

# Unicycle Robot

Fabian Kappeler

June 28, 2007

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Related Work</b>	<b>4</b>
2.1	Basic notions / considerations . . . . .	4
2.1.1	Angles: . . . . .	4
2.1.2	Human riding a unicycle: . . . . .	4
2.1.3	Stability: . . . . .	6
2.2	Set-up . . . . .	6
2.2.1	Bio-Inspired approaches . . . . .	6
2.2.2	Gyroscopic precession . . . . .	7
2.3	Control . . . . .	8
2.4	Further remarks . . . . .	9
2.4.1	Turning approaches . . . . .	9
2.4.2	Sensors . . . . .	9
2.4.3	Friction . . . . .	9
2.5	Functionalities of the unicycle to develop . . . . .	10
<b>3</b>	<b>Calculation of motion dynamics</b>	<b>11</b>
3.1	Generalized coordinates . . . . .	11
3.2	Basics: Lagrange theory . . . . .	13
3.3	Simple calculation of basic equations . . . . .	14
3.4	Calculation of equations of motion, Constraint handling . . . . .	15
3.5	Evaluated configurations . . . . .	16
3.6	Validation of the obtained models . . . . .	17
<b>4</b>	<b>Controllability</b>	<b>19</b>
4.1	Linearization, linearized models, controllability . . . . .	19
4.2	Results . . . . .	20
4.2.1	Disk-actuator . . . . .	20
4.2.2	Rotor . . . . .	21
4.2.3	Inclined rotor . . . . .	21
4.2.4	Combination of rotor and disk . . . . .	22
4.2.5	General remarks . . . . .	22

<b>5 Conclusion, future work</b>	<b>24</b>
References . . . . .	25
<b>A Contents of the CD</b>	<b>27</b>
<b>B Manual to the m-files</b>	<b>28</b>

# Chapter 1

## Introduction

In the last nearly 20 years, several researchers throughout the world developed unicycle robots and associated control systems. The problem of the inherently unstable 3D-Unicycle can be seen as an enhancement of the 1D inverted pendulum control problem.

In 2000 “Joe le pendule” (a mobile inverted pendulum robot, see [2]) has been developped within the “industrial electronics laboratory” at EPFL. This actual project (and following future projects) wants to go a step further. The goal is to develop, build and test a mobile unicycle robot. The robot will be used as a demonstration object for courses related to control (probably mainly in the advanced, linear and nonlinear multi-variable, control courses).

Although a unicycle robot is inherently unstable, it has several advantages over multi-wheeled, statically stable robots: As it has only one wheel (one touching point with the ground) it requires less space. A unicycle is better suited to move on inclined planes. As it is based on dynamic stability it shows better dynamic behavior and mobility ([5], [4]).

As performance limits in mobile robotics are pushed, dynamic effects (and therefore dynamic stability) will become more and more important. Unicycle robots might be the solution. However up to now no real-world application for unicycle robots is known.

# Chapter 2

## Related Work

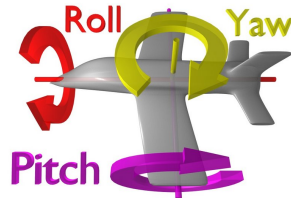
### 2.1 Basic notions / considerations

In this section some basic notions and observations from a human riding a unicycle will be reported. This latter can give an intuition to the problem.

#### 2.1.1 Angles:

A 3D unicycle can be characterized by 3 tilt angles. These are namely the roll angle (lateral tilt, perpendicular to the direction of motion), the pitch angle (longitudinal tilt, parallel to the direction of motion) and the yaw angle (heading, indicating the direction of motion in a fixed coordinate system). For illustration see Fig. 2.1.

If we want to know the position of the unicycle in space we further need to know the xy-coordinates of the robots touching point with the ground.



*Figure 2.1: Illustration of roll, pitch and yaw-angle on an airplane*

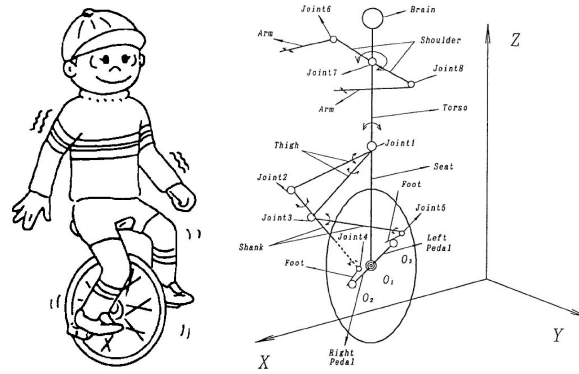
#### 2.1.2 Human riding a unicycle:

*Set-Up:* A human riding a unicycle represents an over actuated system. In fact it can be split into two subsystems:

The first, lower part, is constituted of the riders legs and the unicycle. With his legs, the rider controls the position and speed of the unicycle wheel. This action controls the pitch angle and velocity of the whole system.

The second, upper part, is constituted of the humans wrist, body and arms. By moving the arms,

leaning and turning the body, the rider can influence the roll, pitch and yaw angle of the unicycle.



**Figure 2.2:** Human riding a unicycle (left) and its schematic (right), taken from [11]

*Stabilization approach:* Stabilizing a unicycle means to keep it in a vertical position, thus this means to control roll and pitch angles. Control of yaw is not necessary for stabilizing. To stabilize the unicycle with the above mentioned “human system” three basic approaches can be followed: The first one consists of blocking the wheel and balancing the unicycle just by leaning (swinging) movements of the body and arms. By blocking the wheel, we lose one degree of liberty. It is then no more a unicycle (problem) but can be seen as a 2D inverted pendulum. The second possibility is to decouple longitudinal and lateral stability. Longitudinal stability can be obtained by appropriate control of the wheel, lateral stability by appropriate movements in the lateral plane (only leaning to the left or right, no leaning to the front or back). The third approach uses the wheel (the legs) to stabilize the pitch angle and rotary movements of the body to turn the wheel in the direction the unicycle is falling. With this approach a lateral deviation is turned into a pitch angle, that is finally eliminated by pedaling appropriately.

As a human has all possibilities, the observed movements of the body are a combination of the described approaches. Intuition says that the third approach is probably the most robust (tolerating larger deviations; as the effects of leaning are probably more limited than those of turning forces). This implementation is also more often used in literature (see later section 2.2).

