I)} = j_0 \times id_I$$

where $j_0 : \bigsqcup_n \partial I \to S^2$.

This provides us with inverses; two elements a, a^{-1} are inverses if aa^{-1} is concordant to the identity in the group: the unlink.

With these definitions in place, we can begin discussing the group of interest to us: $\mathcal{C}(n)/Ncl(\mathcal{P}(n))$. $\mathcal{C}(n)$ refers to the group of n-component string links under stacking, where string links are equivalent under concordance. $\mathcal{P}(n)$ is the pure braid subgroup of $\mathcal{C}(n)$. Briefly, we will review the definition of pure braids for those unfamiliar.

Pure braids on n strands are a subgroup of the braid group on n strands, and subsequently, a subgroup of the string link concordance group. Pure braids cannot loop back on themselves in the way that string links may, though. One can think of a pure braid as a parametrization of a falling object, where each strand must continually move down as it moves horizontally. Formally,

Definition 1.7. *Let D be the unit disk, I the unit interval, and $\{p_1, p_2, \ldots, p_k\}$ be n points in the interior of D. An n-component pure braid is a smooth, proper embedding $\sigma : \bigsqcup_n I \to D \times I$ such that*

$$\sigma|_{I_i(0)} = \{p_i\} \times \{0\}$$
$$\sigma|_{I_i(1)} = \{p_i\} \times \{1\}$$
$$\sigma|_{I_i(x)} = \{q_{i,x}\} \times \{x\}$$

where $q_{i,x} \in D$.

A braid can be turned into a link by taking its closure (attaching arcs at the top and bottom).

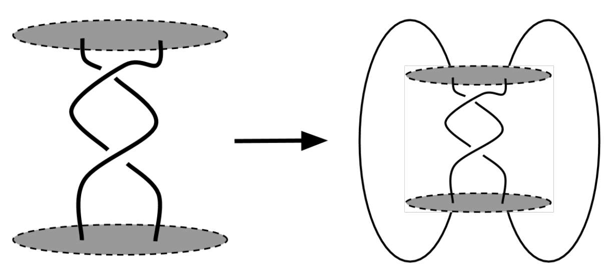


Figure 1: Turning a braid into a link by attaching arcs/strings at the top and bottom

The stacking operation for $\mathcal{P}(n)$ is defined in the same way as the stacking operation in $\mathcal{C}(n)$. With these groups in mind, we can move to the primary focus of our study: whether $\mathcal{C}(n)/Ncl(\mathcal{P}(n))$ is solvable. To understand solvability, we will need to discuss two group-theoretic concepts: commutators and the derived series.

$\mathcal{C}(n)$ is known to be non-abelian because it contains $Ncl(\mathcal{P}(n))$ as a subgroup. Kuzbary [1] shows that it is still non-abelian if you quotient out $Ncl(\mathcal{P}(n))$.

Definition 1.8. *For $a, b \in G$, the commutator of a and b, written $[a, b]$, is the element $aba^{-1}b^{-1} \in G$.*

As the name implies, commutators give a sense of which elements in a group commute. If $[a, b] = id_G$, then $ab = ba$, i.e. a and b commute. In an abelian group, all commutators are equal to the identity, so they do not provide much information. In a non-abelian group, however, looking at iterations of commutators can help measure just how close to being abelian a non-abelian group is.

To this end, we employ the concept of a derived series for a group G, a sequence of nested commutator subgroups.

Definition 1.9. *The commutator subgroup $[G, G]$ is the set of all commutators in G. Formally, $[G, G] = \{aba^{-1}b^{-1} | a, b \in G\}$*

Definition 1.10. *The derived series of G is a sequence of commutator subgroups defined recursively as:*

$$G^{(0)} = G \qquad n = 0$$
$$G^{(n)} = [G^{(n-1)}, G^{(n-1)}] \qquad n \geq 1$$

With this, we can define solvability.

Definition 1.11. *A group G is solvable if $G^{(n)} = 1$ for some $n \geq 1$. In other words, G is solvable if some member of its derived series is the trivial subgroup.*

Recently, Kuzbary showed that the group we are working with is non-Abelian [1]. That is,

Theorem 1.1. *$\mathcal{C}(n)/Ncl(\mathcal{P}(n))$ is non-abelian for every n.*

In our investigation, we sought to expand this result and prove that $\mathcal{C}(n)/Ncl(\mathcal{P}(n))$ is not solvable. In pursuit of this goal, our team utilized two distinct methodologies, assessing string link commutators using surface systems and using the fundamental group of the string link complement. The focus of the project narrowed as we worked only in $\mathcal{C}(n)/Ncl(\mathcal{P}(n))$ for $n = 2$. Conveniently, this group is $\mathcal{C}(2)/\mathcal{P}(2)$, since $\mathcal{P}(\in) = Ncl(\mathcal{P}(\in))$, which is not true for $n > 2$. $\mathcal{P}(n)$ is not a normal subgroup of $\mathcal{C}(n)$. Considering only this group, closures of two-strand pure braids may have a nonzero linking number.

Linking number involves assigning a value (either 1 or -1) to each crossing. Assume that an oriented link with two components, K and J, has a standard projection. If a particular crossing is right-handed – J crosses under K from the right to the left – then the sign of that crossing is 1. But if J crosses under K from the left to the right, the sign of said crossing is -1, shown in figure 2.

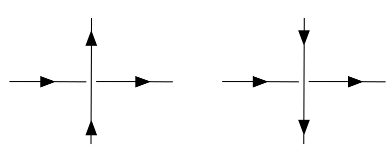


Figure 2: A left hand crossing (left) and right hand crossing (right)

Definition 1.12. *The linking number of K and J, denoted $\ell k(K, J)$, is the sum of the signs of all the crossings where Kpasses under J.*

Note that linking number is symmetric. That is, $\ell k(J, K) = \ell k(K, J)$ for two components J, K.

