

The background features a dark blue gradient. At the bottom, there is a horizontal line representing a surface, with three wavy, light blue shapes below it, resembling water or a base. Above each wavy shape is an inverted pendulum. Each pendulum consists of a light blue oval bob connected by a thin line to a circular pivot point. The pendulums are positioned at different angles, suggesting motion. The title text is overlaid on the upper part of the image.

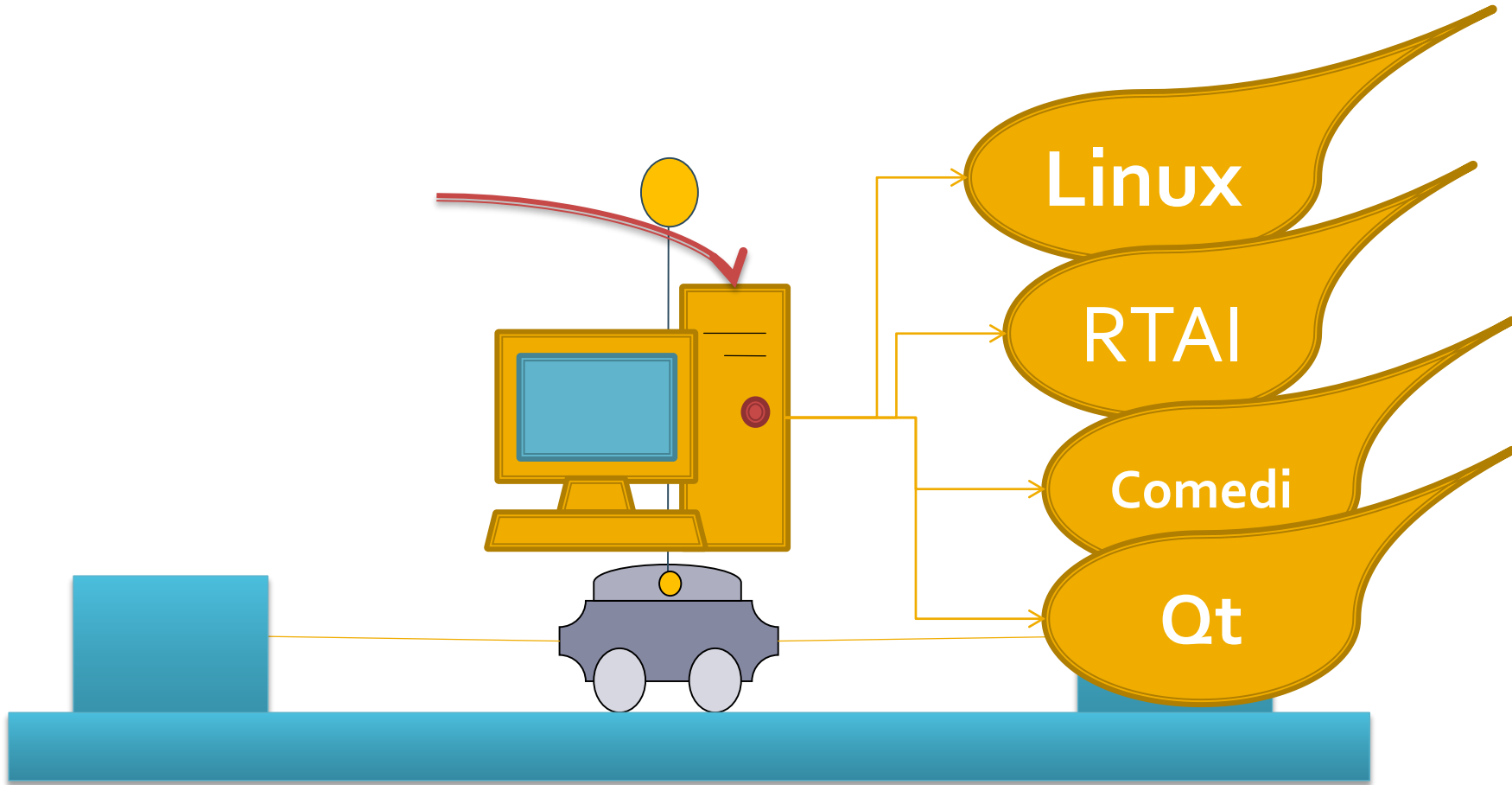
Controllo in RealTime di un pendolo inverso