*Turning methods:* Turning methods are required to control the unicycle’s heading (the yaw direction). So turning is complementary to stabilizing the unicycle in terms that it only concentrates on yaw, whilst stabilizing only concentrates on roll and pitch. With human riders two different methods to make turns can be observed. The first one uses the above mentioned rotary movements of the body to change the wheel’s direction, this time not only for stabilizing but to get a specific direction. These jerky movements allow to turn the wheel practically on the spot. A second method consists of leaning slightly to one side, while pedaling. To keep the dynamic equilibrium an appropriate forward lean is also required. The unicycle will then change direction

toward the leaning side, probably through gyroscopic effects. As this approach needs good control of the riders center of mass it is mainly observed with more experienced / skilled riders.

### 2.1.3 Stability:

Normally a system is said to be stable, when it can be brought to a precise state. As a unicycle is highly unstable, it is impossible to bring it to a specific state (for instance, it will never have all the states equal to zero). Based on observations on human riders, Sheng [11] propose a definition of postural stability.

The first observation they made is, that the postural stability will often be broken, when either pitch or roll angle will become greater than  $16^\circ$ . The second observation is, that if there's no control or too little stabilizing force, stability will get lost in 1s. Thus if postural stability is kept longer than 1s, the applied control is working. Sheng defines that the posture of a unicycle is stabilized if:

1. The state of the unicycle (in particular roll and pitch angles) stay within some specific range.
2. The posture is kept longer than 1s.

*Stable domain:* The range of the stable domain (maximum angles before stability is lost) depends heavily on the maximum deliverable control force. As done by Naveh, the range can be calculated in consequence. This allows to define a working range, useful to know when simplification of the equation of motion is needed (see [10]).

## 2.2 Set-up

The existing approaches to the unicycle problem can be classified into three groups depending on the used actuation principle. Two of them are inspired of a human rider, the third uses gyroscopical precession:

### 2.2.1 Bio-Inspired approaches

These robots use basically the stabilization techniques described in section 2.1.2.

The first type of robots uses moving weights to *change the center of gravity* to the left or to the right, and hereby control the roll angle. The pitch angle is controlled by the wheel.

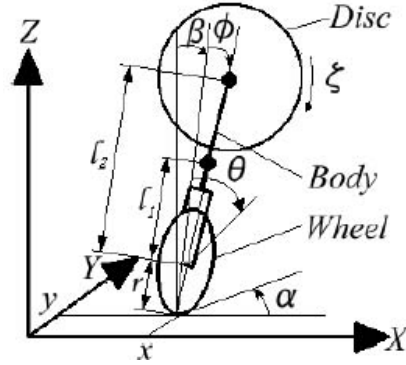
Works are reported that use a simple moving weight (Ozaka) or two swinging arms (Usui). Unfortunately these works are not available on the Internet and written in japanese.

A robot using leaning and rotating arms implemented on lego bricks can be found on "youtube" [1].

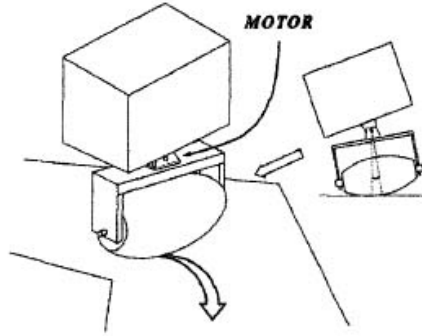
[8] proposes instead of moving weights a *disk rotating perpendicular to the wheel*. This allows to apply momentums in lateral directions and thereby control the roll angle. By intuition, the interaction of the disk with the rest of the robot is simpler than the interaction of moving weights (as there appear additional static and dynamic effects).

Nakajima et al. [9] propose a set-up using an elliptical (rugby-shaped) wheel. The upper part of the robot can be rotated around an axis parallel to the moving direction. By this lean of the upper

body the roll angle is controlled.



**Figure 2.3:** Unicycle with disk to control roll angle, taken from [8]



**Figure 2.4:** Unicycle with rugby-shaped wheel, taken from [9]

The second type of robots uses a *rotating table* (in the xy-plane) on top of the robot that represents the riders torso and arms ([14], [10],[12]). By applying torque to this table, the yaw angle of the wheel changes (due to its smaller inertia). The third stabilization approach of section 2.1.2 can though be applied. [11] uses in addition to the turntable, two closed-loop mechanisms representing the two feet of a human rider.

### 2.2.2 Gyroscopic precession

The third class of robots uses gyro precession to stabilize the roll angle. In fact when applying a torque perpendicular to the axis of rotation of a spinning disk, reaction forces appear perpendicular to the rotation axes. The fundamental equation for gyroscopic precession is:  $T = J \times \omega \times \Omega$

So when a disk, spinning around x-axis, is rotated around y-axis, a torque is generated along the z-axis (see also Figure 2.5.

[5] uses this principle to stabilize the roll angle. On top of the robot two fly-wheels that can be tilted along y-axis rotate in opposite directions along z-axis, thus creating the torque to control the



roll angle. When the gyroscope is in a tilted position the generated torque has not only a rolling but also a yawing component. Thus, two gyroscopes are used to compensate this effect.

Another robot, called “Gyrover”, using a gyroscope is proposed by [4]. The stabilizing mechanism being inside the wheel, it is not a unicycle in the classical way. However, it shows some interesting features. The control mechanism consists of a gyroscope rotating along the y-axis. The stabilization principle is thus not gyroscopic precession, but the gyroscope serving as inertial reference of the vertical plane. The gyroscope can be tilted along the y-axis, making the robot lean. The flywheel’s position near the center of gravity, even enables the robot to righten up from its rest position (laying on a side).

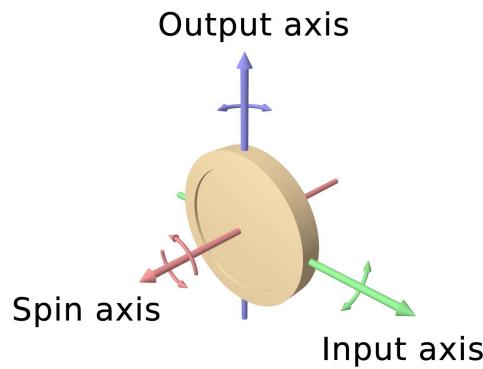
Gyroscopic precession is though also observed. In fact by tilting the flywheel a rotation around the z-axis is induced. This effect is used for directional control (yaw angle) of the Gyrover.