Milnor's invariants give us a natural way to detect subtle higher order linking data. If given an oriented, ordered link L, the Milnor's invariants of L are integers corresponding to a multi-index which can be computed in multiple ways.

In the following sections, we will explore the two methods used to compute Milnor's invariants. With the group presentation method, we use fundamental groups and Magnus expansions to help compute Milnor's invariants. With surface systems, we construct figures from which we can calculate Milnor's invariants as combinations of linking numbers.

2 Computing Milnor's Invariants Using Group Presentations

One way to calculate the Milnor's invariants for a string link is to consider the structure of the fundamental group of the link complement.

Definition 2.1. *[2] Given a link L, the fundamental group of its complement* $\pi_1(S^3\backslash v(L), p)$ *is the set of all homotopy classes* $[f]$ *of loops* $f : I \to S^3\backslash v(L)$ *such that*

$$f(0) = f(1) = p$$

for some fixed $p \in S^3\backslash v(L)$. *The group operation is concatenation of loops.*

Essentially, this is a group of all loops in the space around L which begin and end at a fixed point p. Elements of this group are considered equivalent under homotopy. This means that if one loop can be deformed into another without crossing the link, then those loops are equivalent.

As it happens, there is an easy way to present this fundamental group using the diagram for the link. To each arc in the diagram we assign a generator of the group. This generator is a loop going around that arc oriented so that it has positive linking number with the component it links. Figure 3 shows three such generators near a crossing. It is always assumed that such loops are based at a point p far above the link, but often we will only draw a portion directly under the link diagram for simplicity.

Once generators are determined, we can use crossings to deduce relations for the group. In figure 3, we see a right hand crossing with generators a, b, and c.

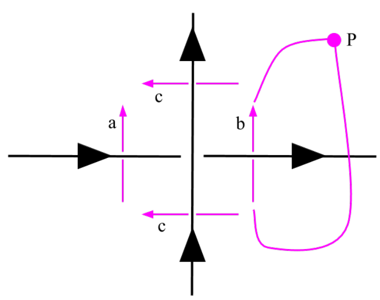


Figure 3: Four fundamental group generators near a crossing

Imagine concatenating these generators to create the loop $bca^{-1}c^{-1}$, shown in figure 4. This loop need only be based at P at its start and end, so we can homotope other meeting points for the generators away from P, as shown. The resulting loop can be pulled out from under the crossing and is in fact homotopic to the constant loop at the base point. Thus, this is the identity in the fundamental group and we can write $bca^{-1}c^{-1} = id_G$.

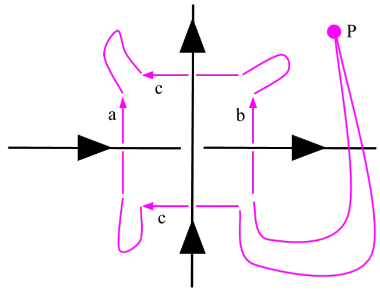


Figure 4: Four fundamental group generators concatenated

This process can be repeated to find a relation for every crossing. Along with the list of generators, this forms a presentation for the fundamental group known as the Wirtinger presentation.

We will try and write the longitude of a component of our link as an element of the fundamental group of the link complement, and we will use this presentation to do so. This longitude can then be used to directly compute the Milnor's invariants of the link.

2.1 0-framed Longitudes

As will be discussed in section 3, each knot bounds many surfaces. One such surface can be found algorithmically using the Seifert algorithm. We will use this fact to define the 0-framed longitude for a knot.

Definition 2.2. *The* 0*-framed longitude of a knot K is the curve formed by the intersection of a Seifert surface for K with the boundary torus $\partial(S^3 \backslash v(K))$.*

This longitude can be written as a word in the fundamental group of the link complement. We would like to write this longitude using only one generator from each component. If we can do so, it is possible to use a mapping known as the Magnus expansion to directly compute Milnor's invariants for the link.

Definition 2.3. *The Magnus expansion is a mapping from the free group with n generators to the ring of power series in n noncommuting variables such that*

- $x_i \to 1 + X_i$
- $x_i^{-1} \to 1 - X_i + X_i^2 - X_i^3 + \dots$

The Magnus expansion of a word w made from n generators can be written $M(w) = 1 + \sum_I \epsilon_I X^I$ where the sum is taken over all possible multi-indices $I = (i_1, \dots, i_m)$ where $1 \le i_j \le n$. In this notation $X^I = X^{i_1} \dots X^{i_m}$. This notation will prove useful for the next definition.

For this group presentation method, the definition of Milnor's invariants we will use depends on this Magnus expansion.

Definition 2.4. *[3] Let L be an n-component link and $G = (\pi(S^3 \backslash v(L), p), *)$ be the fundamental group of its complement. Let l_{i_k} be the i_k^{th} longitude of L and let $R_k(l_{i_k})$ be its image in G/G_k. Using the notation from the previous definition, let $M(R_k(l_{i_k})) = 1 + \sum_I \epsilon_I X^I$. The Milnor's invariants of L are integers $\overline{\mu}(i_1, \dots, i_k)$ where $\overline{\mu}(i_1, \dots, i_k) = \epsilon_I(R_k(l_{i_k}))$.*

This definition gives us a useful way of calculating Milnor's invariants of a link with reference to the fundamental group of its complement. The major difficulty for this approach is getting the longitude in a form where it uses only one generator from each component. As it happens, some generators in the Wirtinger presentation are redundant, so it is possible to reduce the longitude using only information from crossing relations. However, this is often not enough. The next steps involve modding out the fundamental group by elements of its lower central series.