Relatore: Paolo Righettini

Michele Pirovano
matricola 55270

Anno 2008/2009

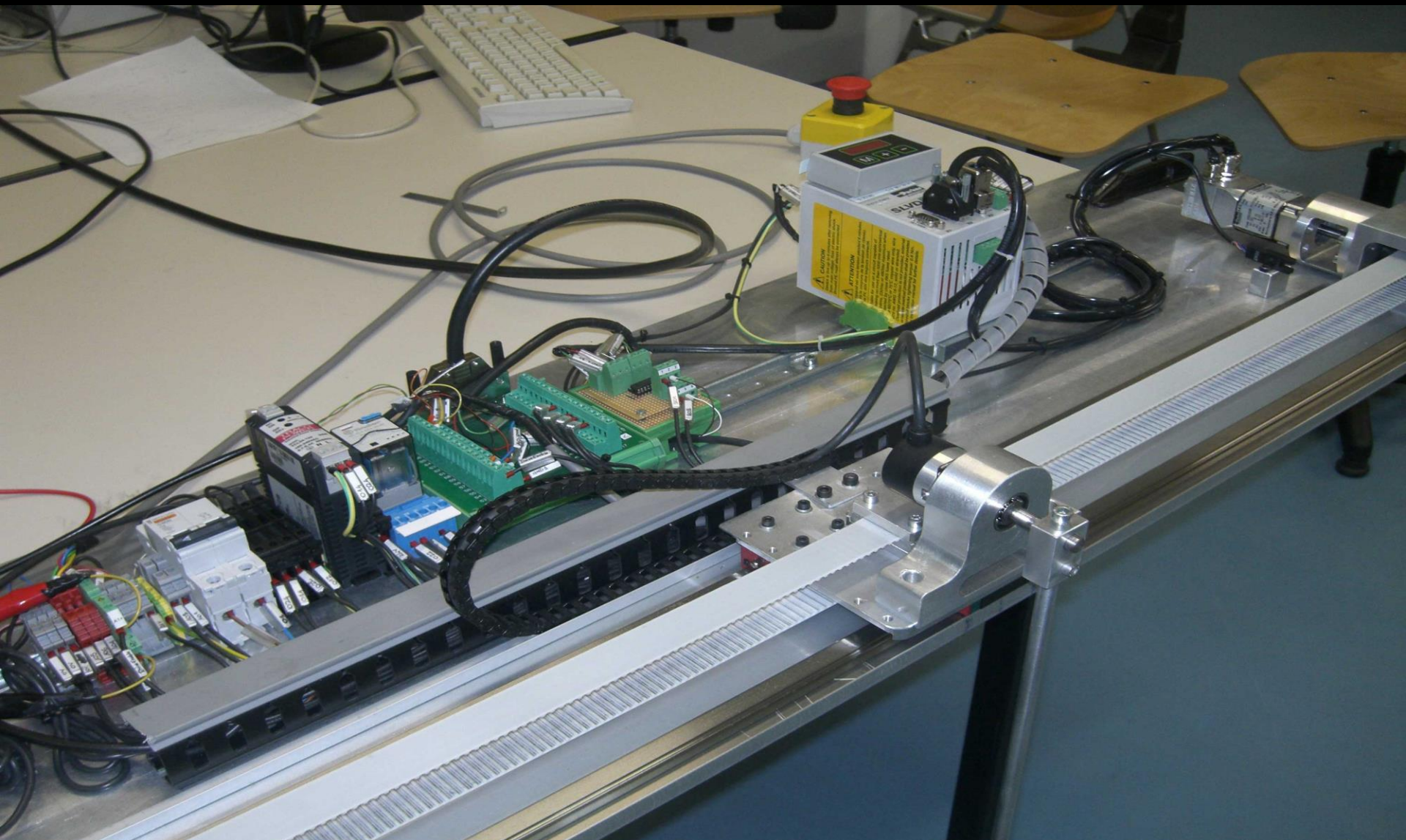
Il problema



Gli argomenti

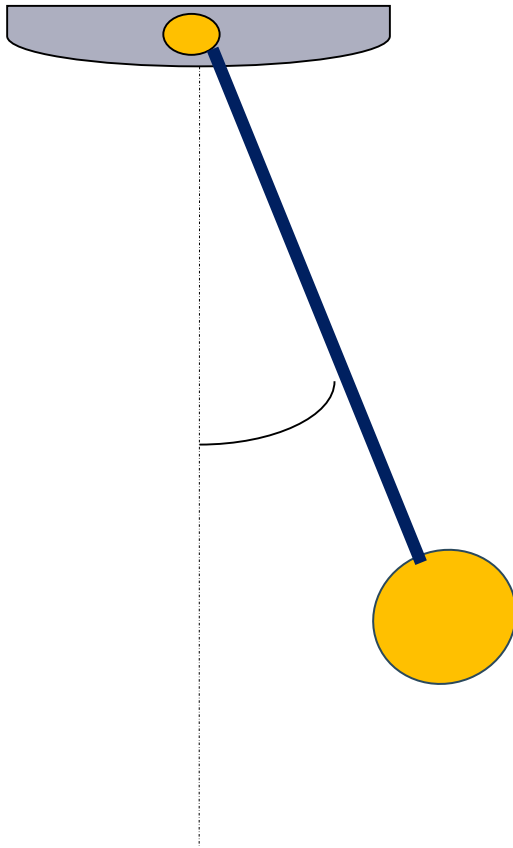
- Banco del pendolo
- Il modello
- Il controllore
- RTAI e Comedi
- Kernel Setup
- Implementazione controllore
- Interfaccia

