Technische Universiteit Eindhoven University of Technology

Department of Mechanical Engineering
Dynamics and Control

Alternative approach in modeling the dynamics of the 'Butterfly' robot

Internship Report
DC 2017.022

Author:
Hao Liang Chen 0799276
Project supervisor:
Prof. Dr. H. Nijmeijer
Department of Mechanical Engineering
Research group Dynamics and Control
Eindhoven Univesity of Technology
Advisor:
Prof. A. Shiriaev
Department of Engineering Cybernetics
Norwegian University of Science and Technology

Version Final

Eindhoven, January 2017

Abstract

For the motion planning of the 'Butterfly' robot, the model must be able to be reduced to a two degrees of freedom model. This reduction happens by applying a virtual holonomic point contact and no-slip constraint. An ad hoc (situational) transformation of coordinates has been used to incorporate these constraints and thus reduce the model. A trajectory has been planned for such a model and with it a suitable controller. Experiments then show that it is possible to let the 'Butterfly' robot track periodic motions such as unidirectional rolling of the ball.

Problems arise when it is desired to extend this 'ad hoc' model to also be applicable (measurable) in situations where the previously mentioned virtual holonomic constraints do not hold. One such situation could be occurence of slip and another one where the rolling object loses contact with the plates. The problem then lies in the ad hoc transformed coordinates which can only be measured when these virtual holonomic constraints are valid. The model thus needs to be changed in such a way that it can be applied in both situations where the virtual holonomic constraints are valid and not.

A solution to this problem would be the modeling of the 'Butterfly' robot in another set of coordinates. A reasonable choice would be to use polar coordinates. With the usage of polar coordinates a new issue arises in which it is difficult (or even impossible) to make analytical expressions for the virtual holonomic constraints. As we wish for an extension of the 'ad hoc' model, it is thus needed that this 'polar coordinate' model can be reduced in the same manner as the 'ad hoc' model. In this report several approaches have been made in trying to get these analytical expressions. The conclusion of these attempts is the issue that the curve which describes the position of the constrained center of the rolling object is not expressible in the angle related to this center of the rolling object, but rather in the angle related to the point contact made when the rolling object and plate touches.

Contents

List of symbols

Symbol	Quantity	Unit	Abbreviation unit
d_{plates}	Distance between the two plates	Meter	m
g	Gravitational acceleration	Meter per square second	m s^{-2}
J_c	Mass moment of inertia ball	Kilogram square meter	kg m^2
J_f	Mass moment of inertia of the plates	Kilogram square meter	kg m^2
J_{pen}	Mass moment of inertia pendulum	Kilogram square meter	kg m^2
m_c	Mass of ball	Kilogram	kg
m_{car}	Mass of cart	Kilogram	kg
m_f	Mass of the two plates	Kilogram	kg
m_{pen}	Mass of pendulum	Kilogram	kg
R	'Effective' radius of ball	Meter	m
R_c	Radius toward center of abll	Meter	m
R_p/δ	Radius toward point contact	Meter	m
R_{real}	Real radius of the ball	Meter	m
s/ s_c	Arc length distance over curve γ_c	Meter	m
s_p	Arc length distance over curve γ_p	Meter	m
w	Normal offset distance center of ball w.r.t. γ_c	Meter	m
x_c	Horizontal position of center of ball	Meter	m
x_{car}	Horizontal position of car	Meter	m
x_p	Horizontal position of point contact between ball and plates	Meter	m
x_{pen}	Horizontal position of mass center pendulum	Meter	m
$\mathcal{L}$	Lagrangian	Joule	J
K	Kinetic Energy	Joule	J
V	Potential Energy	Joule	J
$\vec{n}$	Unit normal vector	-	-
$\vec{\tau}$	Unit tangent vector	-	-
$\vec{v}_c$	Velocity vector of ball	Meter per second	m s^{-1}
$\vec{v}_p$	Velocity vector of point contact	Meter per second	m s^{-1}
$\vec{w}_c$	Rotational velocity vector of the ball with respect to inertial frame	Radian per second	rad s$^-$1
$\vec{w}_f$	Rotational velocity vector of the plates with respect to inertial frame	Radian per second	rad s$^-$1
γ_c	Curve which described constrained center of ball position	-	-
γ_p	Curve which describes point contact between ball and plates	-	-
θ	Angle of plates w.r.t. inertial frame	Radian	rad
θ_p	Angle of pendulum	Radian	rad
ρ	Radius toward constrained center of ball	Meter	m
ϕ	Angle of point contact	Radian	rad
φ	Angle of center of ball	Radian	rad
ψ	Angle of ball w.r.t. angle of plates	Radian	rad

1 Introduction

Humans can interact with objects in many ways. One such way would be the most obvious grasping motion, which can be preferred when transporting small objects. There may be times when such grasping motions can not be made as the object may be too big or too slippery. It is for this reason that other interactions may be preferred, like balancing a slippery ball on top of your hand. These non-grasping interactions can also be called non-prehensile manipulations which also relates to the 'Butterfly' robot. The 'Butterfly' robot does not manipulate the rolling object by grasping it, but manipulates it by letting it roll over a rotating surface (see figure 2).

A reason for researching non-prehensile manipulation in the field of robotics, may be the design of a perfect humanoid. In a society in which every object is made in such a way that humans can interact with it, a humanoid should also be able to interact with these objects when it wants to perform the same tasks as us. Another reason could be related to the properties of the interacting object. The object may for example be too delicate to be grasped and transportation without grasping is then desired.

The 'Butterfly' robot is a robot built for analyzing one kind of non-prehensile manipulation, namely the rolling interaction. The plates of the robot resembles a hand on which a rolling object (in this case a ball) is balanced. In [1] a controller has been designed in which it was possible to control and stabilize a periodic motion on the 'Butterfly' robot. The controller works, but there is a problem with the model used in this work. The model is made in such a way that it is only applicable when certain virtual holonomic constraints, constraints which are not physically present, are active. In case of extending the model to situations in which those virtual holonomic constraints are not active, it is needed to derive the model in other excessive (not necessarily minimal) coordinates.

In this report an attempt has been made in finding excessive coordinates that can extend the model in a way that it can also be applied to situations where the virtual holonomic constraints are not active, while also being able to simulate the constrained behaviour as the 'old' model. With the term 'old' model it is referred to the model used in [1], in which an ad hoc (situational) coordinate transformation is applied. From here on the term 'new' model refers to the model designed with the newfound excessive coordinates (polar coordinates), which can be seen later in the report. The difference between the 'new' and 'old' model lies in the fact that the 'new' model should be applicable in both situations where the virtual holonomic constraints do and do not uphold, while the 'old' model is only applicable in situations where the virtual holonomic constraints do uphold.

The report is built up in the following manner. In chapter 2 an explanation will be given to why it is needed to reduce the model to a two degrees of freedom model, how the 'old' model was derived and what its problem is. The solution to this problem is explained in chapter 3 by using excessive coordinates and deriving a 'new' model with it. This 'new' model should also be able to simulate the behaviour of the 'old' model, which is why attempts have been made in finding analytical expressions of the virtual holonomic constraints in excessive coordinates. This can be seen in chapter 4 and 5. After that a conclusion and recommendation about this 'new' model will follow in chapter 6.

2 Ad hoc transformed model

In previous works ([1],[2] and [3]), an ad hoc (situational) coordinate transformation has been applied in order to reduce the degrees of freedom and apply a motion planning process. This chapter first start by describing why a reduction of the model is needed. Afterwards it will describe the ad hoc transformed model (which is based on chapter 2 of [3]) and will explain what its problems are. Appendix B can be referred to for the steps taken in deriving the equations of motion mentioned in this chapter (with also a guide for a 'Maple' script).

2.1 Why reduction is needed

Motion planning and stabilization of the 'Butterfly' robot as described in [1] requires the degree of freedom of a system to be reduced to two in which one of them is underactuated. The actuated degree of freedom is the one which can influence both degrees of freedom. A virtual holonomic constraint (a constraint which is not physically present) will be proposed between the two degrees of freedom which allows the model to be reduced to a one degree of freedom model. Motion planning will then take place by choosing a feasible trajectory (a periodic motion) on the phase portrait of the passive dynamic (the one related to the unactuated degree of freedom). Stabilization of this motion will take place after this, but this will not be elaborated in this report. If the reader is interested, the following sources can be of interest [1],[2],[3] and [4].

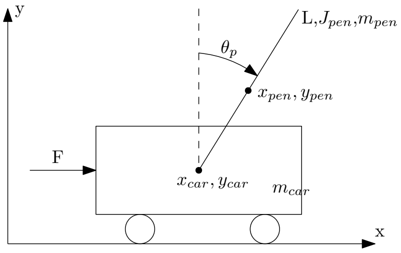

Figure 1: Cart Pendulum

An example will now be worked out to clarify this. In Figure 1 a simple cart pendulum can be seen. This cart pendulum has two degrees of freedom x_{car} (horizontal displacement of the cart) and θ_p (angle of pendulum). Notice that there is only a case of non-prehensile manipulation when there is actuation on the cart (by a force F) and not when there is actuation on the pendulum , although both cases represent underactuation. The equations of motion ('EOM' in short) with actuation on the cart is represented in (2.1) (see appendix A.1 for parameter values and steps taken).

$$EOM \begin{cases} 2\ddot{x}_{car} + \cos(\theta_p)\ddot{\theta}_p - \sin(\theta_p)\dot{\theta}_p^2 = F \\ \ddot{x}_{car}\cos(\theta_p) + \ddot{\theta}_p - g\sin(\theta_p) = 0 \end{cases} \tag{2.1}$$

A virtual holonomic constraint $\Phi(.)$ will be proposed between the two degrees of freedom to reduce the model to a one degree of freedom model. One of the two degrees of freedom will be the 'generating' variable (this degree of freedom determines the value of the other), in this case θ_p will be the generating variable. Equation (2.2) shows the relations between x_{car} and θ_p with virtual holonomic constraint $\Phi(.)$. In [1] and [5] examples of such a virtual holonomic constraint can be found for the 'Butterfly' robot.

$$x_{car} = \Phi(\theta_p) \tag{2.2a}$$

$$\dot{x}_{car} = \Phi^{'}(\theta_p) \cdot \dot{\theta}_p \tag{2.2b}$$

$$\ddot{x}_{car} = \Phi^{''}(\theta_p) \cdot \dot{\theta}_p^2 + \Phi^{'}(\theta_p) \cdot \ddot{\theta}_p \tag{2.2c}$$

Combining the passive dynamic of (2.1) (bottom one) and the virtual holonomic constraints of (2.2) will give an α, β, γ-equation (2.3). This α, β, γ-equation will generate a phase portrait that can be used for the motion planning (more information regarding this phase portrait can be found in appendix A.2). Notice that allocating the actuator to the pendulum (θ_p) will result in a phase portrait with no feasible trajectories, this is also described in appendix A.2.

$$\begin{aligned} \alpha(\theta_p)\ddot{\theta}_p + \beta(\theta_p)\dot{\theta}_p^2 + \gamma(\theta_p) = 0 \\ (1 + \cos(\theta_p)\Phi'(\theta_p))\ddot{\theta}_p + \cos(\theta_p)\Phi''(\theta_p)\dot{\theta}_p^2 - g\sin(\theta_p) = 0 \end{aligned} \tag{2.3}$$

2.2 Modeling of the 'Butterfly' robot

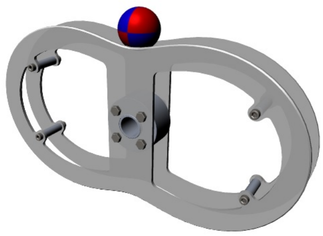

Figure 2: 3-D representation of the 'Butterfly' robot [1]

Figure 2 visualises how the 'Butterfly' robot looks like. It consists of two 'number-eight'-shaped plates which are attached to each other parallel. These plates are rotated with a torque in the middle while having a soft rolling object (in this case a ball) between the two plates. The following assumptions are made about this model:

- The two plates are identical and perfectly aligned which causes the dynamics in the third dimension to be obvious (and this is thus left out).
- The two objects, the plates and the ball, are assumed to be rigid bodies. This will mean that the ball and the plates will be regarded as solid objects which won't deform (line contact between the ball and plate will not occur).
- The plates and ball are assumed to be 'perfect', thus both objects in this case are regarded as smooth objects without imperfections.
- Uniform density/mass distribution assumed in both objects. This way the center of the mass of the ball coincides with the center of the ball and the center of mass of the plates coincides with the center of the plates

Incorporating these assumptions will give a simplified two dimensional representation of this 'Butterfly' robot where the upper right part can be seen on Figure 3. The inertial reference frame is $\underline{\vec{e}}^0$, and the body fixed frames for the plate and ball are respectively $\underline{\vec{e}}^1$ and $\underline{\vec{e}}^2$. There are six degrees of freedom (every two dimensional 'body' has three degrees of freedom) which can be summed up in a coordinate vector $\underline{q}$ as can be seen in (2.4).

$$\underline{q} = \begin{pmatrix} x_f & y_f & \theta & x_c & y_c & \psi \end{pmatrix}^T \tag{2.4}$$

Between the inertial reference frame and the body fixed frame of the plate, the x_f is the horizontal displacement, y_f the vertical displacement and θ the rotation. Between the body fixed frame of the plate and the body fixed frame of the center of the ball, x_c is the horizontal displacement, y_c the vertical displacement and ψ the rotation. Note that the plate does not move ($\dot{x}_f = 0, \dot{y}_f = 0$) and that by alligning the inertial reference frame with the body fixed frame of the plate, a constraint is

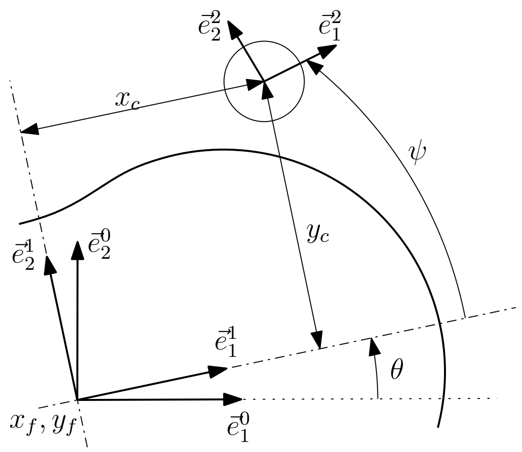

Figure 3: Two dimensional representation of the upper right part of the 'Butterfly' robot

imposed that $x_f = 0$ and $y_f = 0$. Using this constraint, the total degree of freedom of the system can be reduced to four, as can be seen in the following vector.

$$\underline{q}_2 = \begin{pmatrix} \theta & x_c & y_c & \psi \end{pmatrix}^T \tag{2.5}$$

A further reduction of degrees of freedom is needed to apply the method of motion planning as mentioned in the previous chapter. This can be done by adding two more virtual holonomic constraints, namely 1) require contact between the ball and the plate at all time 2) require that the ball does not slip while rolling on the plate. From now on these two constraints will be called for simplicity 'point contact' and 'no-slip' constraint. Notice that these constraints are difficult to express in the degrees of freedom that are stated in $\underline{q}_2$ (this will be seen in chapter 3), which is why an ad hoc transformation of coordinates has been applied. This ad hoc transformation will result in the following degrees of freedom and can be seen in Figure 4.

$$\underline{q}_3 = \begin{pmatrix} \theta & s & w & \psi \end{pmatrix}^T \tag{2.6}$$

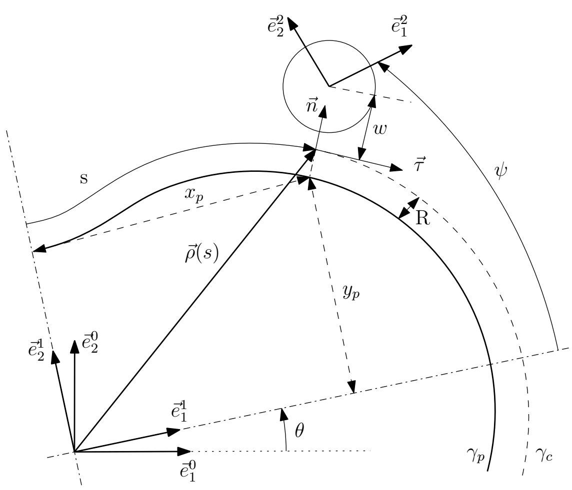

Figure 4: Ad hoc transformated coordinates of 'Butterfly' robot [3]

The 'point contact' between the ball and the plate can be represented by the curve γ_p $(=\begin{pmatrix} x_p & y_p \end{pmatrix}^T)$. As the ball is a rigid body, it always has a constant radius R. Note that this R is not the real radius of the ball as the ball will rest between two plates in which the distance between them will be called d_{plates}. This radius R will then be given by the following equation, in which R_{real} is the real radius of the ball.

$$R = \sqrt{R_{real}^2 - \frac{d_{plates}}{2}^2} \tag{2.7}$$

With the 'point contact' curve γ_p and the rigid body assumption, a new curve can be made which represents the position of the constrained center of the ball (γ_c). The degree of freedom 's' (=s(.)) is then obtained by applying a Frenet coordinate frame, in which 's' is the distance traveled along the curve γ_c. The position of the center of the ball can be described by the vector $\vec{\rho}(s)$, which is a vector towards γ_c from the inertial reference frame, and the degree of freedom 'w' (=w(.)) which is the normal offset of the center of the ball with respect to the curve γ_c.

With these new degrees of freedom it is much easier to formulate the constraints. The 'point constact' constraint assumes that the center of the ball follows the curve (γ_c), thus 'w' is 0. The following constraint can then be imposed.

$$w = 0 \tag{2.8}$$

The 'no-slip' constraint assumes that the distance traveled along the curve γ_p is equal to the rolling distance of the ball. This second constraint can then be seen below. Here the constant 'C' gets a value of 0 when considering an initial position where s and ψ are both 0 (which is used in this report).

$$s = -R\psi + C \tag{2.9}$$

These two constraints can be written as holonomic constraints (position constraints) as been done in section 3.1.1 of [6], which has the following notation.

$$h_i(q_1, ..., q_n, t) = 0 \qquad i = 1, ..., m \tag{2.10}$$

In this case 'm' would be equal to 2, which gives the following two holonomic constraints where h_1 described the 'point contact' constraint and h_2 the 'no-slip' constraint. These where the holonomic constraints as have been used in the previous works ([1],[2] and [3]).

$$h_1 = w \qquad \dot{h}_1 = 0 \rightarrow \dot{w} = 0 \tag{2.11a}$$

$$h_2 = s + R\psi \qquad \dot{h}_2 = 0 \rightarrow \dot{s} + R\dot{\psi} = 0 \tag{2.11b}$$

Note that the 'no-slip' constraint (2.9)/(2.11b) may not be correct. I'm not sure about this as I may have misinterpreted the definition of the degree of freedom 's'. The explanation for why it may be wrong, can be gotten when looking at Figure 5. In these previous works, I think they consider that the distance traveled on γ_p is equal to the distance traveled on γ_c. For a flat surface example this would indeed be true, as the rolling distance from B to B' is equal to the distance traveled on the curve γ_p from A to A' (considering no slip). This traveled distance from A to A' is equal to the traveled distance from C to C', which makes the distance traveled on γ_p equal to the distance traveled on γ_c. The 'Butterfly' robot is not a flat surface, and will cause the distance traveled on γ_c to be bigger than γ_p. It thus not holds that 's' (which I assume to be the distance on γ_c) is then equal to the distance traveled on γ_p, which is stated in (2.9) and (2.11b).

As I'm not sure that the constraint is wrong, it is left unchanged in this chapter and the corresponding appendix (note that this is not the problem of the model I wanted to tackle, just a remark). In the following chapters a distinction is made between these two distances traveled by denoting new variables s_c and s_p (note that everything will stay the same, only 's' in (2.11b) will be replaced by 's_p'). Now that the constraints have been formulated, the degrees of freedom of the model can be reduced to two and it is also possible to simulate the four degrees of freedom model with the constraints. The equations of motion via an Euler-Lagrange method can be found in [3]. Due to the lack of steps given or a 'Maple' script in [3], these have been provided in appendix B.

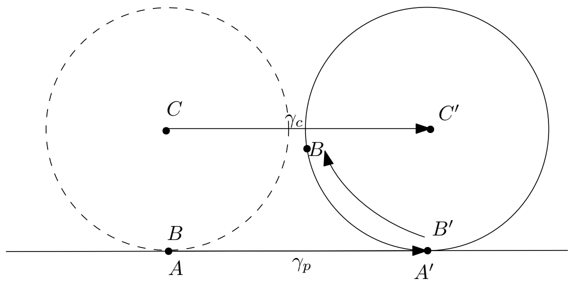

Figure 5: Rolling distance flat surface

2.3 Problems with the ad hoc transformed model

With the virtual holonomic constraints, a four degrees of freedom model can be reduced to a two degrees of freedom model. The reduced degrees of freedom are not set, and it depends on the motion that is desired to decide which degrees of freedom should be used to reduce the model. Possible reduced degrees of freedom are displayed in (2.12).

$$\underline{q}_4 = \begin{pmatrix} \theta & s \end{pmatrix}^T \tag{2.12a}$$

$$\underline{q}_5 = \begin{pmatrix} \theta & \psi \end{pmatrix}^T \tag{2.12b}$$

The first and main problem, is that the degree of freedom 's' can only be measured in the reduced model with active virtual holonomic constraints. The reason for this, is that it is difficult (or even impossible) to assign 's' a value when the ball for example does not have contact with the plates. One way to get 's' (assuming active constraints) is then by measuring the rotations of the ball ψ. Another way is to introduce a new angle φ which is measured from x_c, y_c. Notice that in case the 'point contact' constraint is active, every possible x_c, y_c combination has only one unique φ related to it. The degree of freedom 's' can be expressed in this new angle φ (s(φ)) which leads to another reduced model with degrees of freedom used as in (2.13). A model with these reduced degrees of freedom can be seen in Figure 6. In appendix B the equation of motion of this reduced model with $\underline{q}_r$ can be found. Note that this is the reduction which was used for planning a motion with unidirectional rolling of the ball, as was described in [1].

$$\underline{q}_r = \begin{pmatrix} \theta & \varphi \end{pmatrix}^T \tag{2.13}$$

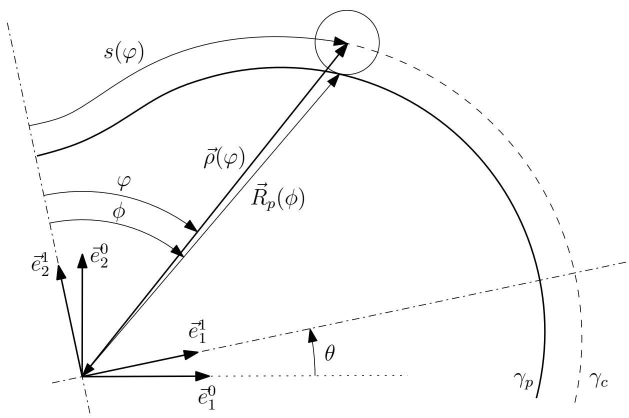

Figure 6: θ,φ degrees of freedom of the 'Butterfly' robot

The radius (and corresponding angle) of the plates of the 'Butterfly' robot is given as $\delta(\phi)$ [1].

With this radius and angle, a vector can be made which describes this plate $\vec{\delta}$. Note that when the ball has contact with the plate, this is equal to the point contact vector $\vec{R}_p(\phi)$ which also describes curve γ_p. There is no expression for curve γ_c with the angle of the center of the ball φ, which thus makes $\vec{\rho}(\varphi)$ unknwon. The only possible way to define γ_c is then in the angle ϕ with the point contact vector $\vec{R}_p$, knowing that the center of the ball is always a normal distance R away from the curve γ_p. This we can also see in the equation below.

$$\begin{aligned}
\delta(\phi) &= 0.1095 - 0.0405\cos(2\phi) \\
\vec{R}_p(\phi) &= \begin{pmatrix}\sin(\phi)\delta(\phi) & \cos(\phi)\delta(\phi)\end{pmatrix}^T \\
\vec{\rho}(\varphi(\phi)) &= \vec{R}_p(\phi) + R\vec{n} \\
s(\varphi) &= \int_0^{\varphi} || \frac{d\vec{\rho}(\varphi)}{d\varphi} || d\varphi
\end{aligned} \tag{2.14}$$

The second (minor) problem, is the unavailability of analytical expressions for the variables presented in (3.8). A look-up-table has been made which relates the degree of freedom $\varphi(\phi)$ with the variable $\rho(\phi)$ (both can be calculated), to get $\rho(\varphi)$. One of the issues presented in [1] is the fact that not all virtual holonomic constraints may always be valid as slip for example may occur, which would mean that the model may be inaccurate from time to time. Having no analytical expressions for the variables used as $\rho(\varphi)$ makes the model even more inaccurate. For the particular shape and rolling object used in [1], it was possible to design a robust enough controller in such a way that tracking was possible. For other shapes and rolling objects there is no guarantee that it is possible to make such a robust controller.

The main issue that is present, is that modeling without the constraints is difficult (or impossible) in the degrees of freedom 's' and 'w'. What kind of value would you assign 's' when the ball slips for example? We can't take the the usual approach that the rolling distance is equal to 's' anymore. The other issue is that the model is not that accurate as no analytical expression is present for $\rho(\varphi)$. We will see that this last issue also forms a problem for the 'new' model, and that it is important to get such an analytical expression if we want the model to be more accurate. The real challenge in this 'Butterfly' robot is actually then to model it as accurate as possible.

