Claptrap simulation and control.

Matthieu Vigne

November 13, 2019

Contents

1   Introduction

In order to design controllers for a complex system such as Claptrap, knowing the equation of motion and being able to simulate them is crucial. This document covers both of these aspects. In a first part, it presents the kinematics of the system, and how to simulate it easily, assuming perfect, non-slipping wheel-ground contact, using a dynamics library like *pinocchio*. In a second (upcomming) part, an analysis of these equations is performed, in order to design and test plausible controllers.

2   Claptrap Kinematics and Simulation

2.1   System overview

The dynamics of Claptrap are just a particular case of rigid body dynamics. Luckily, this is a very well studied problem, with efficient state-of-art algorithms available to solve this. In particular, Roy Featherstone's "Rigid Body Dynamics" is a well-known reference in the field ; the algorithms presented therein (and others) are all efficiently implemented in the open-source library *pinocchio*, that we will be using for doing the simulation.

    All that we need to apply these algorithms is to give them a description of the kinematics (i.e. geometry) of the system, and the relevant dynamic
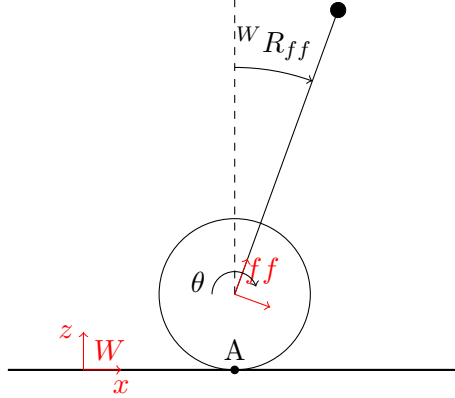
Figure 1: Claptrap kinematics.

parameters (mass and inertia of each rigid body constituting the system) : all this can be done using a URDF model.

For the sake of illustration, we will consider a 2D version of Claptrap without arms: indeed, the only difficulty comes from ground contact and freeflyer consideration ; adding internal degrees of freedom is not a problem at all. Then, Claptrap reduces to a simple pendulum on wheel system, as depicted in Figure 1.

Let θ be the angle of the wheel with respect to the bar. Note that this is a relative angle, and not the angle of the bar with respect to the ground. Obviously this variable is not enough to characterize the motion of the system: being a mobile base system, six extra degrees of freedom (three in rotation, three in translation) characterize its position is space. The freeflyer can be seen as representing the pose of any point in the system: for simplicity, we will take here the center of the wheel as origin, with orientation matching the upper link bar: this is depicted as the ff (freeflyer) frame in Figure 1

We call $\boldsymbol{q}_{ff}$ the coordinates of the freeflyer. While only six degrees of freedom are present, in practice a rotation is represented in pinocchio using a quaternion: thus $\boldsymbol{q}_{ff}$ is a vector of dimension 7 ; its derivative however, $\boldsymbol{v}_{ff}$, needs only to contain six components. In detail (but this is not really important), these vectors write:

$$\boldsymbol{q}_{ff} \triangleq \begin{pmatrix} {}^{W}p_{ff} \\ {}^{W}q_{ff} \end{pmatrix} \qquad \boldsymbol{v}_{ff} \triangleq \begin{pmatrix} {}^{ff}q_{W}{}^{W}\dot{p}_{ff} \\ \boldsymbol{\omega}_{ff} \end{pmatrix} \tag{1}$$

where $\boldsymbol{\omega}_{ff}$ is the angular velocity associated to ${}^{W}R_{ff}$, i.e. ${}^{W}\dot{R}_{ff} = {}^{W}R_{ff}[\boldsymbol{\omega}_{ff}]_{\times}$ with $[.]_{\times}$ the skew-symmetric (cross product) operator.

The generalized coordinates describing this them are simply obtained by concatenating the freeflyer with the internal joints. Note that in prac-

2

tice, the internal joints structure is given by the URDF, and the freeflyer is added when loading the model in *pinocchio*: this is because these joints are slightly different in nature, the URDF means to characterized kinematic relationships inside a system, and the freeflyer encodes the relationship with the outside. In our example, we simply have

$$
\boldsymbol{q} \triangleq \begin{pmatrix} \boldsymbol{q}_{ff} \\ \theta \end{pmatrix} \qquad \boldsymbol{v} \triangleq \begin{pmatrix} \boldsymbol{v}_{ff} \\ \dot{\theta} \end{pmatrix} \tag{2}
$$

2.2 Implementing ground contact

Lagrange's laws of motion tell us that the inverse dynamics of such a system writes

$$
H(\boldsymbol{q})\dot{\boldsymbol{v}} + C(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{f}_{ext} \tag{3}
$$

where H is the joint-space inertia matrix, C the matrix representing Coriolis and centrifugal forces, $\boldsymbol{g}$ gravity, and $\boldsymbol{F}_{ext}$ the vector of external forces. These forces obviously include motor torques, but also ground forces that we need to consider here.