## 2.3 Control

In this paragraph the highlight will be set on the control techniques applied in previous papers. In fact, early works decoupled the problem and used basic (PD) controllers ([9], [11]). Vos [14] developed a new LQG structure to control the system. Linearized state feedback controllers were used by [5] and [8].

The unicycles equation of motion are highly non-linear and contain couplings between different motions. Unfortunately by linearizing these equations the couplings vanish, diminishing thereby the robustness of control. This is probably the reason why early works were only partially successful, if at all.

So, Naveh [10] uses non-linearized, yet simplified, equation of motion for the control. Other researchers ([12]) go even further, using fuzzy intelligent control with soft computing based on GA (genetic algorithms) and FNN (feedforward neural network).



**Figure 2.5:** Gyro with spinning axis, axis of applied torque and axis of reaction torque, from Wikipedia

## 2.4 Further remarks

### 2.4.1 Turning approaches

The two in section 2.1.2 described turning methods can be used with robotic unicycle. Reported works use mainly the first approach (jerky rotary movements), which is easier to implement. However there exist papers using the second approach:

[8] uses this principle to change the yaw-direction of an underactuated unicycle (wheel plus disk in perpendicular plane as actuator). However it is difficult to stabilize the unicycle at a more or less precise roll angle. So as to avoid the just mentioned stabilization problem, they add sinusoidal wave input to the state feedback control of both the wheel and the disk. This generates coordinated lean-forward-backward motion of the unicycle, forcing thereby a change in yaw-direction.

### 2.4.2 Sensors

To control a real unicycle, the robots posture must be determined. This is done via appropriate sensors. Two categories of sensors can be used:

- accelerometers and
- (rate) gyroscopes

However, they both have drawbacks. Accelerometers are heavily disturbed by the (often high-frequency) stabilization motions of the unicycle. Gyroscopes are more accurate in the high frequency domain, but suffer from drift (less accuracy in lower frequency domain). The drift rate depends heavily on the quality of the sensor. Most reported experimental set-ups used gyros.

Some researchers [11] circumvented the gyro-drift problem, by conducting the experiment only during short time intervals (for instance 8s; if the stability isn't broken after 1s, the control is supposed to work, see 2.1.3). In that time the drift error won't get too big.

[8] uses a combination of one gyro and two accelerometers per angle to determine, to compensate the respective drawbacks.

### 2.4.3 Friction

As the unicycle has only one contact point to the ground, changing the direction (the yaw angle) is a non-trivial problem. Different approaches have been presented in section 2.1.2. However, turnings are heavily dependent on the wheel-surface interaction. Naveh [10] proposes a combination of Coulomb and viscous friction torques in the wheel yawing direction:  $M_f = -f_\psi \dot{\psi} - F_\psi \text{sign}(\dot{\psi})$ . It is important to have good representation of the friction at that point. This can be done by calculation and adjustment of the parameters using on-line measurement (see [14], [13]).

Most of the papers finally neglect the surface friction but use a non-linear bang-bang control to overcome the nonlinear effects of it ([10]).

## 2.5 Functionalities of the unicycle to develop

As this semester project is the first on that topic the goal is to deliver the fundamentals for following projects. Thus the work was mainly concentrated on the development of the equations of motion and basic study of some actuation principles.

The first goal in control of the unicycle will be to stabilize it in an upright position (if possible with control of all states, even if that isn't necessary), the next step then to impose simple trajectories (straight lines, curves, turns), respectively to implement a cascade controller that brings the unicycle to a specific xy-coordinate and a specific orientation.

The difference between the last two points is that the first one can be done in open-loop. Principles could be investigated that make the robot do simple trajectories (for example making it lean to a side and to the front in order to do a curved trajectory by feed-forward commands). At this stage the robot, can be commanded with a remote control.

The second step closes the loop. This cascade control could work in quasi static domain (i.e. to do a curve the unicycle is practically upright, the yaw is changed in small increments while the robot is moving forwards) or in more dynamic domain (leaning to the side and the front to do a smooth curve, as described in section 2.1.2; this approach eventually needs more sophisticated stabilizing controllers <sup>1</sup>). The cascade controller allows more complicated trajectory following than simple open loop steering.

The main characteristics of the set-up that has to be developed are:

- stabilize the unicycle in upright position,
- controlling as much states as possible
- allow turning at the spot

---

<sup>1</sup>eventually non-linear controlling techniques could be useful, as they take into account more information of the real model than linearized approaches and should thus be more robust

# Chapter 3

## Calculation of motion dynamics

In this chapter the determination of the equations of motion will be treated. These equations are obtained by the Lagrange approach. Therefore we start with the definitions for the modeling (the set of state variables) and the assumptions. Then a short reminder of the underlying theory and finally the obtained results will be presented. The entire development of the equations of motion can be found in the corresponding files on the CD.

### 3.1 Generalized coordinates

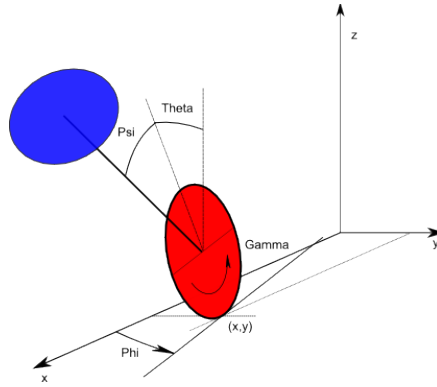
To describe the position of the unicycle in space we need to define some generalized coordinates. If we start with the wheel,  $x$  and  $y$  define its touching point with the ground and  $\phi$  gives the orientation. At rest, the unicycle is supposed to look into x-direction, so  $\phi$  is defined as the angle (in mathematical positive sense) between the x-axis and the unicycle's actual heading. Further angles are needed to describe a possible sideward lean of the wheel ( $\theta$ ) and its actual rotation position around its axis ( $\gamma$ ).

By adding a body to the wheel, an additional angle ( $\psi$ ) is necessary to describe the lean to front, resp. back. The angles  $\theta$ ,  $\psi$  and  $\phi$  correspond to roll, pitch and yaw angles described in section 2.1.1. There are other solutions to describe the position of a rotated body (different sets of euler angles). The one chosen is the most intuitive in this case and allows straight forward use, for example with (rate gyro-)sensors oriented in the principal planes of the unicycle.