Definition 2.5. *The lower central series of a group G is the sequence of nested commutator subgroups $G_1 \trianglerighteq G_2 \trianglerighteq G_3 \trianglerighteq \ldots$ such that $G_1 = G$ and $G_n = [G, G_{n-1}]$ for $n \geq 2$.*

Rewriting the 0-framed longitude of a link component mod G_q and taking its Magnus expansion will yield some power series. If that power series has a non-constant term, then the coefficient of the first non-constant term is the first non-vanishing Milnor's invariant for the link. Milnor's invariants are defined as follows [1].

Definition 2.6. *Let $L \subset S^3$ be an oriented, ordered link and let $G = \pi_1(S^3 \backslash v(L), *)$ be the fundamental group of its complement. The Milnor's invariants of L are integers $\bar{\mu}(i_1...i_k)$ each corresponding to a multi-index $(i_1, ..., i_k)$ where $i + j \in \{1, ..., n\}$. Let l_{i_k} be the ${i_k}^{th}$ longitude of L and let $R_k(l_{i_k})$ be its image in G/G_k. Expressed in the generators found in Theorem 2.36 of Kuzbary [1], this group element corresponds to a word w in meridians $x_1, ..., x_n$. The Magnus expansion of this word is $M(w) = 1 + \Sigma_{I \in I} X^I$ where the sum is taken over all possible multi-indices $I = (j_1, ..., j_m)$ and X^I is shorthand for $x^{j_1}....X^{j_m}$. Then,*

$$\bar{\mu}((i_1...i_k) =\in_I (R_k(l_{i_k}))$$

this integer is well-defined if all the Milnor's invariants of order less than k are 0. Otherwise, this integer is defined to be the residue class modulo

$$\Delta = gcd\{\bar{\mu}(\tilde{I})\}$$

where $\tilde{I}$ is obtained from $i_1, ..., i_k$ by removing one index and cyclically permuting the other indices.

Many of our results deal with this rewriting process and attempt to make it easier, giving us tools to prove that there exist string link commutators of arbitrary size with non-zero Milnor's invariants. Since Milnor's invariants are defined modulo other Milnor's invariants of smaller weight, we are only concerned with the first non-zero Milnor's invariants.

3 Computing Milnor's Invariants Using Surface Systems

3.1 Seifert Surfaces

Seifert's theorem states that every knot is the boundary of an orientable surface [4].

Definition 3.1. *Let k be a knot and S be an oriented surface such that the boundary of S is k. Then, S is a Seifert surface for k.*

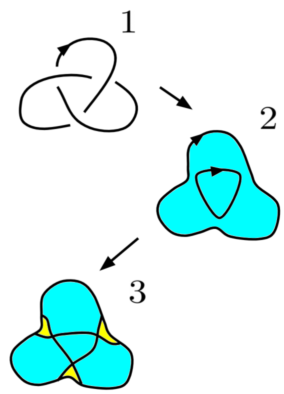
Seifert's algorithm for generating a Seifert surface for a given knot k is as follows.

1. Orient the knot k.

2. Resolve each crossing (that is, smooth the knot so it no longer has any crossing) in a manner that respects the orientation.

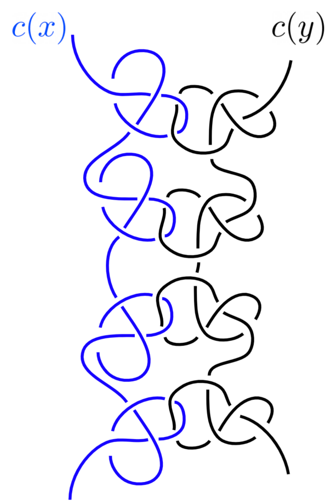
3. At the location of each crossing that was smoothed, add a twisted band.

Example 3.1. *Finding a Seifert surface for the trefoil using Seifert's algorithm.*

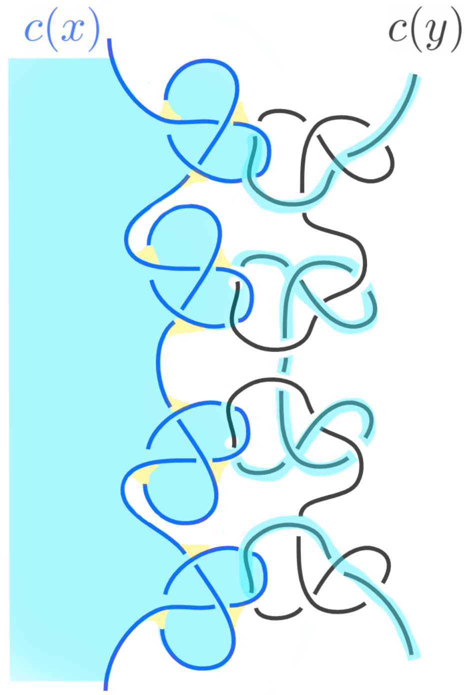


To compute the Milnor's invariants of a string link commutator using surface systems, we will perform a variation on Seifert's algorithm. Consider the string link below.

Example 3.2. *Finding a Seifert surface for the component c(x).*



To find a Seifert surface for the component of the link on the left, we can take the closure of the string link, then select one knotted section of the component and perform Seifert's algorithm as if the knot were closed. We will do this for each knotted section until the entire component bounds a surface. Then, to ensure that the surface bounded by this component is disjoint from the second component of the link, we can add hollow tubes for the other component to pass through in a manner that respects the orientation of the surface.