To represent ground contact, we make the following assumption:

Assumption 1 *The contact between the ground and the wheel is assumed to be punctual, happening at point A. Furthermore, we assume that the wheel does not slip on the ground. Finally, we only consider flat ground.*

Ideal contact (i.e. no slippage at point A) implies that the velocity of point A is zero. The position of point A in turn is function of $\boldsymbol{q}$: thus, this constraint takes the generic form $\boldsymbol{g}(\boldsymbol{q}) = 0$. Let $J(\boldsymbol{q})$ be the Jacobian of $\boldsymbol{g}$. This Jacobian appears in two important places:

- first, it enables us to rewrite the non-linear constraint on $\boldsymbol{q}$ as a linear constraint on its derivative, $J(\boldsymbol{q})\boldsymbol{v} = 0$, and then on its second derivative:

$$
J(\boldsymbol{q})\dot{\boldsymbol{v}} + \dot{J}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} = 0 \tag{4}
$$

- second, its transpose links the ground reaction force to the vector of generalized forces $\boldsymbol{f}_{ext}$ in (3). More specifically, $\boldsymbol{f}_{ext}$ is the sum of two effects: the motor torque $\boldsymbol{tau}$ applied on the joints, and the ground reaction force $\boldsymbol{f}_g$ (a vector in the Cartesian space) projected onto the joints using $J^T(\boldsymbol{q})$:

$$
\boldsymbol{f}_{ext} \triangleq J^T(\boldsymbol{q})\boldsymbol{f}_g + \boldsymbol{\tau} \tag{5}
$$

Then, using (3), (4) and (5), the constraint equation of dynamics writes:

$$
\begin{aligned}
H(\boldsymbol{q})\dot{\boldsymbol{v}} + C(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = J^T(\boldsymbol{q})\boldsymbol{f}_g + \boldsymbol{\tau} \\
J(\boldsymbol{q})\dot{\boldsymbol{v}} + \dot{J}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} = 0
\end{aligned} \tag{6}
$$

This is a coupled system of equations of unknown $\dot{\boldsymbol{v}}$ and $\boldsymbol{f}_g$ (the state and input torque $\boldsymbol{q}$, $\boldsymbol{v}$, $\boldsymbol{\tau}$ being given).

This system is easily solvable, as it is in fact linear:

$$
\begin{pmatrix} H(\boldsymbol{q}) & J^T(\boldsymbol{q}) \\ J(\boldsymbol{q}) & 0 \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{v}} \\ \boldsymbol{f}_g \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} - C(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{q}) \\ -\dot{J}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} \end{pmatrix} \tag{7}
$$

and the first matrix is invertible, granted that $J(\boldsymbol{q})$ is full-ranked - in other words, that the constraints are linearly independent. This is the case in practice for a well-posed problem - indeed it not being full-ranked implies either an unsolvable problem (over-constraint), or that one of the constraint is a linear combination of the others and can thus be removed.

A kinematics and dynamics library like pinocchio directly gives us the right-hand side vector, and the joint space inertia matrix $H(\boldsymbol{q})$. In our particular case however, it won't directly give us $J(\boldsymbol{q})$, as point A is not a fixed point in the structure: pinocchio can easily compute the jacobian of any point defined in the URDF, but A is a point moving along the wheel rim. We will thus compute here this Jacobian by hand.

The position of point A in the freeflyer frame ff is $R_{ff}^T \begin{pmatrix} 0 \\ 0 \\ -r \end{pmatrix}$ where r is the radius of the wheel, thus its velocity writes

$$
\boldsymbol{v}_A = v_{ff} + (\boldsymbol{\omega}_{ff} + \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix}) \times R_{ff}^T \begin{pmatrix} 0 \\ 0 \\ -r \end{pmatrix} \tag{8}
$$