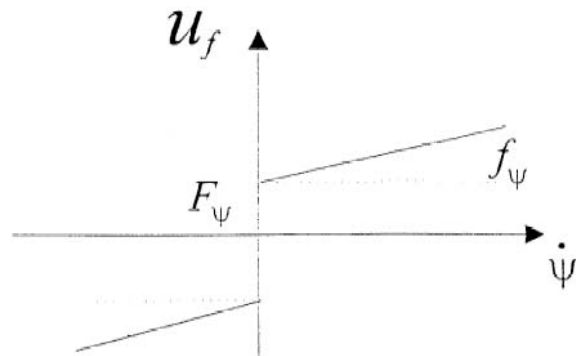
The generalized coordinates are visualized in fig. 3.1.

This configuration with wheel and body serves as point of departure for different configurations, that will be evaluated in the following chapters. The idea is that different actuator modules can be added to the basic configuration easily. State variables that are necessary for the control will be introduced in the respective sections.

The main *assumption* concerns the wheel. In fact, we suppose rolling without slipping, making the unicycle a non-holonomic vehicle. Further on viscous friction is modeled at the motors and between the wheel and the ground. Non-linear friction (the static friction part, see fig. 3.2) between wheel and ground is neglected at the moment.



**Figure 3.1:** Generalized coordinates for Unicycle with wheel and body



**Figure 3.2:** Model of non-linear friction between wheel and ground, note that  $\psi$  has to be replaced by  $\phi$  in our annotations. Figure taken from [10]

## 3.2 Basics: Lagrange theory

<sup>1</sup> Two approaches exist, to derive the equations of motion. The first one is to divide the concerned object into subparts and to write for each subpart the corresponding newton-euler equations (equilibrium of force, resp. of momentum). The system of equations can then be simplified iteratively. This approach not only delivers the equations of motion, but also the interaction forces between subparts.

The second is to use the Lagrange approach. For this approach the Lagrangian of the system needs to be determined. By a procedure of particular derivation steps, the equations of motion are obtained. This approach is easier than the first approach, but does not give any information about the interacting forces. As in this project, we are only interested in the equations of motion, the Lagrange approach will be used.

First of all, the *Lagrangian* of the system must be determined. The Lagrangian corresponds to the total kinetic energy (translational and rotational) minus the potential energy:

$$L(\vec{q}, \vec{\dot{q}}) = T_{kin,trans}(\vec{q}, \vec{\dot{q}}) + T_{kin,rot}(\vec{q}, \vec{\dot{q}}) - V(\vec{q}, \vec{\dot{q}})$$

$\vec{q}, \vec{\dot{q}}$  corresponds to the vector of generalized coordinates, resp. its derivatives.

The *equations of motion* are obtained by deriving for each generalized coordinate and finally solving the system of equation for the second derivatives of the generalized coordinates:

$$\frac{d}{dt} \left[ \frac{\partial L(\vec{q}, \vec{\dot{q}})}{\partial \dot{q}_k} \right] - \left[ \frac{\partial L(\vec{q}, \vec{\dot{q}})}{\partial q_k} \right] = F_k \quad , \quad k = 1, 2, \dots, n$$

$F_k$  are non-conservative external forces.

To determine these forces, only one generalized coordinate is free at a moment. We then consider which of the external forces have an effect on a small displacement  $\partial q_k$  of that coordinate.

It is important to notice that in mobile robotics no subpart is entirely fixed. This implies that a force created by an actuator has an effect not only on the part connected to the actuator arm, but also on that connected to the actuator basis (action-reaction, couple-counter-couple problem). This counter-effects appear automatically, when the determination of generalized force is done correctly.

In the case of the unicycle we need to take into account two *constraints*. In fact, as we assume roll without slip, the system has not 7, but only five degrees of freedom. The coordinates of the contact point are defined, by the yaw direction and wheel speed throughout time.

$$\dot{x} = \dot{\gamma} * R * \cos(\Phi) \tag{3.1}$$

$$\dot{y} = \dot{\gamma} * R * \sin(\Phi)$$

When deriving the equations of motions, these constraints need to be included. This is done by adding sort of a constraint force with a Lagrange multiplier per constraint to the generalized forces:

$$\frac{d}{dt} \left[ \frac{\partial L(\vec{q}, \vec{\dot{q}})}{\partial \dot{q}_k} \right] - \left[ \frac{\partial L(\vec{q}, \vec{\dot{q}})}{\partial q_k} \right] = F_k + \sum_i \lambda_i \frac{\partial c_i}{\partial q_k} \quad , \quad k = 1, 2, \dots, n, \quad i = 1, 2, \dots, m$$

---

<sup>1</sup>see also [7]

m is the number of constraints.

In the obtained set of equations, the terms of the constraint coordinates (in our case  $\dot{x}$ ,  $\dot{y}$ , resp.  $\ddot{x}$  and  $\ddot{y}$ ) need to be replaced by their constraint equivalent (3.1). m equations are then used to determine the Lagrange multipliers ( $\lambda_i$ ). After substitution, the rest of the equations can be solved for the unconstrained generalized coordinates.

### 3.3 Simple calculation of basic equations

Once the Lagrangian of the model is determined, the determination of the equation of motion is straight forward, all the (analytical) calculation can be done by computer. However some basic equations need to be programmed by hand. These are in particular, the equations describing the position of the center of mass and the generalized rotation vector for each subpart of the unicycle. From the first one we obtain the speed, needed to calculate the translational kinetic energy, as well as the height to calculate the potential energy. The second will be used to calculate the rotational kinetic energy.

To easily calculate the unicycles position in a fixed referential (needed to apply Lagrange), we start with the coordinates describing the unicycle in a mobile referential, that is fixed to the unicycle itself. The origin of the referential is fixed at the point of contact wheel-ground. The x-axis is directing to the front of the unicycle (the direction the unicycle is moving, when the wheel turns in positive direction), the z-axis is oriented in direction of the unicycle's body, the y-axis is oriented in consequence.