2.4 Summary

A reduction of the model is needed to be able to plan a motion. It is necessary that the model is reduced to a two degrees of freedom model in which one of the degrees of freedom can be related to the actuated part, while the other is related to the unactuated part. For the 'Butterfly' robot an ad hoc transformation of coordinates is applied in order to reduce the model to such a shape.

It can be seen that the ad hoc transormed model has two issues. The first issue, that the model can only be applied correctly in situations where the rolling object does not slip and always maintains contact with the plates. These conditions are required in order to give meaningful values to the coordinates 's' and 'w'. In case these conditions do not uphold, for example when the ball is in the air, it is not possible to assign a correct value to the coordinates 's' and 'w'.

The second issue, the unavailability of an analytical expression for ρ in the reduced degree of freedom φ. This is needed to be able to express the dynamics of the model more accurately, as a look-up-table won't always provide smooth derivatives. Solving both these issues will provide a more accurate model, for which the first issue is of course the most important one. It is for this reason that the next chapter will focus on searching this new set of coordinates. We will see however that solving the first issue also requires solving the second issue, which may not prove to be so simple.

3 Model in excessive coordinates

For the ad hoc transformed model, which will be called 'old' model from now on, it was possible to let the system track a periodic motion as can be seen in [1]. The 'old' model however is not perfect, and is only applicable in situations where the virtual holonomic constraints 'point contact' and 'no-slip' are active. In this chapter it is tried to derive a 'new' model in excessive (not necessarily minimal) coordinates, which should also be applicable in situations where these virtual holonomic constraints do not uphold. Usage of excessive coordinates however brings another issue in which it is difficult to formulate the virtual holonomic constraints analytically. The chapter first starts with modeling in cartesian coordinates and states its difficulty with writing the virtual holonomic constraints. A 'new' model in polar coordinates is then suggested in which it is seen that it is possible to write the virtual holonomic constraints, only if it is possible to get an analytical expression for $\phi(\varphi)$.

3.1 Model in Cartesian coordinates

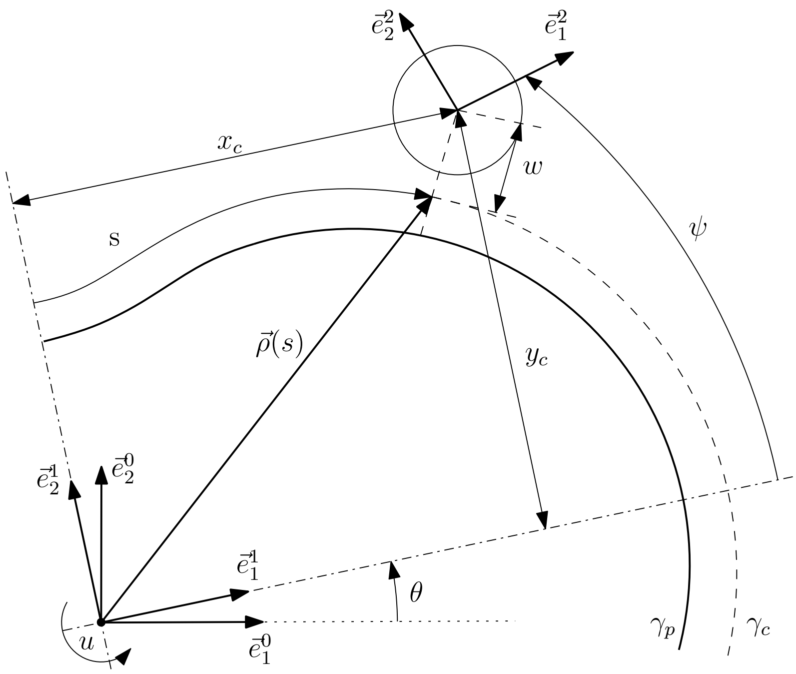

Figure 7: 'Butterfly' robot in cartesian coordinates

Figure 7 gives a representation of the 'Butterfly' robot in Cartesian coordinates. This model uses the following degrees of freedom $\underline{q}_2 = \begin{pmatrix}\theta & x_c & y_c & \psi\end{pmatrix}^T$, in which the degrees of freedom are already described in chapter 2.2. An Euler-Lagrange method can be applied to derive the equations of motions, which can be seen in (3.1) (the notations are used as described in [6]). Herein $\underline{Q}_{nc}$ are the non conservative forces (e.g. actuator forces), $\underline{W\lambda}$ consists of the holonomic and non-holonomic velocity constraints which are linear in the velocity terms (otherwise the shape of (3.1) can not be adopted as stated in chapter 3 page 55 of [6]) and $\mathcal{L}$ is the Lagrangian which is built up from the kinetic energy 'K' and potential energy 'V'.

$$\begin{aligned}(\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\underline{q}}_2} - \frac{\partial\mathcal{L}}{\partial\underline{q}_2})^T &= \underline{M}(\underline{q}_2)\ddot{\underline{q}}_2 + \underline{C}(\underline{q}_2,\dot{\underline{q}}_2)\dot{\underline{q}}_2 + \underline{G}(\underline{q}_2) = \underline{Q}_{nc} + \underline{W\lambda} \\ \mathcal{L} &= K - V\end{aligned} \tag{3.1}$$

Using the assumption of uniform mass distribution, the center of mass of the plates should coincide with the center position of the plates. This will make the potential force of the plates negligible as it has no influence on the system. The kinetic energy of the plates are solely due to rotation as no translation is possible. Equation (3.2) gives the kinetic energy of the plates (K_f), herein J_f is the inertia of the plates and $\vec{w}_f$ the rotational velocity of the plates.

$$\begin{aligned} K_f &= \frac{1}{2} J_f(\vec{w}_f \cdot \vec{w}_f) \\ \vec{w}_f &= \begin{pmatrix} 0 & 0 & \dot{\theta} \end{pmatrix} \underline{\vec{e}}^0 \end{aligned} \tag{3.2}$$

The ball will have both a kinetic and a potential energy. The kinetic energy of the center of the ball (K_c) consists of a translational and a rotational component, which can be seen in (3.3).

$$K_c = \frac{1}{2} m_c(\vec{v}_c \cdot \vec{v}_c) + \frac{1}{2} J_c(\vec{w}_c \cdot \vec{w}_c) \tag{3.3}$$

The translational velocity of the center of the ball ($\vec{v}_c$) can be gotten by taking the time derivative of the position of the center of the ball ($\vec{R}_c$) (note that ${}^{10}\vec{w} = \vec{w}_f$).

$$\begin{aligned} \vec{R}_c &= \begin{pmatrix} x_c & y_c & 0 \end{pmatrix} \underline{\vec{e}}^1 = \underline{R}_c^{1^T} \underline{\vec{e}}^1 \\ \frac{d\vec{R}_c}{dt} = \vec{v}_c &= \underline{\dot{R}}_c^{1^T} \underline{\vec{e}}^1 + \underline{R}_c^{1^T} \underline{\dot{\vec{e}}}^1 = \underline{\dot{R}}_c^{1^T} \underline{\vec{e}}^1 + {}^{10}\vec{w} \times \vec{R}_c \\ &= \begin{pmatrix} \dot{x}_c & \dot{y}_c & 0 \end{pmatrix} \underline{\vec{e}}^1 + {}^{10}\vec{w} \times \begin{pmatrix} x_c & y_c & 0 \end{pmatrix} \underline{\vec{e}}^1 = \begin{pmatrix} \dot{x}_c - \dot{\theta} y_c & \dot{y}_c + \dot{\theta} x_c & 0 \end{pmatrix} \underline{\vec{e}}^1 \end{aligned} \tag{3.4}$$

The rotational velocity of the center of the ball ($\vec{w}_c$) can be gotten by taking the time derivative of the total angle difference between the body fixed frame of the ball and the inertial reference frame.

$$\vec{w}_c = \begin{pmatrix} 0 & 0 & \dot{\psi} + \dot{\theta} \end{pmatrix} \underline{\vec{e}}^0 \tag{3.5}$$

Under the assumption that the ball also has uniform mass distribution, the center of mass of the ball will coincide with the center of the ball. This will then give the following potential energy of the center of the ball (V_c), in which 'g' is the gravitational component and $\underline{A}^{10}$ is a rotation matrix that relates the angle difference between the body fixed frame of the plate and the inertial reference frame.

$$V_c = m_c \vec{g} \cdot (\underline{R}_c^{1^T} \underline{\vec{e}}^1) \tag{3.6a}$$

$$\vec{g} = \begin{pmatrix} 0 & g & 0 \end{pmatrix} \underline{\vec{e}}^0 = g \vec{e}_2^0 \tag{3.6b}$$

$$\underline{\vec{e}}^1 = \underline{A}^{10} \underline{\vec{e}}^0 = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \underline{\vec{e}}^0 \tag{3.6c}$$

Adding all the kinetic and potential energy terms will result in the Lagrangian as stated in (3.7). The $\underline{M}(\underline{q}_2), \underline{C}(\underline{q}_2, \underline{\dot{q}}_2)$ and $\underline{G}(\underline{q}_2)$ terms in (3.1) can be found in appendix C.1(a Maple script can be made by following the steps in appendix B.2).

$$\begin{aligned} \mathcal{L} &= K_c + K_f - V_c \\ &= \frac{m_c}{2}((x_c^2 + y_c^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c})\dot{\theta}^2 + (2\frac{J_c}{m_c}\dot{\psi} - 2(y_c\dot{x}_c - x_c\dot{y}_c))\dot{\theta} + \frac{J_c}{m_c}\dot{\psi}^2 + \dot{x}_c^2 + \dot{y}_c^2 - 2m_c\vec{g} \cdot (\underline{R}_c^{1^T} \underline{A}^{10})) \end{aligned} \tag{3.7}$$

The non-conservative term has only one input as the only external force that is present is the torque on the plates, which will be called 'u'.

$$\underline{Q}_{nc} = \begin{pmatrix} u & 0 & 0 & 0 \end{pmatrix}^T \tag{3.8}$$

The last term $\underline{W}\lambda$ can be gotten from the virtual holonomic constraints 'point contact' and 'no-slip', where . The previous 'point contact' constraint (2.8) requires that w is 0, which thus means that the center of the ball is always on the constrained curve γ_c (described by $\vec{\rho}$). Assuming that no-slip is present and that there is thus a relation between 's' and (x_c, y_c), this can then be formulated as the following.

$$\sqrt{x_c^2 + y_c^2} = ||\vec{\rho}(s(x_c, y_c))|| \tag{3.9}$$

The 'no-slip' constraint (2.9) can then be gotten in Cartesian coordinates by having a relation between 's' and (x_c, y_c). This can be given in the following velocity constraint (this was much easier to depict).

$$\begin{aligned} \dot{s} &= \sqrt{\dot{x}_c + \dot{y}_x} \\ \sqrt{\dot{x}_c + \dot{y}_x} &= -R\dot{\psi} \end{aligned} \tag{3.10}$$

The main issue with these constraints is the difficulty to reduce the dynamics with it. For this model a logical choice of reduced degrees of freedom would be $\underline{q}_5 = \begin{pmatrix} \theta & \psi \end{pmatrix}^T$ as x_c or y_c can't give good information about the position of the ball when used seperately. It is then desired to have two virtual holonomic constraints in which for example one constraint gives the relation $x_c = f(\psi)$ and the other constraint the relation $y_c = g(x_c)$. Equation (3.9) may then give the relation $y_c = g(x_c)$ and (3.10) the relation $x_c = f(\psi)$. Assuming that $\rho(s(x_c, y_c))$ will not be a simple expression as the derivative of 's' has already square root terms in it, (3.9) will be difficult (or even impossible) to solve in the shape of $y_c = g(x_c)$. Without an analytical expression for $y_c = g(x_c)$, it is then also impossible to solve (3.10) as $x_c = f(\psi)$ and it is thus not able to reduce the model to a two degrees of freedom model.

Even though the equations of motions are rather easily derived, it is seen that the constraints are difficult to formulate. It is for this reason that the 'old' model used an ad hoc transformation of coordinates. This is not the solution that is desired in this report as the flaw of the 'old' model has already been pointed out. Another possible choice of coordinates is the usage of polar coordinates. This way it is possible to replace the less practical coordinates x_c and y_c (in the sense that they're not useful as reduced coordinates) as an angle φ and corresponding radius R_c. As φ was also chosen as the reduced coordinate for the reduced 'old' model, it thus emphasizes that it is not a bad choice. The 'new' model that arises with polar coordinates can be reduced to the same degrees of freedom as the 'old' model, while also having coordinates that can be measured in situations where the virtual holonomic constraints are not valid.

3.2 Model in polar coordinates

Transforming the Cartesian coordinates into polar coordinates will give the degrees of freedom as can be seen in (3.11). Those degrees of freedom can be found back in Figure 8, wherein R_c is the distance between the inertial reference frame and the center of the ball. Herein also a distinction is made between the distances traveled along the curves γ_p and γ_c, respectively named s_p and s_c .

$$\underline{q}_n = \begin{pmatrix} \theta & R_c & \varphi & \psi \end{pmatrix}^T \tag{3.11}$$

The relations between (x_c, y_c) and (R_c, φ) can be seen in (3.12).

$$x_c = R_c \sin(\varphi) \tag{3.12a}$$

$$y_c = R_c \cos(\varphi) \tag{3.12b}$$

Notice that another possible choice of coordinates would be $\underline{q}_o = \begin{pmatrix} \theta & R_p & \phi & R_c & \varphi & \psi \end{pmatrix}^T$. This transformation of coordinates would only be of benefit if it is desired to get ϕ as one of the two reduced degrees of freedom. In practice ϕ will not get measured, as it is impossible to measure just point contact. If it is however desired to get ϕ, then the following four constraints must be used : $R_p(\phi), R_c(\phi), \varphi(\phi), \psi(\phi)$. This has not been worked out in this report as it was not of interest.

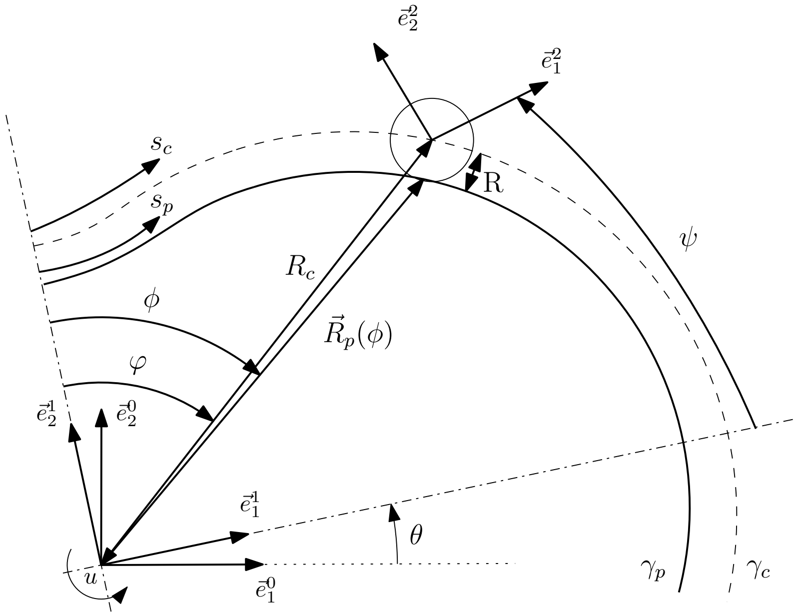

Figure 8: 'Butterfly' robot in polar coordinates

3.2.1 Equation of motion four degrees of freedom model

The same equation as (3.1) can be used with coordinate vector $\underline{q}_n$ for deriving the equations of motion. The Lagrangian can be derived in the same manner and deriving this should give the same answer as inserting the new expressions of x_c and y_c (3.12) into (3.7), which gives (3.13).

$$\mathcal{L} = \frac{m_c}{2}((R_c^2+\frac{J_c}{m_c}+\frac{J_f}{m_c})\dot{\theta}^2+(2\frac{J_c}{m_c}\dot{\psi}-2R_c^2\dot{\varphi})\dot{\theta}+\frac{J_c}{m_c}\dot{\psi}^2+R_c^2\dot{\varphi}^2+\dot{R}_c^2-2gR_c(\sin(\varphi)\sin(\theta)+\cos(\varphi)\cos(\theta))) \tag{3.13}$$

With the Lagrangian defined we can derive the equation of motion via the Euler-Lagrange method. This can be seen below, in which the terms for the respective degree of freedom are worked out seperately.

Degree of freedom θ

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = m_c((R_c^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c})\dot{\theta} + \frac{J_c}{m_c}\dot{\psi} - R_c^2\dot{\varphi}) \tag{3.14}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = m_c((R_c^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c})\ddot{\theta} + \frac{J_c}{m_c}\ddot{\psi} - R_c^2\ddot{\varphi} + 2R_c\dot{R}_c(\dot{\theta} - \dot{\varphi})) \tag{3.15}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m_c g R_c(\sin(\varphi)\cos(\theta) - \cos(\varphi)\sin(\theta)) \tag{3.16}$$

Degree of freedom R_c

$$\frac{\partial \mathcal{L}}{\partial \dot{R}_c} = m_c\dot{R}_c \tag{3.17}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{R}_c} = m_c\ddot{R}_c \tag{3.18}$$

$$\frac{\partial \mathcal{L}}{\partial R_c} = m_c(R_c\dot{\theta}^2 - 2R_c\dot{\varphi}\dot{\theta} + R_c\dot{\varphi}^2 - 2g(\sin(\varphi)\sin(\theta) + \cos(\varphi)\cos(\theta))) \tag{3.19}$$

Degree of freedom φ

$$\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} = m_c(R_c^2(\dot{\varphi} - \dot{\theta})) \tag{3.20}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} = m_c(R_c^2(\ddot{\varphi} - \ddot{\theta}) + 2R_c\dot{R}_c(\dot{\varphi} - \dot{\theta})) \tag{3.21}$$

$$\frac{\partial \mathcal{L}}{\partial \varphi} = -m_c g R_c(\cos(\varphi)\sin(\theta) - \sin(\varphi)\cos(\theta)) \tag{3.22}$$

Degree of freedom ψ

$$\frac{\partial \mathcal{L}}{\partial \dot{\psi}} = J_c(\dot{\theta} + \dot{\psi}) \tag{3.23}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\psi}} = J_c(\ddot{\theta} + \ddot{\psi}) \tag{3.24}$$

$$\frac{\partial \mathcal{L}}{\partial \psi} = 0 \tag{3.25}$$

Equation of motion

$$\underline{M}_n(\underline{q}_n) = m_c \begin{pmatrix} (R_c^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c}) & 0 & -R_c^2 & \frac{J_c}{m_c} \\ 0 & 1 & 0 & 0 \\ -R_c^2 & 0 & R_c^2 & 0 \\ \frac{J_c}{m_c} & 0 & 0 & \frac{J_c}{m_c} \end{pmatrix} \tag{3.26}$$

$$\underline{C}_n(\underline{q}_n, \dot{\underline{q}}_n) = m_c \begin{pmatrix} R_c\dot{R}_c & R_c(\dot{\theta} - \dot{\varphi}) & -R_c\dot{R}_c & 0 \\ R_c(\dot{\varphi} - \dot{\theta}) & 0 & R_c(\dot{\theta} - \dot{\varphi}) & 0 \\ -R_c\dot{R}_c & R_c(\dot{\varphi} - \dot{\theta}) & R_c\dot{R}_c & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.27}$$

$$\underline{G}_n(\underline{q}_n) = \begin{pmatrix} m_c g R_c(\sin(\varphi)\cos(\theta) - \cos(\varphi)\sin(\theta)) \\ m_c g(\sin(\varphi)\sin(\theta) + \cos(\varphi)\cos(\theta)) \\ m_c g R_c(\cos(\varphi)\sin(\theta) - \sin(\varphi)\cos(\theta)) \\ 0 \end{pmatrix} \tag{3.28}$$

The nonconservative force $\underline{Q}_{nc}$ will not change and can be seen in (3.8). The 'point contact' constraint can be formulated as that R_c should be equal to $\rho(\varphi)$ $(=||\vec{\rho}(\varphi)||)$, which constraints R_c on the curve γ_c. The 'no-slip' constraint can be found by stating that the rolling distance of the ball is equal to a traveled distance '$s_p(\varphi)$' (assuming that the ball starts rolling at s_p=0 and $\psi = 0$). Note that the reduced 'new' model uses the same degrees of freedom as the reduced 'old' model, namely $\underline{q}_r = \begin{pmatrix}\theta & \varphi\end{pmatrix}^T$. With these two constraint expressions it is possible formulate them in the same way as (2.10) and (2.11), as holonomic constraints $h_{1,n}$ and $h_{2,n}$ which stands respectively for the new 'point contact' and 'no-slip' holonomic constraints. With them formulated in this way, it is possible to derive $\underline{W}\underline{\lambda}$ as below.

$$h_{1,n} = R_c - \rho(\varphi) \tag{3.29a}$$

$$h_{2,n} = s_p(\varphi) + R\psi \tag{3.29b}$$

$$\underline{W}^T = \frac{\partial \begin{pmatrix} h_{1,n} & h_{2,n} \end{pmatrix}^T}{\partial \underline{q}_n} = \begin{pmatrix} \frac{dh_{1,n}}{d\theta} & \frac{dh_{1,n}}{dR_c} & \frac{dh_{1,n}}{d\varphi} & \frac{dh_{1,n}}{d\psi} \\ \frac{dh_{2,n}}{d\theta} & \frac{dh_{2,n}}{dR_c} & \frac{dh_{2,n}}{d\varphi} & \frac{dh_{2,n}}{d\psi} \end{pmatrix} = \begin{pmatrix} 0 & 1 & -\frac{d\rho(\varphi)}{d\varphi} & 0 \\ 0 & 0 & \frac{ds_p(\varphi)}{d\varphi} & R \end{pmatrix} \tag{3.29c}$$

$$\underline{\lambda} = \begin{pmatrix} \lambda_1 & \lambda_2 \end{pmatrix}^T \tag{3.29d}$$

For getting $\underline{\lambda}$, a look can be given at (B.46).

3.2.2 Equation of motion two degrees of freedom model

To reduce the dynamics to a two degrees of freedom model ($\underline{q}_r = \begin{pmatrix}\theta & \varphi\end{pmatrix}^T$), it is needed to get the 'point contact' constraint $R_c(\varphi)$ (=$\rho(\varphi)$) and the 'no-slip' constraint $\psi(\varphi)$ (= -$\frac{s_p(\varphi)}{R}$). Suppose that these constraints can be obtained in their desired format, then the time derivatives of these constraints can be seen in (3.30) (note that $R_c' = \frac{dR_c}{d\varphi}, R_c'' = \frac{d^2R_c}{d\varphi^2}, \psi' = \frac{d\psi}{d\varphi}, \psi'' = \frac{d^2\psi}{d\varphi^2}$).

$$\dot{R}_c(\varphi) = R_c'\dot{\varphi} \qquad \ddot{R}_c(\varphi) = R_c'\ddot{\varphi} + R_c''\dot{\varphi}^2 \tag{3.30a}$$

$$\dot{\psi}(\varphi) = \psi'\dot{\varphi} \qquad \ddot{\psi}(\varphi) = \psi'\ddot{\varphi} + \psi''\dot{\varphi}^2 \tag{3.30b}$$

The Lagrangian can now be obtained by inserting these constraints and time derivatives into (3.13), which results in (3.31).

$$\begin{aligned}\mathcal{L} =& \frac{m_c}{2}((R_c^2(\varphi) + \frac{J_c}{m_c} + \frac{J_f}{m_c})\dot{\theta}^2 + (2\frac{J_c}{m_c}\psi' - 2R_c^2(\varphi))\dot{\varphi}\dot{\theta} + (\frac{J_c}{m_c}(\psi')^2 + R_c^2(\varphi) + (R_c')^2)\dot{\varphi}^2 \\ &- 2gR_c(\varphi)(\sin(\varphi)\sin(\theta) + \cos(\varphi)\cos(\theta)))\end{aligned} \tag{3.31}$$

Degree of freedom θ

$$\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = m_c((R_c^2(\varphi) + \frac{J_c}{m_c} + \frac{J_f}{m_c})\dot{\theta} + (\frac{J_c}{m_c}\psi' - R_c^2(\varphi))\dot{\varphi}) \tag{3.32}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = m_c((R_c^2(\varphi) + \frac{J_c}{m_c} + \frac{J_f}{m_c})\ddot{\theta} + (\frac{J_c}{m_c}\psi' - R_c^2(\varphi))\ddot{\varphi} + 2R_c(\varphi)R_c'\dot{\varphi}\dot{\theta} + (\frac{J_c}{m_c}\psi'' - 2R_c(\varphi)R_c')\dot{\varphi}^2) \tag{3.33}$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = -m_c gR_c(\varphi)(\sin(\varphi)\cos(\theta) - \cos(\varphi)\sin(\theta)) \tag{3.34}$$

Degree of freedom φ

$$\frac{\partial\mathcal{L}}{\partial\dot{\varphi}} = m_c((\frac{J_c}{m_c}\psi' - R_c^2(\varphi))\dot{\theta} + (\frac{J_c}{m_c}(\psi')^2 + R_c^2(\varphi) + (R_c')^2)\dot{\varphi}) \tag{3.35}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\varphi}} =& m_c((\frac{J_c}{m_c}\psi' - R_c^2(\varphi))\ddot{\theta} + (\frac{J_c}{m_c}(\psi')^2 + R_c^2(\varphi) + (R_c')^2)\ddot{\varphi} \\ &+ (\frac{J_c}{m_c}\psi'' - 2R_cR_c')\dot{\varphi}\dot{\theta} + 2(\frac{J_c}{m_c}\psi'\psi'' + R_cR_c' + R_c'R_c'')\dot{\varphi}^2)\end{aligned} \tag{3.36}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\varphi} =& m_c(R_c(\varphi)R_c'\dot{\theta}^2 + (\frac{J_c}{m_c}\psi'' - 2R_c(\varphi)R_c')\dot{\varphi}\dot{\theta} + (\frac{J_c}{m_c}\psi'\psi'' + R_c(\varphi)R_c' + R_c'R_c'')\dot{\varphi}^2 \\ &- gR_c'(\sin(\varphi)\sin(\theta) + \cos(\varphi)\cos(\theta)) - gR_c(\varphi)(\cos(\varphi)\sin(\theta) - \sin(\varphi)\cos(\theta)))\end{aligned} \tag{3.37}$$

Equation of motion

$$\underline{M}_r(\underline{q}_r) + \underline{C}_r(\underline{q}_r, \underline{\dot{q}}_r)\underline{\dot{q}}_r + \underline{G}_r(\underline{q}_r) = \begin{pmatrix}u & 0\end{pmatrix}^T \tag{3.38}$$

$$\underline{M}_r(\underline{q}_r) = m_c\begin{pmatrix}(R_c^2(\varphi) + \frac{J_c}{m_c} + \frac{J_f}{m_c}) & (\frac{J_c}{m_c}\psi' - R_c^2(\varphi)) \\ (\frac{J_c}{m_c}\psi' - R_c^2(\varphi)) & (\frac{J_c}{m_c}(\psi')^2 + R_c^2(\varphi) + (R_c')^2)\end{pmatrix} \tag{3.39}$$

$$\underline{C}_r(\underline{q}_r, \underline{\dot{q}}_r) = m_c\begin{pmatrix}R_c(\varphi)R_c'\dot{\varphi} & R_c(\varphi)R_c'\dot{\theta} + (\frac{J_c}{m_c}\psi'' - 2R_c(\varphi)R_c')\dot{\varphi} \\ -R_c(\varphi)R_c'\dot{\theta} & (\frac{J_c}{m_c}\psi'\psi'' + R_c(\varphi)R_c' + R_c'R_c'')\dot{\varphi}\end{pmatrix} \tag{3.40}$$

$$\underline{G}_r(\underline{q}_r) = \begin{pmatrix}m_c gR_c(\varphi)(\sin(\varphi)\cos(\theta) - \cos(\varphi)\sin(\theta) \\ m_c gR_c'(\sin(\varphi)\sin(\theta) + \cos(\varphi)\cos(\theta)) + m_c gR_c(\varphi)(\cos(\varphi)\sin(\theta) - \sin(\varphi)\cos(\theta)))\end{pmatrix} \tag{3.41}$$

3.2.3 Constraints

In order to simulate the constrained four degrees of freedom model accurately or to reduce it to an accurate two degrees of freedom model, analytical expressions of the 'point contact' constraint $R_c(\varphi)$ (=$\rho(\varphi)$) and the 'no-slip' constraint $\psi(\varphi)$ (= -$\frac{s_p(\varphi)}{R}$) are needed.