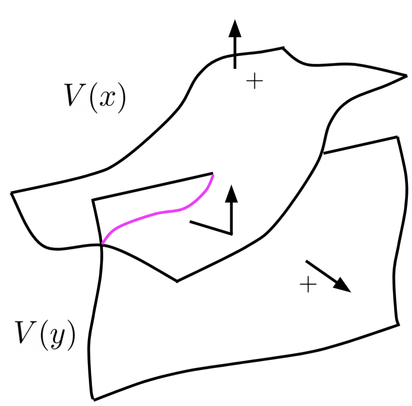


Seifert surfaces are not unique, so although adding genus like this changes the surface, its boundary is still the component $c(x)$ of the string link.

3.2 Surface Systems Method

For a curve $c(x)$ on a surface or surfaces, the positive push off of $c(x)$ is denoted $c^+(x)$. We arrive at the positive push by pushing a curve off the surface(s) it lays on in a direction that is positive relative to all those surfaces [3].

Example 3.3. *The following image indicates the direction in which one should "push" the intersection curve $c(xy)$ off the surfaces to arrive at the positive push off of the surfaces: $c^+(xy)$.*



We use n-bracketing to index all the curves and surfaces in a surface system, which Cochran defines as follows [5].

Definition 3.2. *The set of n-bracketings (in m variables) B_n is given inductively by:*

- $B_1 = \{x,y,z...\}$ *and*

- $B_n = \{(\sigma,\omega) \mid \sigma \in B_k, \omega \in B_{n-k}, 1 \le k \le n-1\}$

Using the two previous terms, Cochran [5] defines a surface system as follows.

Definition 3.3. *A surface system of length n for L is a pair $(C, \mathcal{V})$ of sets satisfying:*

- *There exists a coherent subset S of $\cup B_i$ such that if $\omega(\sigma) < n$ then $\sigma \in S$.*
- *$\mathcal{V}$ is a set of compact, oriented, transversly indersecting, 2-dimensional (possibly empty) submanifolds $V(\sigma)$ of $E(L)$, bijectively indexed by $\sigma \in S$ such that $V(\alpha,\beta)$ is $V(\beta,\alpha)$ with the opposite orientation.*
- *C is a set of closed, oriented (possibly empty) 1-dimensional submanifolds of E(L) containing the longitudes c(x),c(y),c(z),... and all of whose other elements are bijectively indexed by (β,α) where β,α are in S. Specifically, C is the set consisting of the longitudes of L together with all $c(\beta,\alpha)$ where $c(\beta,\alpha)$ is $V(\beta) \cap V(\alpha)$. By convention, $V(\alpha) \cap V(\alpha)$ is empty. These intersections are oriented according to the convention that the ordered triple (orientation of $c(\beta,\alpha)$, positive normal to $V(\beta)$, positive normal to $V(\alpha)$) be the chosen orientation of S^3. These positive normals shall be chosen so that the ordered pairs (orientation of the surface, positive normal) give the ambient orientation. We shall let $c^+(\beta,\alpha)$ denote either a (+,+), (+,-),(-,+), or a (-,-) push-off of $c(\beta,\alpha)$ with respect to $V(\beta)$ and $V(\alpha)$ such that the interior of the annulus spanning c and c^+ misses all elements of C. Define $c^+(x)$ to be $c(x)$ and similarly for other longitudes. Since $c(\beta,\alpha) = -c(\alpha,\beta)$, it is required that these push-offs satisfy $c^+(\beta,\alpha) = -c^+(\alpha,\beta)$.*
- *$\delta V(\sigma) = c^+(\sigma)$.*
- *Suppose $w(\alpha)+w(\beta) \le n$; then, $c(\sigma) \cap V(\beta)$ is empty unless $c(\sigma) \subset V(\beta)$. Thus, $c(\sigma) \cap c(\beta)$ is empty unless the images of the curves coincide. Furthermore, it is convenient to impose the condition that if $w(\beta) < n$ then each component of $\delta V(\beta)$ is the entire boundary of a component of $V(\beta)$.*

Cochran also shows that we can compute the lowest weight non-vanishing (non-zero) Milnor's invariants of any link using surface systems [5].

Proposition 3.1. *If a complete system of weight n exists for L then the value of $\bar{\mu}(i_1 i_2 ... i_n)$ where $i_1 \neq i_n$ is $(-1)^n \Sigma \ell k(c(\alpha), c^+(\beta))$, a summation of minimal n-linkings. Suppose $\gamma_i = (\ell k(c(\sigma), c^+(\gamma))_i$, $i = 1, ..., r$ are minimal coset representatives of the (non-trivial) equivalence classes of n-linkings. Then, $\bar{\mu}(i_1 i_2 ... i_n)$ is a linear combination $\Sigma A_i \gamma_i$ of these n-linkings. These coefficients may be computed (independent of L) by formally calculating the matrix Massey product.*

These coefficients can be calculated using formal Massey products as detailed in Cochran. However, the examples we have studied are simple enough that finding a single non-zero linking is sufficient.
This method is most effective for finding the first non-vanishing Milnor's invariant; if a Milnor's invariant is zero, we can perform more iterations of the process, but otherwise must stop upon reaching the first non-zero Milnor's invariant.

Definition 3.4. *A linking $\ell k(c(a), c(b))$ is minimal if (a,b) is minimal.*

Cochran provides a table of minimal linkings (that is, how many Milnor's invariants of each weight there are and what combination of linking numbers they are equal to). This is helpful in identifying which linear combination of linking numbers one should take to calculate a given Milnor's invariant.