The coordinates in the mobile reference can then be transfered into the fixed reference by three rotations (corresponding to roll, pitch and yaw angles) and a translation (corresponding to the ground touching point)<sup>2</sup>. The rotations are done by three rotation matrices, that express the coordinates respectively after a rotation around y-, x- and z-axis (see below). The translation is done by simply adding the x-y-coordinates of the mobile reference origin to the turned mobile coordinates.

$$rotX(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}, \quad rotY(\psi) = \begin{pmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{pmatrix},$$

$$rotZ(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$RPY(\theta, \psi, \phi) = rotY(\psi) * rotX(\theta) * rotZ(\phi)$$

To determine the rotational kinetic energy, the rotation vectors (determined in different intermediate mobile references) need to be transfered onto the subparts primary axis of inertia. This is done by the inverse of the above described procedure (the rotation matrices are applied in inverse order and with angles of opposite sign).

---

<sup>2</sup>see also [3]

$$RPYinv(\theta, \psi, \phi) = rotZ(-\phi) * rotX(-\theta) * rotY(-\psi)$$

### 3.4 Calculation of equations of motion, Constraint handling

For simple systems, the equations of motion can be calculated with the approach described in 3.2. However, for more complicated systems (and the unicycle evidently is one of them), the set of equations becomes too complicated to solve it for the second derivatives. We then need to put the solution of Lagrange formalism into a more explicit form.

In fact, as

$$\frac{\partial f(q, \dot{q})}{\partial t} = \frac{\partial f}{\partial q} \dot{q} + \frac{\partial f}{\partial \dot{q}} \ddot{q}$$

the first term of Lagrange formalism becomes:

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{q}} = \frac{\partial^2 L}{\partial \dot{q}^2} \ddot{q} + \frac{\partial^2 L}{\partial \dot{q} \partial q} \dot{q}$$

By regrouping we obtain:

$$\frac{\partial^2 L}{\partial \dot{q}^2} \ddot{q} = \frac{\partial L}{\partial q} - \frac{\partial^2 L}{\partial \dot{q} \partial q} \dot{q} + F + \lambda^T \frac{\partial c}{\partial \dot{q}}$$

where  $I = \frac{\partial^2 L}{\partial \dot{q}^2}$  represents the matrix of inertia of the system,  $V_m = \frac{\partial^2 L}{\partial \dot{q} \partial q}$  is the centripetal and coriolis matrix. The other terms are replaced for simplicity of notation with  $dLdq = \frac{\partial L}{\partial q}$  and  $A = \frac{\partial c}{\partial \dot{q}}$ .

$$I\ddot{q} = dLdq - V_m\dot{q} + F + A^T\lambda$$

To eliminate the constraint terms, we follow the development done in [6]. First we determine the matrix S, so that

$$S^T(q)A^T(q) = 0$$

This matrix S contains the linearly independent vectors spanning the null space of  $A(q)$ . We furthermore note, that a auxiliary vector time function  $v(t)$  exists, such that for all  $t$

$$\dot{q} = S(q)v(t)$$

In our case this is equal to

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \\ \dot{\gamma} \\ \dot{\delta} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & R\cos(\phi) & 0 \\ 0 & 0 & 0 & R\sin(\phi) & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \\ \dot{\gamma} \\ \dot{\delta} \end{pmatrix}$$



The equations of motion can thus be rewritten as

$$S^T I S \dot{v} = -S^T (I \dot{S} + V_m S) v + S^T (dLdq + F)$$

Note that the A-matrix has vanished. There are now only n-m equations left. By adding the constraint equations, the system becomes complete again.

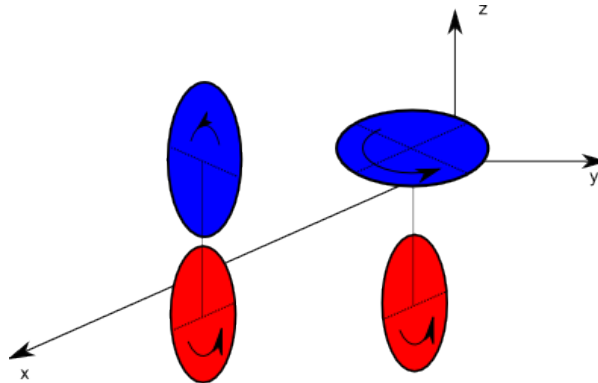
### 3.5 Evaluated configurations

The basic goal of this study is to find a set-up that fulfills the requirements cited in 2.5. Therefore two basic actuation principles have been modeled. The first one uses a turntable in the xy-plane on top of the unicycle as actuator. The second approach uses a rotary element as well, but in the plane perpendicular to the wheel (see fig. 3.3).

The center of mass of the rotary elements isn't necessarily located at the center of rotation. By simply changing the eccentricity distance, the models can represent four physical setup: Namely a unicycle with

- a symmetrical rotary disk in the xy-plane
- a simple mass (or several asymmetrically arranged masses) in the xy-plane
- a symmetrical rotary disk in the xz-plane or
- a (or several) swinging arm in the xz-plane

The eccentric actuators are visualized in fig. 3.4.



**Figure 3.3:** Actuation principles: disk (left) and rotor (right)

As the determination of the equations of motion, as described in the previous chapters, is totally analytical, I started development under Mathematica. The second part of work, validation and simulation of the obtained models, is numerical integration. The most powerful tool for that part is Matlab.

So the first models have been calculated with Mathematica. The obtained models are highly non-linear and complicated. Analytical inversion of the inertia matrix (and thus calculation of the state accelerations) is possible. However the resulting equations take several pages to display<sup>3</sup>.

<sup>3</sup>Hint to the use of Mathematica: Although it might be seducing, use "simplify" only at the right places. Even there think twice, the equations are just too complicated that (full-)simplify will end in useful time

Furthermore, the exportation to Matlab posed problems. As Mathematica uses a special notation (everything is based on lists) kind of a conversion needs to be done (brackets have to be removed, some replaced by matrix element separators “,” others with matrix line separators “;”). This conversion has to be done at least partially by hand, which is non-neglectible with that size of equations. In addition, Matlab wasn’t able to read the whole matrices of the system at once. Reading was possible element by element and the matrix then could be reconstructed. However, again this had to be done at least partially by hand, so finally I decided to stop development under Mathematica and switch to Matlab also for the analytical work.