Point contact

The 'point contact' constraint assumes that R_c is equal to the length of vector $\vec{\rho}$ (3.42c). The only way of getting an analytical expression of $\vec{\rho}$ is to formulate it in the coordinate ϕ, thus $\vec{\rho}(\phi)$. This can be seen by (3.42a) in which $\vec{R}_p$ is described in the coordinate ϕ by the shape of the plate $\delta(\phi)$ given in (2.14). Knowing that $\vec{\rho}$ or γ_c is a normal distance 'R' away from γ_p, gives the addition $R\vec{n}$. The normal vector $\vec{n}$ is defined by the tangent vector $\vec{\tau}$ (3.42b) (which can be seen in Figure 4). Here v_p can be seen as the velocity of the point contact. With this all, it is possible to formulate the constraint in the same way as (3.29a) which can be seen in (3.42d) and its derivative in (3.42e). From it, it can be seen that an analytical expression is available if $\phi(\varphi)$ is known, which is yet unknown.

$$||\vec{\rho}(\phi)|| = ||\vec{R}_p(\phi) + R\vec{n}(\phi)|| \tag{3.42a}$$

$$\vec{n}(\phi) = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T \times \vec{\tau}, \quad \vec{\tau} = \frac{\vec{v}_p}{||\vec{v}_p||}, \quad \vec{v}_p = \frac{\partial \vec{R}_p(\phi)}{\partial \phi} \tag{3.42b}$$

$$R_c = ||\vec{\rho}(\phi(\varphi))|| \tag{3.42c}$$

$$h_{1,n} = R_c - ||\vec{\rho}(\phi(\varphi))|| \tag{3.42d}$$

$$\dot{h}_{1,n} = 0 \rightarrow \dot{R}_c - \frac{d||\vec{\rho}||}{d\phi}\frac{d\phi}{d\varphi}\dot{\varphi} = 0 \tag{3.42e}$$

No-slip

This time a differentiation between s_p and s_c is made. The correct 'no-slip' constraint would use s_p, thus it is also used here. The constraint can be seen in (3.43a) which can be reformulated in the same way as (3.29b) as can be seen in (3.43b).

$$s_p = \int_0^{\phi} ||\frac{d\vec{R}_p(\phi)}{d\phi}||d\phi = \int_0^{\varphi} ||\frac{d\vec{R}_p(\phi(\varphi))}{d\varphi}||d\varphi = -R\psi \tag{3.43a}$$

$$h_{2,n} = \int_0^{\varphi} ||\frac{d\vec{R}_p(\phi(\varphi))}{d\varphi}||d\varphi + R\psi \tag{3.43b}$$

$$\dot{h}_{2,n} = 0 \rightarrow ||\frac{d\vec{R}_p(\phi(\varphi))}{d\varphi}||\dot{\varphi} + R\dot{\psi} = 0 \tag{3.43c}$$

From the equations it can be seen that the expression $\phi(\varphi)$ is again needed to make an analytical expression for the 'no-slip' constraint in φ.

3.3 Summary

A model made in Cartesian coordinates may be the most obvious and practical choice, but reducing the model seems to be difficult (or even impossible). It is for this reason that a model is made in polar coordinates, which showed a much easier process reducing the model. The problem that then arises, is that the reduction needs an analytical expression for $\phi(\varphi)$. It is desired to reduce the model in φ as it makes much more sense (and is also easier) to measure this angle in expreriments instead of ϕ. When it is possible to find such an expression, a 'new' model can be created that can be applied to situations where the virtual holonomic constraints do not hold, while also being able to be reduced to the same model as the 'old' model.

4 Expressing angle of the center of the ball in angle of point contact

Before deriving $\phi(\varphi)$, the inverse $\varphi(\phi)$ is first derived. The reason for doing this, is that $\varphi(\phi)$ is much easier to derive and that the inverse of $\varphi(\phi)$ may then be taken for getting $\phi(\varphi)$. Also the attempts made for deriving $\varphi(\phi)$ may then be applied for deriving $\phi(\varphi)$. First a geometrical approach is taken which has as advantage that it is easier to see how the solution is built up. A disadvantage of such a geometrical approach is that time is disregarded, but this will only play a role when the rolling object is not in the shape of a ball (e.g. an ellips). Second an analytical approach is taken which has as advantage that it will give a general solution which includes the variable time.

4.1 Geometrical approach

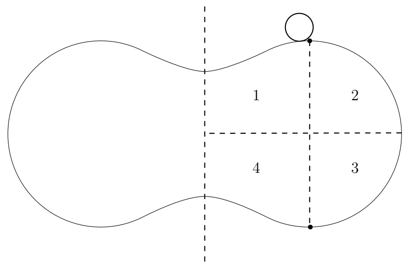

Figure 9: 'Butterfly' robot and important sections

The 'Butterfly' robot is half symmetrical, so it is enough to consider only the right part of it. This right part consists of four sections in which the relation between $\varphi(\phi)$ differ a bit. These four sections are represented in Figure 9. The division between the sections are made for the following reasons

- The division between the upper half and lower half are made due to the reason that $\phi > \varphi$ in the upper half, while $\phi < \varphi$ in the lower half. This division happens at an angle of $\phi = 0.5\pi$

- The division between section 1 and 2 (and thus also between 3 and 4), is made due to the reason that the tangent vector changes direction (e.g. from upward to downward). This division happens at an angle in which $\tau_y(\phi) = 0$, which happens around $\phi \approx 0.22\pi$ and $\phi \approx 0.78\pi$.

4.1.1 Section 1

When looking at Figure 10, it can be seen that φ is equal to ∠CAD (angle of corner A of triangle ACD). By regarding the △ACD (triangle ACD), the following expression can be made for φ:

$$\varphi = \angle CAD = \pi - \angle ADC - \angle ACD \tag{4.1}$$

To get ∠ADC an expression for α_1 must first be gotten. Notice that α_1 is related to the tangent vector $\vec{\tau}$ by the following expression in which $\vec{R}_p$ is the position vector from the inertial reference frame to the point contact of the ball (note that δ is described in (2.14)).

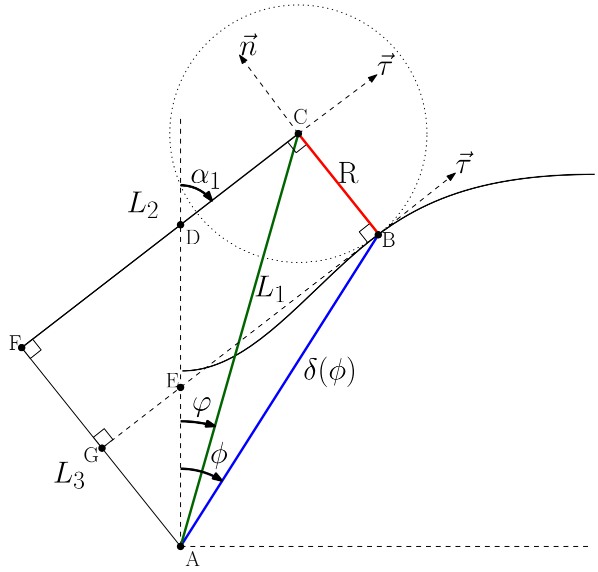

Figure 10: Section 1

$$
\begin{aligned}
\vec{R}_p &= \begin{pmatrix} \sin(\phi)\cdot\delta(\phi) \\ \cos(\phi)\cdot\delta(\phi) \\ 0 \end{pmatrix} \\
\vec{\tau} &= \frac{\frac{d\vec{R}_p}{d\phi}}{||\frac{d\vec{R}_p}{d\phi}||} \\
\alpha_1 &= \arctan(\frac{\tau_x}{\tau_y})
\end{aligned} \tag{4.2}
$$

With α_1 defined, ∠ADC can be formulated by the following expression.

$$\angle ADC = \pi - \alpha_1 \tag{4.3}$$

The second unknown ∠ACD can be gotten from the lines L_2 (CF) and L_3 (AF).

$$\angle ACD = \arctan(\frac{L_3}{L_2}) \tag{4.4}$$

To get L_2, △ABG should be regarded in which side BG has the same length as L_2. Notice that L_3 can also be gotten via this triangle as L_3 is equal to AG+FG, while FG is equal to the 'effective radius' of the ball 'R'.

$$
\begin{aligned}
L_2 &= \delta(\phi)\cos(\angle ABG) \\
L_3 &= \delta(\phi)\sin(\angle ABG) + R
\end{aligned} \tag{4.5}
$$

$$\angle ABG = \angle ABE \tag{4.6a}$$
$$\angle ABE = \pi - \phi - \angle AEB \tag{4.6b}$$
$$\angle AEB = \angle ADC = \pi - \alpha_1 \tag{4.6c}$$

This will eventually deliver the following expression for φ.

$$\varphi = \pi - \angle ADC - \angle ACD = \alpha_1 - \arctan(\frac{L_3}{L_2}) \tag{4.7}$$

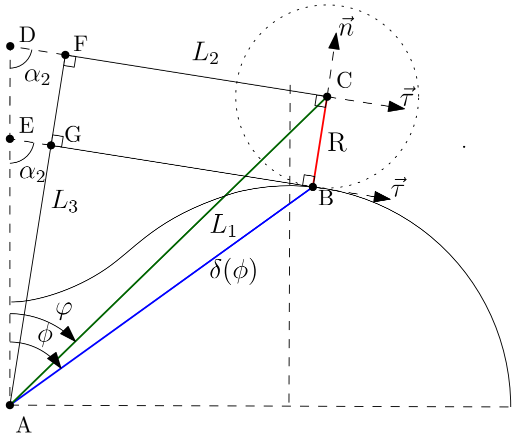

Figure 11: Section 2

4.1.2 Section 2

$$\angle ABG = \pi - \alpha_2 - \phi \tag{4.8a}$$

$$\alpha_2 = \arctan(\frac{\tau_x}{-\tau_y}) \tag{4.8b}$$

$$L_2 = \delta(\phi)\cos(\angle ABG) \tag{4.8c}$$

$$L_3 = \delta(\phi)\sin(\angle ABG) + R \tag{4.8d}$$

$$\varphi = \pi - \alpha_2 - \arctan(\frac{L_3}{L_2}) \tag{4.8e}$$

4.1.3 section 3

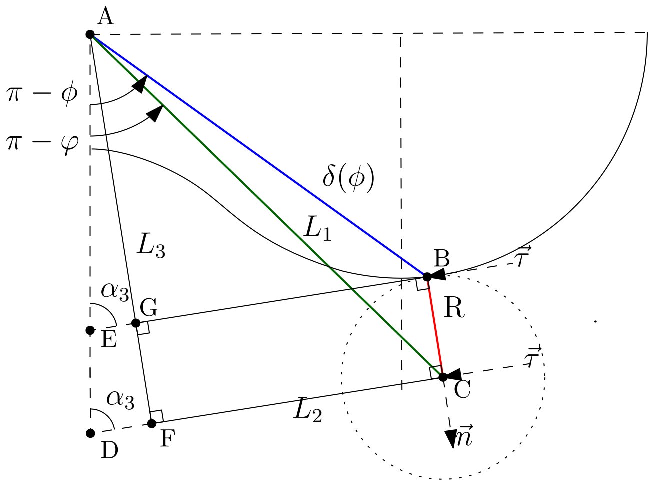

Figure 12: Section 3

$$\angle ABG = \angle ABE = \pi - \alpha_3 - (\pi - \phi) = \phi - \alpha_3 \tag{4.9a}$$
$$\alpha_3 = \arctan(\frac{-\tau_x}{-\tau_y}) \tag{4.9b}$$
$$L_2 = \delta(\phi)\cos(\angle ABG) \tag{4.9c}$$
$$L_3 = \delta(\phi)\sin(\angle ABG) + R \tag{4.9d}$$
$$\pi - \varphi = \pi - \alpha_3 - \arctan(\frac{L_3}{L_2}) \rightarrow \varphi = \alpha_3 + \arctan(\frac{L_3}{L_2}) \tag{4.9e}$$

4.1.4 Section 4

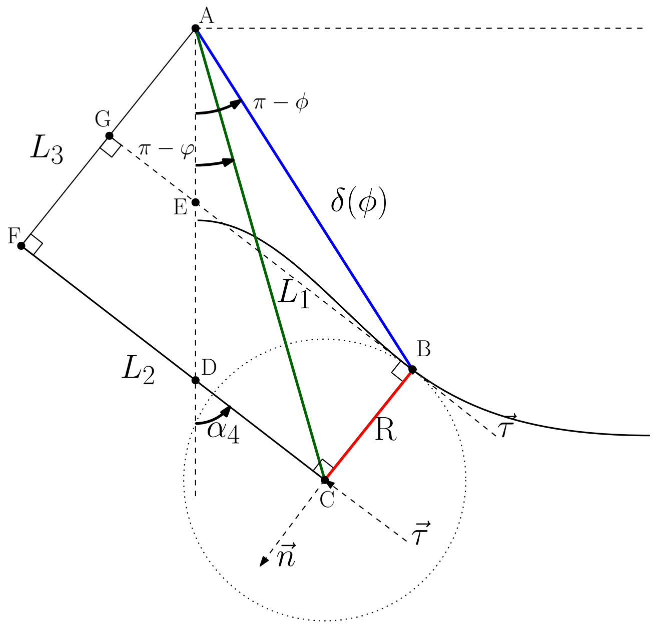

Figure 13: Section 4

$$\angle ABG = \angle ABE = \pi - (\pi - \alpha_4) - (\pi - \phi) = -\pi + \phi + \alpha_4 \tag{4.10a}$$
$$\alpha_4 = \arctan(\frac{-\tau_x}{\tau_y}) \tag{4.10b}$$
$$L_2 = \delta(\phi)\cos(\angle ABG) \tag{4.10c}$$
$$L_3 = \delta(\phi)\sin(\angle ABG) + R \tag{4.10d}$$
$$\pi - \varphi = \pi - (\pi - \alpha_4) - \arctan(\frac{L_3}{L_2}) \rightarrow \varphi = \pi - \alpha_4 + \arctan(\frac{L_3}{L_2}) \tag{4.10e}$$

4.1.5 Total equation

For the right part of the 'Butterfly' robot, the following holds with $L_3 = \delta(\phi)\sin(\angle ABG) + R$ and $L_2 = \delta(\phi)\cos(\angle ABG)$.

$$0 < \phi <\approx 0.22\pi \begin{cases} \angle ABG = \alpha_1 - \phi \\ \varphi = \alpha_1 - \arctan\frac{L_3}{L_4} \end{cases} \tag{4.11}$$

$$\approx 0.22\pi < \phi < 0.5\pi \begin{cases} \angle ABG = \pi - \alpha_2 - \phi \\ \varphi = \pi - \alpha_2 - \arctan \frac{L_3}{L_4} \end{cases} \tag{4.12}$$

$$0.5\pi < \phi <\approx 0.79\pi \begin{cases} \angle ABG = \phi - \alpha_3 \\ \varphi = \alpha_3 + \arctan \frac{L_3}{L_4} \end{cases} \tag{4.13}$$

$$\approx 0.79\pi < \phi < \pi \begin{cases} \angle ABG = -\pi + \phi + \alpha_4 \\ \varphi = \pi - \alpha_4 + \arctan(\frac{L_3}{L_2}) \end{cases} \tag{4.14}$$

The 'Butterfly' robot is half symmetrical, which means that for an angle $\pi > \phi > 2\pi$ the same relations will hold as for angle $0 > \phi > \pi$. Noticing that α_i with i$\in\{1, 2, 3, 4\}$ is always the same relation $\frac{\tau_x}{\tau_y}$ which is made positive, makes it possible to define a general variable β.

$$\beta = |\frac{\tau_x}{\tau_y}| = \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 \tag{4.15}$$

When resolving the left part of the 'Butterfly' robot, it should be noted that $(\phi - \pi)$ must be used instead of ϕ when calculating ∠ABG. This can be seen on Figure 14 in which there are two triangles △ABG and △A'B'G'. To make sure that ∠ABG = ∠A'B'G', it can be seen that $(\phi' - \pi)$ is used instead of ϕ'. Taking into account all these remarks, the following can then be concluded for the left part of 'Butterfly' robot.

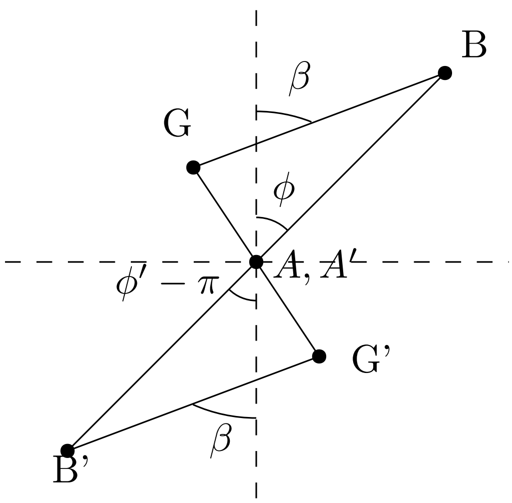

Figure 14: Remark ∠ ABG when working on the left part of the 'Butterfly' robot

$$\pi < \phi <\approx 1.22\pi \begin{cases} \angle ABG = \beta - \phi + \pi \\ \varphi = \pi + \beta - \arctan \frac{L_3}{L_4} \end{cases} \tag{4.16}$$

$$\approx 1.22\pi < \phi_p < 1.5\pi \begin{cases} \angle ABG = 2\pi - \beta - \phi \\ \varphi = 2\pi - \beta - \arctan \frac{L_3}{L_4} \end{cases} \tag{4.17}$$

$$1.5\pi < \phi_p <\approx 1.79\pi \begin{cases} \angle ABG = \phi - \pi - \beta \\ \varphi = \pi + \beta + \arctan \frac{L_3}{L_4} \end{cases} \tag{4.18}$$

$$\approx 1.79\pi < \phi_p < 2\pi \begin{cases} \angle ABG = -2\pi + \phi + \beta \\ \varphi = 2\pi - \beta + \arctan(\frac{L_3}{L_2}) \end{cases} \tag{4.19}$$

There are eight equations $\varphi(\phi)$ needed to describe the whole 'Butterfly' robot. It may be better to only use one equation by combining all these eight equations. The following equation can be gotten when these equations are combined. It has to be noted that this combined equation is undefined on the angles in which the sections switch, like $\phi = 0.5\pi$.

$$\begin{cases} \angle BAE = \phi - (1 - \frac{\sin(2\phi)}{|\sin(2\phi)|})(\frac{-\pi+2\phi}{2}) - (1 - \frac{\sin(\phi)}{|\sin(\phi)|})\frac{\pi}{2} \\ \angle AEB = (1 + \frac{\sin(\phi_p)}{|\sin(\phi_p)|}\frac{v_{p,y}}{|v_{p,y}|})\frac{\pi}{2} - \frac{\sin(\phi_p)}{|\sin(\phi_p)|}\frac{v_{p,y}}{|v_{p,y}|}\beta \\ \angle ABG = \pi - \angle BAE - (1 - \frac{\sin(\phi)}{|\sin(\phi)|})\frac{\pi}{2}) - \angle AEB \\ L_4 = \delta(\phi)\cos(\angle ABG) \\ L_3 = \delta(\phi)\sin(\angle ABG) + R \\ \varphi = \pi - \angle AEB + (1 - \frac{\sin(2\phi)}{|\sin(2\phi)|})(\frac{-\pi+2\angle AEB}{2}) - \frac{\sin(2\phi)}{|\sin(2\phi)|}\arctan\frac{L_3(\phi)}{L_4(\phi)} + (1 - \frac{\sin(\phi)}{|\sin(\phi)|})\frac{\pi}{2} \end{cases} \tag{4.20}$$

An explanation of this equation and 'Maple' script can be found in appendix D. A 'Matlab' script is also given as a measure to check the output of the Maple script. To clarify the dismissal of time as mentioned before, imagine the rolling object as an ellips. This ellips has no constant radius and its 'effective radius' (distance from the point of contact towards the center of the ellips) will vary depending on its initial condition. The geometrical approach can not be taken anymore as time is disregarded in it.

