Modelling a Cart

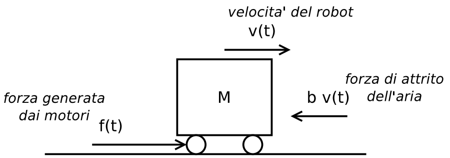
Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

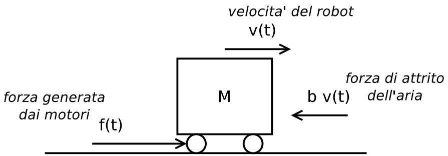
Robotic Systems

A Cart

- Let's consider a **cart** moving over a 1-dimensional space
- We suppose that is has a mass M and is equipped with electric motors
- At the instant $T = 0$ we turn-on the motors, so they provide a certain **traction force** $f(t)$
- What is the behaviour of the cart?
- What is the trend of the speed over time?
- What is the trend of the position over time?



Modelling the Cart

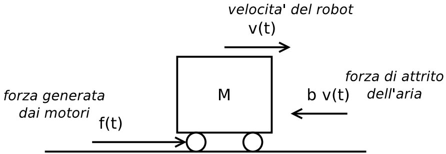


The Second Principle of Dynamics

Let us use the second principle of dynamics: the sum of the forces in a rigid body is equal to the mass' body times the acceleration.

$$\sum F(t) = M\, a(t)$$

Modelling the Cart



$$f(t) - b\, v(t) = M\, a(t)$$

$$f(t) - b\, v(t) = M\, \frac{dv(t)}{dt}$$

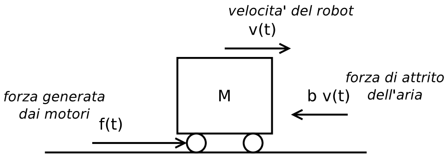
The “Dotted” Notation

Let’s introduce the “dotted notation” widely used in physics

Given $x(t)$ as a function of time, we denote:

$$
\begin{aligned}
\dot{x} &= \frac{dx(t)}{dt} \\
\ddot{x} &= \frac{d^2x(t)}{dt^2} \\
&\ldots
\end{aligned}
$$

Modelling the Cart



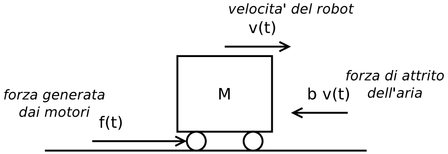
$$f(t) - b\,v(t) = M\,a(t)$$

$$f(t) - b\,v(t) = M\,\frac{dv(t)}{dt}$$

$$f - b\,v = M\,\dot{v}$$

Here **f** is the input (stimulus) and **v** is the output (effect)

Modelling the Cart



Since we know that **speed** is the derivative of **position (space)**, we can write:

$$\begin{cases} \dot{v} = -\frac{b}{M}v + \frac{1}{M}f \\ \dot{p} = v \end{cases}$$

It is a **system of first-order differential equation** that, if solved analytically, gives the analytical expression of v and p as a function of time and stimulus $f(t)$:

$$v = f1(t, f(t))$$
$$p = f2(t, f(t))$$

But... the analytical way is the sole solution way that we can considered?

Approximating the Cart

We know that:

$$f'(t) \quad = \quad \lim_{\Delta t \to 0} \frac{f(t + \Delta t) + f(t)}{\Delta t}$$

If we consider a Δt **really small** we can do the following approximation:

$$f'(t) \quad \simeq \quad \frac{f(t + \Delta t) + f(t)}{\Delta t}$$

Modelling the Cart

Using the approximation, the model:

$$\begin{cases} \dot{v} & = & -\frac{b}{M} v + \frac{1}{M} f \\ \dot{p} & = & v \end{cases}$$

becomes:

$$\begin{cases} \frac{v(t+\Delta t) - v(t)}{\Delta T} & = & -\frac{b}{M} v(t) + \frac{1}{M} f(t) \\ \frac{p(t+\Delta t) - p(t)}{\Delta T} & = & v(t) \end{cases}$$

in other words

$$\begin{cases} v(t+\Delta t) & = & (1 - \frac{b \Delta T}{M}) v(t) + \frac{\Delta T}{M} f(t) \\ p(t+\Delta t) & = & p(t) + v(t) \Delta T \end{cases}$$

If we consider a discretisation (or sampling) using the time interval Δt, these relations give the value of v and p at the **next time interval**, or the **updating laws for v and p**

The last relations can be implemented in software in order to simulate the behaviour of the cart

Implementing the Cart

Our cart is an **object** (so we define a class) that has some properties, speed and position, and a behaviour, the updating laws:

```
class Cart:


    def __init__(self, _mass, _friction):
        self.M = _mass
        self.b = _friction
        self.speed = 0
        self.position = 0


    def evaluate(self, delta_t, _force):
        new_speed = (1 - self.b * delta_t / self.M) * self.speed + \
                         delta_t * _force / self.M
        new_position = self.position + self.speed * delta_t

        self.speed = new_speed
        self.position = new_position
```

Each time we call the method **`evaluate()`** the next values of v and p are computed, therefore a call to that method implies that a time interval elapsed

(see **`tests/test_cart.py`** and **`tests/test_cart_plot.py`**)

Modelling a Cart

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
`santoro@dmi.unict.it`

Robotic Systems