Banco del pendolo

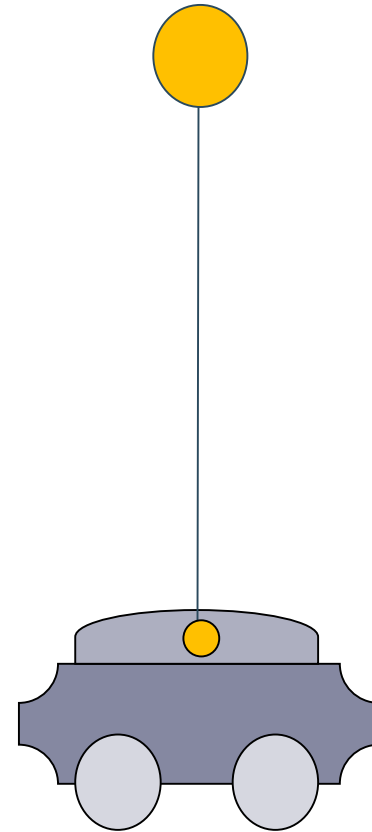


Il pendolo

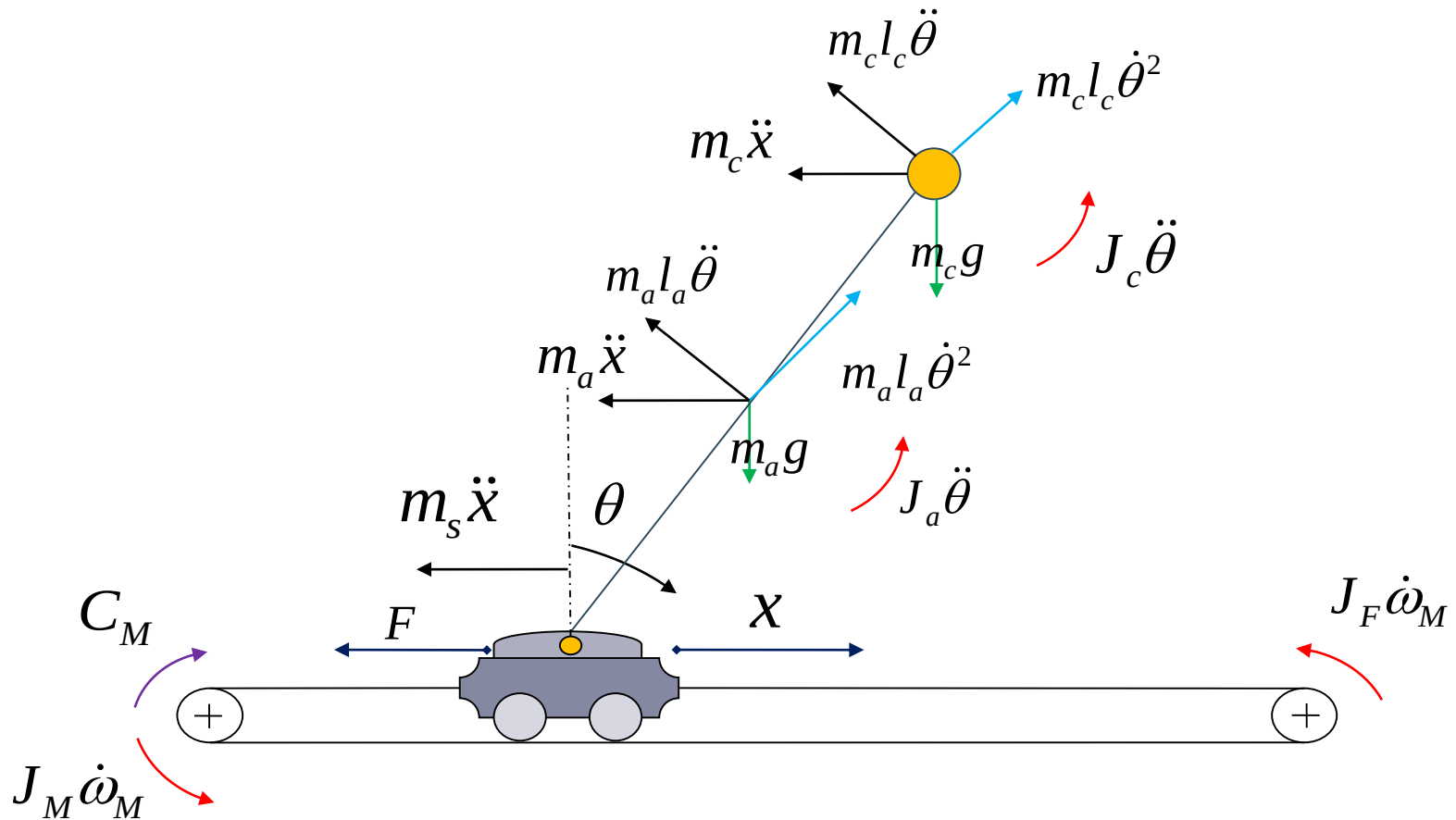
Pendolo semplice



Pendolo inverso



Il modello



Dinamica del sistema

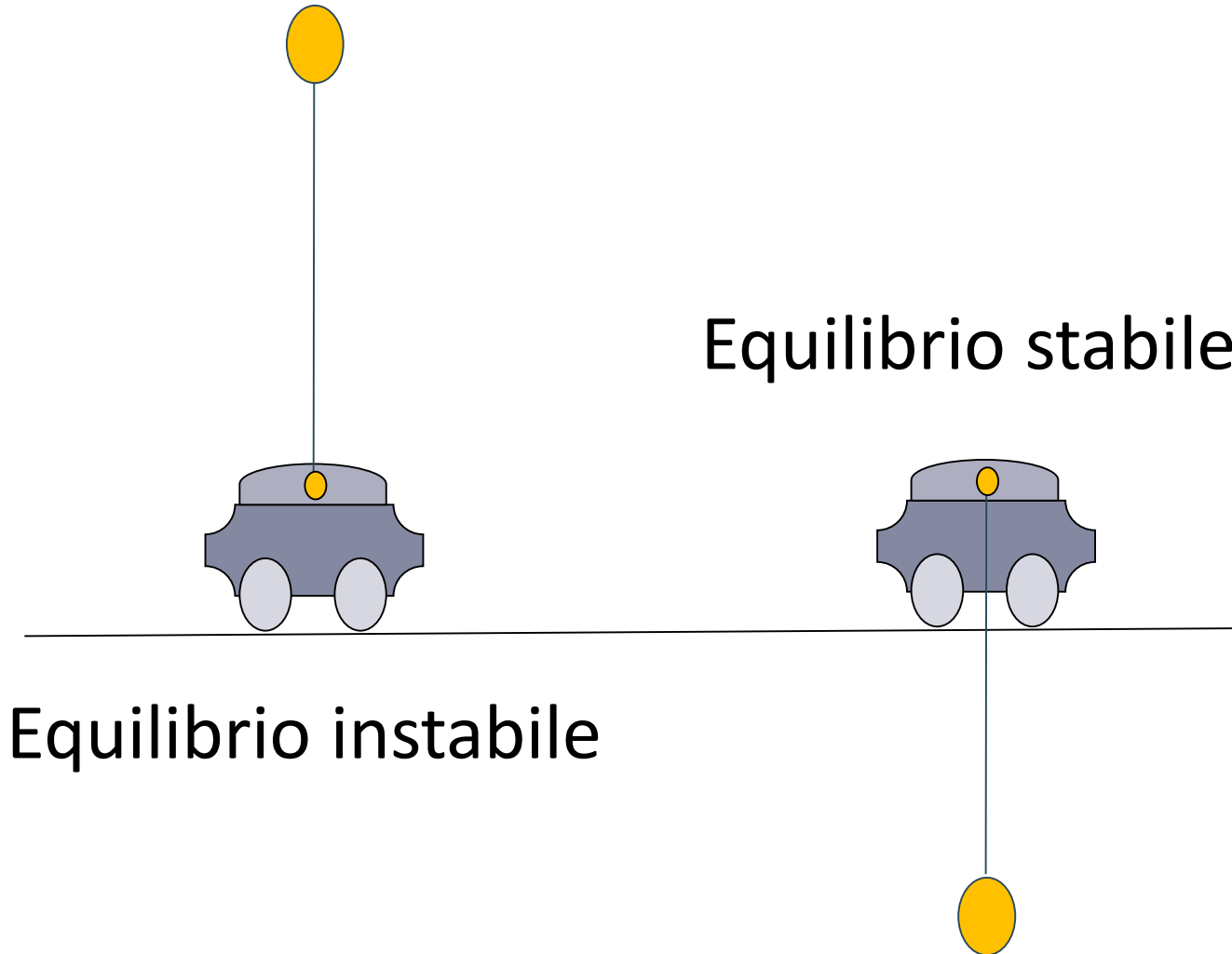
$$K_1 = m_a l_a^2 + J_a + m_c l_c^2 + J_c$$

$$K_2 = m_a l_a + m_c l_c$$

$$K_3 = m_s + m_a + m_c + \frac{J_M + J_F}{R^2}$$

$$\begin{bmatrix} K_1 & K_2 \cos \theta \\ K_2 \cos \theta & K_3 \end{bmatrix} \cdot \begin{bmatrix} \ddot{\theta} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} K_2 g \sin \theta \\ \frac{C_M}{R} + K_2 \dot{\theta}^2 \sin \theta \end{bmatrix}$$

Equilibrio



Linearizziamo...

$$\begin{bmatrix} K_1 & K_2 \\ K_2 & K_3 \end{bmatrix} \cdot \begin{bmatrix} \ddot{\theta} \\ \ddot{\chi} \end{bmatrix} = \begin{bmatrix} K_2 g \theta \\ \frac{C_M}{R} \end{bmatrix}$$

Matrici di stato

$$\mathbf{Z} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}$$
$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ C_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ C_3 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ C_2 \\ 0 \\ C_4 \end{bmatrix}$$