Definition 3.5. *A linking of weight m or an m-linking for $(C, \mathcal{V})$ is $\ell k(c(\alpha), c^+(\beta))$ where $w(\alpha) + w(\beta) = n$. This will often be abbreviated by $\ell k(c(\alpha), c(\beta))$.*

The weight of a Milnor's invariant can be thought of as the number of digits in the Milnor's invariant, where a 1 refers to the first component and a 2 refers to the second component. For example, $\bar{\mu}_{\hat{c}}(12)$ has a weight of two and $\bar{\mu}_{\hat{c}}(1122)$ has a weight of 4. Alternatively, we can think of weight as a measure of complexity; with each iteration of this process, the Milnor's invariants we calculate increase in weight. Formally,

Note: The length of a surface system (described in Cochran's initial definition of a surface system) is distinct from the weight of a surface system.
As mentioned above, Cochran shows that there exist only a certain number of Milnor's invariants of each weight, and some of these may be equal [5]. It is necessary to calculate all unique Milnor's invariants of a given weight before moving on to the next iteration of the process.

3.3 Procedure

Given these definitions, we can outline what the process of calculating the Milnor's invariants of a link using surface systems may look like. For a commutator C in $\mathcal{C}(2)/P(2)$, the first few iterations of the process would be as follows:

1. Let C be a commutator. Take the closure of C: the link $\hat{c}$ and denote its components c(x) and c(y). Then,

$$\bar{\mu}_{\hat{c}}(12) = \ell k(c(x), c(y))$$

2. If this linking number is 0, then generate surfaces $V(x), V(y)$ bounded by $c(x), c(y)$ respectively. The surface $V(x)$ must be disjoint from the component $c(y)$ and, similarly, the surface $V(y)$ must disjoint from the component $c(x)$; this can be achieved by adding hollow tubes as outlined previously.

3. Find the curve of intersection of $V(x)$ and $V(y)$. Denote this curve c(xy). Then,

$$\bar{\mu}_{\hat{c}}(112) = \ell k(c(x), c^+(xy))$$

$$\bar{\mu}_{\hat{c}}(212) = \ell k(c(y), c^+(xy))$$

and

$$\bar{\mu}_{\hat{c}}(1212) = \ell k(c(xy), c^+(xy))$$

4. If all of these linking numbers are 0, then draw the surface $V(xy)$ bounded by $c(xy)$ so that it is disjoint from c(x) and c(y). More hollow tubes may be needed.

5. Identify the following curves of intersection:

   (a) $c(xyx)$: the curve of intersection of $V(xy)$ and $V(x)$
   (b) $c(xyy)$: the curve of intersection of $V(xy)$ and $V(y)$

   Then, similarly, linking numbers of various combinations of these curves will be Milnor's invariants of these links. This process of generating surfaces and intersecting them may continue until the first non-zero Milnor's invariant appears.

4 Results

4.1 Group Theoretic Results

In order to state the next few results, we will need to establish some additional technical terms.

Definition 4.1. *Let a_i and a_{i+1} be generators of consecutive arcs in a knot diagram. Assume these arcs are oriented so that there exists a crossing separating them where the a_i arc goes into the crossing and the a_{i+1} arc goes away from it. We will denote this crossing by $a_i \to a_{i+1}$.*

Definition 4.2. *Given consecutive arcs with generators a_i and a_{i+1} respectively, we will use g_{i+1} to refer to the generator corresponding to the overstrand in $a_i \to a_{i+1}$. Furthermore, let $\epsilon_{i+1} \in \{-1, 1\}$ refer to the 'handedness' of that crossing, where $\epsilon_{i+1} = 1$ if $a_i \to a_{i+1}$ is right handed and $\epsilon_{i+1} = -1$ otherwise.*

Figure 5 shows an ambiguously handed crossing $a_i \to a_{i+1}$ with the appropriate labelings.

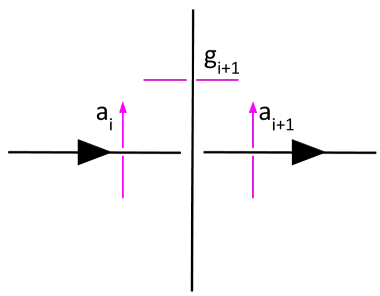


Figure 5: An illustration of $a_i \to a_{i+1}$

Our first major result will allow us to quickly write the longitude of a link component without needing to identify a surface bound by the component.

Theorem 4.1. *Given a knot diagram with n arcs where $a_n \to a_1$ exists and $a_i \to a_{i+1}$ exist for $1 \leq i \leq n-1$, then the 0-framed longitude of the knot is given by $l = a_1^{-\epsilon_2} g_2^{\epsilon_2} a_2^{-\epsilon_3} g_3^{\epsilon_3} \dots a_{n-1}^{-\epsilon_n} g_n^{\epsilon_n} a_n^{-\epsilon_1} g_1^{\epsilon_1}$*

Proof. Recall that a 0-framed longitude will lie on a surface bounded by the knot. We will assume our knot was given a Seifert surface via Seifert's algorithm, and we will trace along that surface, recording the longitude as we go. This surface is orientable due to the nature of Seifert's algorithm, so we will assume without loss of generality that the positive side of this surface is oriented clockwise. That is, we will assume the positive side of the surface sits to the right of the knot.

Note that if the surface would normally appear just as a negative side to the left of our knot, we can create a 'bump' in the surface to ensure the longitude always sits to the right of our knot, as shown in figure 6

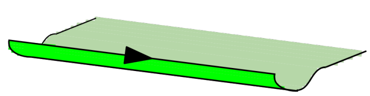


Figure 6: Illustration of the 'bump'.