4.2 Analytical approach

The coordinates of point contact with respect to the inertial reference frame is given by the following equations (see Figure 8 for more clarification).

$$x_p(t) = R_p(t) \cdot \sin(\phi(t)) \tag{4.21a}$$
$$y_p(t) = R_p(t) \cdot \cos(\phi(t)) \tag{4.21b}$$

In the same manner the location of the center of the ball can be written as the following.

$$x_c(t) = R_c(t) \cdot \sin(\varphi(t)) \tag{4.22a}$$
$$y_c(t) = R_c(t) \cdot \cos(\varphi(t)) \tag{4.22b}$$

The relation between $\vec{R}_p$ and $\vec{R}_c$ (note $||\vec{R}_p|| = R_p, ||\vec{R}_c|| = R_c$) in case of point contact can be written as follows, in which $\vec{n}_p$ is the normal vector of the point contact.

$$\vec{R}_c = \vec{R}_p + R \cdot \vec{n}_p \tag{4.23}$$

knowing that $\vec{n}_p$ is perpendicular to $\vec{\tau}_p$ and always pointing outwards (as can be seen on e.g. Figure 4), it can be calculated by rotating the $\vec{\tau}_p$ 90 degrees anti-clockwise in case of clockwise rotation of the ball. For $\vec{\tau}_p$ the following expression holds, where v_p denotes the velocity of the point contact.

$$v_{x,p}(t) = \dot{R}_p(t) \cdot \sin(\phi(t)) + R_p \cdot \dot{\phi}(t) \cdot \cos(\phi(t)) \tag{4.24a}$$
$$v_{y,p}(t) = \dot{R}_p(t) \cdot \cos(\phi(t)) - R_p \cdot \dot{\phi}(t) \cdot \sin(\phi(t)) \tag{4.24b}$$
$$||v_p(t)|| = \sqrt{R_p(t)^2\dot{\phi}(t) + \dot{R}_p(t)^2} \tag{4.24c}$$
$$\vec{\tau}_p(t) = \frac{\vec{v}_p(t)}{||\vec{v}_p(t)||} \tag{4.24d}$$

Rotating $\vec{\tau}$ by 90 degrees anti-clockwise will then give

$$\begin{aligned} \vec{n}_p(t) &= \begin{pmatrix} \cos(0.5\pi) & -\sin(0.5\pi) \\ \sin(0.5\pi) & \cos(0.5\pi) \end{pmatrix} \cdot \vec{\tau}_p(t) \\ &= \begin{pmatrix} \frac{R_p(t)\cdot\dot{\phi}(t)\cdot\sin(\phi(t)) - \dot{R}_p(t)\cdot\cos(\phi(t))}{\sqrt{R_p(t)^2\dot{\phi}(t)+\dot{R}_p(t)^2}} \\ \frac{\dot{R}_p(t)\cdot\sin(\phi(t)) + R_p(t)\cdot\dot{\phi}(t)\cdot\cos(\phi(t))}{\sqrt{R_p(t)^2\dot{\phi}(t)+\dot{R}_p(t)^2}} \end{pmatrix} \end{aligned} \tag{4.25}$$

Assuming that R_p(t) is equal to $\delta(\phi(t))$ (2.14) as they both describe the same curve γ_p, will give $R_p(t) = \delta(\phi(t)) = R_p(\phi(t))$. Inserting this new R_p into the previous equation will give the following.

$$\dot{R}_p(\phi(t)) = \frac{dR_p}{d\phi}\dot{\phi}(t) \tag{4.26a}$$

$$\vec{n}_p(t) = \frac{\dot{\phi}(t)}{|\dot{\phi}(t)|}\begin{pmatrix} \frac{R_p(\phi(t))\cdot\sin(\phi(t)) - \frac{dR_p}{d\phi}\cos(\phi(t))}{\sqrt{R_p(t)^2 + \frac{dR_p}{d\phi}^2}} \\ \frac{R_p(\phi(t))\cdot\cos(\phi(t) + \frac{dR_p}{d\phi}\sin(\phi(t))}{\sqrt{R_p(\phi(t))^2 + \frac{dR_p}{d\phi}^2}} \end{pmatrix} \tag{4.26b}$$

Filling (4.26b) into (4.23) will give an expression for $\vec{R}_c(\phi(t))$ which can then be used to describe x_c and y_c according to (4.22).

$$x_c(t) = R_c(t)\sin(\varphi(t)) = R_p(\phi(t))\sin(\phi(t)) + R\frac{R_p(\phi(t))\cdot\sin(\phi(t)) - \frac{dR_p}{d\phi}\cos(\phi(t))}{\sqrt{R_p(\phi(t))^2 + \frac{dR_p}{d\phi}^2}}\frac{\dot{\phi}(t)}{|\dot{\phi}(t)|} \tag{4.27a}$$

$$y_c(t) = R_c(t)\cos(\varphi(t)) = R_p(\phi(t))\cos(\phi(t)) + R\frac{R_p(\phi(t))\cdot\cos(\phi(t) + \frac{dR_p}{d\phi}\sin(\phi(t))}{\sqrt{R_p(\phi(t))^2 + \frac{dR_p}{d\phi}^2}}\frac{\dot{\phi}(t)}{|\dot{\phi}(t)|} \tag{4.27b}$$

With (4.27) the following φ(t) expression can then be gotten.

$$\varphi(t) = \arctan(\frac{x_c(t)}{y_c(t)}) \tag{4.28}$$

When 't' (note this is not real 'time') is taken equal as ϕ, the following relation can be gotten

$$\varphi(\phi) = \arctan(\frac{R_p(\phi)\sin(\phi)\sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2} - R\frac{\dot{\phi}}{|\dot{\phi}|}(-R_p(\phi)\cdot\sin(\phi) + \frac{dR_p}{d\phi}\cos(\phi))}{R_p(\phi)\cos(\phi)\sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2} + R\frac{\dot{\phi}}{|\dot{\phi}|}(R_p(\phi)\cdot\cos(\phi) + \frac{dR_p}{d\phi}\sin(\phi))}) \tag{4.29}$$

Notice that this equation only gives solutions φ ranging from 0 to 0.5π. π should be added when dealing with higher angles of ϕ as input, as can be seen on (4.30).

$$\begin{aligned} 0 > \phi > 0.5\pi &: \varphi = \varphi(\phi) \\ 0.5\pi > \phi > \pi &: \varphi = \pi + \varphi(\phi) \\ \pi > \phi > 1.5\pi &: \varphi = \pi + \varphi(\phi) \\ 1.5\pi > \phi > 2\pi &: \varphi = 2\pi + \varphi(\phi) \end{aligned} \tag{4.30}$$

It is also possible to get $R_c(t)$. This can be done by solving the following equation

$$R_c(t)^2(\sin^2(\varphi(t)) + \cos^2(\varphi(t))) = x_c(t)^2 + y_c(t)^2 \tag{4.31}$$

Replacing 't' for ϕ will then result in the following (notice that for unidirectional rolling the term $\frac{\dot{\phi}}{|\dot{\phi}|} = 1$).

$$R_c(\phi) = \sqrt{\frac{(R^2\frac{\dot{\phi}}{|\dot{\phi}|}^2 + R_p(\phi)^2)\sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2} + 2R_p(\phi)^2 R\frac{\dot{\phi}}{|\dot{\phi}|}}{\sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2}}} \tag{4.32}$$

Now with $R_c(\phi)$ and ϕ, the curve γ_c can be parameterized in ϕ. Using 'Maple' (appendix D.4) the following value of R_c can be gotten with $R_p(\phi) = 0.1095 - 0.0405\cos(2\phi)$.

$$
\begin{aligned}
Var1 &= 0.0001176524999 + (0.1095 - 0.0405\cos(2\phi))^2 \\
R_c(\phi) &= \sqrt{Var1 + \frac{0.02169354742((0.1095 - 0.0405\cos(2\phi))^2)}{\sqrt{(0.1095 - 0.0405\cos(2\phi))^2 + 0.00656100(\sin(2\phi)^2)}}}
\end{aligned}
\tag{4.33}
$$

The same can be done for the angle $\varphi(\phi)$ as can be seen below. Note that the fraction goes on in the second row as the fraction was too long to fit on one line.

$$
\begin{aligned}
\varphi = \arctan(&\frac{(0.1095 - 0.0405\cos(2\phi))\sin(\phi)\sqrt{(0.1095 - 0.0405\cos(2\phi))^2 + 0.00656100\sin(2\phi)^2}}{(0.1095 - 0.0405\cos(2\phi))\cos(\phi)\sqrt{(0.1095 - 0.0405\cos(2\phi))^2 + 0.00656100\sin(2\phi)^2}} \\
&\frac{-0.0008785886705\cos(\phi)\sin(2\phi) + 0.01084677371(0.1095 - 0.0405\cos(2\phi))\sin(\phi)}{0.0008785886705\sin(\phi)\sin(2\phi) + 0.01084677371(0.1095 - 0.0405\cos(2\phi))\cos(\phi)})
\end{aligned}
\tag{4.34}
$$

4.3 Summary

Two methods have been discussed for getting $\varphi(\phi)$. The geometrical approach divided the 'Butterfly' robot in several sections where the geometrical relations were analyzed. For all these sections a $\varphi(\phi)$ relation can be gotten, which can eventually be combined into one equation which is valid for most of the time. The most important part is not necessary this total equation, but rather the geometrical relations for getting $\phi(\varphi)$ which seems to be sort of the same for all sections. These geometrical relations may then also be applied for the analysis in next chapter.

For the analytical approach usage has been made of the fact that the curve γ_c (and thus the center of the ball) is always a normal distance 'R' away from the curve γ_p (thus its point contact). Simply with this knowledge, an equation can be derived for all sections (with some minor adjustments to the output). As time does not play a role since the ball radius is constant, it thus does not really matter whether the geometrical or analytical approach is taken as they would give the same result. The approaches for getting the result is however different and may deliver different results in the analysis for the next chapter. It shows that there is not just a single way in getting an answer, but most often multiple ways.

5 Expressing angle of point contact in angle of the center of the ball

In this chapter attempts are made in getting the relation $\phi(\varphi)$. Being able in getting an analytical expression for $\phi(\varphi)$ will result in being able to make a 'better' model than the 'old' model, in the sense that it can be applied in more situations. It is however seen that such an analytical expression is rather difficult to get and that the reason for this is the inability to express γ_c in φ. The first attempt is to take an inverse relation of the previously gotten $\varphi(\phi)$. After that, attempts similar as in previous chapter have been made. This time the analytical attempt is extended in which also the arc length 's$_p$' and trigonometric polynomials are taken into account.

5.1 Inverse relation

From previous chapter, expressions of $\varphi(\phi)$ have been derived. For simple equations it is often possible to take the inverse relation of it with respect to a certain variable, for example the equation below in which the variables 'a' and 'b' play a role.

$$a(b) = b + 2 \tag{5.1a}$$
$$b(a) = a - 2 \tag{5.1b}$$

Initially it is thought that this could also be done for $\varphi(\phi)$, but with a somewhat more difficult inverse relation. In order to solve this, 'Maple' has been used with a 'solve' command as can be seen below. Hereby the previously gotten $\varphi(\phi)$ is renamed to a certain function 'f(.)'.

$$\varphi(\phi) = f(\phi) \tag{5.2a}$$
$$solve(\varphi = f(\phi), \phi) \tag{5.2b}$$

This has been done for both analytical and geometrical expressions from previous chapter. For the geometrical expression only one section and its $\varphi(\phi)$ (e.g. (4.7)) is considered, as the total equation described in (4.20) is a bit too complex.

Geometrical solution $\varphi(\phi)$ Section 1

$$\begin{aligned} \alpha_1 &= -\arctan\left(\frac{1.620\cos(\phi)^3 - 2.08\cos(\phi)}{1.62\sin(\phi)\cos(\phi)^2 - \sin(\phi)}\right) \\ \varphi &= \alpha_1 + \arctan\left(\frac{0.01084677371 + (-0.1095 + 0.0405\cos(2\phi))\sin(\phi - \alpha_1)}{(-0.1095 + 0.0405\cos(2\phi))\cos(\phi - \alpha_1)}\right) \end{aligned} \tag{5.3}$$

Equation (5.3) is based on (4.7) and is derived with 'Maple' using a 'simplify' command. The 'Maple' script can be made by making small adjustments in the script given in appendix D.2. It is seen that the $\varphi(\phi)$ expression is a rather complex trigonometric function, which makes it difficult (or even impossible) to solve by hand. Even with the help of 'Maple', it is not possible to take the inverse relation of this expression. This is most likely due to the fact that ϕ components are embedded deeply into all these sinus/cosinus expressions. Trying to take it apart in the format of $\phi(\varphi)$ with only one φ on the left hand side is quite the task (if it is even possible).

Analytical solution $\varphi(\phi)$

When taking a look at (4.34), it is seen that it is equally complex as (5.3). It is probably for the same reasons as the geometrical solution that 'Maple' could not solve this.

5.2 Geometrical attempts

A similar approach as in section 4.1 is taken in which a look is given at triangles and its relations. In this chapter, two of multiple attempts are shown as they have been regarded as the best (in the sense that they looked promising) attempts. Other geometrical attempts can be seen in appendix E as they were less promising.

5.2.1 Express terms in φ (example section 1)

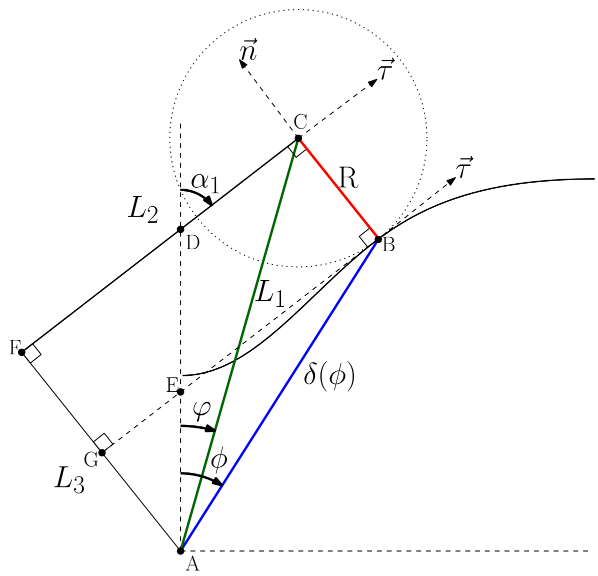

Figure 15: Section 1

Unlike as in section 4.1, it is now focused on trying to express ϕ in φ. A look can then be given at Figure 15, in which now the focus is on $\triangle$ABE.

$$\phi = \pi - \angle ABE - \angle AEB = \alpha_1 - \arctan\left(\frac{L_3 - R}{L_2}\right) \tag{5.4}$$

Getting α_1, L_2 and L_3 expressed in φ would solve (5.4). This gives the following expressions

$$L_3 = L_1 \cdot \sin(\angle ACD) \tag{5.5a}$$

$$L_2 = L_1 \cdot \cos(\angle ACD) \tag{5.5b}$$

$$\angle ACD = \alpha_1 - \varphi \tag{5.5c}$$

$$\vec{R}_c = \begin{pmatrix} \sin(\varphi) \cdot L_1 \\ \cos(\varphi) \cdot L_1 \\ 0 \end{pmatrix} \tag{5.5d}$$

$$\vec{\tau} = \frac{\frac{d\vec{R}_p}{d\phi}}{||\frac{d\vec{R}_p}{d\phi}||} = \frac{\frac{d\vec{R}_c}{d\varphi}}{||\frac{d\vec{R}_c}{d\varphi}||} \tag{5.5e}$$

From these expressions, it can be seen that L_1 is important as it is used in all three variables α_1, L_2 and L_3 (note that α depends on $\vec{\tau}$ which then depends on R_c). Being able to parameterise L_1 in φ would then let us solve (5.4) in the format that is desired. Note that L_1 is equal to R_c and that a solution of $L_1(\varphi)$ would mean a description of the curve γ_c (or R_c) in φ. The problem that is present, is that L_1 can not be solely expressed in φ. One way of getting L_1 could be done via (5.6), but this equation depends on variables L_2 and L_3 which as we have seen depends on L_1 directly and indirectly (via α).

$$L_1 = \sqrt{L_2^2 + L_3^2} \tag{5.6}$$

Another option is then to rewrite as many components as possible in φ while allowing some ϕ components in it. This however is also unsolvable as there will always remain an α_1 term in it due to the angles and geometric relations used. When a look is given at the complexity of this α_1 expression, it is seen that it is difficult to take ϕ apart from it.

$$\alpha_1 = \arctan\left(\frac{1.62\cos(\phi)^3 - 2.08\cos(\phi)}{1.62\sin(\phi)\cos(\phi)^2 - \sin(\phi)}\right) \tag{5.7}$$

Eventually it all comes down to the problem that L_1(or R_c, γ_c) is not parameterisable in the variable φ. So if $\gamma_c(\varphi)$ was known beforehand instead of $\gamma_p(\phi)$, then this problem was a lot easier to solve.

5.2.2 Change constrained center of ball curve (γ_c)

The problem that is now present, is that $L_1(\varphi)$ (or $\gamma_c(\varphi)/R_c(\varphi)$) is unknown. The idea is to change the model in such a way that a new constrained center of ball curve $\gamma_{c,new}(\varphi)$, for which we can set up an analytical expression in φ, can be made while still upholding the same relation $\varphi(\phi)$ as described in chapter 4. An example of this can be seen in Figure 16. Here the actual point contact curve (γ_p) will become the new constrained center of ball curve ($\gamma_{c,new}$), which will also deliver a new point contact curve ($\gamma_{p,new}$) on which a ball with a smaller effective radius R_3 (dashed circle) will roll. Note that the normal ball with effective radius 'R' and the smaller dashed ball with effective radius R_3 have the same angles φ and ϕ. A remark should be made that also for the smaller ball, the center of the ball is a normal distance R_3 away from the curve ($\gamma_{p,new}$) (even though this may not be depicted that clearly on the picture).

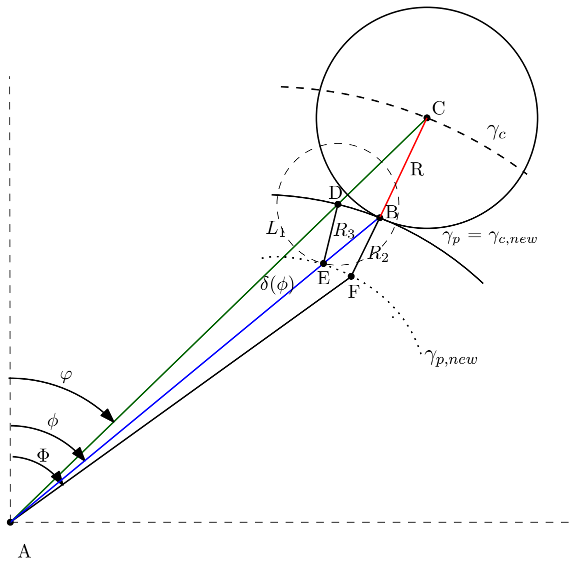

Figure 16: Model change for new curves

Initially it was thought that this smaller ball would also have a constant effective radius R_3. From calculations it can be determined that the effective radius of the smaller ball changes in order to fulfill the earlier set relation $\varphi(\phi)$. This would thus mean that $R_3 \neq R_2$ and that the effective radius of the smaller ball is angle dependent. To get the expression $\phi(\varphi)$ it was thought about using point E, which will give the following relation

$$\phi = \arctan(\frac{E_x}{E_y}) \tag{5.8}$$

Now it is desired to get the expression of the vector from inertial reference frame towards E in φ coordinate, thus

$$\vec{E}(\varphi) = \vec{D}(\varphi) - \vec{R}_3(\varphi) \tag{5.9}$$

Here $||\vec{D}||$ is easily expressed in φ as it is equal to $\delta(\varphi)$ (note that it just describes the old point contact curve γ_p, but this time reparameterized in φ). The only unknown would then be $\vec{R}_3(\varphi)$, thus getting $R_3(\varphi)$ would solve this equation. To get the equation of R_3, a look is given at R_2 and the angle Φ (note that $\vec{R}_2$ and $\vec{R}$ are in the same direction). Taking the same geometrical approaches as in chapter 4.1 will then give (5.10) (α, L_2 and L_3 have similar expressions as in section 5.1).

$$\Phi = -\arctan(\frac{L_3(\phi) - R - R_2}{L_2(\phi)}) + \pi - \alpha(\phi) \tag{5.10}$$

Noticing the similarity between △ABF and △ADE, allows the equation to be extended for ϕ.

$$\phi = -\arctan(\frac{L_3(\varphi) - R - R_3}{L_2(\varphi)}) + \pi - \alpha(\varphi) \tag{5.11}$$

The problem now is that $R_3(\varphi)$ cannot be gotten from (5.11). There is always one ϕ component on the left hand side which makes it complicated. Notice that it is possible to express R_3 as a function of ϕ when using the relation $\varphi(\phi)$ (e.g. (4.7)) in combination with (5.11). This however will not give any help in getting a $\phi(\varphi)$ expression. The problem that is now present, is that the constraint $\varphi(\phi)$ causes a non-constant effective radius R_3 of the ball which complicates the whole process. So making a new constraint curve $\gamma_{c,new}$ is useless when $\gamma_{p,new}$ is restricted in a certain way (here thus by needing to fulfill previous $\varphi(\phi)$ relation).

5.3 Analytical attempts

As the geometrical attempts could not solve the $\phi(\varphi)$ problem, a look has been given at analytical approaches. The first attempt was by doing the same approach as described in section 4.2. This approach was not satisfying and for this reason attempts were also done based on the arc length parameter 's_p' and trigonometric approximations.

5.3.1 Expressing R_p(t')

The problem that was determined in section 5.2.1 was that there was no expression for $R_c(\varphi)$. Being able to derive this expression will immediately lead to an expression of $\phi(\varphi)$. In section 4.2 an expression for $R_c(\phi)$ was determined by replacing the variable 't' in $R_c(t)$ with the variable ϕ. Note that simply replacing this variable 't' with 'φ' would not give an analytical expression for $R_c(\varphi)$ as it will then also depend on the variable $\phi(\varphi)$ (4.27).

The same approach as in section 4.2 can be taken, but this time the focus will be on R_p and a new variable '(t')' is used instead of 't'. Via $\vec{R}_p$ it is possible to gain an expression of ϕ. Being able to denote $R_p(\varphi)$ will allow us to get an expression for $\phi(\varphi)$. The vector for the point contact can be written as follows

$$\vec{R}_p(t') = \vec{R}_c(t') - R\vec{n}_c(t') \tag{5.12}$$

When working this expression out as in section 4.2, this will give a $R_p(t')$ expression which is a function of $\varphi(t')$ and $R_c(t')$.

$$R_p(t') = f(R_c(t'), \varphi(t')) \tag{5.13}$$

When replacing '(t')' with φ, it can be seen that it is necessary to know $R_c(\varphi)$ to be able to solve $R_p(\varphi)$ and thus $\phi(\varphi)$. So the same problem as in section 5.2.1 is noticed here.

5.3.2 Reparameterization in arc length

Another approach is to determine if it is possible to denote the arc length 's_p' in the variable φ. The only possible analytical notations of the arc length s_p and s_c can be seen in (5.14).

$$s_p = \int_0^{\phi} \|\frac{d\vec{R}_p(\phi)}{d\phi}\| d\phi \tag{5.14a}$$

$$s_c = \int_0^{\phi} \|\frac{d\vec{R}_c(\phi)}{d\phi}\| d\phi \tag{5.14b}$$

the idea of the approach would be the one given below, in which it is tried to relate ϕ and φ via the arclengths s_p and s_c. However from (5.14b) it can be seen that s_c is not expressible as φ as it requires knowledge of $R_c(\varphi)$.

$$R_c(\phi) \to \phi(s_p) \to s_p(s_c) \to s_c(\varphi) \tag{5.15}$$

5.3.3 Velocity arc length and reparameterization in time

It may be that the velocity of the arc length could be of more help. Using it, it may be possible to derive expressions of $\phi(t)$ and $\varphi(t)$. The derivative of the arc length s_p can be derived via the second fundamental theorem of calculus.

$$\begin{aligned} s_p &= \int_0^{\phi} \|\frac{d\vec{R}_p(\phi)}{d\phi}\| d\phi \\ s_p &= \int_0^{\phi} \|\frac{d\vec{R}_p(\phi)}{d\phi}\| dt \frac{d\phi}{dt} \\ \dot{s}_p &= \|\frac{d\vec{R}_p(\phi)}{d\phi}\| \dot{\phi} \end{aligned} \tag{5.16}$$

When assuming unit velocity of s_p, the following $\dot{\phi}$ can be found which is a function of ϕ itself.

$$\dot{s}_p = \|\frac{d\vec{R}_p(\phi)}{d\phi}\| \dot{\phi} = 1 \tag{5.17a}$$

$$\dot{\phi}(t) = \frac{1}{\|\frac{d\vec{R}_p(\phi)}{d\phi}\|} = f(\phi) \tag{5.17b}$$

Solving (5.17b) will then deliver an expression for $\phi(t)$. Substituting this in the previous gotten relation $\varphi(\phi)$ will then give $\varphi(\phi(t))$, thus an expression of $\varphi(t)$. As the two variables ϕ and φ are now defined in the variable 'time', it may be possible to determine $\varphi(t)$ components in $\phi(t)$.

$$\vec{R}_p(\phi) = \begin{pmatrix} \sin(\phi)(0.1095 - 0.0405\cos(2\phi)) \\ \cos(\phi)(0.1095 - 0.0405\cos(2\phi)) \\ 0 \end{pmatrix} \tag{5.18}$$

Using the shape of the 'Butterfly' robot as given in (5.18), the following expression for $\dot{\phi}(t)$ with $\dot{s}_p = 1$ can be derived.

$$\begin{aligned} \|\frac{d\vec{R}_p}{d\phi}\| &= \sqrt{0.0225 - 0.019683\cos(\phi)^4 + 0.0019440\cos(\phi)^2} \\ \dot{\phi}(t) &= \frac{1}{\sqrt{0.0225 - 0.019683\cos(\phi)^4 + 0.0019440\cos(\phi)^2}} \end{aligned} \tag{5.19}$$

This expression is in a rather difficult format for which I would not know how to solve it. simpler cases like $\dot{x} = x$ could be solved with a solution like $x = e^t$, but $\dot{\phi}$ is not in this format.

5.3.4 Trigonometric polynomial

The curve γ_p can be defined in ϕ by the radius of the plate

$$R_p = \delta(\phi) = 0.1095 - 0.0405\cos(2\phi) \tag{5.20}$$

Equation (5.20) can be seen as a trigonometric polynomial, as the main component of the function is the 'cos(2ϕ)' term. If it is possible to express $R_c(\varphi)$ (and thus also $\gamma_c(\varphi)$) in the same manner as (5.20), then the problem of accurate modeling would also be solved. From Figure 17 it can be seen that the curve γ_c has a similar shape as the curve γ_p, but then with a little more offset. As γ_c has a similar shape as γ_p, it may not be that wrong to think that $R_c(\varphi)$ would have a similar expression as (5.20).