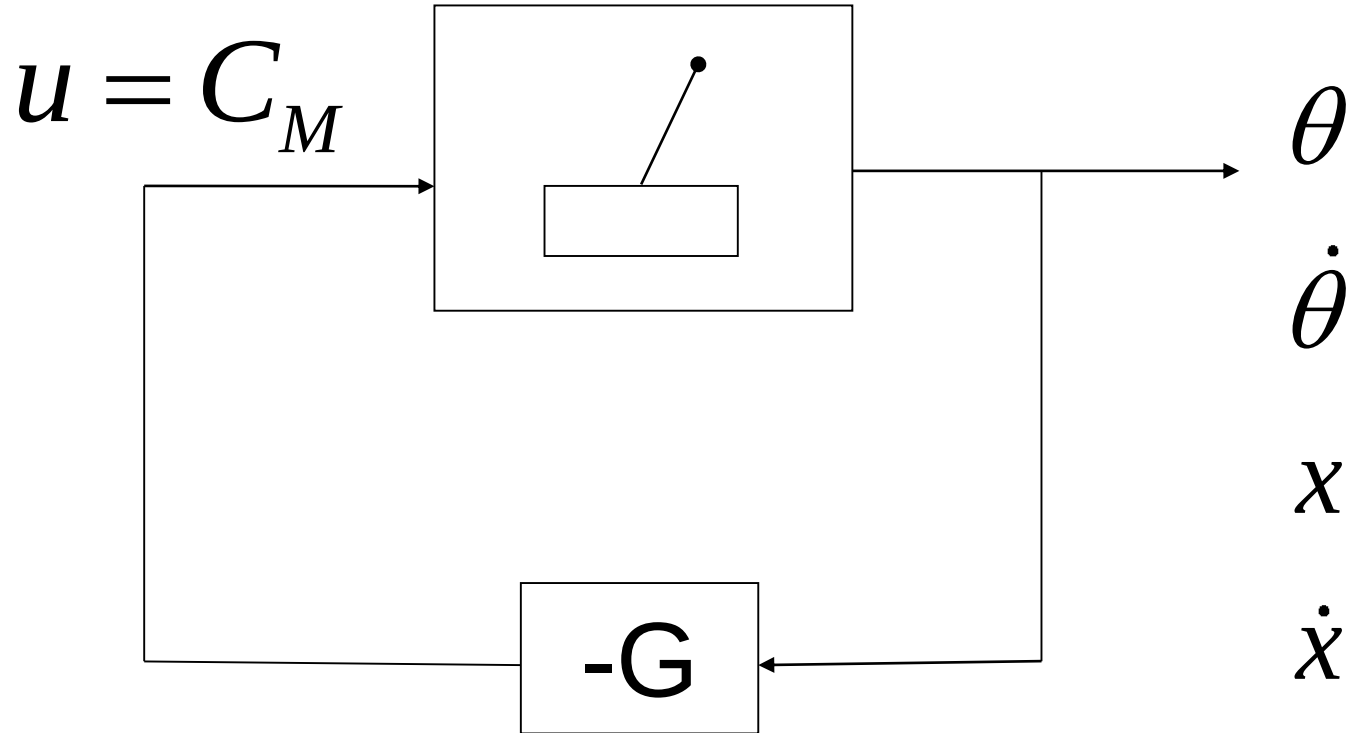
Stabilità dell'equilibrio

$$\det(sI - A) = \det \begin{bmatrix} s & -1 & 0 & 0 \\ -C_1 & s & 0 & 0 \\ 0 & 0 & s & -1 \\ -C_2 & 0 & 0 & s \end{bmatrix} = 0$$

INSTABILE

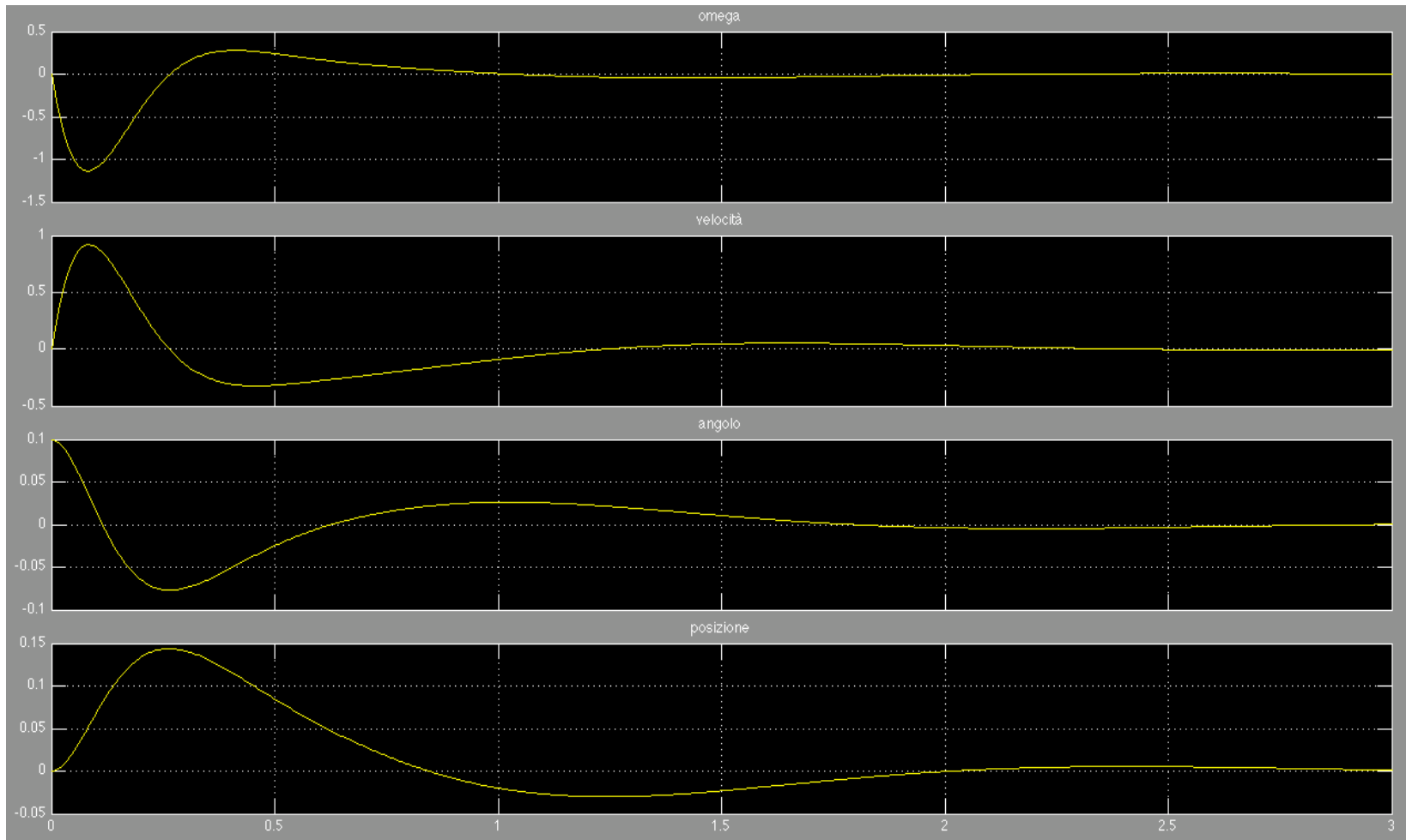
$$s_1 = 0; \quad s_2 = 0; \quad s_3 = \sqrt{C_1}; \quad s_4 = -\sqrt{C_1}$$