Thus, the Jacobian writes

$$
J(\boldsymbol{q}) = \begin{pmatrix} I_3 & \left[R_{ff}^T \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \right]_{\times} & \left[R_{ff}^T \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \right]_{\times} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \end{pmatrix} \tag{9}
$$

where $[]_{\times}$ is the skew-symmetric operator. Notice that through the first identity term, this matrix is indeed full-ranked. Finally, its derivative writes

$$
\dot{J}(\boldsymbol{q}, \boldsymbol{v}) = \begin{pmatrix} 0_3 & \left[-[\boldsymbol{\omega}_{ff}]_{\times} R_{ff}^T \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \right]_{\times} & \left[-[\boldsymbol{\omega}_{ff}]_{\times} R_{ff}^T \begin{pmatrix} 0 \\ 0 \\ r \end{pmatrix} \right]_{\times} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \end{pmatrix} \tag{10}
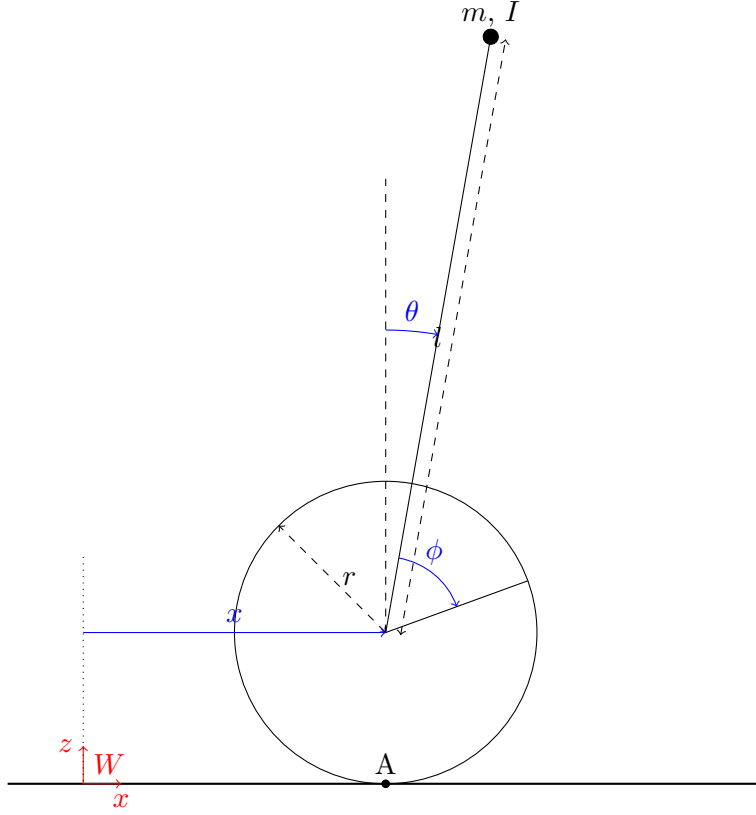$$

We now have all we need to implement a simulator !

Figure 2: Sagittal dynamics of Claptrap

3 2D sagittal dynamics: a Segway model

In order to design a controller for this system, we still want to know its equation of dynamics. Assuming we can decouple motion in the sagittal and frontal plane, we here study the dynamics of the sagittal (forward) motion, that is, the dynamics of a Segway.

3.1 Computing the equation of dynamics

Figure 2 shows the parametrization of the system. We consider a two-body system: a wheel of radius r, mass m_w and inertia J linked to a body of mass m, length l and inertia (about its center of mass) I. We call θ the angle of the body, ϕ the angle of rotation of the wheel from the body (that is, the angle read by an encoder placed on the wheel motor) and x the cartesian position of the center of the wheel.

Assuming like before a non-slip hypothesis, we have the following con-

straint:

$$\dot{x} = r(\dot{\theta} + \dot{\phi}) \tag{11}$$

The Jacobian associated to this constraint is $J \triangleq \begin{pmatrix} 1 & -r & -r \end{pmatrix}$. The kinetic and potential energy for this system writes:

$$K = \frac{1}{2} \left(m_r \dot{x}^2 + J(\dot{\theta} + \dot{\phi})^2 + I\dot{\theta}^2 + m(\dot{x}^2 + 2l\cos\theta \dot{x}\dot{\theta} + l^2\dot{\theta}^2) \right) \tag{12}$$

$$E_p = mlg\cos\theta$$

Let λ be the tangent ground reaction force applied at contact point A, and τ the motor torque input. The Lagrange equation of dynamics for this system then writes:

$$\begin{pmatrix} m + m_w & lm\cos(\theta) & 0 \\ lm\cos(\theta) & l^2 m + I + J & J \\ 0 & J & J \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{\theta} \\ \ddot{\phi} \end{pmatrix} + \begin{pmatrix} -\dot{\theta}^2 lm\sin(\theta) \\ -glm\sin(\theta) \\ 0 \end{pmatrix} = J^T\lambda + \begin{pmatrix} 0 \\ 0 \\ \tau \end{pmatrix} \tag{13}$$