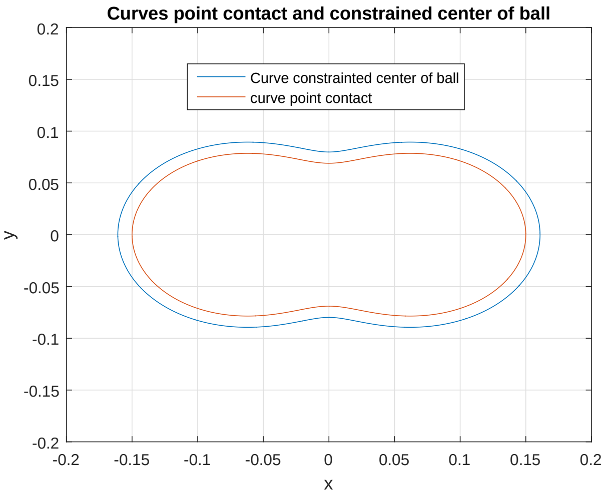

Figure 17: Curves of the 'Butterfly' robot

It is thus checked if R_c can be approximated by the following trigonometric polynomial in which the coefficients have yet to be determined.

$$R_{c,approx}(\varphi) = a_0 + \sum_{n=1}^{N}(a_n\cos(n\varphi) + b_n\sin(n\varphi)) \tag{5.21}$$

Using 'Matlab' (See appendix E.2.1 for more insight) will show that a degree of N=2 will give a good estimate towards the real solution $R_c(\phi)$ in which the highest error is around $7{\cdot}10^{-4}$ m. A higher degree will lessen the error, but will not change the shape of the polynomial drastically as old terms will keep reappearing in higher order approximations. This can be seen in (5.22) in which a general solution for higher order approximation is given.

$$\begin{aligned} &N = 2 \rightarrow R_{c,approx}(\varphi) = a_0 + a_2\cos(2\varphi) + b_1\sin(\varphi) \\ &N > 2 \rightarrow R_{c,approx}(\varphi) = a_0 + \sum_{n=1}^{N}((1-(n \mod 2))a_n\cos(n\varphi) + (n \mod 2)b_n\sin(n\varphi)) \end{aligned} \tag{5.22}$$

Now it is checked if the approximation confirms with the real solution analytically. It is thus checked if those terms like 'cos(2φ)' from $R_{c,approx}$ also appear in the real solution R_c. As $R_c(\phi)$ and $\varphi(\phi)$ were determined before, simply looking if terms like 'cos(2$\varphi(\phi)$)' are present in $R_c(\phi)$ would satisfy this check. This has been done with 'Maple', but none of the attempts (see appendix E.2.2) delivered the result that was desired. Another idea was to look at the derivatives of the expressions and then integrate afterwards. The following derivative was used.

$$\frac{\frac{dR_c(\phi)}{d\phi}}{\frac{d\varphi(\phi)}{d\phi}} = \frac{dR_c}{d\varphi}(\phi) \tag{5.23}$$

When it is possible to rewrite $\frac{dR_c}{d\varphi}$ as a function of φ instead of ϕ, then it is possible to take the integral of this in the following manner to get the desired $R_c(\varphi)$.

$$R_c(\varphi) = \int_0^{\varphi} \frac{dR_c}{d\varphi}(\varphi) d\varphi \tag{5.24}$$

From 'Matlab' the following trigonometric polynomial can be derived from which it can be seen that this is equal to the derivative of (5.22) when ignoring the coefficient values.

$$\begin{aligned} N = 2 \rightarrow \frac{dR_c}{d\varphi}_{approx} &= a_1 \cos(\varphi) + b_2 \sin(2\varphi) \\ N > 2 \rightarrow \frac{dR_c}{d\varphi}_{approx} &= \sum_{n=1}^{N} ((n \mod 2) a_n \cos(n\varphi) + (1 - (n \mod 2)) b_n \sin(n\varphi)) \end{aligned} \tag{5.25}$$

The same attempts as for $R_{c,approx}$ have been made in 'Maple', but this also did not give the desired result. It may be that my 'Maple' skills were not sufficient enough to solve this problem. Another reason could be that the shape of the 'Butterfly' robot is too complex for getting a $R_c(\varphi)$. A simpler shape like an ellips may deliver a solution to this problem, but this has not been (thoroughly) checked in this report due to lack of time.

For an ellips it holds that $(\frac{x}{a})^2 + (\frac{y}{b})^2 = 1$. Using $x = R_{p,ellips} \sin(\phi)$ and $y = R_{p,ellips} \cos(\phi))$ the following can be said for $R_{p,ellips}$

$$R_{p,ellips} = \frac{ab}{\sqrt{(a\cos(\phi))^2 + (b\sin(\phi))^2}} \tag{5.26}$$

Taking an 'a' value of 2, 'b' value of 1 and an effective ball radius of 0.1 will give the following figure.

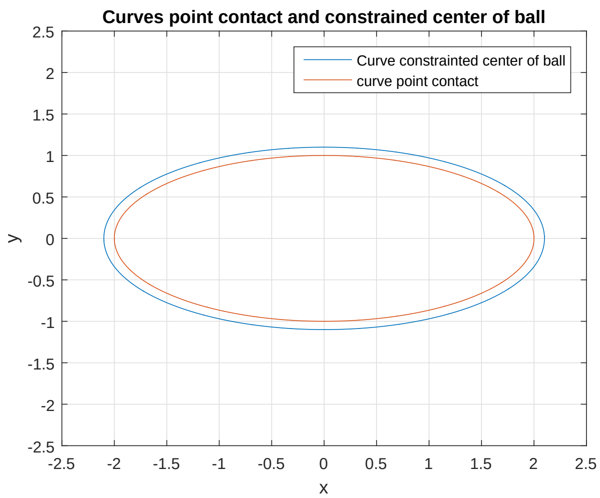

Figure 18: Curves of the ellips shaped 'Butterfly' robot

From this, it can be seen that the curve of γ_c also looks like an ellips. It is then checked if a similar expression can be gotten for $R_{c,ellips}$, which can be seen in (5.27). To check if it is possible to write R_c in such a format is left for future works.

$$R_{c,ellips} = \frac{a_c b_c}{\sqrt{(a_c \cos(\varphi))^2 + (b_c \sin(\varphi))^2}} \tag{5.27}$$

5.4 Summary

Several attempts have been made in trying to get an analytical expressions for $\phi(\varphi)$. The issue with most attempts was the fact that they needed an analytical expression for $\gamma_c(\varphi)$ or $R_c(\varphi)$. As the only information available beforehand was the plate shape in the angle ϕ, it was not possible to get these expressions. Even considering the plate shape as a new (imaginative) $\gamma_c(\varphi)$ with a smaller (imaginitive) ball did not prove to be succesfull as it needed to fulfill the relation $\varphi(\phi)$ defined in previous chapter. This would require that the smaller (imaginative) ball would not have a constant radius and it was thus unsolvable.

As the main problem changed to getting an expression for $R_c(\varphi)$, a last attempt with trigonometric polynomials have been made. The idea was that as the plate shape could be described by a trigonometric polynomial $\delta(\phi)/R_p(\phi)$ (2.14), γ_c which has a similar shape as the plate could then also be described by such a trigonometric polynomial. Via 'Matlab' an approximation $R_{c,approx}(\varphi)$ has been derived which does seem to be in correspondence with a graph made from $R_c(\phi)$ and $\varphi(\phi)$ (for which we do have analytical expressions).

As higher degree polynomials of the approximation will have the same terms reoccuring as 'cos(2φ)', a nice way of validation would then be to check if these terms also reappear in $R_c(\phi)$ by using the known relation $\varphi(\phi)$. Using 'Maple' several attempts have been made for $R_{c,approx}$ and $\frac{dR_c}{d\varphi}_{approx}$. The results however do not show that reappearance of these terms of the approximations are true. This may be due to my insufficient skills in 'Maple', or due to the difficult shape of the 'Butterfly' robot. A recommendation for the future would then also be to look at simpler shapes like ellipses. An even better recommendation would be to put more emphasis on describing the trajectory of the center of the ball $\gamma_c(\varphi)$, as now more emphasis is put on the shape of the plate as the only information provided was $\delta(\phi)$ from [1].

6 Conclusion and recommendation

6.1 Conclusion

In chapter 2 the 'old' model was introduced and also the reason for why an ad hoc transformation of coordinates was applied. The main problem that this ad hoc transformation brings, is the restricted usage of the model. The coordinates only give meaningful values when the ball is in contact with the plates and when the ball does not slip. When these conditions are not valid (and this will happen in reality as stated in [1]), the model will become unaccurate/unapplicable. A secondary problem which may not be directly linked to this ad hoc transformation, is the inability to express $\gamma_c(\varphi)$ which also makes the model less accurate.

To bypass the main problem, a new set of coordinates is proposed in chapter 3, namely polar coordinates. A new problem then arises that this 'new' model can't be reduced unless analytical expressions are present for $\phi(\varphi)$. Being able to solve this new problem also solves the secondary problem in which an analytical expression for $\gamma_c(\varphi)$ is needed. This will then bring a 'new' model which is accurate and can also be applied in both situations where the virtual holonomic constraints are active and not.

In chapter 4 the inverse relation $\varphi(\phi)$ is first sought to get insight in the approaches that may be applied for getting $\phi(\varphi)$. This chapter shows that there are multiple ways of getting the result, as both the geometrical and analytical approaches are valid. In chapter 5 attempts are then made in getting $\phi(\varphi)$, for which it can be seen that the main issue is the inability to express $\gamma_c(\varphi)$. For this reason a last attempt with trigonometric polynomials have been made in trying to get such a $\gamma_c(\varphi)$ approximation.

This approximation always has a certain format in which certain terms reappear. To be absolutely sure that this approximation was valid, it was thus checked if these approximated terms also reappear in the analytical expression of $R_c(\phi)$. Using 'Maple' it has thus been concluded that these terms do not reappear and that the approximation is not that accurate. The highest accuracy is desired, to make sure that it is possible to make a robust controller for other shapes of the plates and rolling objects. So even though the 'Butterfly' robot looks quite simple and the problem of finding a new set of coordinates does not seem that demanding, it can be concluded that this problem is not simple and that it would demand quite the effort.

6.2 Recommendation

The first recommendation is the higher emphasis on describing $\gamma_c(\varphi)$. Now every relation is derived from the information of the plates, namely $\delta(\phi)$ [1]. From the shape of the plates, the shape of the motion of the center of the ball is derived ($\gamma_c(\phi)$). It may be however better to do it the other way around, namely describing the motion of the center of the ball first and from it the shape of the plates. From $\gamma_c(\varphi)$ it would then be possible to get $\gamma_p(\varphi)$ by considering that these curves are always a normal distance 'R' away from each other. The 'new' model can then be reduced to a two degrees of freedom model while also being able to track the ball when it is for example in the air.

Another recommendation is to check if simpler plate shapes may get a $\phi(\varphi)$ expression. A circle would then be a bit too simple, but an ellips might prove to be challenging enough. If the curve $\gamma_{c,ellips}$ also has a shape of an ellips, then checking if (5.27) is valid is enough. Whilst doing this, it may be helpful getting more knowledge of the commands in 'Maple' as my approaches may not be sufficient enough. An even further step could be considering other rolling objects like an ellips, in which the distance between γ_c and γ_p becomes time dependent. For this it may be helpful to analyze the analytical approaches as they can consider time in it.

7 References

References

[1] M. Surov, A. Shiriaev, L. Freidovich, S. Gusev, L. Paramonov (2015). Case study in non-prehensile manipulation: planning and orbital stabilization of one-directional rollings for the "Butterfly" robot. 2015 IEEE international conference on Robotics and Automation (ICRA), May 26-30(2015), 1484-1489. Pages 1, 2, 3, 5, 6, 7, 8,

[2] S.J.A.M. van den Eijnden (2015). Orbital stabilization and robustness analysis of the Butterfly robot with pre-defined periodic trajectories. Eindhoven University of Technology. Pages 2, 5

[3] M. Francke (2015). Motion planning and analysis of the underactuated butterfly robot. Eindhoven University of Technology. Pages 2, 4, 5,

[4] A. Shiriaev, L. Freidovich, S. Gusev(2009). Transverse Linearization for Controlled Mechanical systems with several passive degrees of freedom. 2009 American Control Conference, june 10-12 (2009), 3039-3044. Pages 2

[5] K. Lynch, N. Shiroma, H. Arai, K. Tanie (1998). The roles of shape and motion in dynamic manipulation : The butterfly example. 1998 IEEE international conference on Robotics and Automation (IRCA), may (1998), 1958-1963. Pages 2

[6] N. van de Wouw(2015). Lecture notes and Excersises for this part on Multibody dynamics 2015, Multibody and Nonlinear dynamics. Eindhoven University of Technology. Pages 5, 8,

[7] Hassan K. Khalil (2014), Nonlinear systems Third Edition, Harlow : Pearson Education Limited. Pages

A Non-prehensile manipulation cart pendulum example

In Chapter 2.1, some background information for the motion planning of the 'Butterfly' robot is given. In it, an example of a simple cart pendulum is used as clarification. This appendix will give more information about the steps taken to get the equations of motion and about the phase portrait that is used to find a feasible trajectory.

A.1 Equations of motion

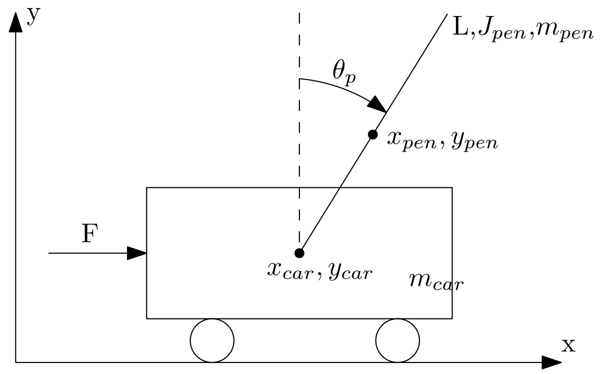

Figure 19: simple cart pendulum

The cart pendulum that is used can be seen on Figure 19. The cart with a mass m_{car} is driven by a force F in the horizontal direction. The pendulum has a length L, intertia J_{pen} and it center off mass can be considered at half way the length of the pendulum with a mass of m_{pen}. The Euler-Lagrange method can be applied to get the equations of motion, wherein the $x_{(.)}$ and $y_{(.)}$ denote the position of the center of mass of the component, θ_p the angle of the pendulum and 'g' the gravitational acceleration.

$$\mathcal{L} = K - V \tag{A.1a}$$

$$K_{car} = \frac{1}{2} m_{car} \dot{x}_{car}^2 \tag{A.1b}$$

$$K_{pen} = \frac{1}{2} m_{pen} (\dot{x}_{pen}^2 + \dot{y}_{pen}^2) + \frac{1}{2} J_{pen} \dot{\theta}_p^2 \tag{A.1c}$$

$$V_{pen} = m_{pen} g y_{pen} \tag{A.1d}$$

x_{pen} and y_{pen} can be rewritten in polar coordinates.

$$x_{pen} = x_{car} + \frac{L}{2} \sin(\theta_p) \tag{A.2a}$$

$$y_{pen} = \frac{L}{2} \cos(\theta_p) \tag{A.2b}$$

Which will then result in the following Lagrangian.

$$\mathcal{L} = \frac{1}{2} m_{car} \dot{x}_{car}^2 + \frac{1}{2} m_{pen} (\dot{x}_{pen}^2 + \dot{y}_{pen}^2) + \frac{1}{2} J_{pen} \dot{\theta}_p^2 + m_{pen} g y_{pen} \tag{A.3a}$$

$$= \frac{1}{2}(m_{pen} + m_{car}) \dot{x}_{car}^2 + \frac{1}{2} m_p L \dot{x}_{car} \dot{\theta}_p \cos(\theta_p) + \frac{1}{2}(\frac{m_{pen} L^2}{4} + J_{pen}) \dot{\theta}_p^2 - \frac{L}{2} m_{pen} g \cos(\theta_p) \tag{A.3b}$$

With a coordinate vector $\underline{q} = \begin{pmatrix} x_{car} & \theta_p \end{pmatrix}^T$, The Euler-Lagrange equation can be set up as the following.

$$\frac{d}{dt}(\frac{\delta \mathcal{L}}{\delta \dot{\underline{q}}}) - \frac{\delta \mathcal{L}}{\delta \underline{q}} = \begin{pmatrix} F \\ 0 \end{pmatrix} \tag{A.4}$$

$$\frac{\delta \mathcal{L}}{\delta x_{car}} = 0 \qquad \frac{\delta \mathcal{L}}{\delta \dot{x}_{car}} = (m_{pen} + m_{car})\dot{x}_{car} + \frac{1}{2} m_{pen} L \cos(\theta_p)\dot{\theta}_p \tag{A.5a}$$

$$\frac{\delta \mathcal{L}}{\delta \theta} = -\frac{1}{2} m_{pen} L \dot{x}_{car}\dot{\theta}_p \sin(\theta_p) + \frac{L}{2} m_{pen} g \sin(\theta_p) \quad \frac{\delta \mathcal{L}}{\delta \dot{\theta}_p} = \frac{1}{2} m_{pen} L \dot{x}_{car} \cos(\theta_p) + (\frac{m_{pen} L^2}{4} + J_{pen})\dot{\theta}_p \tag{A.5b}$$

Which will give the following equation of motion.

$$EOM \begin{cases} (m_{pen} + m_{car})\ddot{x}_{car} + \frac{1}{2} m_{pen} L \cos(\theta_p)\ddot{\theta}_p - \frac{1}{2} m_{pen} L \sin(\theta_p)\dot{\theta}_p^2 = F \\ \frac{1}{2} m_{pen} L \ddot{x}_{car} \cos(\theta_p) + (\frac{m_{pen} L^2}{4} + J_{pen})\ddot{\theta}_p - \frac{L}{2} m_{pen} g \sin(\theta_p) = 0 \end{cases} \tag{A.6}$$

Taking L = 2, $m_{car} = m_{pen} = 1$ and $J_{pen} = 0$ will then give the shape as is used in chapter 2.

$$EOM \begin{cases} 2\ddot{x}_{car} + \cos(\theta_p)\ddot{\theta}_p - \sin(\theta_p)\dot{\theta}_p^2 = F \\ \ddot{x}_{car} \cos(\theta_p) + \ddot{\theta}_p - g \sin(\theta_p) = 0 \end{cases} \tag{A.7}$$

A.2 Phase portrait

In order to get the phase portrait, the passive dynamic is analyzed. This passive dynamic is written as the bottom part of (A.7). It is desired to reduce the passive dynamic in a format in which only one degree of freedom is present. This is done by assuming that there exists a certain relation (virtual holonomic constraint) between θ_p and x_{car}, depending on the motion that is desired (note that not all motions may have such a relation). This can be seen on (A.8), in which θ_p is chosen as the generating variable.

$$x_{car} = \Phi(\theta_p) \tag{A.8a}$$

$$\dot{x}_{car} = \Phi'(\theta_p)\dot{\theta}_p \tag{A.8b}$$

$$\ddot{x}_{car} = \Phi''(\theta_p)\dot{\theta}_p^2 + \Phi'(\theta_p)\ddot{\theta}_p \tag{A.8c}$$

Combining the passive dynamic with the upper relations will give what is called an α, β, γ-equation.

$$(1 + \cos(\theta_p)\Phi'(\theta_p))\ddot{\theta}_p + \cos(\theta_p)\Phi''(\theta_p)\dot{\theta}_p^2 - g \sin(\theta_p) = 0 \tag{A.9a}$$

$$\alpha(\theta_p)\ddot{\theta}_p + \beta(\theta_p)\dot{\theta}_p^2 + \gamma(\theta_p) = 0 \tag{A.9b}$$

As this equation is time invariant (output does not depend on time), it is possible to analyze trajectories based on a phase portrait. On this phase portrait it is then possible to find a feasible trajectory (periodic trajectory). Rewriting (A.9) in state space will give the following.

$$\frac{d}{dt}\begin{pmatrix} \theta_p \\ \dot{\theta}_p \end{pmatrix} = \begin{pmatrix} \dot{\theta}_p \\ -\frac{\beta(\theta_p)}{\alpha(\theta_p)}\dot{\theta}_p^2 - \frac{\gamma(\theta_p)}{\alpha(\theta_p)} \end{pmatrix} \tag{A.10}$$

In this state space, we are interested in the points where $\gamma(\theta_p) = 0$ and $\dot{\theta}_p = 0$ (note that $\dot{\theta}_p, \ddot{\theta}_p$ are 0 then), these are called the critical points. Analyzing behaviour around these critial points will give trajectory information of the whole system (using vector fields). For linear systems the qualitative behaviours around an equilibrium point is checked by the eigenvalues. It is seen that in general cases for non-linear systems, the qualitative behaviours can also be checked via a linearization around this equilibrium point chapter 2 of [7]. In this case, the critical points would be present at angles θ_p=kπ for 'k' being any integer, as this would give $\gamma(\theta_p) = 0$. Analyzing the behaviour around these points for the linearized system of (A.10) would then give all feasible trajectories. Note that this only holds for one degree of freedom systems, which due to the virtual holonomic constraint is also thus applicable to our system.

It can be shown that taking x_{car} as the generating variable would not matter. A new relation $\theta_p = \Phi_2(x_{car})$ is then created, which would give

$$(\cos(\Phi_2(x_{car})) + \Phi_2'(x_{car}))\ddot{x}_{car} + \Phi_2''(x_{car})\dot{x}_{car}^2 - g\sin(\Phi_2(x_{car})) = 0 \tag{A.11}$$

The same analysis can then be done as before, in which the behaviour around the critical points $g\sin(\Phi_2(x_{car}))$=0 are analyzed. Notice how the actuator is allocated in such a way that it is possible for a relation to exist between x_{car} and θ_p. Pushing the cart would make the pendulum sway, so it is a non-prehensile manipulation. A non-prehensile manipulation is not present when this actuator is allocated to the pendulum, which would produce a torque (note that the system is still underactuated, this means that not all underactuated systems are non-prehensile). This would thus mean that the motion planning process as described in [1] (with phase portrait analysis) is not possible. To clarify this, a new a, β, γ-equation can be set up, but this time the passive dynamic would be the upper equation of (A.7) as the bottom equation now has a torque input.

$$(2\Phi'(\theta_p) + \cos(\theta_p))\ddot{\theta}_p + (2\Phi''(\theta_p) - \sin(\theta_p))\dot{\theta}_p^2 = 0 \tag{A.12}$$

Notice that all points for $\dot{\theta}_p = 0$ are equilibrium points due to the lack of $\gamma(\theta_p)$. There are thus no feasible trajectories as there are no periodic motions seen on the phase portrait.

B Ad hoc transformed 'Butterfly' model ('old' model)

In this appendix elaboration regarding the model as written in [3] will be given. This appendix will begin by giving the analytical steps in deriving the 'old' model. Note that in [3] an extra term $x_6 = \vec{\tau} \cdot \vec{\kappa}$ is kept, which causes extra terms like x_9 (not the same x_9 as in this appendix !) to appear when taking the derivative of x_6. This should not happen, as the two vectors $\vec{\tau}$ and $\vec{\kappa}$ are perpendicular to each other which causes the inner product to vanish. Attention has been paid to this small detail, which causes the model to differ a bit from the one in [3]. The variables that are used, are written in such a way that it (mostly) corresponds with the variables used in [3] and [1]. This appendix concludes with a guide in writing a 'Maple' script for deriving the equations of motion.