Controllore

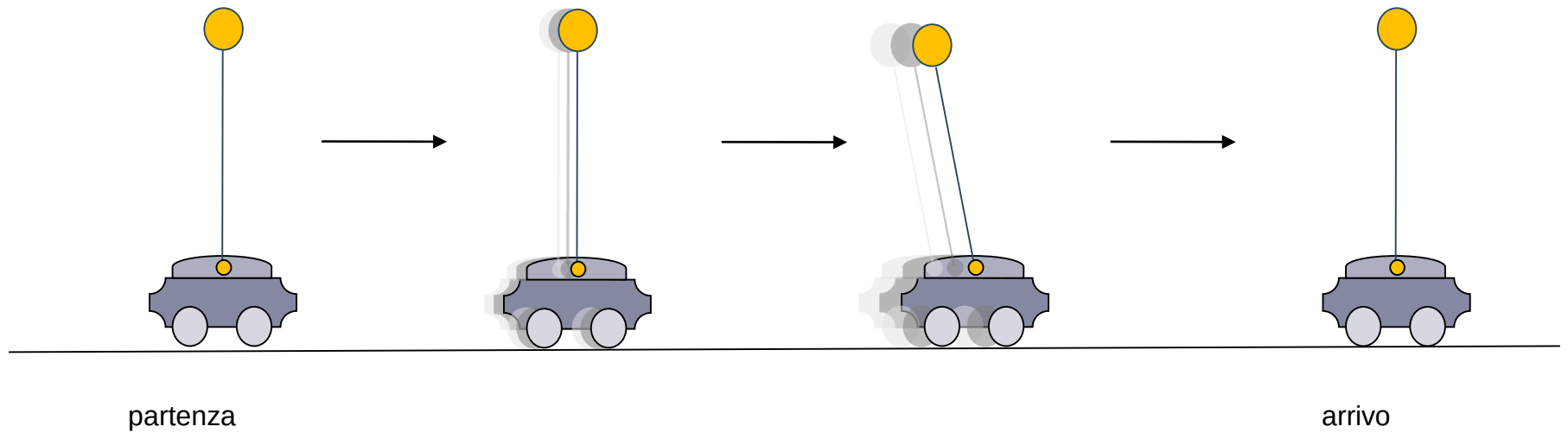


$$\dot{X} = AX + B(-GX)$$

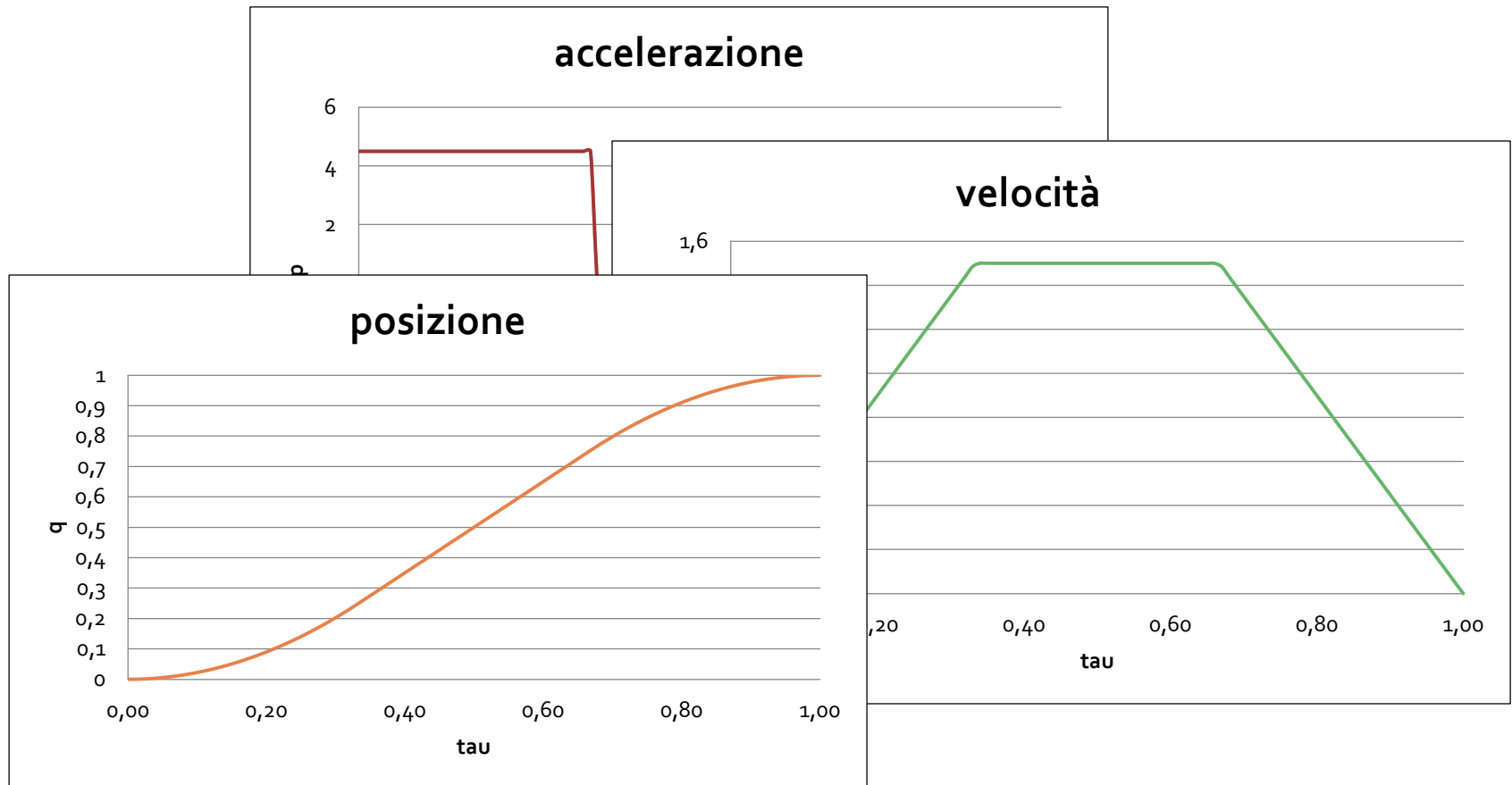
Simulazione



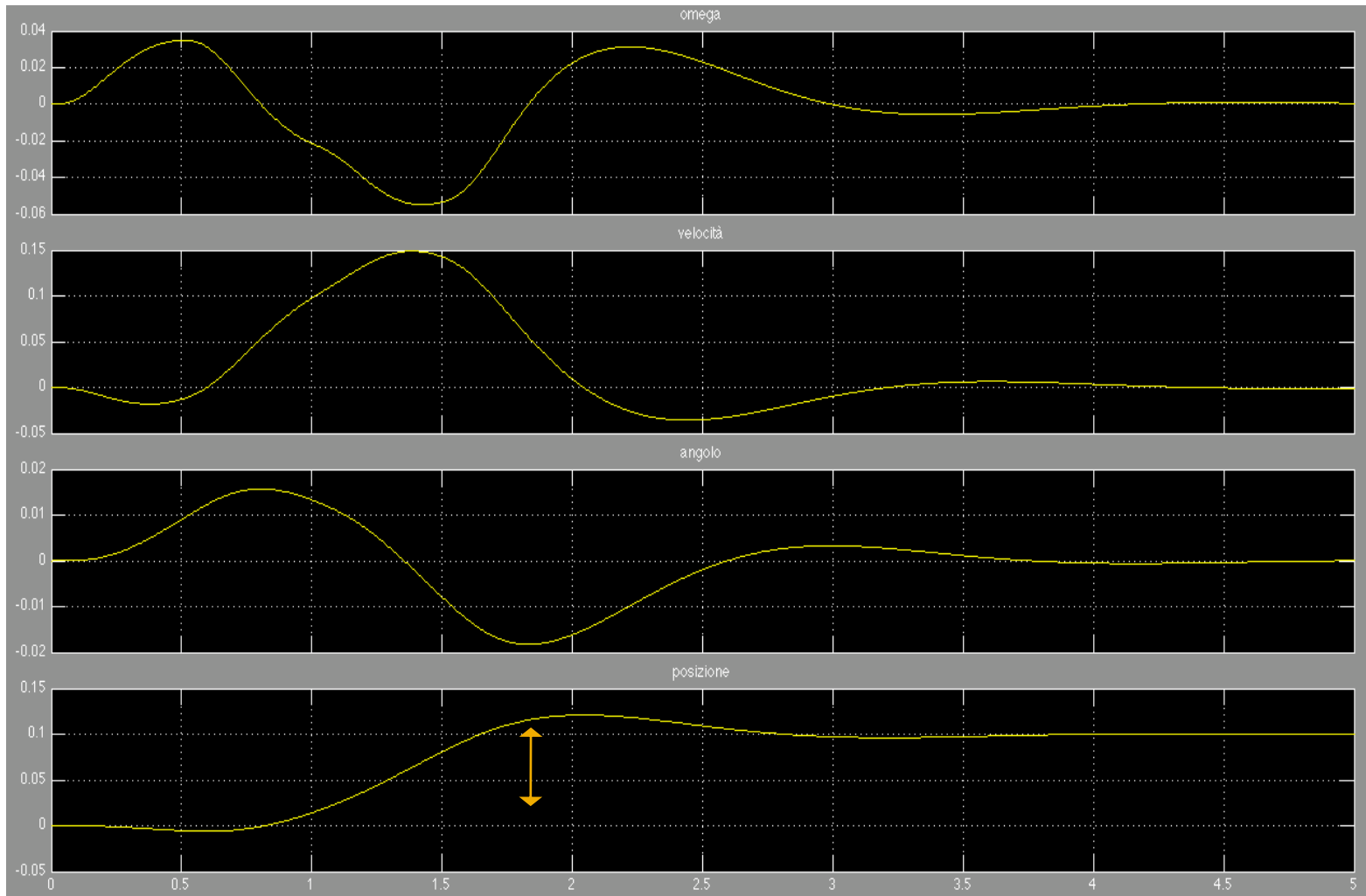
Movimento



Profili di moto



Simulazione



Implementazione

Kernel real-time

- RTAI
- Comedi
- Kernel Setup

Programma real-time

- Modulo pendolo
- Modulo controllo
- Modulo acquisizione

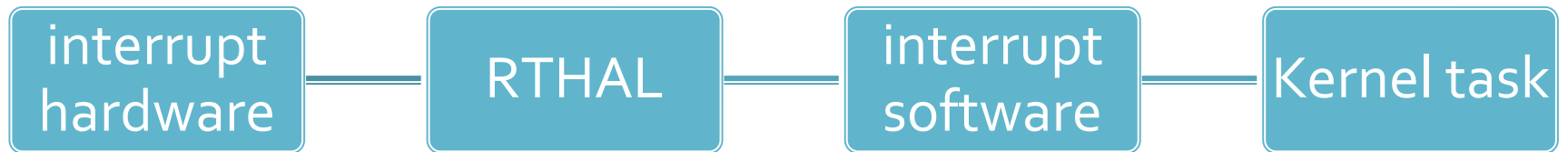
Interfaccia

- Grafica
- Librerie Qt
- Comunicazione

RTAI

RealTime Application Interface

RTAI



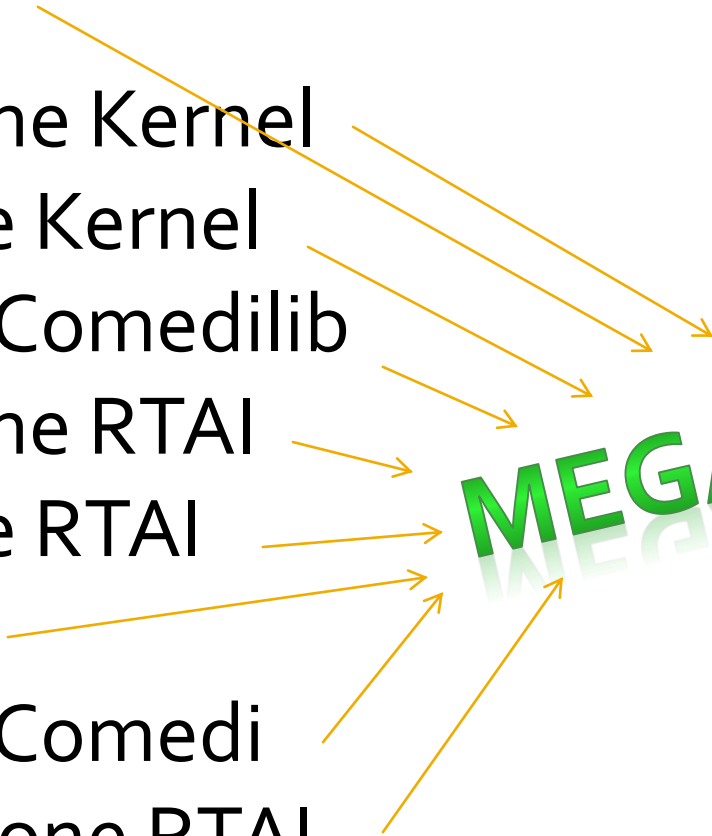
Comedi

Control and Measurement Device Interface

Kernel Setup

- Patch RTAI
- Configurazione Kernel
- Compilazione Kernel
- Installazione Comedilib
- Configurazione RTAI
- Compilazione RTAI
- Test Kernel
- Installazione Comedi
- Riconfigurazione RTAI

MEGABATCH

A diagram consisting of nine orange arrows pointing from the list items to the word 'MEGABATCH'. The arrows originate from the following items: 'Patch RTAI', 'Configurazione Kernel', 'Compilazione Kernel', 'Installazione Comedilib', 'Configurazione RTAI', 'Compilazione RTAI', 'Test Kernel', 'Installazione Comedi', and 'Riconfigurazione RTAI'. The word 'MEGABATCH' is written in a large, bold, green, sans-serif font, tilted at an angle, and has a faint reflection below it.