Therefore, we can begin tracing the knot to the right of an arc and consider what happens at each crossing. As long as we follow the same arc, our longitude will never pass under the knot. Therefore, we need only consider crossings where the arc we are tracing passes under some other arc.

First, suppose we reach a right-hand crossing $a_i \to a_{i+1}$, as shown in figure 7. Note that the surface must appear locally in the quadrants as shown in order to maintain proper orientations when Seifert's algorithm is applied. It is possible that quadrants 2 and 4 contain a surface sitting below quadrant 1 or 3, but such a region is not local to this particular crossing, and thus not relevant to the longitude at this point.

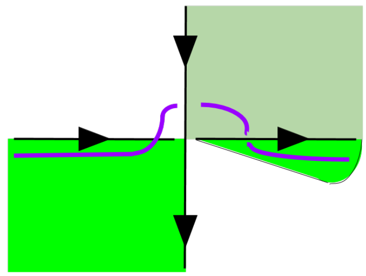


Figure 7: A 0-framed longitude passing by a RHC

The longitude (shown in purple) will pass under the arc for g_{i+1} as the surface twists, and then under the arc for a_{i+1} to reach the bump shown on the right. This is equivalent to the word $g_{i+1}a_{i+1}^{-1}$ in G. However, this crossing also gives us the relation $g_{i+1}a_{i+1}^{-1} = a_i^{-1}g_{i+1}$, and we will record the latter expression to keep in line with the statement of the theorem.

Similarly, we may reach a left-hand crossing $a_j \to a_{j+1}$ as shown in figure 8.

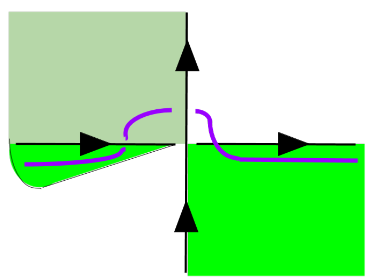


Figure 8: A 0-framed longitude passing by a LHC

In this case, the longitude passes under the arc for a_j and then under the arc for g_{j+1}. This corresponds to the word $a_j g_{j+1}^{-1}$ in G.

Thus, we have found that as we trace the knot, we must record $(a_i^{-1}g_{i+1})^{\epsilon}$ whenever we pass under a crossing $a_i \to a_{i+1}$, where ϵ corresponds to the handedness of the crossing, as described earlier in our definitions. If we begin tracing alongside the arc for a_1, then the full word will come out to be $l = a_1^{-\epsilon_2}g_2^{\epsilon_2}a_2^{-\epsilon_3}g_3^{\epsilon_3}\ldots a_{n-1}^{-\epsilon_n}g_n^{\epsilon_n}a_n^{-\epsilon_1}g_1^{\epsilon_1}$, as desired. □

Note that the above theorem applies only to knots. We need a slight adjustment if we are to apply it for links.

Corollary 4.1. *Let L_1 be a component of a link L. Then the 0-framed longitude of L_1 is given by $l_1 = x_1^{\epsilon_2}x_2^{\epsilon_3}\ldots x_{n-1}^{\epsilon_n}x_n^{\epsilon_1}$, where*

$$x_i = a_i^{-1}g_{i+1} \quad \text{if } a_i \to a_{i+1} \text{ includes arcs from exactly one component}$$
$$x_i = g_{i+1} \quad \text{if } a_i \to a_{i+1} \text{ includes arcs from two components}$$
$$g_{n+1} = g_1$$

Proof. Again we imagine tracing along the component of our link as in theorem 4.1. For crossings $a_i \to a_{i+1}$ where all arcs come from the same component, nothing changes. For crossings $a_i \to a_{i+1}$ where g_{i+1} is a generator of a different component than a_i or a_{i+1}, there will not be a twist in the surface bound by L_1 as there would be in other crossings. See figure 9. Therefore, only g_{i+1} needs to be recorded.

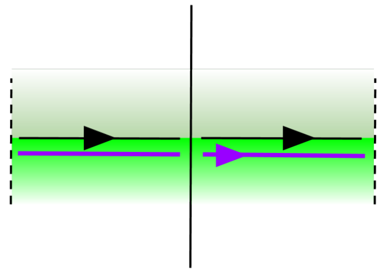


Figure 9: An ambiguously handed crossing between two components. L_1 on bottom.

□

If one wishes, they can use this result to quickly record a 0-framed longitude for a component in their link by tracing along it and recording generators as described. If one is working with a string link commutator, however, there is an even easier way to quickly record the 0-framed longitude for a component. This is due to the following result.

Theorem 4.2. *If L is a string link commutator, then the 0-framed longitude of each component is equal to its blackboard longitude.*

Proof. Let L_1 be an arbitrary component of our string link commutator L, and assume we have a knot diagram where L_1 has n arcs. Let $\{a_1, \dots, a_n\}$ be the generators corresponding to arcs of L_1. Assume $a_n \to a_1$ exists and $a_i \to a_{i+1}$ exist for $1 \leq i \leq n-1$. From these crossings, we can derive the relations $a_n g_1^{\epsilon_1} a_1^{-1} g_1^{-\epsilon_1} = id$ and $a_i g_{i+1}^{\epsilon_{i+1}} a_{i+1}^{-1} g_{i+1}^{-\epsilon_{i+1}} = id$ for $1 \leq i \leq n-1$.