B.1 Analytical steps

B.1.1 Lagrangian

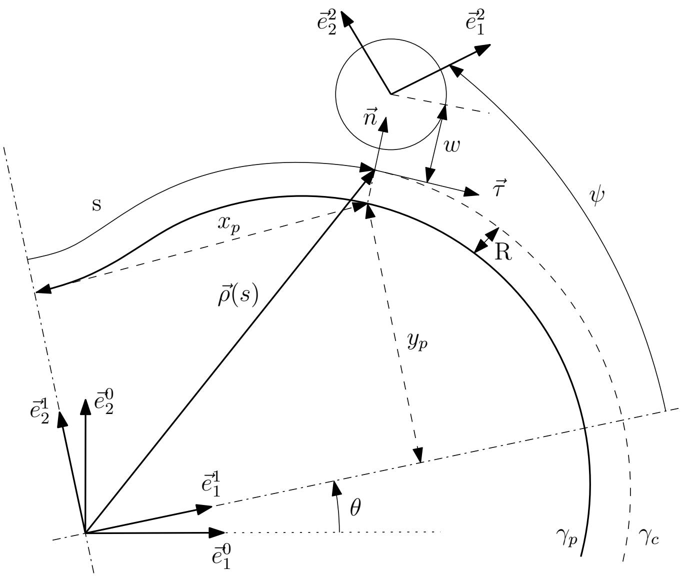

Figure 20: Ad hoc transformated coordinates of 'Butterfly' robot [3]

Figure 20 shows the model that is used in which the degrees of freedom are $\underline{q} = \begin{pmatrix} \theta & s & w & \psi \end{pmatrix}^T$. The Lagrangian can be seen in (B.1) in which J_f and J_c are the mass moment of inertia of the plates and ball respectively, $\vec{w}_f$ and $\vec{w}_c$ the rotational velocities of the plates and the ball, $\vec{v}_c$ the translational velocity of the ball, $\vec{R}_c$ the position of the center of the ball and $\vec{\rho}$ the position of where the constrained center of ball should be. More detailed expressions of the terms in (B.1) can be seen in (B.2).

$$\mathcal{L} = \frac{1}{2}(J_f(\vec{w}_f \cdot \vec{w}_f) + J_c(\vec{w}_c \cdot \vec{w}_c) + m_c(\vec{v}_c(s) \cdot \vec{v}_c(s))) - m_c\vec{g} \cdot (\Pi(\theta)\vec{R}_c(s)) \tag{B.1}$$

$$\vec{w}_f = \begin{pmatrix} 0 & 0 & \dot{\theta} \end{pmatrix} \underline{\vec{e}}^0 \tag{B.2a}$$

$$\vec{w}_c = \begin{pmatrix} 0 & 0 & \dot{\theta} + \dot{\psi} \end{pmatrix} \underline{\vec{e}}^0 \tag{B.2b}$$

$$\vec{R}_c(s,w) = \vec{\rho}(s) + w\vec{n}(s) \tag{B.2c}$$

$$\vec{v}_c(s,w) = \frac{d\vec{R}_c(s,w)}{dt} + \vec{w}_f \times \vec{R}_c(s,w) \tag{B.2d}$$

$$\Pi(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.2e}$$

$$\vec{g} = \begin{pmatrix} 0 & g & 0 \end{pmatrix} \underline{\vec{e}}^0 \tag{B.2f}$$

$$\tag{B.2g}$$

Furthermore for describing the equations of motion, it is necessary to mention the vectors $\vec{\tau}, \vec{n}, \hat{k}$, $\vec{\kappa}$ and $\vec{\xi}$. These vectors can be described in (B.3) and will be used extensively throughout the appendix. For more information about the vectors, the reader may check [3].

$$\hat{k} = \vec{e}_3^{\,0} = \vec{e}_3^{\,1} = \vec{e}_3^{\,2} \tag{B.3a}$$

$$\vec{\tau}(s) = \frac{d\vec{\rho}}{ds} \tag{B.3b}$$

$$\vec{n}(s) = \hat{k} \times \vec{\tau} \qquad \hat{k} = \vec{\tau} \times \vec{n} \qquad \vec{\tau} = \vec{n} \times \hat{k} \tag{B.3c}$$

$$\vec{\kappa}(s) = \frac{d\vec{\tau}}{ds} = \kappa\vec{n} \tag{B.3d}$$

$$\vec{\xi}(s) = \frac{d\vec{\kappa}}{ds} \tag{B.3e}$$

From this part on, relations are worked out which can not be found in [3]. The first step is to describe the Lagrangian ($\mathcal{L}$) (B.1) in the state $\underline{q}$. It is only needed to work out v_c, as the rotational velocity expressions can be found in (B.2). The v_c expression as in (B.2d) consists of two terms, the first term with a time derivative and the second term with a cross product. for the second term, usage is made of the 'distributive property of multiplication over addition for cross products' : $(\vec{a} \times (\vec{b} + \vec{c}) = \vec{a} \times \vec{b} + \vec{a} \times \vec{c})$. Note that scalar multiplications can be taken apart in a way as $a\vec{b} \times c\vec{d} = ac(\vec{b} \times \vec{d})$. The second term can then be worked out as in (B.4).

$$\vec{w}_f \times \vec{R}_c = \dot{\theta}\hat{k} \times (\vec{\rho} + w\vec{n}) = \dot{\theta}\hat{k} \times \vec{\rho} + w\dot{\theta}(\hat{k} \times \vec{n}) = \dot{\theta}(\hat{k} \times \vec{\rho}) - w\dot{\theta}\vec{\tau} \tag{B.4}$$

The first term can be expressed as (B.5), in which expressions of (B.3) are used. This equation is built up by using the cross product rule. Note that the time derivative of variables as a function of 's', is equal to the derivative w.r.t. 's' times $\dot{s}$. Furthermore it holds that $\frac{d\hat{k}}{dt}$=0.

$$\begin{aligned} \frac{d\vec{R}_c}{dt} &= \frac{d\vec{\rho}(s)}{dt} + \dot{w}\vec{n} + w(t)\frac{d\vec{n}}{dt} = \frac{d\vec{\rho}(s)}{ds}\dot{s} + \dot{w}\vec{n} + w(t)\frac{d}{dt}(\hat{k} \times \vec{\tau}) \\ &= \dot{s}\vec{\tau} + \dot{w}\vec{n} + w(t)(\frac{d\hat{k}}{dt} \times \tau + \hat{k} \times \frac{d\vec{\tau}}{dt}) = \dot{s}\vec{\tau} + \dot{w}\vec{n} + w(t)(\hat{k} \times \frac{d\vec{\tau}}{ds}\dot{s}) \\ &= \dot{s}\vec{\tau} + \dot{w}\vec{n} + w(t)\dot{s}(\hat{k} \times \vec{\kappa}) \end{aligned} \tag{B.5}$$

Using (B.2d), (B.4) and (B.5) results in the following expression for $\vec{v}_c$. Here the parameters 'a','b','c' and 'd' are used as substitutions to make the calculations easier.

$$\begin{aligned} \vec{v}_c &= \frac{d\vec{R}_c}{dt} + \vec{w}_f \times \vec{R}_c \\ &= (\dot{s} - w\dot{\theta})\vec{\tau} + \dot{w}\vec{n} + w\dot{s}(\hat{k} \times \vec{\kappa}) + \dot{\theta}(\hat{k} \times \vec{\rho}) \\ &= a\vec{\tau} + b\vec{n} + c(\hat{k} \times \vec{\kappa}) + d(\hat{k} \times \vec{\rho}) \end{aligned} \tag{B.6}$$

Using the simplified velocity term, the inner product of it can be made as (B.7). Note that the 'distributive property of multiplication over addition' also holds for the inner product.

$$\begin{aligned}\vec{v}_c \cdot \vec{v}_c =& a^2(\vec{\tau}\cdot\vec{\tau}) + ab(\vec{\tau}\cdot\vec{n}) + ac(\tau\cdot(\hat{k}\times\vec{\kappa})) + ad(\vec{\tau}\cdot(\hat{k}\times\vec{\rho})) + ab(\vec{n}\cdot\vec{\tau}) + b^2(\vec{n}\cdot\vec{n}) + bc(\vec{n}\cdot(\hat{k}\times\vec{\kappa}))\\ &+ bd(\vec{n}\cdot(\hat{k}\times\vec{\rho})) + ac((\hat{k}\times\vec{\kappa})\cdot\vec{\tau}) + bc((\hat{k}\times\vec{\kappa})\cdot\vec{n}) + c^2((\hat{k}\times\vec{\kappa})\cdot(\hat{k}\times\vec{\kappa})) + cd((\hat{k}\times\vec{\kappa})\cdot(\hat{k}\times\vec{\rho}))\\ &+ ad((\hat{k}\times\vec{\rho})\cdot\vec{\tau}) + bd((\hat{k}\times\vec{\rho})\cdot\vec{n}) + cd((\hat{k}\times\vec{\rho})\cdot(\hat{k}\times\vec{\kappa})) + d^2((\hat{k}\times\vec{\rho})\cdot(\hat{k}\times\vec{\rho}))\end{aligned} \tag{B.7}$$

By using the following properties (B.7) can be simplified to (B.8).

- Unit vectors $\vec{n}, \vec{\tau}$ and $\hat{k}$ will give the value 1 when an inner product with themselves is used, e.g. $\vec{n}\cdot\vec{n}=1$.
- Vectors $\vec{n}, \vec{\tau}$ and $\hat{k}$ are perpendicular to each other. The inner product with one to another will result in 0 value, e.g. $\vec{n}\cdot\vec{\tau}=0$.
- Scalar triple product property : $\vec{a}\cdot(\vec{b}\times\vec{c}) = \vec{b}\cdot(\vec{c}\times\vec{a}) = \vec{c}\cdot(\vec{a}\times\vec{b})$. An example would then be : $(\hat{k}\times\vec{\kappa})\cdot(\hat{k}\times\vec{\rho}) = \vec{\rho}\cdot((\hat{k}\times\vec{\kappa})\times\hat{k})$.
- Vector triple product property : $\vec{a}\times(\vec{b}\times\vec{c}) = \vec{b}(\vec{a}\cdot\vec{c}) - \vec{c}(\vec{a}\cdot\vec{b})$ and $(\vec{a}\times\vec{b})\times\vec{c} = -\vec{c}\times(\vec{a}\times\vec{b})$. An example would then be : $(\hat{k}\times\vec{\kappa})\times\hat{k} = -\hat{k}\times(\hat{k}\times\vec{\kappa}) = -\hat{k}(\hat{k}\cdot\vec{\kappa}) + \vec{\kappa}(\hat{k}\cdot\hat{k})$.
- $\vec{\kappa}$ is in the same direction as $\vec{n}$ as it can be expressed as $\kappa\vec{n}$. This will give e.g. : $\vec{n}\cdot(\hat{k}\times\vec{\kappa}) = \vec{n}\cdot(\hat{k}\times\kappa\vec{n}) = -\kappa(\vec{n}\cdot\vec{\tau})$.

$$\begin{aligned}\vec{v}_c\cdot\vec{v}_c =& a^2 - 2ac(\hat{k}\cdot(\vec{\tau}\times\vec{\kappa})) + 2ad(\hat{k}\cdot(\vec{\rho}\times\vec{\tau})) + b^2 + 2bd(\vec{\rho}\cdot\vec{\tau}) + c^2||\vec{\kappa}||^2\\ &+ 2cd(\vec{\rho}\cdot\vec{\kappa}) + d^2||\vec{\rho}||^2\end{aligned} \tag{B.8}$$

To facilitate the calculations, it has been chosen to substitute terms into one variable as has been done in [3]. This can be seen in (B.9).

$$x_1 = \hat{k}\cdot(\vec{\tau}\times\vec{\kappa}) \tag{B.9a}$$
$$x_2 = \hat{k}\cdot(\vec{\rho}\times\vec{\tau}) \tag{B.9b}$$
$$x_3 = \vec{\rho}\cdot\vec{\tau} \tag{B.9c}$$
$$x_4 = \vec{\rho}\cdot\vec{\kappa} \tag{B.9d}$$

The Lagrangian in its desired format can then be worked out by combining (B.1), (B.2), (B.8) and (B.9) which results in (B.10). Note that substitutions with (B.9) will happen more often throughout this appendix.

$$\begin{aligned}\mathcal{L} =& \frac{1}{2}((J_f + J_c)\dot{\theta}^2 + J_c(\dot{\psi}^2 + 2\dot{\theta}\dot{\psi}) + m_c(\vec{v}_c\cdot\vec{v}_c)) - m_c\vec{g}\cdot(\Pi(\theta)(\vec{\rho}+w\vec{n}))\\ =& \frac{m_c}{2}((\frac{J_f}{m_c} + \frac{J_c}{m_c} + w^2 + ||\vec{\rho}||^2 - 2wx_2)\dot{\theta}^2 + ((2x_1w^2 + (2x_4 - 2)w + 2x_2)\dot{s} + 2x_3\dot{w} + 2\frac{J_c}{m_c}\dot{\psi})\dot{\theta}\\ &+ (||\vec{\kappa}||^2w^2 - 2x_1w + 1)\dot{s}^2 + \dot{w}^2 + \frac{J_c}{m_c}\dot{\psi}^2 - 2\vec{g}\cdot(\Pi(\theta)(\vec{\rho}+w\vec{n})))\end{aligned} \tag{B.10}$$

B.1.2 Equations of motion for four degrees of freedom model

$$\frac{d}{dt}\frac{d\mathcal{L}}{d\underline{\dot{q}}} - \frac{d\mathcal{L}}{d\underline{q}} = \begin{pmatrix} u & 0 & 0 & 0 \end{pmatrix}^T \tag{B.11}$$

Degree of freedom θ

$$\frac{d\mathcal{L}}{d\dot{\theta}} = m_c((\frac{J_f}{m_c} + \frac{J_c}{m_c} + w^2 + ||\vec{\rho}||^2 - 2wx_2)\dot{\theta} + (x_1 w^2 + (x_4 - 1)w + x_2)\dot{s} + x_3\dot{w} + \frac{J_c}{m_c}\dot{\psi}) \tag{B.12}$$

The first row of (B.11) can be described with the degree of freedom θ. The first term can then be gotten by taking the time derivative of (B.12). It has to be taken into account that these substituted terms $x_\#$ will also have a time derivative. These time derivatives are given in (B.14) and the new terms in that equation can be expressed as (B.13).

$$x_5 = \hat{k} \cdot (\vec{\tau} \times \vec{\xi}) \tag{B.13a}$$
$$x_6 = \hat{k} \cdot (\vec{\rho} \times \vec{\kappa}) \tag{B.13b}$$
$$x_7 = \vec{\rho} \cdot \vec{\xi} \tag{B.13c}$$

$$\frac{dx_1}{dt} = \frac{d\hat{k}}{dt} \cdot (\vec{\tau} \times \vec{\kappa}) + \hat{k} \cdot \frac{d(\vec{\tau} \times \vec{\kappa})}{dt} = \hat{k} \cdot (\frac{d\vec{\tau}}{dt} \times \vec{\kappa} + \vec{\tau} \times \frac{d\vec{\kappa}}{dt}) = (\hat{k} \cdot (\vec{\tau} \times \vec{\xi}))\dot{s} = x_5\dot{s} \tag{B.14a}$$
$$\frac{dx_2}{dt} = \hat{k} \cdot \frac{d(\vec{\rho} \times \vec{\tau})}{dt} = (\hat{k} \cdot (\vec{\rho} \times \vec{\kappa}))\dot{s} = x_6\dot{s} \tag{B.14b}$$
$$\frac{dx_3}{dt} = \frac{d\vec{\rho}}{dt} \cdot \vec{\tau} + \vec{\rho} \cdot \frac{d\vec{\tau}}{dt} = \dot{s}(1 + (\vec{\rho} \cdot \vec{\kappa})) = \dot{s}(1 + x_4) \tag{B.14c}$$
$$\frac{dx_4}{dt} = \frac{d\vec{\rho}}{dt} \cdot \vec{\kappa} + \vec{\rho} \cdot \frac{d\vec{\kappa}}{dt} = \dot{s}(\vec{\rho} \cdot \vec{\xi}) = \dot{s}x_7 \tag{B.14d}$$
$$\frac{\mathrm{d}||\vec{\rho}||^2}{dt} = \frac{d(\vec{\rho} \cdot \vec{\rho})}{dt} = 2\dot{s}(\vec{\rho} \cdot \vec{\tau}) = 2\dot{s}x_3 \tag{B.14e}$$

Using (B.14), the time derivative can be calculated as (B.15).

$$\begin{aligned}\frac{d}{dt}\frac{d\mathcal{L}}{d\dot{\theta}} = m_c((&\frac{J_f}{m_c} + \frac{J_c}{m_c} + w^2 + \rho^2 - 2wx_2)\ddot{\theta} + (x_1 w^2 + (x_4 - 1)w + x_2)\ddot{s} + x_3\ddot{w} + \frac{J_c}{m_c}\ddot{\psi} \\ &+ (2w\dot{w} + 2\dot{s}x_3 - 2w\dot{s}x_6 - 2x_2\dot{w})\dot{\theta} + (2x_1 w\dot{w} + w^2 x_5\dot{s} + x_7 w\dot{s} + (x_4 - 1)\dot{w} + x_6\dot{s})\dot{s} \\ &+ (1 + x_4)\dot{s}\dot{w})\end{aligned} \tag{B.15}$$

The second term is given in (B.16).

$$\frac{d\mathcal{L}}{d\theta} = -m_c\vec{g} \cdot (\Pi'(\theta)(\vec{\rho} + w\vec{n})) \tag{B.16}$$

Using (B.15) and (B.16) gives the first row of (B.11).

$$\frac{d}{dt}\frac{d\mathcal{L}}{d\dot{\theta}} - \frac{d\mathcal{L}}{d\theta} = \underline{M}_1^T(q)\ddot{q} + \underline{C}_1^T(q, \dot{q})\dot{q} + G_1(q) = u \tag{B.17}$$

$$\underline{M}_1^T = m_c \begin{pmatrix} (\frac{J_f}{m_c} + \frac{J_c}{m_c} + w^2 + \rho^2 - 2wx_2) & (x_1 w^2 + (x_4 - 1)w + x_2) & x_3 & \frac{J_c}{m_c} \end{pmatrix} \tag{B.18}$$

$$C_{1,1} = m_c((w - x_2)\dot{w} + (x_3 - wx_6)\dot{s}) \tag{B.19a}$$
$$C_{1,2} = m_c((x_3 - wx_6)\dot{\theta} + (x_1 w + x_4)\dot{w} + (x_5 w^2 + x_7 w + x_6)\dot{s}) \tag{B.19b}$$
$$C_{1,3} = m_c((w - x_2)\dot{\theta} + (x_1 w + x_4)\dot{s}) \tag{B.19c}$$
$$C_{1,4} = 0 \tag{B.19d}$$

$$G_1 = m_c\vec{g} \cdot (\Pi'(\theta)(\vec{\rho} + w\vec{n})) \tag{B.20}$$

Degree of freedom 's'

$$\frac{\partial \mathcal{L}}{\partial \dot{s}} = m_c((x_1 w^2 + (x_4 - 1)w + x_2)\dot{\theta} + (||\vec{\kappa}||^2 w^2 - 2x_1 w + 1)\dot{s}) \tag{B.21}$$

$$x_8 = \vec{\kappa} \cdot \vec{\xi} \tag{B.22}$$

$$\frac{d||\vec{\kappa}||^2}{dt} = \frac{d(\vec{\kappa} \cdot \vec{\kappa})}{dt} = 2\dot{s}(\vec{\kappa} \cdot \vec{\xi}) = 2\dot{s}x_8 \tag{B.23}$$

The second row of (B.11) can be expressed by the degree of freedom 's'. Taking the time derivative of (B.21) will give the first term in (B.24).

$$\begin{aligned}\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{s}} =& m_c((x_1 w^2 + (x_4 - 1)w + x_2)\ddot{\theta} + (||\vec{\kappa}||^2 w^2 - 2x_1 w + 1)\ddot{s} \\ &+ (x_5 w^2 \dot{s} + 2x_1 w\dot{w} + x_7 w\dot{s} + (x_4 - 1)\dot{w} + x_6\dot{s})\dot{\theta} \\ &+ (2x_8 w^2 \dot{s} + 2||\vec{\kappa}||^2 w\dot{w} - 2x_5 w\dot{s} - 2x_1\dot{w})\dot{s})\end{aligned} \tag{B.24}$$

For the second term it is important to know that a lot of variables are dependent of 's'. Taking the derivative with respect to 's' is almost the same as taking the derivative with respect to time, only now there are no $\dot{s}$ terms (e.g. $\frac{dx_1}{dt} = \dot{s}x_3$ and $\frac{dx_1}{ds} = x_3$).

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial s} =& m_c((x_3 - wx_6)\dot{\theta}^2 + ((x_5 w^2 + x_7 w + x_6)\dot{s} + (1 + x_4)\dot{w})\dot{\theta} + (w^2 x_8 - x_5 w)\dot{s}^2 \\ &- \vec{g} \cdot (\Pi(\theta)(\vec{\tau} + w(\hat{k} \times \vec{\kappa}))))\end{aligned} \tag{B.25}$$

Using (B.24) and (B.25) will then give (B.26).

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{s}} - \frac{\partial \mathcal{L}}{\partial s} = \underline{M}_2^T(q)\ddot{q} + \underline{C}_2^T(q, \dot{q})\dot{q} + G_2(q) = 0 \tag{B.26}$$

$$\underline{M}_2^T = m_c \begin{pmatrix}(x_1 w^2 + (x_4 - 1)w + x_2) & (||\vec{\kappa}||^2 w^2 - 2x_1 w + 1) & 0 & 0\end{pmatrix} \tag{B.27}$$

$$C_{2,1} = m_c((x_1 w - 1)\dot{w} + (wx_6 - x_3)\dot{\theta}) \tag{B.28a}$$
$$C_{2,2} = m_c((x_8 w^2 - x_5 w)\dot{s} + (||\vec{\kappa}||^2 w - x_1)\dot{w}) \tag{B.28b}$$
$$C_{2,3} = m_c((x_1 w - 1)\dot{\theta} + (||\vec{\kappa}||^2 w - x_1)\dot{s}) \tag{B.28c}$$
$$C_{2,4} = 0 \tag{B.28d}$$

$$G_2 = m_c\vec{g} \cdot (\Pi(\theta)(\vec{\tau} + w(\hat{k} \times \vec{\kappa}))) \tag{B.29}$$

Degree of freedom 'w'

$$\frac{\partial \mathcal{L}}{\partial \dot{w}} = m_c(x_3\dot{\theta} + \dot{w}) \tag{B.30}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{w}} = m_c(x_3\ddot{\theta} + \ddot{w} + (1 + x_4)\dot{s}\dot{\theta}) \tag{B.31}$$

$$\frac{\partial \mathcal{L}}{\partial w} = m_c((w - x_2)\dot{\theta}^2 + (2x_1 w + x_4 - 1)\dot{s}\dot{\theta} + (||\vec{\kappa}||^2 w - x_1)\dot{s}^2 - \vec{g} \cdot (\Pi(\theta)\vec{n})) \tag{B.32}$$

Combining (B.31) and (B.32) will give the third row of (B.11) and can be seen in equation (B.33).

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{w}} - \frac{\partial \mathcal{L}}{\partial w} = \underline{M}_3^T(q)\ddot{q} + \underline{C}_3^T(q,\dot{q})\dot{q} + G_3(q) = 0 \tag{B.33}$$

$$\underline{M}_3^T = m_c \begin{pmatrix} x_3 & 0 & 1 & 0 \end{pmatrix} \tag{B.34}$$

$$C_{3,1} = m_c((1 - x_1 w)\dot{s} + (x_2 - w)\dot{\theta}) \tag{B.35a}$$
$$C_{3,2} = m_c((1 - x_1 w)\dot{\theta} + (x_1 - ||\vec{\kappa}||^2 w)\dot{s}) \tag{B.35b}$$
$$C_{3,3} = 0 \tag{B.35c}$$
$$C_{3,4} = 0 \tag{B.35d}$$

$$G_3 = m_c\vec{g} \cdot (\Pi(\theta)\vec{n}) \tag{B.36}$$

Degree of freedom ψ

$$\frac{\partial \mathcal{L}}{\partial \dot{\psi}} = J_c(\dot{\psi} + \dot{\theta}) \tag{B.37}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\psi}} = J_c(\ddot{\psi} + \ddot{\theta}) \tag{B.38}$$

$$\frac{\partial \mathcal{L}}{\partial \psi} = 0 \tag{B.39}$$

Combining (B.38) and (B.39) will give the last row of (B.11) and can be seen in (B.40).