Moduli Kernel

pendolo

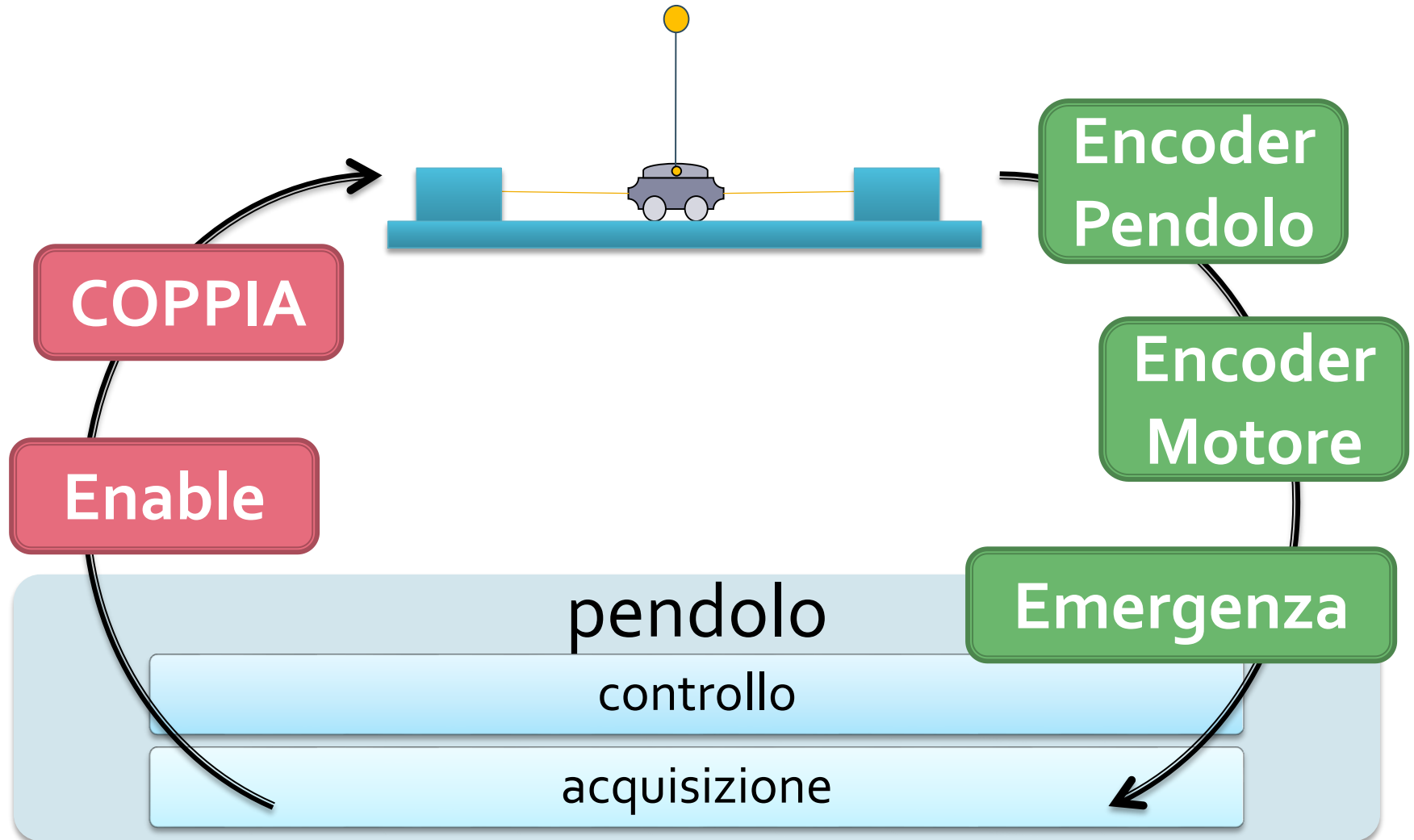
controllo

acquisizione

Moduli Kernel

- **pendolo** – modulo principale, contiene le funzioni di gestione dei task real-time
- **controllo** – modulo che implementa il controllore
- **acquisizione** – modulo che si occupa dell'acquisizione dei dati

Segnali



Modulo Pendolo

- `init_module(void)`
 - `rt_set_periodic_mode();`
 - `rt_task_init(&task, task_handler, 1, STACK_SIZE, TASK_PRIORITY, 1, 0);`
 - `tick_period=start_rt_timer(nano2count(TICK_PERIOD));`
 - `rt_task_make_periodic(&task, rt_get_time() + tick_period, tick_period);`
- `cleanup_module(void)`
- `task_handler(int t)`