Because the constraint (11) is so simple, we can actually easily inject it in the previous equation, to remove the redundant variable x, together with the force parameter λ. This yields the following constraint dynamics:

$$H(\theta) \begin{pmatrix} \ddot{\theta} \\ \ddot{\phi} \end{pmatrix} - \begin{pmatrix} \left(\dot{\theta}^2 lmr + glm \right) \sin(\theta) \\ \dot{\theta}^2 lmr \sin(\theta) \end{pmatrix} = \begin{pmatrix} 0 \\ \tau \end{pmatrix} \tag{14}$$

with:

$$H(\theta) \triangleq \begin{pmatrix} 2\,lmr\cos(\theta) + l^2 m + (m + m_w)r^2 + I + J & lmr\cos(\theta) + (m + m_w)r^2 + J \\ lmr\cos(\theta) + (m + m_w)r^2 + J & (m + m_w)r^2 + J \end{pmatrix} \tag{15}$$

Linearizing the dynamics about $\theta = 0$, $\dot{\theta} = 0$, we obtain the following dynamics:

$$H_{lin} \begin{pmatrix} \ddot{\theta} \\ \ddot{\phi} \end{pmatrix} = \begin{pmatrix} mlg\theta \\ \tau \end{pmatrix} \tag{16}$$

with:

$$H_{lin} \triangleq \begin{pmatrix} 2\,lmr + l^2 m + (m + m_w)r^2 + I + J & lmr + (m + m_w)r^2 + J \\ lmr + (m + m_w)r^2 + J & (m + m_w)r^2 + J \end{pmatrix} \tag{17}$$

Note that here, we choose to write the dynamics in terms of ϕ, the encoder readings. It may be interesting also to write it in terms of $\psi \triangleq \theta + \phi = \frac{x}{r}$: indeed, ψ represents the forward motion of the system (for

a static, standstill equilibrium, we want $\theta = 0$ and $\psi = cte$). With this convention, (17) takes the following simpler form:

$$\begin{pmatrix} ml^2 + I & lmr \\ lmr & (m + m_w)r^2 + J \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} mlg\theta - \tau \\ \tau \end{pmatrix} \tag{18}$$

3.2 Canonical form and controllability

Let us now study this equation of dynamics (18). The first thing one can notice is that we have two outputs, θ and ψ, but only one input, the torque τ. Is this one input enough to actually control both outputs ? This question is called controllability, and is easily answered by putting the system in a canonical form.

Let us define the following state vector:

$$\boldsymbol{x} \triangleq \begin{pmatrix} \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{pmatrix} \tag{19}$$

Then (18) rewrites as a linear time-invariant first order system:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\tau \tag{20}$$

with

$$A \triangleq \begin{pmatrix} 0 & 1 & 0 & 0 \\ H_1^{-1}mlg & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ H_2^{-1}mlg & 0 & 0 & 0 \end{pmatrix} \qquad B \triangleq \begin{pmatrix} 0 \\ -H_1^{-1} + H_2^{-1} \\ 0 \\ -H_3^{-1} + H_4^{-1} \end{pmatrix} \tag{21}$$

where H_i^{-1} are the coefficients of the inverse of H in (18):

$$H^{-1} \triangleq \begin{pmatrix} H_1^{-1} & H_2^{-1} \\ H_3^{-1} & H_4^{-1} \end{pmatrix} \tag{22}$$

The Kalman criterion tells us that the system is *controllable* - that is, that given a starting point $\boldsymbol{x}_a$ and a target point $\boldsymbol{x}_b$, it is possible to find a input τ that brings the system from $\boldsymbol{x}_a$ to $\boldsymbol{x}_b$ - if and only if the commandability matrix $\mathcal{C}$ is full ranked, where:

$$\mathcal{C} \triangleq \begin{pmatrix} B & AB & A^2B & A^3B \end{pmatrix} \tag{23}$$

In this context, given that the physical parameters are all strictly positive, it is easy to check that this condition is satisfied granted that $H_1^{-1} \neq H_1^{-2}$, which translates into $J + (m + m_w)r^2 \neq mlr$. If we assume that the

body is much heavier that the wheel ($m >> m_w$), this simplifies into $l \neq r$, a condition obviously met in practice.

Thus, the system is indeed controllable: we can find control laws that will stabilize this segway. It's already a result we know of course (otherwise there wouldn't be segways around), but it's nice to check that the equations are indeed coherent.