Rearranging these relations, we have $a_n g_1^{\epsilon_1} = g_1^{\epsilon_1} a_1$ and $a_i g_{i+1}^{\epsilon_{i+1}} = g_{i+1}^{\epsilon_{i+1}} a_{i+1}$ for $1 \leq i \leq n-1$. We could also have derived the relations $a_n^{-1} g_1^{\epsilon_1} a_1 g_1^{-\epsilon_1} = id$ and $a_i^{-1} g_{i+1}^{\epsilon_{i+1}} a_{i+1} g_{i+1}^{-\epsilon_{i+1}} = id$ which give us $a_n^{-1} g_1^{\epsilon_1} = g_1^{\epsilon_1} a_1^{-1}$ and $a_i^{-1} g_{i+1}^{\epsilon_{i+1}} = g_{i+1}^{\epsilon_{i+1}} a_{i+1}^{-1}$ for $1 \leq i \leq n-1$ when rearranged. In summary,

$$\begin{aligned}
&a_n g_1^{\epsilon_1} = g_1^{\epsilon_1} a_1 \\
&a_n^{-1} g_1^{\epsilon_1} = g_1^{\epsilon_1} a_1^{-1} \\
&a_i g_{i+1}^{\epsilon_{i+1}} = g_{i+1}^{\epsilon_{i+1}} a_{i+1} \\
&a_i^{-1} g_{i+1}^{\epsilon_{i+1}} = g_{i+1}^{\epsilon_{i+1}} a_{i+1}^{-1}
\end{aligned}$$

Now, consider the 0-framed longitude of L_1, given by corollary 4.1 as

$$l_1 = x_1^{\epsilon_2} x_2^{\epsilon_3} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$

where

$x_i = a_i^{-1} g_{i+1}$ if $a_i \to a_{i+1}$ includes arcs from exactly one component

$x_i = g_{i+1}$ if $a_i \to a_{i+1}$ includes arcs from two components

$g_{n+1} = g_1$

Let $x_k = g_{k+1}$ be the first x_i to correspond to a two component crossing. Using the relations listed above, we can make the following changes:

$$l_1 = x_1^{\epsilon_2} x_2^{\epsilon_3} \dots x_k^{\epsilon_{k+1}} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$
$$l_1 = a_1^{-\epsilon_2} g_2^{\epsilon_2} a_2^{-\epsilon_3} g_3^{\epsilon_3} \dots g_{k+1}^{\epsilon_{k+1}} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$
$$l_1 = g_2^{\epsilon_2} a_2^{-\epsilon_2} a_2^{-\epsilon_3} g_3^{\epsilon_3} \dots g_{k+1}^{\epsilon_{k+1}} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$
$$l_1 = g_2^{\epsilon_2} a_2^{-\epsilon_2} g_3^{\epsilon_3} a_3^{-\epsilon_3} \dots g_{k+1}^{\epsilon_{k+1}} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$
$$l_1 = g_2^{\epsilon_2} g_3^{\epsilon_3} a_3^{-\epsilon_2} a_3^{-\epsilon_3} \dots g_{k+1}^{\epsilon_{k+1}} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$
$$\vdots$$
$$l_1 = g_2^{\epsilon_2} g_3^{\epsilon_3} \dots a_k^{-(\epsilon_2+\epsilon_3+\dots+\epsilon_k)} g_{k+1}^{\epsilon_{k+1}} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$
$$l_1 = g_2^{\epsilon_2} g_3^{\epsilon_3} \dots g_{k+1}^{\epsilon_{k+1}} a_{k+1}^{-(\epsilon_2+\epsilon_3+\dots+\epsilon_k)} \dots x_{n-1}^{\epsilon_n} x_n^{\epsilon_1}$$

Continuing this pattern, we arrive at

$$l_1 = g_2^{\epsilon_2} g_3^{\epsilon_3} \dots g_n^{\epsilon_n} g_1^{\epsilon_1} a_1^{-((\sum_{i=1}^n \epsilon_i) - \epsilon_{k_1} - \epsilon_{k_2} - \dots - \epsilon_{k_p})}$$

where $\{x_{k_1}, x_{k_2}, \dots, x_{k_p}\}$ are all x_i's corresponding to two component crossings.

From here, note that since our link is a commutator, it has an equal number of right and left hand crossings. So, $\sum_{i=1}^n \epsilon_i = 0$. Additionally, since it is a commutator, $\ell k(L_1, L_i) = 0$ for all $i \neq 1$. Thus, $\epsilon_{k_1} + \epsilon_{k_2} + \dots + \epsilon_{k_p} = 0$.

Therefore, the 0-framed longitude of our knot is $l_1 = g_2^{\epsilon_2} g_3^{\epsilon_3} \dots g_n^{\epsilon_n} g_1^{\epsilon_1} a_1^{-(0-0)} = g_2^{\epsilon_2} g_3^{\epsilon_3} \dots g_n^{\epsilon_n} g_1^{\epsilon_1}$, which is precisely the blackboard longitude.

To see that this is the blackboard longitude, imagine tracing the knot as described in theorem 4.1, but only record the generators for arcs you pass under. This will give you the blackboard longitude for the knot, and will result in the word given above.

□

From this proof, we now have an even quicker way of recording 0-framed longitudes for components of string link commutators. It is echoed in the following theorem.

Corollary 4.2. *Let L be a commutator of string links. Let L_1 be an arbitrary component of L, and assume we have a knot diagram where L_1 has n arcs. Let $\{a_1, \dots, a_n\}$ be the generators corresponding to arcs of L_1. Assume $a_n \to a_1$ exists and $a_i \to a_{i+1}$ exist for $1 \leq i \leq n-1$. Then the 0-framed longitude of L_1 is given by $l_1 = g_2^{\epsilon_2} g_3^{\epsilon_3} \dots g_n^{\epsilon_n} g_1^{\epsilon_1}$*

Notably, this result applies to any knot with an equal number of right and left hand crossings, as well as any link component which has an equal number of right and left hand self-crossings. Additionally, the result does not require one to start tracing at a specific point, so one could write $l_1 = g_i^{\epsilon_i} g_{i+1}^{\epsilon_{i+1}} \dots g_n^{\epsilon_n} g_1^{\epsilon_1} \dots g_{i-1}^{\epsilon_{i-1}}$ for any $1 \leq i \leq n$.