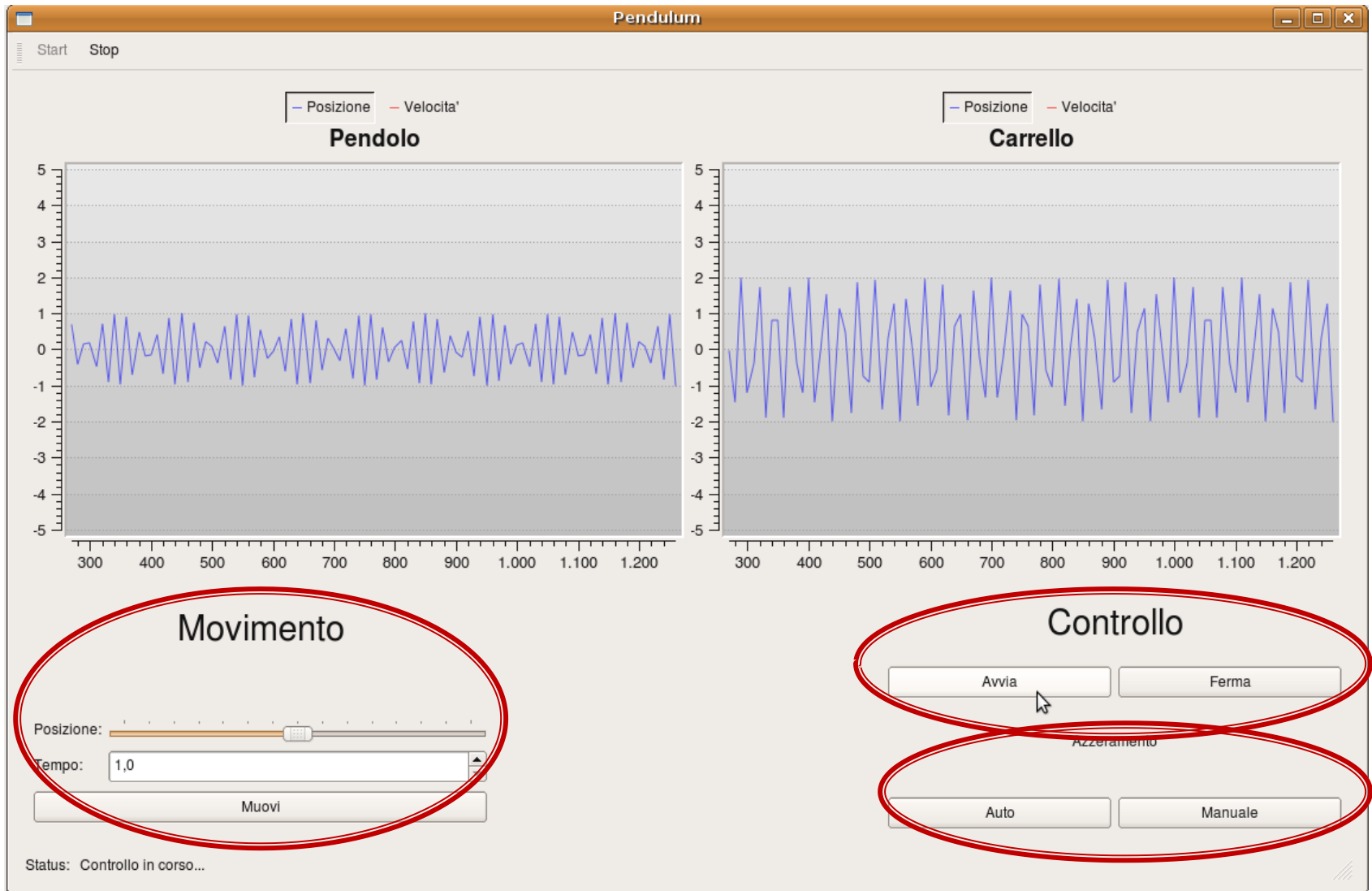
Modulo Controllo

- controllo (Lettura lettura, Scrittura* scrittura)
- getStato(&pos, &vel, &ang, &ome, lettura.m_enc, lettura.p_enc)
- getFeedbackCoppia(pos, vel, ang, ome, &coppia)
- filter(Filter f, double u[3], double y[3])

Modulo Acquisizione

- lettura_dati(&lettura);
- scrittura_dati(&scrittura);
- **Funzioni kcomedilib**
 - instr.insn = INSN_READ;
 - instr.n = 1;
 - instr.data = &val;
 - instr.subdev = m_enc_subdev;
 - instr.chanspec=CR_PACK(m_enc_chan,range,aref);
 - ret = comedi_do_insn(device, &instr);

Interfaccia Grafica



librerie Qt

```
void MainWindow::setUpGui()
{
    // Creazione Pannelli
    createToolBar();
    createStatusBar();
    createPlotPanel();
    createControlPanel();
    createMovePanel();
    createZeroPanel();

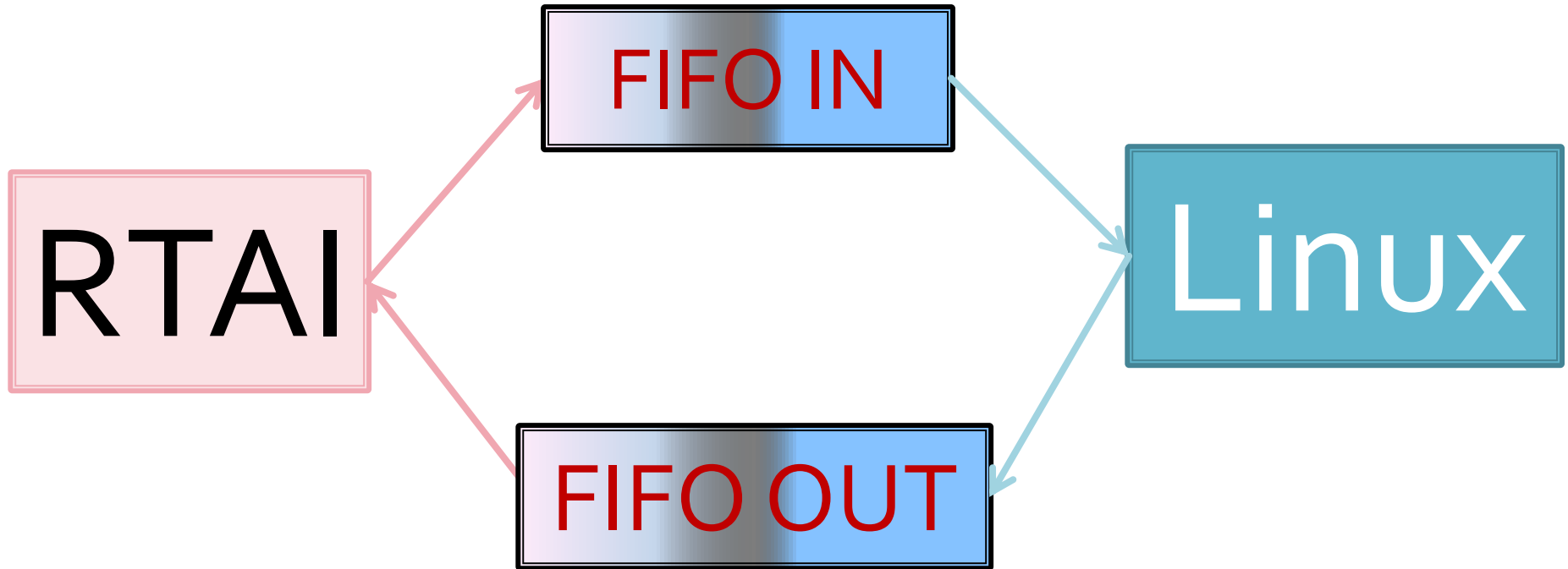
    // Creazione Main Widget
    QWidget *mainWidget = new QWidget;
    this->setCentralWidget(mainWidget);

    // Disposizione Componenti
    QGridLayout *guiBox = new QGridLayout;
    QSpacerItem *spacer = new QSpacerItem(300,100,QSizePolicy::Minimum,QSizePolicy::Minimum);
    guiBox->addWidget(plotPanel, 0, 0, 1, 3);
    guiBox->addWidget(movePanel, 1, 0, 2, 1);
    guiBox->addItem(spacer, 1, 1, 2, 1);
    guiBox->addWidget(controlPanel, 1, 2);
    guiBox->addWidget(zeroPanel, 2, 2);
    guiBox->addWidget(statusBar, 3, 0, 1, 3);

    guiBox->setSpacing(10);
    guiBox->columnStretch(2);

    mainWidget->setLayout(guiBox);
}
```

Comunicazione



Grazie per l'attenzione!