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\psi}} - \frac{\partial \mathcal{L}}{\partial \psi} = \underline{M}_4^T(q)\ddot{q} + \underline{C}_4^T(q,\dot{q})\dot{q} + G_4(q) = 0 \tag{B.40}$$

$$\underline{M}_4^T = m_c \begin{pmatrix} \frac{J_c}{m_c} & 0 & 0 & \frac{J_c}{m_c} \end{pmatrix} \tag{B.41}$$

$$C_{4,1} = 0 \tag{B.42a}$$
$$C_{4,2} = 0 \tag{B.42b}$$
$$C_{4,3} = 0 \tag{B.42c}$$
$$C_{4,4} = 0 \tag{B.42d}$$

$$G_4 = 0 \tag{B.43}$$

Four degrees of freedom equations of motion

Combining (B.17), (B.26), (B.33) and (B.40) will give the equations of motion as described in (B.44) in which $\underline{h}$ are the virtual holonomic constraints 'point contact' and 'no-slip'.

$$\begin{aligned} \underline{M}(\underline{q})\ddot{\underline{q}} + \underline{C}(\underline{q},\dot{\underline{q}})\dot{\underline{q}} + \underline{G}(\underline{q}) &= \begin{pmatrix} u & 0 & 0 & 0 \end{pmatrix}^T + \underline{W}(\underline{q},t)\underline{\lambda} \\ \underline{W}^T(\underline{q},t) &= \frac{\partial \underline{h}(\underline{q},t)}{\partial \underline{q}} \end{aligned} \tag{B.44}$$

$$\underline{M}(\underline{q}) = \begin{pmatrix} \underline{M}_1^T \\ \underline{M}_2^T \\ \underline{M}_3^T \\ \underline{M}_4^T \end{pmatrix}, \quad \underline{C}(\underline{q},\dot{\underline{q}}) = \begin{pmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} \\ C_{2,1} & C_{2,2} & C_{2,3} & C_{2,4} \\ C_{3,1} & C_{3,2} & C_{3,3} & C_{3,4} \\ C_{4,1} & C_{4,2} & C_{4,3} & C_{4,4} \end{pmatrix}, \quad \underline{G}(\underline{q}) = \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} \tag{B.45}$$

Note that (B.44) can only be used when the holonomic and non-holonomic velocity constraints are linear in the generalized velocities (as stated in chapter 3 (page 55) [6]). Here $\underline{W}\underline{\lambda}$ is related to the constraint forces. $\underline{\lambda}$ can be gotten from (B.46), which can be found in chapter 5 (page 104) of [6]. Here the constraints $h_1 = w$ and $h_2 = s + R\psi$ are used.

$$\underline{\lambda} = (\underline{W}^T(\underline{q}_3,t)\underline{M}^{-1}(\underline{q}_3)\underline{W}(\underline{q}_3,t))^{-1}(\underline{W}^T(q_3,t)\underline{M}^{-1}(\underline{H}(\underline{q}_3,\dot{\underline{q}}_3) - \underline{S}(\underline{q}_3)\underline{\tau}) - \underline{\bar{w}}(\underline{q}_3,\dot{\underline{q}}_3,t) \tag{B.46a}$$

$$\underline{H}(\underline{q}_3,\dot{\underline{q}}_3) = \underline{C}(\underline{q}_3,\dot{\underline{q}}_3)\dot{\underline{q}}_3 + \underline{G}(\underline{q}_3) \tag{B.46b}$$

$$\underline{S}(\underline{q}_3)\underline{\tau} = \begin{pmatrix} u & 0 & 0 & 0 \end{pmatrix}^T \tag{B.46c}$$

$$\underline{\tilde{w}}(\underline{q},t) = \frac{\partial \underline{h}(\underline{q},t)}{\partial t}(= 0 \textit{ in our case}) \tag{B.47a}$$

$$\underline{\bar{w}}(\underline{q},\dot{\underline{q}},t) = (\frac{\partial \underline{W}^T(\underline{q},t)}{\partial t} + \frac{\partial \underline{W}^T(\underline{q},t)\dot{\underline{q}}}{\partial \underline{q}} + \frac{\partial \underline{\tilde{w}}(\underline{q},t)}{\partial \underline{q}})\dot{\underline{q}} + \frac{\partial \underline{\tilde{w}}(\underline{q},t)}{\partial t}(= 0 \textit{ in our case}) \tag{B.47b}$$

B.1.3 Two degrees of freedom equations of motion

To get the two degrees of freedom model as described in [1] with a reduced state of $\underline{q}_r = \begin{pmatrix} \theta & \varphi \end{pmatrix}^T$, it is first reduced to the reduced state $\underline{q}_2 = \begin{pmatrix} \theta & s \end{pmatrix}^T$. This is done with the two constraints 'point contact' and 'no-slip', which give the following constrained expressions for 'w' and ψ.

$$w = 0 \rightarrow \dot{w} = 0 \tag{B.48a}$$

$$\psi = -\frac{s}{R} \rightarrow \dot{\psi} = -\frac{\dot{s}}{R} \tag{B.48b}$$

Inserting these two constraint expressions into (B.10), will give the following Lagrangian.

$$\mathcal{L} = \frac{m_c}{2}((\frac{J_f}{m_c} + \frac{J_c}{m_c} + ||\vec{\rho}||^2)\dot{\theta}^2 + (2x_2\dot{s} - 2\frac{J_c}{m_c}\frac{\dot{s}}{R})\dot{\theta} + \dot{s}^2 + \frac{J_c}{m_c}\frac{\dot{s}}{R}^2 - 2\vec{g}\cdot(\Pi(\theta)\vec{\rho})) \tag{B.49}$$

Assuming that 's' can be expressed in the variable φ, this will result in the reduced state $\underline{q}_r = \begin{pmatrix} \theta & s(\varphi) \end{pmatrix}^T$. Note that variables that were dependent on 's', will now become dependent on φ. Taking time derivatives of such variables, will cause the usual $\dot{s}$ to be replaced with $s'\dot{\varphi}$ in which $s' = \frac{ds}{d\varphi}$ (and $s'' = \frac{d^2s}{d\varphi^2}$). This will then change the Lagrangian to the following form.

$$\mathcal{L} = \frac{m_c}{2}((\frac{J_f}{m_c} + \frac{J_c}{m_c} + ||\vec{\rho}||^2)\dot{\theta}^2 + (2x_2s'\dot{\varphi} - 2\frac{J_c}{m_c}\frac{s'\dot{\varphi}}{R})\dot{\theta} + (s'\dot{\varphi})^2 + \frac{J_c}{m_c}\frac{s'\dot{\varphi}}{R}^2 - 2\vec{g}\cdot(\Pi(\theta)\vec{\rho})) \tag{B.50}$$

Degree of freedom θ

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = m_c((\frac{J_f}{m_c} + \frac{J_c}{m_c} + ||\vec{\rho}||^2)\dot{\theta} + x_2 s'\dot{\varphi} - \frac{J_c}{m_c R} s'\dot{\varphi}) \tag{B.51}$$

Note that $\vec{\rho}(s(\varphi))$ is used, so $\frac{d\vec{\rho}}{dt} = \dot{s}x_3 = s'x_3\dot{\varphi}$. Also take into account that the derivative of three products is : $(abc)' = a'bc + ab'c + abc'$.

$$\begin{aligned}\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} =& m_c((\frac{J_f}{m_c} + \frac{J_c}{m_c} + ||\vec{\rho}||^2)\ddot{\theta} + ((x_2 - \frac{J_c}{m_c R})s')\ddot{\varphi} \\ &+ 2s'x_3\dot{\varphi}\dot{\theta} + x_6(s')^2\dot{\varphi}^2 + x_2 s''\dot{\varphi}^2 - \frac{J_c}{m_c R}s''\dot{\varphi}^2)\end{aligned} \tag{B.52}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m_c \vec{g} \cdot (\Pi'(\theta)\vec{\rho}) \tag{B.53}$$

Combining (B.52) and (B.53) will give (B.54).

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} - \frac{\partial \mathcal{L}}{\partial \theta} = \underline{M}^T_{r,1}(\underline{q}_r)\ddot{\underline{q}}_r + \underline{C}^T_{r;1}(\underline{q}_r, \dot{\underline{q}}_r)\dot{\underline{q}}_r + G_{r,1}(\underline{q}_r) = u \tag{B.54}$$

$$\underline{M}^T_{r,1} = m_c \left((\tfrac{J_f}{m_c} + \tfrac{J_c}{m_c} + ||\vec{\rho}||^2) \quad ((x_2 - \tfrac{J_c}{m_c R})s') \right) \tag{B.55}$$

$$C_{r;1,1} = m_c x_3 s' \dot{\varphi} \tag{B.56a}$$

$$C_{r;1,2} = m_c(x_3 s'\dot{\theta} + (x_6(s')^2 + (x_2 - \frac{J_c}{m_c R})s'')\dot{\varphi}) \tag{B.56b}$$

$$G_{r,1} = m_c \vec{g} \cdot (\Pi'(\theta)\vec{\rho}) \tag{B.57}$$

Degree of freedom φ

$$\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} = m_c((x_2 - \frac{J_c}{m_c R})s'\dot{\theta} + (1 + \frac{J_c}{m_c R^2})(s')^2\dot{\varphi}) \tag{B.58}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} =& m_c((x_2 - \frac{J_c}{m_c R})s'\ddot{\theta} + (1 + \frac{J_c}{m_c R^2})(s')^2\ddot{\varphi} \\ &+ x_6(s')^2\dot{\varphi}\dot{\theta} + (x_2 - \frac{J_c}{m_c R})s''\dot{\varphi}\dot{\theta} + 2(1 + \frac{J_c}{m_c R^2})s's''\dot{\varphi}^2)\end{aligned} \tag{B.59}$$

$$\frac{\partial \mathcal{L}}{\partial \varphi} = m_c(x_3 s'\dot{\theta}^2 + x_6(s')^2\dot{\varphi}\dot{\theta} + (x_2 - \frac{J_c}{m_c R})s''\dot{\varphi}\dot{\theta} + (1 + \frac{J_c}{m_c R^2})s's''\dot{\varphi}^2 - \vec{g} \cdot (\Pi(\theta)s'\vec{\tau})) \tag{B.60}$$

Combining (B.59) and (B.60) will give (B.61).

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} - \frac{\partial \mathcal{L}}{\partial \varphi} = \underline{M}^T_{r,2}(\underline{q}_r)\ddot{\underline{q}}_r + \underline{C}^T_{r;2}(\underline{q}_r, \dot{\underline{q}}_r)\dot{\underline{q}}_r + \underline{G}_{r,2}(\underline{q}_r) = 0 \tag{B.61}$$

$$\underline{M}^T_{r,2} = m_c \left(((x_2 - \tfrac{J_c}{m_c R})s') \quad ((1 + \tfrac{J_c}{m_c R^2})(s')^2) \right) \tag{B.62}$$

$$C_{r;2,1} = -m_c x_3 s' \dot{\theta} \tag{B.63a}$$

$$C_{r;2,2} = m_c((1 + \frac{J_c}{m_c R^2})s's''\dot{\varphi}) \tag{B.63b}$$

$$G_{r,2} = m_c \vec{g} \cdot (\Pi(\theta)\vec{\tau}s') \tag{B.64}$$

Equations of motion for the two degrees of freedom model

Combining (B.54) and (B.61) will result in (B.65). Filling all the expressions in (B.66), will show that this equations of motion is of similar shape as the one stated in [3] or [1].

$$\underline{M}_r(\underline{q}_r)\underline{\ddot{q}}_r + \underline{C}_r(\underline{q}_r, \underline{\dot{q}}_r)\underline{\dot{q}}_r + \underline{G}_r(\underline{q}_r) = \begin{pmatrix} u & 0 \end{pmatrix}^T \tag{B.65}$$

in which

$$\underline{M}_r(\underline{q}_r) = \begin{pmatrix} \underline{M}_{r,1}^T \\ \underline{M}_{r,2}^T \end{pmatrix}, \quad \underline{C}_r(\underline{q}_r, \underline{\dot{q}}_r) = \begin{pmatrix} C_{r;1,1} & C_{r;1,2} \\ C_{r;2,1} & C_{r;2,2} \end{pmatrix}, \quad \underline{G}_r(\underline{q}_r) = \begin{pmatrix} G_{r,1} \\ G_{r,2} \end{pmatrix} \tag{B.66}$$

B.2 Guide 'Maple' script for four degrees of freedom equations of motion

Instead of just giving the script, this subsection will mainly discuss the important parts of the script in such a way that it can be applied to all other four degrees of freedom models, such as the ones in chapter 3. An explanation will be given for the actions taken. This will be done in chronological order in such a way that just 'glueing' these steps together will give a fine working script (notice that only a part of the M,C and G matrix is given here as the other parts look similar). Note that this may not be the most optimal script as my skills in Maple are not that high.

Settings

```
restart:
with(Student[VectorCalculus]): with(VectorCalculus): with(plots): with(LinearAlgebra):
    with(ArrayTools): with(CodeGeneration): with(Physics): with(DifferentialGeometry):
    with(JetCalculus): BasisFormat(false): Setup(mathematicalnotation = true)
```

The first step is to reset everything with 'restart', just like you would use 'clc,clear,close all' in 'Matlab'. After that u can add packages (like in 'Latex') which then provides u with more tools. For more information about these packages, the reader should go to the 'Maple' site.

Degrees of freedom definitions

```
(th, sr, ps, wr) := (theta(t), s(t), psi(t), w(t)):
(dth, ds, dps, dw) := (diff(th, t), diff(sr, t), diff(ps, t), diff(wr, t)):
(ddth, dds, ddps, ddw) := (diff(dth, t), diff(ds, t), diff(dps, t), diff(dw, t)):
```

- ':=' $\rightarrow$ is used to assign a value to a certain variable. So you can recall the variable later and the same value will be given.
- 'theta(t)' $\rightarrow$ by adding '(t)' u make the variable 't' dependent. 'Maple' can then use it for other operations like differentiating with respect to 't' for which the answer will be for example $\frac{d}{dt}\theta(t)$.
- (x1, x2, x3 ,x4) := (x6, x7, x8, x9) $\rightarrow$ is the same as x1:=x6, x2:=x8 etc.
- Row ending with ':' $\rightarrow$ This way the answer will not be shown, just like in 'Matlab' with ';'.

Vector definitions (position, velocity etc.) and Lagrangian

```
wc := <0, 0, dth + dps> :     # Rotational velocity of ball w.r.t. frame 0
wf := <0,0,dth> :             # Rotational velocity of butterfly plate w.r.t. frame 0
rh := <rho[1](sr), rho[2](sr), 0>:   # position vector rho(s) (Constrained center of
                                      ball w.r.t. frame 1)

# tau ; tangent vector to curve s
drh := diff(rh, sr):
(D(rho[1])(sr), (D(rho[2])(sr), (D(rho[3])(sr)) := (tau[1](sr), tau[2](sr), tau[3](sr)):
```

```
ta := drh;
# Don't know why, but have to define it in this way to make sure that the definitions
    defined above (e.g. D(rho[1])(sr) := tau[1](sr) ) will also appear like that in 'ta'

# Kappa ; differential of tau w.r.t. s
dta := diff(ta, sr):
(D(tau[1])(sr), (D(tau[2])(sr), (D(tau[3])(sr)) := (kappa[1](sr), kappa[2](sr), kappa
    [3](sr)):
kap := dta:

# xi; differential of kappa w.r.t. s
dka := diff(kap, sr);
(D(kappa[1])(sr), (D(kappa[2])(sr), (D(kappa[3])(sr) := zeta[1](sr), zeta[2](sr), zeta
    [3](sr);
xi := dka;
# Don't define de vector components as the same name as vector !! e.g. xi = <xi[1],xi
    [2],0>, will bring trouble (don't know what anymore) , but that's why I defined zeta
    [#].


k := <0,0,1>:  # hat{k}
N := CrossProduct(k, ta): # Normal vector
r := rh + N*wr:           # Position vector of center of ball
vb := simplify( diff(r, t) + CrossProduct(wf,r) ): Velocity vector of center of ball
g_vec := <0,g,0> :
P := Matrix(3, 3, [[cos(th), -sin(th), 0], [sin(th), cos(th), 0], [0, 0, 1]]):
P_acc := diff(P, th):

# Define Lagrangian
T[frame] := (1/2)*J[f]*(wf.wf):      # Note that (x.x)  is the same as DotProduct(x,x,
                                     conjugate=false); so just the dotproduct.
T[ball_rotation] := (1/2)*J[c]*(wc.wc):
T[ball_velocity] := (1/2)*m[c]*(vb.vb):
Kin_simple := {ta.kap = 0}:       # Simplify siderelation
Kin := simplify(T[frame]+T[ball_rotation]+T[ball_velocity], Kin_simp);:
Pot := simplify(m[c]*(g_vec.(P.r)), Kin_simp):
Lag := Kin-Pot:
```

- # → with # you can make comments, just like in 'Matlab' with '%'.
- Kin_simple → in 'Maple' a command 'simplify' exists which simplifies the expression. We can also add side-relations into the 'simplify' command as 'simplify(expression,siderelation)'. With side-relations we can add certain properties to the expression which 'Maple' could not have known beforehand, for example that a+b=e : 'simplify(a+b+c,{a+b=e})' = e+c. Note that you should not forget to use brackets when incorporating siderelations and also that the command 'simplify' can not handle siderelations with square root terms. In our case we wanted to add the side relation that the inner product of tau and kappa is equal to 0.

M Matrix

```
# First row : Degree of freedom 'theta'

dKddq1 := simplify(diff(Lag, dth), {ta.kap=0}):# dL/(d dot(theta) )    , dot(theta) =
                                                d/dt theta
dtdKddq1 := diff(dKddq1, t);      # d/dt dL/(d dot(theta))

# M matrix first row components
M11 := simplify(diff(dtdKddq1, ddth));
M12 := simplify(diff(dtdKddq1, dds));
M13 := simplify(diff(dtdKddq1, ddw));
M14 := diff(dtdKddq1, ddps);
# M(firstrow)ddot{q} = M11 ddot(theta) + M12 ddot(s) + M13 ddot(w) + M14 ddot(psi)
```

The M matrix is connected to the acceleration term in the equations of motion as $M(q)\ddot{q}$. An acceleration term can only occur in the equations of motion via the term $\frac{d}{dt}\frac{d\mathcal{L}}{d\dot{q}}$. As the acceleration is only of the order one, taking the derivative of $\frac{d}{dt}\frac{d\mathcal{L}}{d\dot{q}}$ with respect to the acceleration term will give the M-matrix components.

C Matrix

```
# Substitution relations
coeff_subs := {diff(psi(t), t) = dum_4, diff(s(t), t) = dum_2, diff(theta(t), t) = dum_1
    , diff(w(t), t) = dum_3, diff(psi(t), t, t) = dum_8, diff(s(t), t, t) = dum_6, diff(
    theta(t), t, t) = dum_5, diff(w(t), t, t) = dum_7}:
Back_subs := {dum_1 = diff(theta(t), t), dum_2 = diff(s(t), t), dum_3 = diff(w(t), t),
    dum_4 = diff(psi(t), t)}:

#  C Matrix first row
dKdq1 := simplify(diff(Lag, th), {ta.kap = 0}):    # dL/dtheta
EOM1 := dtdKddq1-dKdq1:                          # Equation of motion first row (theta DOF)

dummy_EOM1 := subs(coeff_subs, EOM1);
C11 := simplify(subs(Back_subs, (dummy_EOM1-coeff(dummy_EOM1, dum_1, 0))/dum_1));

dummy_EOM12 := subs(coeff_subs, EOM1-C11*(diff(th, t)));
C12 := simplify(subs(Back_subs, (dummy_EOM12-coeff(dummy_EOM12, dum_2, 0))/dum_2));

dummy_EOM13 := subs(coeff_subs, EOM1-C11*(diff(th, t))-C12*(diff(sr, t)));
C13 := simplify(subs(Back_subs, (dummy_EOM13-coeff(dummy_EOM13, dum_3, 0))/dum_3));

dummy_EOM14 := subs(coeff_subs, EOM1-C11*(diff(th, t))-C12*(diff(sr, t))-C13*(diff(wr, t
    )));
C14 := simplify(subs(Back_subs, (dummy_EOM14-coeff(dummy_EOM14, dum_4, 0))/dum_4));
```

To get the C-matrix components, the following idea was applied. First only one degree of freedom is focused on. In this case the degree of freedom θ which gives the following equation that we call 'EOM1'.

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} - \frac{\partial \mathcal{L}}{\partial \theta} = EOM1 \tag{B.67}$$

Substitute all velocity and acceleration terms of this 'EOM1' with a 'dummy' variable called 'dum_#' which results in the new expression 'dummy_EOM1'. In Maple this can be done with the 'subs(siderelation, expression)' command. We define it in this 'dummy' variable to be able to use the command 'coeff', which can not be used with differentiated variables. An example of such a substitution is seen below

$$\dot{\theta} + \dot{s} + s \rightarrow dum\_1 + dum\_2 + s = dummy\_EOMX \tag{B.68}$$

If we want $C_{1,1}\dot{\theta}$, then we need to find all the terms with variable 'dum_1'($=\dot{\theta}$) in it. This we can get by eliminating all terms which do not have 'dum_1' in it. To get all the terms that do not have 'dum_1' in it, we use the 'coeff' command : coeff(dummy_EOM1,dum_1,0). We are then able to get $C_{1,1}$ by the following expression.

$$C_{1,1}\dot{\theta} = C_{1,1}dum\_1 = \frac{dummy\_EOMX - coeff(dummy\_EOMX, dum\_1, 0)}{dum\_1}dum\_1 \tag{B.69}$$

For $C_{1,2}\dot{s}$ we use a different equation 'dummy_EOM12'. To prevent terms like '$2\dot{s}\dot{\theta}$' to appear twice in the total C_1 expression, it has been decided to eliminate all previous $C_{1,\#}$ components from the current 'dummy' equation. To clarify this problem, assume that EOM1 $= 2\dot{s}\dot{\theta}$. When not eliminating previous C terms, we would get $C_{1,1} = 2\dot{s}$ and $C_{1,2} = 2\dot{\theta}$. The total $C_1\dot{q}$ = EOM1 $= 4\dot{s}\dot{\theta}$ is not equal to the defined $2\dot{s}\dot{\theta}$. This idea was made to deal with problems in which higher order $\dot{s}$ terms for example would appear.

G Matrix + siderelation example

```
G1 := simplify(dtdKddq1-dKdq1-Multiply(`<|>`(M11, M12, M13, M14), `<,>`(ddth, dds, ddw,
    ddps))-Multiply(`<|>`(C11, C12, C13, C14), `<,>`(dth, ds, dw, dps)))
M11_simp := {rho[1](sr)^2+rho[2](sr)^2 = rho(sr)^2, tau[1](sr)^2+tau[2](sr)^2 = 1, k .
    CrossProduct(rh, ta) = x2}
```

The G matrix is the term left after eliminating the M and C matrix from the equation of motion. The siderelation M11_simp has equations in it which would simplify the M11 expression. With these siderelation equations it is possible to replace certain terms with the variables x_1, x_2, x_3, etc. , as has been done in the analytical steps in appendix B.1. A remark has to be made that it is better to do apply these siderelations at the end of the script, thus after defining the whole M, C and G matrix. This is due to the problem that the the G matrix could become a mess when substracting unsimplified terms 'dtdKddq1' with simplified terms like 'M11'.

C 'New' Model 'Butterfly' robot

In this appendix, the equations of motion for the cartesian coordinates is given with coordinate vector $\underline{q} = \begin{pmatrix} \theta & x & y & \psi \end{pmatrix}^T$

C.1 Equations of motion for the cartesian four degrees of freedom model

The Lagrangian was given earlier in (3.7).

$$\begin{aligned}\mathcal{L} &= K_c + K_f - V_c \\ &= \frac{m_c}{2}((x^2 + y^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c})\dot{\theta}^2 + (2\frac{J_c}{m_c}\dot{\psi} - 2(y\dot{x} - x\dot{y}))\dot{\theta} + \frac{J_c}{m_c}\dot{\psi}^2 + \dot{x}^2 + \dot{y}^2 - 2m_c\vec{g}\cdot(\underline{R}_c^{1^T}\underline{A}^{10}))\end{aligned} \tag{C.1}$$

Degree of freedom θ

$$\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = m_c((x^2 + y^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c})\dot{\theta} + \frac{J_c}{m_c}\dot{\psi} - y\dot{x} + x\dot{y}) \tag{C.2}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = m_c((x^2 + y^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c})\ddot{\theta} + \frac{J_c}{m_c}\ddot{\psi} - y\ddot{x} + x\ddot{y} + 2(x\dot{x} + y\dot{y})\dot{\theta}) \tag{C.3}$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = -m_c\vec{g}\cdot(\underline{R}_c^{1^T}\frac{d\underline{A}^{10}}{d\theta}) = m_c g(x\cos(\theta) - y\sin(\theta)) \tag{C.4}$$

Degree of freedom x

$$\frac{\partial\mathcal{L}}{\partial\dot{x}} = m_c(-y\dot{\theta} + \dot{x}) \tag{C.5}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{x}} = m_c(-y\ddot{\theta} + \ddot{x} - \dot{y}\dot{\theta}) \tag{C.6}$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = m_c(x\dot{\theta}^2 + \dot{y}\dot{\theta} - \vec{g}\cdot(\frac{d\underline{R}_c^{1^T}}{dx}\underline{A}^{10})) \tag{C.7}$$

Degree of freedom y

$$\frac{\partial\mathcal{L}}{\partial\dot{y}} = m_c(x\dot{\theta} + \dot{y}) \tag{C.8}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{y}} = m_c(x\ddot{\theta} + \ddot{y} + \dot{x}\dot{\theta}) \tag{C.9}$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = m_c(y\dot{\theta}^2 - \dot{x}\dot{\theta} - \vec{g}\cdot(\frac{d\underline{R}_c^{1^T}}{dy}\underline{A}^{10})) \tag{C.10}$$

Degree of freedom ψ

$$\frac{\partial\mathcal{L}}{\partial\dot{\psi}} = J_c(\dot{\theta} + \dot{\psi}) \tag{C.11}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{y}} = J_c(\ddot{\theta} + \ddot{\psi}) \tag{C.12}$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = 0 \tag{C.13}$$

Unconstrainted equations of motion

$$\underline{M}(\underline{q})\ddot{\underline{q}} + \underline{C}(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + \underline{G}(\underline{q}) = \begin{pmatrix} u & 0 & 0 & 0 \end{pmatrix}^T \tag{C.14}$$

$$\underline{M}(\underline{q}) = m_c \begin{pmatrix} (x^2 + y^2 + \frac{J_c}{m_c} + \frac{J_f}{m_c}) & -y & x & \frac{J_c}{m_c} \\ -y & 1 & 0 & 0 \\ x & 0 & 1 & 0 \\ \frac{J_c}{m_c} & 0 & 0 & \frac{J_c}{m_c} \end{pmatrix} \tag{C.15}$$

$$\underline{C}(\underline{q}, \dot{\underline{q}}) = m_c \begin{pmatrix} (x\dot{x} + y\dot{y}) & (x\dot{\theta}) & (y\dot{\theta}) & 0 \\ -(x\dot{\theta} - \dot{y}) & 0 & \dot{\theta} & 0 \\ (-y\dot{\theta} + \dot{x}) & \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{C.16}$$

$$\underline{G}(\underline{q}) = \begin{pmatrix} m_c g(x\cos(\theta) - y\sin(\theta)) \\ m_c g \sin(\theta) \\ m_c g \cos(\theta) \\ 0 \end{pmatrix} \tag{C.17}$$

D Expressing angle of the center of the ball in angle of point contact

D.1 Total equation

$$\begin{cases} \angle ABG = \pi - (\phi - (1 - \frac{\sin(2\phi)}{|\sin(2\phi)|})(\frac{-\pi+2\phi}{2}) - (1 - \frac{\sin(\phi)}{|\sin(\phi)|})\frac{\pi}{2}) - ((1 + \frac{\sin(\phi_p)}{|\sin(\phi_p)|}\frac{v_{p,y}}{|v_{p,y}|})\frac{\pi}{2} - \frac{\sin(\phi_p)}{|\sin(\phi_p)|}\frac{v_{p,y}}{|v_{p,y}|}\beta) \\ L_4 = \delta(\phi)\cos(\angle ABG) \\ L_3 = \delta(\phi)\sin(\angle ABG) + R \\ \varphi = \pi - \angle AEB + (1 - \frac{\sin(2\phi)}{|\sin(2\phi)|})(\frac{-\pi+2\angle AEB}{2}) - \frac{\sin(2\phi)}{|\sin(2\phi)|}\arctan\frac{L_3(\phi)}{L_4(\phi)} + (1 - \frac{\sin(\phi)}{|\sin(\phi)|})\frac{\pi}{2} \end{cases} \tag{D.1}$$

This is the equation as has been given before in chapter 4.1. In this appendix, an explanation is given about the derivation of this equation. First it is started with explaining $\angle ABG$ (look at the figures in chapter 4.1 to see what this angle means).

$$\angle ABG = \pi - \angle BAE - \angle AEB \tag{D.2}$$

The following properties hold for ∠BAE.

$$\begin{aligned} 0 > \phi > 0.5\pi &: \angle BAE = \phi \\ 0.5\pi > \phi > \pi &: \angle BAE = \pi - \phi \\ \pi > \phi > 1.5\pi &: \angle BAE = \phi - \pi \\ 1.5\pi > \phi > 2\pi &: \angle BAE = 2\pi - \phi (= -\phi) \end{aligned} \tag{D.3}$$

As ∠BAE will be inserted in a 'cos' or 'sin' term, it can be noted that the expression '2π-ϕ' can be seen as as '$-\phi$'. This would then give the following expression for ∠ BAE.

$$\angle BAE = \phi - (1 - \frac{\sin(2\phi)}{|\sin(2\phi)|})(\frac{-\pi + 2\phi}{2}) - (1 - \frac{\sin(\phi)}{|\sin(\phi)|})\frac{\pi}{2} \tag{D.4}$$

Here the second term makes sure that there is a differentiation between the upper and lower part of the 'Butterfly' robot. The third term makes sure that the symmetry properties between the right and left part of the 'Butterfly' robot are obtained, as 'sin(ϕ)' only gives a negative value when considering the left part of the 'Butterfly' robot.