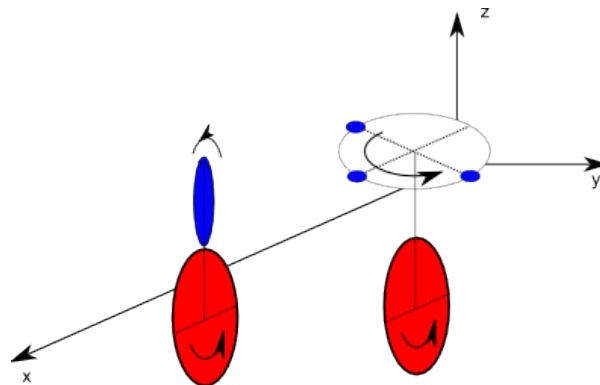
Thus the calculation has been redone under Matlab directly. Matlab, however, is not able to invert the inertia matrix analytically. This step had to be facilitated by replacing the physical parameters by their numerical values. Calculation of the equations of motion (for a specific set-up), necessary for the controller design, becomes feasible.

### 3.6 Validation of the obtained models

The obtained models have been inspected visually in open-loop simulation. It is clear that this isn’t a rigorous validation, however there’s no other possibility.

There are some particular initial conditions, where validation is simple.

- when the unicycle stands straight upright it has to stay there. This will never be observed in reality as there’s always a small displacement in any direction or disturbing forces that make the unicycle at least not stay there for long-time.
- when the unicycle is perfectly up-right in roll direction but with a slight displacement on pitch (slightly leaning to the front or back), the body will fall in that direction, while the wheel will be pushed in the opposite direction by the body’s weight.
- with a slight displacement on roll, but perfect alignment of pitch angle, the unicycle will simply fall to that side
- of course all above movements are symmetric around the vertical axis and thus also independent of the unicycle’s heading (yaw)



**Figure 3.4:** *Eccentric actuation principles: arm (left) and asymmetric rotor (right)*

This last point is also true for symmetric actuation set-ups in closed-loop simulation.

The above mentioned points have been verified on the models in open-loop simulation. The interested reader is directed toward the CD where the validation can be seen on video.

# Chapter 4

## Controllability

In this chapter the controllability of the proposed set-ups is evaluated. This first analysis evaluates the controllability of the linearized models. Although recent papers showed that linearization eliminates couplings and thus reduces the robustness of the controller, this approach was chosen for this first feasibility analysis for simplicity and time constraint reasons. A short reminder on the theory is given before the results are presented.

### 4.1 Linearization, linearized models, controllability

The obtained equations of motion are of the form:

$$\dot{x}(t) = f[x(t), u(t), t]$$

$$y(t) = g[x(t), u(t), t]$$

For simplicity reasons we want to transform it into a stationary linearized model:

$$\dot{x}(t) = A * x(t) + B * u(t)$$

$$y(t) = C * x(t) + D * u(t)$$

The A and B matrices are obtained by deriving the equations of motions for the states, resp. the entries:

$$A = \frac{\partial}{\partial x} f(x, u)$$

$$B = \frac{\partial}{\partial u} f(x, u)$$

We suppose that all states are observable, thus the C-matrix is identity and no observer is needed. There is no direct link from the entries to the observed states so the D-matrix is zero.

As the system is non-holonomic the xy-coordinates are not directly controllable. This primary analysis only controls the euler-angles of the unicycle and position and speed of the used actuators. This first controller aims to stabilize the unicycle in an upright position. The uncontrolled xy-coordinates have to be controlled later by a second (cascadic) controller that assigns appropriate

references to the first stabilizing controller (order of commands to change the orientation and wheelspeed to arrive at a specific xy-point<sup>1</sup>).

The controllability of a linearized model can be determined by the rank of the matrix of governability. The system is fully controllable only if the rank is equal to the number of states:

$$\text{rank}(G) = \text{rank}[I * B \quad A * B \quad \dots \quad A^{n-1}B] = n$$

If  $\text{rank}(G) < n$  some states are not controllable. However, this doesn't tell which state it is. This needs to be determined by subsequent suppression of states and observing the rank of the corresponding controllability matrix.

The linearization has been done around the point  $\vec{x} = 0$ .

For simplicity reasons and as this is fully sufficient for the feasibility study, (analog) lqr-controller have been designed for the feedback loop: the design is fully automatic and it allowed to do all the integration (simulation) within Matlab, without having to pass by Simulink.

## 4.2 Results

In this section the results of the evaluation of the different set-ups will be presented. As a general result can be cited, that effectively linearization (at least around the up-right position) eliminates the couplings and thus leaves us with a system of two independent inversed pendulums.

### 4.2.1 Disk-actuator

The linearized system using a symmetric disk-actuator presents as follows:

$$\begin{pmatrix} \ddot{\theta} \\ \ddot{\psi} \\ \ddot{\phi} \\ \ddot{\gamma} \\ \ddot{\delta} \\ \ddot{\theta} \\ \ddot{\psi} \\ \ddot{\phi} \\ \ddot{\gamma} \\ \ddot{\delta} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 18.2898 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3187 \\ 0 & 53.0928 & 0 & 0 & 0 & 0 & -1.7821 & 0 & 1.7821 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -32.8371 & 0 & 0 \\ 0 & -60.5444 & 0 & 0 & 0 & 0 & 2.9094 & 0 & -2.9094 & 0 \\ -18.2898 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5.4207 \end{bmatrix} * \begin{pmatrix} \theta \\ \psi \\ \phi \\ \gamma \\ \delta \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \\ \dot{\gamma} \\ \dot{\delta} \end{pmatrix} +$$

---

<sup>1</sup>see literature for steering of non-holonomic (unicycle-like) robots

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -6.3740 \\ -35.6425 & 0 \\ 0 & 0 \\ 58.1888 & 0 \\ 0 & 108.4148 \end{bmatrix} * \begin{pmatrix} TGamma \\ TDelta \end{pmatrix}$$

The controllability matrix is of rank 8, two states (yaw:  $\phi$  and  $\dot{\phi}$ ) are uncontrollable. In consequence this linearized model allows stabilization, but no complete control of the unicycle. A different set-up needs to be found for direct control of the orientation of the unicycle. However, using knowledge about the couplings, indirect control of yaw is possible (i.e. imposing sinusoids on both the wheel and the disk to force a coordinated sideways-front-back lean, making the unicycle turn, see [8] for more details).

The fact to use an asymmetric disk (for example a stick), doesn't change much (same rank, yaw stays uncontrollable).