4.2 Additional Results

As a result of our research, we were able to identify a class of string link commutators that are always concordant to the two-component unlink and, subsequently, have no non-zero Milnor's invariants.

Lemma 4.1. *For every symmetric weight $m \geq 2$ commutator $c \in \mathcal{C}(2)^{(m)}$ that is the composition of string links $a_1, a_2, ..., a_{m+1}$ and their inverses, if $a_1 = a_3$, $a_2 = a_4$, $a_5 = a_7$, and so on for all a_n, then its closure $\hat{c}$ is concordant to the two-component unlink.*

Proof. Consider as a base case a weight 2 commutator $c_0 = [[a_1, a_2], [a_3, a_4]]$. If $a_1 = a_3$ and $a_2 = a_4$ then we can substitute to arrive at:

$$c_0 = a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1}$$

By definition, $a_2 {a_2}^{-1}$ is concordant to the identity on this group. Similarly, $a_1 {a_1}^{-1}$ is concordant to the unlink.

$$\begin{aligned}
c_0 &= a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_1 a_2 {a_1}^{-1} a_1 {a_2}^{-1} {a_1}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_1 a_2 {a_2}^{-1} {a_1}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_1 {a_1}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 a_2 {a_1}^{-1} {a_2}^{-1} a_2 a_1 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 a_2 {a_1}^{-1} a_1 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 a_2 {a_2}^{-1} {a_1}^{-1} \\
&\cong a_1 {a_1}^{-1} \\
&\cong id_{\mathcal{C}(2)/\mathcal{P}(2)}
\end{aligned}$$

Then, beginning from the $a_2 {a_2}^{-1}$ in the center of c_0, we can simplify to show that c_0 is concordant to the two-component unlink.

Let $c_{m-2} = [A_{m-3}, B_{m-3}]$ be a commutator with weight $m > 2$ and A_{m-3}, B_{m-3} be commutators concordant to the two-component unlink. Then, A_{m-3} can be written as $[\alpha_1, \alpha_2] \cong id_{\mathcal{C}(2)/\mathcal{P}(2)}$ for some α_1, α_2 in $\mathcal{C}(2)/\mathcal{P}(2)$. We can write this as follows.

$$A_{m-3} = \alpha_1 \alpha_2 {\alpha_1}^{-1} {\alpha_2}^{-1} \cong id_{\mathcal{C}(2)/\mathcal{P}(2)}$$

Then,

$$\alpha_1 \cong \alpha_2$$

So, we can write

$$\begin{aligned}
A_{m-3} &\cong \alpha_1 \alpha_1 {\alpha_1}^{-1} {\alpha_1}^{-1} \\
A_{m-3} &\cong \alpha_1 {\alpha_1}^{-1}
\end{aligned}$$

Similarly, B_{m-3} can be written as $[b_1, b_2] \cong id_{\mathcal{C}(2)/\mathcal{P}(2)}$ for some b_1, b_2 in $\mathcal{C}(2)/\mathcal{P}(2)$, implying

$$b_1 \cong b_2$$

and

$$\begin{aligned}
B_{m-3} &\cong b_1 b_1 {b_1}^{-1} {b_1}^{-1} \\
&\cong b_1 {b_1}^{-1}
\end{aligned}$$

Then, we may substitute to arrive at:

$$c_{m-2} \cong \alpha_1 {\alpha_1}^{-1} b_1 {b_1}^{-1} {\alpha_1}^{-1} \alpha_1 {b_1}^{-1} b_1$$

$$c_{m-2} \cong id_{\mathcal{C}(2)/\mathcal{P}(2)}$$

Thus, if each commutator A_{m-3}, B_{m-3} and its inverse is concordant to the two-component unlink, then c_{m-2} will also be concordant to the two-component unlink. □

Corollary 4.3. *For every weight $m \geq 2$ commutator $c \in \mathcal{C}(2)^{(m)}$ that is the composition of string links $a_1, a_2, ..., a_{m+1}$ and their inverses, if $a_1 = a_3$, $a_2 = a_4$, $a_5 = a_7$, and so on for all a_n, then its closure $\hat{c}$ has no non-vanishing Milnor's invariants.*

Proof. Let c be a commutator in $\mathcal{C}(2)^{(m)}$ that is the composition of string links $a_1, a_2, ..., a_{m+1}$ and their inverses. If $a_1 = a_3$, $a_2 = a_4$, $a_5 = a_7$, and so on for all a_n, then, by Lemma 4.1, its closure $\hat{c}$ is concordant to the two-component unlink, which has no non-vanishing Milnor's invariants. Cochran shows that Milnor's invariants are a concordance invariant [5] . By definition, if c is concordant to the two-component unlink, then c has no non-vanishing Milnor's invariants. □

References

[1] M. Kuzbary, *Link Concordance and Groups.* PhD thesis, Rice University, May 2019.

[2] A. Hatcher, "Algebraic topology," 2001.

[3] C. A. Otto, *The* (n)*-Solvable Filtration of the Link Concordance Group and Milnor's* $\overline{\mu}$*-Invariants.* PhD thesis, Rice University, April 2011.

[4] C. Livingston, "Knot theory," 1993.

[5] T. D. Cochran, "Derivatives of links: Milnor's concordance invariants and massey's products," 1990.