For ∠AEB the following properties hold

$$\begin{aligned} \sim 1.79 > \phi >\sim 0.22\pi \vee \sim 0.79\pi > \phi >\sim 1.22\pi &: \angle AEB = \pi - \beta \\ \sim 0.22\pi > \phi >\sim 0.79\pi \vee \sim 1.22\pi > \phi >\sim 1.79\pi &: \angle AEB = \beta \end{aligned} \tag{D.5}$$

$$\angle AEB = (1 + \frac{\sin(\phi_p)}{|\sin(\phi_p)|}\frac{v_{p,y}}{|v_{p,y}|})\frac{\pi}{2} - \frac{\sin(\phi_p)}{|\sin(\phi_p)|}\frac{v_{p,y}}{|v_{p,y}|}\beta \tag{D.6}$$

Notice that the y-direction of the velocity plays an important role here. This y-directional velocity will be used to differentiate between section 1 and section 2 (as has been explained in chapter 4.1 with $\vec{\tau}$). For the right part of the 'Butterfly' robot, the '$\pi - \beta$' term as in (D.5) always occurs when the y-directional velocity is positive. For the left part this $\pi - \beta$ term occurs when the y-directional velocity is negative. To differentiate the properties between the right and left part, the 'sin(ϕ' term is added.

For the angle φ, the following holds for the right part of the 'Butterfly' robot (note that for the left part this angle is equal, but with an addition of π).

$$\begin{aligned} 0 > \phi >\approx 0.22\pi &: \varphi = \beta - \arctan\frac{L_3}{L_4} \\ \approx 0.22\pi > \phi > 0.5\pi &: \varphi = \pi - \beta - \arctan\frac{L_3}{L_4} \\ 0.5\pi > \phi >\approx 0.79\pi &: \varphi = \beta + \arctan\frac{L_3}{L_4} \\ \approx 0.79\pi > \phi > \pi &: \varphi = \pi - \beta + \arctan(\frac{L_3}{L_2}) \end{aligned} \tag{D.7}$$

Expressing the term 'β' or '$\pi - \beta'$ with $\angle$AEB defined in (D.5) and taking into account all the '+' and '-' signs for the corresponding angle ϕ, will give the summarized equation of (D.7) expressed in the last row of equation (D.1).

D.2 'Maple' script geometric approach (4.20) 'Butterfly' robot

Note that φ=phi[c] and ϕ=phi[p].

```
restart;
with(Student[VectorCalculus]); with(VectorCalculus); with(plots); with(LinearAlgebra);
    with(ArrayTools); with(CodeGeneration); with(Physics); with(DifferentialGeometry);
    with(JetCalculus); BasisFormat(false); Setup(mathematicalnotation = true)

# Variables
Rb  := 16.55*10^(-3):
dis := 25*10^(-3):
Rs  := sqrt(Rb^2-((1/2)*dis)^2);  # theoretical radius of ball
# Butterfly definition
delta(phi[p]) := 0.1095-0.0405*cos(2*phi[p]);
# Vector definitions
d    :=  delta(phi[p]):
x[p]:=  d*sin(phi[p]):
y[p]:=  d*cos(phi[p]):
r[p]:=  <x[p],y[p],0>:
v[p]:=  diff(r[p],phi[p]):
tau :=  v[p]/(norm(v[p],2,conjugate=false):
beta     :=  simplify(arctan( abs(tau[1])/abs(tau[2]) ) ):
AEB    := simplify((1/2)*((1+sin(phi[p])*(1/abs(sin(phi[p])))*v[p][2]*(1/abs(v[p][2])))*
    Pi)-sin(phi[p])*(1/abs(sin(phi[p])))*v[p][2]*(1/abs(v[p][2]))*gamma_old):
BAE := phi[p] - (1 - sin(phi[p])/abs(sin(phi[p])))*0.5*Pi - (1 - sin(2*phi[p])/abs(sin
    (2*phi[p])))*0.5*(-Pi + 2*phi[p]):
L3   :=  d*sin( Pi - AEB - BAE ) + Rs:
L4   :=  d*cos( Pi - AEB - BAE):
phi[c]  :=    simplify( -sin(2*phi[p])/abs(sin(2*phi[p]))*arctan(L3/L4) + Pi - AEB + (1 -
    sin(2*phi[p])/abs(sin(2*phi[p])))*0.5*(-Pi + 2*AEB) + (1 - sin(phi[p])/abs(sin(phi[p
    ])))*0.5*Pi )
eval(phi[c],phi[p]=0.2*Pi)
```

D.3 'Matlab' script function $\varphi(\phi)$ 'Butterfly' robot

```
function [phic_Case2, Rho_real] = phic_Butterfly(Phi)
if Phi > pi                         % Half-symmetry
Phiv     = Phi - pi;
else
Phiv     = Phi;
end
d        = 25*10^-3;             % Distance butterfly plates
Rb       = 16.55*10^-3;          % Real radius ball
```

```
R        = sqrt(Rb^2 - (d/2)^2); % Effective radius ball
k        = [0;0;1];              % Vector pointing outwards
delta    = 0.1095 - 0.0405*cos(2*Phiv);
del_vec = [ sin(Phiv)*delta;
cos(Phiv)*delta;
0];
ddeldphi      = [ cos(Phiv)*delta+0.0810*sin(Phiv)*sin(2*Phiv);
-sin(Phiv)*delta+0.0810*cos(Phiv)*sin(2*Phiv);
0];
tau           = ddeldphi*1/(norm(ddeldphi));
n             = cross(k,tau);
Rho_real      = del_vec + n*R;
if Phi > pi
phic_Case2 = pi + atan2(Rho_real(1),Rho_real(2));
Rho_real = -Rho_real
else
phic_Case2 = atan2(Rho_real(1),Rho_real(2));
end
```

D.4 'Maple' Script Analytical approach ((4.29) and (4.32)) 'Butterfly' robot

Note φ = phic and ϕ = phi[p].

```
restart;
with(Student[VectorCalculus]); with(VectorCalculus); with(plots); with(LinearAlgebra);
    with(ArrayTools); with(CodeGeneration); with(Physics); with(DifferentialGeometry);
    with(JetCalculus); BasisFormat(false); Setup(mathematicalnotation = true)

# position vectors
x[p](t) := Rp(t)*sin(phi[p](t)):
y[p](t) := Rp(t)*cos(phi[p](t)):
x[c](t) := Rc(t)*sin*(phi[c](t)):
y[c](t) := Rc(t)*cos(phi[c](t)):
r[p] := <x[p](t), y[p](t), 0>: # Point contact vector
v[p] := diff(r[p],t): # velocity vector of the point contact
tau(t) := simplify( v[p] / (norm(v[p],2,conjugate=false)) ): # Tangent vector
# See shape of normal vector
n(t) := Multiply(<cos(Pi/2), -sin(Pi/2), 0; sin(Pi/2), cos(Pi/2), 0; 0,0,1>, tau(t)):
# replace d/dt Rp expression with Rp_acc in normal vector
nv := simplify( subs( d/dt Rp(t) = Rp_acc*(d/dt phi[p](t)), n(t) )):
# copy the expressions of nv into nv2 in which (d/dt phi[p])/|(d/dt phi[p])| is
    seperated as 'dir'
nv2 := < ( Rp(t)sin(phi[p](t)) - cos(phi[p](t))Rp_acc )/sqrt( (Rp(t)^2 + Rp_acc^2) ) *
    dir, ( Rp(t)cos(phi[p](t)) + sin(phi[p](t))Rp_acc )/sqrt( (Rp(t)^2 + Rp_acc^2) )*dir
    , 0 >:
# position vector of center of ball
r[c] := r[p] + Multiply(Rd,nv2) # in which Rd is theoretical radius ball

# Definitions of Rc and phic(=varphi in our case)
Rcsquare := simplify(expand(r[c][1]^2 + r[c][2]^2)) : # expand just used so it looks
    better I think
Rc_real := sqrt(Rcsquare) :
phic := arctan( r[c][1] / r[c][2] ):

# Specifications of the objects we use
Rb := 16.55*10^-3 : # Real radius ball
d := 25*10^-3 : Distance between the two plates
Rd := sqrt(Rb^2 - (dis/2)^2): # Theoretical radius of disk
dir := 1:  # unidirectional clockwise rotation makes (d\dt phi[p])/|(d\dt phi[p])| =1
d := 0.1095 - 0.0405cos(2phi[p]) : # Shape of butterfly plate
Rp(t) := d:   # Point contact curve is shape of butterfly plate
Rp_acc := diff(Rp(t),phi[p]):
```

```
#  Make sure that definitions of Rc and phic are correct : change all phi[p](t) in phi[p
    ], as maple thinks theyre two different things (may mess up differentiating
    processes)
Rc_real := subs(phi[p](t) = phi[p], Rc_real):
phic := subs(phi[p](t) = phi[p], phic):

# input and result
phip := 0.25*Pi:
Rc_value := eval(Rc_real, phi[p]=phip)
phic_value := eval(phic, phi[p] = phip)
```

E Expressing angle of the point contact in angle of the center of the ball

E.1 Geometric relations

E.1.1 Watch relations of previous $\mathbf{R}_{\#}$'s to predict $\mathbf{R}_2(\varphi)$

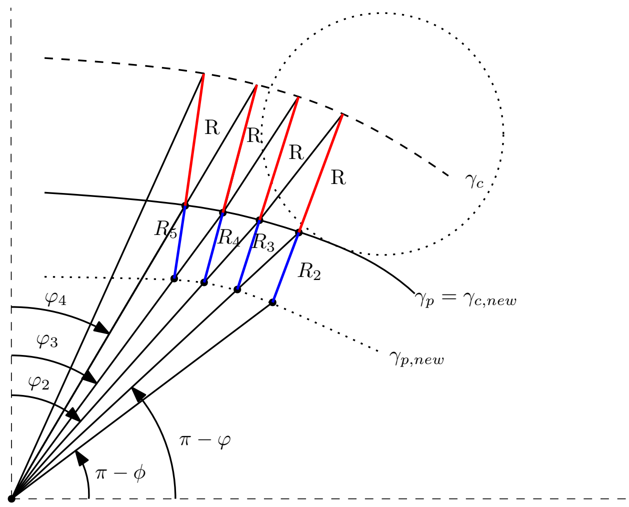

Figure 21: Attempt by relations of previous $R_{\#}$

Figure 21 is built up in the same way as figure 16, only now more angles and $R_{\#}$'s are added (notice that the figure is made in such a way that the current angle of the center of the ball is equal to the point contact angle of the ball in an earlier position). This idea builts up from the attempt of section 5.2.2 in which new curves $\gamma_{p,new}$ and $\gamma_{c,new}$ are defined. From section 5.2.2 we know that $R_{\#}$ (blue lines in Figure 21) is not constant over the angle φ, while 'R' (red line) is constant. From chapter 4 we have determined an expression in which we can rewrite φ in ϕ. If we rename this expression $\varphi(\phi)$ as $f(\phi)$, then we can make the following relations using the angles on Figure 21.

$$\begin{aligned} \varphi(\phi) &= f(\phi) \\ \varphi_2(\varphi) &= f(\varphi) \\ \varphi_3(\varphi_2) &= f(\varphi_2) \\ \varphi_4(\varphi_3) &= f(\varphi_3) \end{aligned} \tag{E.1}$$

In section 5.2.2 the problem was that it was not possible to find an expression for R_2 (as in Figure 21) in φ (note that the R_2 used in this appendix is not equal to the R_2 used in section 5.2.2). An expression for R_2 in ϕ however can be gotten, which we will call g(ϕ). A similar expression can then also be derived for R_3 in φ, which will be called g(φ) (due to similarities of triangles). With the relations of (E.1), we can then get the following $R_{\#}$ relations.

$$\begin{aligned} R_3 &= g(\varphi) \\ R_4 &= g(\varphi_2(\varphi)) \\ R_5 &= g(\varphi_3(\varphi_2(\varphi))) \end{aligned} \tag{E.2}$$

With these radius expressions in φ, we can then calculate $R_2(\varphi)$ as in (E.3).

$$R_2(\varphi) = R_3(\frac{R_3}{R_4} - (\frac{R_4}{R_5} - \frac{R_3}{R_4})) \tag{E.3}$$

This equation is based on the following idea : The change of R_2 compared to R_3, will be as big as the previous change (assuming that the ball rolls to the right) R_3 compared to R_4 minus the change of the relation before that ($\frac{R_4}{R_5} - \frac{R_3}{R_4}$). So if $\frac{R_4}{R_5}$ was bigger than $\frac{R_3}{R_4}$, then we assume this relation to set fort in a way that $\frac{R_3}{R_4}$ should also be bigger than $\frac{R_2}{R_3}$. The problem with this method is the assumption that the same kind of development will occur for $R_{\#}$. Equation (E.3) is however highly non-linear, thus we can not assume that the same development will occur in the future.

E.1.2 Try to get $R_2(\varphi)$ with the knowledge that R ('real' radius of ball) is constant

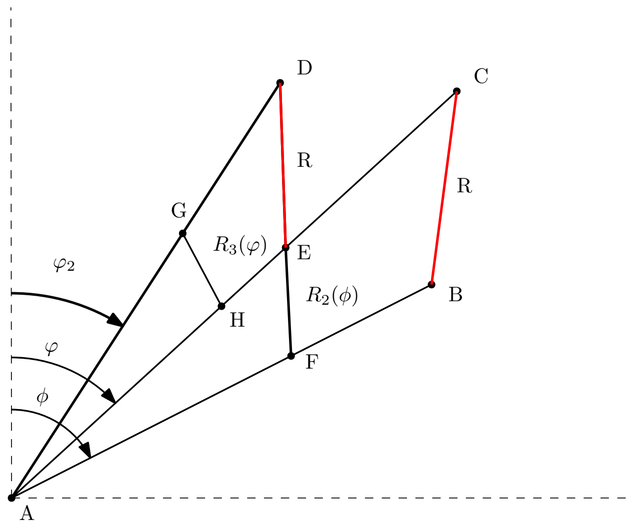

Figure 22: Attempt geometric relations constant R

This attempt is also built up from section 5.2.2 as we can see that the new radius $R_{\#}$ is used. We try to look at the constant 'R' and determine the relations from it (note that AE=$\delta(\varphi)$ and AG = $\delta(\varphi_2)$).

$$\begin{cases} ||\vec{AD} - \vec{AE}|| & = R \\ ||\vec{AC} - \vec{AB}|| & = R \end{cases} \tag{E.4}$$

$$\begin{gathered} ||\vec{AD} - \vec{AE}|| = ||\vec{AC} - \vec{AB}|| \\ ||\vec{AD}(\varphi) - \vec{AE}(\varphi)|| = ||\vec{AE}(\varphi) + \vec{EC} - \vec{AF}|| \end{gathered} \tag{E.5}$$

Here $R_2(\varphi)$ can be determined when an expression for $\vec{AF}(\varphi)$ is present.

$$\vec{R}_2(\varphi) = \vec{AE}(\varphi) - \vec{AF}(\varphi) \tag{E.6}$$

$\vec{AF}(\varphi)$ may then be gotten from (E.5). This would have been possible, if $\vec{AF}$ was the only unknown. The problem now is that there is another unknown $\vec{EC}$, which thus makes this problem unsolvable.

E.1.3 Use the horizontal and vertical position of the point contact to get $\phi(\varphi)$

From Figure 23 it can be seen that the the following relation holds.

$$\tan(\phi) = \frac{B_x}{B_y} \tag{E.7}$$

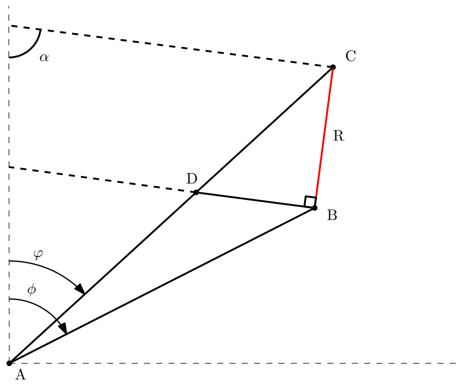

Figure 23: Attempt horizontal and vertical positions of B and D

The B position has the following relation to the D position (which is connected to φ). Note that α is a function of ϕ as AB $= \delta(\phi) = R_p$.

$$\begin{aligned} B_x - D_x &= BD_x \\ B_y - D_y &= BD_y \\ \tan(\varphi) &= \frac{D_x}{D_y} \\ ||\vec{BD}|| &= \frac{R}{\tan(-0.5\pi + \varphi + \alpha(\phi))} \end{aligned} \tag{E.8}$$

Combining (E.7) and (E.8) would then give (E.9).

$$\tan(\phi) = \frac{BD_x + D_y \tan(\varphi)}{BD_y + D_y} \tag{E.9}$$

Notice that BD in this expression contains $\alpha(\phi)$, which makes it impossible to solve (E.9) as $\alpha(\phi)$ is a difficult expression (5.7).

E.2 Trigonometric approximation

E.2.1 'Matlab' script explanation

The 'Matlab' script is based on (E.10), in which it tries to find a trigonometric polynomial (Ax) which approximates the real solution that is stored in vector B. It does this by determining the values of the coefficients $a_0, .., a_N, b_1, .., b_N$ stored in the vector 'x'. The accuracy of this approximation depends on the degree of polynomial that is desired, which can be set by the value of 'N'. The solutions used in the B vector are the expressions of the analytical approach given in section 4.2.

$$\begin{aligned} &Ax = B \rightarrow x = A \backslash B \\ &B = R_c(\phi) \; or \; \frac{dR_c}{d\varphi}(\phi) \\ &A = \begin{pmatrix} 1 & \cos(\varphi(\phi(1))) & ... & \cos(N\varphi(\phi(1))) & \sin(\varphi(\phi(1))) & ... & \sin(N\varphi(\phi(1))) \\ 1 & \cos(\varphi(\phi(2))) & ... & \cos(N\varphi(\phi(2))) & \sin(\varphi(\phi(2))) & ... & \sin(N\varphi(\phi(2))) \\ ... & ... & ... & ... & ... & ... & ... \\ 1 & \cos(\varphi(\phi(end))) & ... & \cos(N\varphi(\phi(end))) & \sin(\varphi(\phi(end))) & ... & \sin(N\varphi(\phi(end))) \end{pmatrix} \end{aligned} \tag{E.10}$$

E.2.2 Attempts in 'Maple'

For these attempts, the variables (e.g. $R_c(\phi)$) of the analytical approach (section 4.2) are used. Here the 'Maple' script of appendix D.4 has been used and extended. From running the 'Matlab' script explained in the previous subsection, the following shapes for the approximations can be gotten.

$$\begin{aligned} R_{c,approx}(\varphi) &= a_0 + \sum_{n=1}^{N}((1-(n \mod 2))a_n\cos(n\varphi) + (n \mod 2)b_n\sin(n\varphi)) \\ \frac{dR_c}{d\varphi}_{approx}(\varphi) &= \sum_{n=1}^{N}((n \mod 2)a_n\cos(n\varphi) + (1-(n \mod 2))b_n\sin(n\varphi)) \end{aligned} \tag{E.11}$$

'Match' command

The 'Match(expression=pattern,variabel,'return argument')' command will check if the given pattern input exists for the expression used. The pattern in (E.12) was used as an input. A polynomial degree of N = 2 has been chosen as the approximation was a good estimate of the real $R_c(\phi)$. The expression is less restrictive in the sense that it does not require 'cos(3φ)' or higher degree terms to be present. An 'x' (residual) term has been added to compensate this.

$$R_{c,approx} = a_0 + a_2\cos(2\varphi(\phi)) + b_1\sin(\varphi(\phi)) + x \tag{E.12}$$

Adding the following after appendix D.4 will then give the answer that no such pattern exists (output = 'false').

```
guess := a0+a2*cos(2*phic2)+b1*sin(phic)+x;
match(Rc_real = guess, phi[p], 'parameters');
```

The same can be done for the derivative which will also result in a 'false' output.

$$\frac{dR_c}{d\varphi}_{approx} = a_1\cos(\varphi(\phi)) + b_2\sin(2\varphi) + x \tag{E.13}$$

```
dRdvarphi := simplify((diff(Rc_real, phi[p]))*(1/(diff(phic, phi[p]))));
guess2 := a0+a1*cos(phic)+b2*sin(2*phic)+x;
match(dRdvarphi = guess2, phi[p], 'parameters');
```

'Solve' command

The 'solve(equations, variables)' command can also be used for checking the approximation. This command assigns values to the coefficients of (E.12) (e.g. a_1 or x) when solving $R_c = R_{c,approx}$. If these coefficient values are reasonable, then this would also prove that the approximation is good. The following commands can then be used, in which the 'guess'-equations are the same as the patterns mentioned in equation (E.12) and (E.13).

```
solve(guess = Rc_real, {a0, a2, b1, x});
solve(guess2 = dRdvarphi, {a1, b2, x});
```

The problem however is that it won't give all coefficients a constant value. For most coefficients it will give answers like $a_0 = a_0$ (meaning it could be anything), but there will be one variable which gets a non-constant value like $b_1 = b_1(a_0, a_2, x, \phi)$. This would mean that this method does not work as all coefficients (except x) should be constant values.

'Simplify' command with siderelations

As explained in appendix B.2, we can add siderelations to make the equation 'simpler'. The problem is, is that the 'sin($\varphi(\phi)$)' and 'cos($\varphi(\phi)$)' terms in (E.12) and (E.13) all consist of square root terms. It has been noted before that the 'simplify' command does not work when dealing with square root terms in the siderelations. This will also explain why the command below will not work, as the siderelations contain square root terms.

```
r[c] := subs(phi[p](t) = phi[p], r[c]);
sinpc := simplify(r[c][1]*(1/Rc_real));
cospc := simplify(r[c][2]*(1/Rc_real));
sin2pc := 2*sinpc*cospc;
simp1 := {sin2pc = s2};   # Siderelation
simplify(dRdvarphi, simp1);
```

Another way is thus to replace these square root terms with something else. To do this, a look is given at the symbolic expressions given in (E.14) (section 4.2).

$$\begin{aligned} \varphi(\phi) &= \arctan\left(\frac{R_p(\phi)\sin(\phi) + \frac{R(R_p\sin(\phi) - \cos(\phi)(\frac{d}{d\phi}R_p(\phi))}{\sqrt{R_p^2(\phi)+(\frac{d}{d\phi}R_p(\phi))^2}}}{R_p(\phi)\cos(\phi) + \frac{R(R_p\cos(\phi) + \sin(\phi)(\frac{d}{d\phi}R_p(\phi))}{\sqrt{R_p^2(\phi)+(\frac{d}{d\phi}R_p(\phi))^2}}}\right) \\ R_c(\phi) &= \sqrt{\frac{(R^2 + R_p(\phi)^2)\sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2} + 2R_p(\phi)^2 R}{\sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2}}} \end{aligned} \tag{E.14}$$

Taking the quadratic expression of $R_c(\phi)$ will not eliminate the square root terms. To eliminate the square root terms left behind, the following substitution is done.

$$C_2 = \sqrt{R_p(\phi)^2 + \frac{dR_p}{d\phi}^2} \tag{E.15}$$

Using this substitution and making the terms involving (E.14) quadratic (e.g. 'sin($\varphi(\phi)$)'), will eliminate all the square root terms. This allows us to set up siderelations in which we want to simplify expressions like 'sin$(\varphi(\phi))^2=\mu$', in which μ is the replacement variable. The result however is not what we wanted, as 'Maple' can not simplify the R_c into such replacement variables. The problem could be due to that terms need to be added for the siderelation to be present and that it can not be really called a simplification. If for example you had the siderelation '(a+d) = e' and the expression 'a+b=c', then the 'simplify' command would not give an answer as 'e - d + b = c' while it may be desired to get the variable 'e' in the expression.