### 4.2.2 Rotor

This set-up is the most used in literature. However, the controllability matrix of the linearized model is only of rank 7. The roll ( $\theta$  and  $\dot{\theta}$ ) and the yaw (this time only  $\phi$ ) are uncontrollable. To regain control of the yaw direction, the control of the rotor ( $\delta$  and  $\dot{\delta}$ ) needs to be omitted. With a symmetrical rotor the position doesn't matter anyway. The speed however would be nice to control (just to prevent it to grow infinitely).

Here again, an asymmetric rotor doesn't change much.

We can conclude that the system with a rotor cannot be stabilized by linearized control methods. However, a rotor allows control of the yaw direction, a mixed system between rotor and disk should thus be able to completely control the unicycle.

### 4.2.3 Inclined rotor

The idea of this set-up is to unify rotor and disk in only one actuator. The (symmetric) rotor of section 4.2.2 is inclined, so that it is kind of a mix between rotor (at inclination angle = 0) and disk (at inclination angle =  $\frac{\pi}{2}$ ).

To examine different possibilities of inclination angles, three parameters have been tested:

- the smallest eigenvalue of S1 (the matrix in Riccati's equation)
- the couple necessary to stabilize the unicycle from a position with 0.2rad offset in roll and pitch direction
- and the feasibility of turning at the spot

The idea behind these evaluation criteria is to estimate the controlling performance. The eigenvalues give an indication of the facility to access the states of the model. Thus, the bigger the smallest eigenvalue, the better the “worst” state is controllable. If the ratio between biggest and smallest eigenvalue is close to one, the controllability of the states is equilibrated.

Another way to evaluate the performance of a set-up is to determine its “bassin of attraction”, this is the maximal initial conditions it just still arrives to stabilize. However this determination needs several simulation runs (with initial conditions in different directions of the state space). The second criteria is kind of an alternative to complete determination of the bassin of attraction. In fact the closer the initial condition is to the boundary of the bassin, the bigger are the forces necessary to stay inside. Thus by comparing the maximal control forces on different set-ups for a given initial condition we get an estimation on the limits.

The last criteria just wants to make sure, that the set-up fulfills all the criteria defined in 2.5. Stabilization is tested with the first two criteria, turning at a spot with this last one.

The results are as follows:

The bigger the inclination, the bigger (the better) the smallest eigenvalue (0.0018 at 30°, 0.0034 at 60°). Also the biggest eigenvalue tends to decrease. However it is clear, that more the rotor is inclined, the more difficult it is to turn the unicycle at a spot, at high angles a try to turn the unicycle (even when at the beginning it is straight upright) might force the unicycle to fall.

Simulation couldn’t be done for all the set-ups (a conditioning error stops it before reasonable time spans), so the second criteria couldn’t be evaluated. In the successful simulations (at 30°) the set-up wasn’t able to stabilize the unicycle.

#### 4.2.4 Combination of rotor and disk

The last evaluated set-up is a real combination of rotor and disk. This actuation principle is more complicated than the others as it has a supplementary actuator. The system is, however, not over-actuated but possesses in principle for each degree of freedom one specific actuator. As a consequence of the two actuators, the system is heavier and has a bigger inertia.

With this configuration again simulation stops before reaching the wished time span due to conditioning errors. However at that time the system is in an upright position, the control seems to work (the system was taken to the equilibrium point).

#### 4.2.5 General remarks

In opposition to what was presented at the final presentation, within this final report (writing finished one week later) no clear conclusion can be drawn on the performance of the last two set-ups. The originally proposed set-up, the inclined rotor, doesn’t seem that promising anymore (in the only successful simulation the unicycle falls). The combined actuation (rotor and disk) seems to work (at least simulation stops when it is near the straight upright position), however no guaranty can be given that it stays there.

Where has this change come from: Unfortunately after the final presentation, when preparing the m-files for release, I discovered a mistake in the dynamic modeling. The mistake was situated at the actuator level, thus wasn’t detected by the validation done in open-loop (as the numerical

values of states corresponding to the actuators stay close to zero and thus have no influence on the global behavior of the unicycle). All results that are finally presented in this report (included the validation) are obtained with the corrected equations of motion, they are thus valid.

Why does the simulation stop: At this moment I only can guess. The most probable case is that those points are singularities. This means that very small change in a state variable provoke very large displacement in others. The exact origin of these problems need to be determined. In the worst case scenario, the chosen set of euler coordinates might have been a bad choice for simulation (although it is the most intuitive and straight forward in use with a real unicycle). A solution might thus be to chose another set of euler angles (eventually there are transformations between two sets?!) or to completely change strategy and use quaternions instead of rotation matrices to determine the equations of motion.

Although this problems remain, we can conclude that control of the unicycle is possible.



# Chapter 5

## Conclusion, future work

Within this semester project the foundations of the unicycle development have been laid. The equations of motion have been derived and should now be correct. This development was more than once really close to the limits of the used software tools. Several times simplifications had to be done, tricks needed to be found to lighten the work- or memory-load for the programs. Several basic set-ups have been evaluated and the feasibility has been confirmed.

All the simple set-ups using only one actuator, in addition to the wheel itself, were not able to control the unicycle entirely. The only working set-up for the moment is the combination of rotor and disk. Starting from that point, further set-ups (using two actuators like the proposed combined method) could be evaluated.

For the moment all physical parameters are quite arbitrary: Friction coefficients are table based estimations. Dimensions and masses of the unicycle are guesses and do not necessarily correspond to real unicycles. For further work it would probably be useful to determine real parameters as far as they are defined by the purchased material, to refine the model with these parameters and to design the actuators in consequence.

For this feasibility study analog lqr-controllers have been used. A real controller will certainly be discretized. So the results should be validated under this circumstances. Furthermore in this study we supposed easy access to all states. The real unicycle will certainly have less sensors than states, thus observer design will be necessary, modifying slightly the dynamics of the regulator. This will also have to be taken into account in future studies. To improve the robustness of the control non-linear control techniques could be applied. Certainly the cascade control system to command the unicycle with coordinates and orientation needs to be developed.

In parallel the real unicycle could be conceived and constructed. Special attention certainly needs to be paid to the choice of the position (angular) sensors and appropriate observers. Dealing with 'real world' problems such as noise and drift will become important in the end.

With the equations of motions, now hopefully in bug free state, the more interesting part of work begins.

As mentioned before the work on that project was more than once quite a challenge. Therefore I'd like to thank my assistants Damien Perritaz, Davide Buccieri and Yvan Michellod as well as MER Denis Gillet for their aid and ideas throughout this project. I've learned a lot and appreciated much the spirit and ambiance in the lab.

# Bibliography

- [1] Four wheeled robot unicycle. [http://www.robots-dreams.com/2006/12/four\\_wheeled\\_ro.html](http://www.robots-dreams.com/2006/12/four_wheeled_ro.html).
- [2] Joe le pendule. <http://leiwww.epfl.ch/joe/>.
- [3] Bleuler, Clavel, Baur, Bouri, and Breguet. *Robotique, Microrobotique*. Polycopie EPFL, 2004.
- [4] H. Benjamin Brown and Yangsheng Xu. A single wheel, gyroscopically stabilized robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3658–3663, April 1996.
- [5] Minh-Quan Dao and Kang-Zhi Liu. Gain-scheduled stabilization control of a unicycle robot. In *JSME International Journal*, volume 48, pages 649–655, January 2005.
- [6] R. Fierro and F. L. Lewis. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3805–3810, New Orleans, LA, December 1995.
- [7] Max-Olivier Hongler. *Modelisation dynamique*. Polycopie EPFL, 2007.
- [8] Sumiko Majima and Takashi Kasai. A controller for changing the yaw direction of an underactuated unicycle robot. In *International Conference on Technology and Automation*, volume 5, pages 73–88, October 2005.
- [9] Ryo Nakajima, Takahashi Tsubouchi, Shin’chi Yuta, and Eiji Koyanagi. A development of a new mechanism of an autonomous unicycle. In *Intelligent Robots and Systems*, volume 2, pages 906–912, September 1997.
- [10] Yoav Naveh, Pinhas Z.Bar-Yoseph, and Yoram Halevi. Nonlinear modeling and control of a unicycle. In *Dynamics and Control*, volume 9, pages 279–296, October 1999.
- [11] Zaiquan Sheng and Kazuo Yamafuji. Postural stability of a human riding a unicycle and its emulation by a robot. In *Robotics and Automation*, volume 13, pages 709–720, October 1997.
- [12] S.V. Ulyanov, S. Watanabe, V.S. Ulyanov, K. Yamafuji, L.V. Litvintseva, and G.G. Rizzotto. Soft computing for the intelligent robust control of a robotic unicycle with a new physical measure for mechanical controllability. In *Soft Computing*, volume 2, pages 73–88, June 1998.
- [13] D.W. Vos. *Dynamics and nonlinear adaptive control of an autonomous unicycle:theory and experiment*. PhD thesis, Massachusetts Institute of Technology, MIT.

- [14] D.W. Vos and A.H. Von Flotow. Dynamics and nonlinear adaptive control of an autonomous unicycle:theory and experiment. In *Decision and Control*, volume 1, pages 182–187, December 1990.

# Appendix A

## Contents of the CD

The CD contains all the files developed or generated during this project. In particular:

- This report in pdf-format
- The development files under Mathematica
- The development files as well as intermediate results generated automatically (by the development files) under Matlab. For further description of these files see Appendix [B](#)
- The papers presented in the state of the art of this report
- Some videos of the simulation results

# Appendix B

## Manual to the m-files

In this chapter I will give a quick overview of the developed m-files, the order and how to use them:

1. `Mod_Lag_development`: this file calculates the lagrangian of a submodule. If you want to study other actuators, add them here. The structure for each module is the same: first parameters and variables are defined, then translational and rotational kinetic energy and potential energy are calculated. Finally  $W_{kinrot} + W_{kintrans} - W_{pot}$  gives the complete submodule Lagrangian.  
If this file has run once, the module Lagrangian are saved in a separate m-file (`Modules.m`). Therefore it is not necessary to relaunch it for each new evaluation (as the development takes some time, this can be economized).
2. `dynamics`: This file derives the equations of motion (expressed in the matrix of inertia and the right-hand side of the equations). The constraint handling is done automatically. The final result is saved in a separate m-file.  
As a parameter you can indicate, what kind of model (brute, rotor, disk, inclined disk or combined actuation) you want to use. If you want to add a model, just add a case to the if's.
3. `Parametres_Physiques`: m-file containing the physical parameters.
4. `Linearization_exc_direct`: This file does the linearization of the model (so that it can be used afterwards to design a lqr-controller). This linearization needs to be done line by line (as else the memory space of Matlab fills up, making the programm crash) and the nominal state values are directly replaced. Thus the obtained matrices are completely numeric.  
Here again you need to chose the number corresponding to your model (so to call the right m-file containing the non-linear model)
5. `Control`: Now the lqr-controller can be calculated. Again choose the right number of your model (save the feedback coefficients to the right place) and choose if you want to reduce the model. (All models need reduction, thus this could also be automated...)
6. `Main`: Finally you can launch integration of your model with this file. Again chose the right model number. Then you can change the initial condition (X) and the time span of integration (TSPAN). The state evolution is automatically plotted in a graph and in a

rudimentary 3D-Simulator. The subsequent images of this simulators are put together and saved in “Unicycle\_movie.avi” in the Matlab-directory.

With the last parameter of the function called in ode45, the user can simulate in open (0) or in closed-loop (1) mode.

Further m-files have been developed that are not used for simulation or that the user at least doesn't call directly:

- UC\_affichage\_unicycle: This is the m-file that controls the simulation window, included the video generation.
- actuator\_extremes: This file calculates the maximum and minimum torques applied to the system
- reduce\_model: eliminates specific uncontrollable states, depending on the parameter.
- RPY and RPYinv: returns the rotation matrices for coordinate transformation toward fixed reference or toward mobile reference (inv).
- Evaluation\_XY: function called by ode45 for the integration of the corresponding system
- Gov: calculates the matrix of controllability
- filesave: allows to save an analytical expression in a m-file
- verif\_rotation: file programmed to verify that the fact that the uncontrollable yaw with rotor actuator doesn't come from a bug in the calculation. In fact, the equation of motion of the unicycle in straight upright position is calculated. The only degrees of freedom are (angular) position and velocity of the unicycle (yaw:  $\phi, \dot{\phi}$ ) and the rotor ( $\delta, \dot{\delta}$ ).