Qui c'è il notebook di Mathematica in cui ho fatto i calcoli per il pendolo singolo:
https://www.wolframcloud.com/obj/matteobonna/Published/Calcoli.nb

Singolo pendolo

Scrivo la lagrangiana e trovo le equazioni del moto. Questa parte sono solo calcoli, li ho fatti con mathematica:

```
CM0 = {x, 0};
CM1 = CM0 + {l1cm * Sin[θ], l1cm * Cos[θ]};

V0 = D[CM0, t];
V1 = D[CM1, t];

T = 1 / 2 * (M * V0.V0 + m1 * V1.V1 ) + 1 / 2 * (I1 * wt * wt );
V = g * (m1 * CM1[[2]]);

L = T - V;

eq1 = (D[D[L, v], t] - D[L, x] == u) // Simplify;
eq2 = (D[D[L, wt], t] - D[L, θ] == -η1 * wt) // Simplify;

motionEquations = Solve[eq1 && eq2, {x''[t], θ''[t]}][[1]] // Simplify;
```

La lagrangiana è questa

raditionalForm=

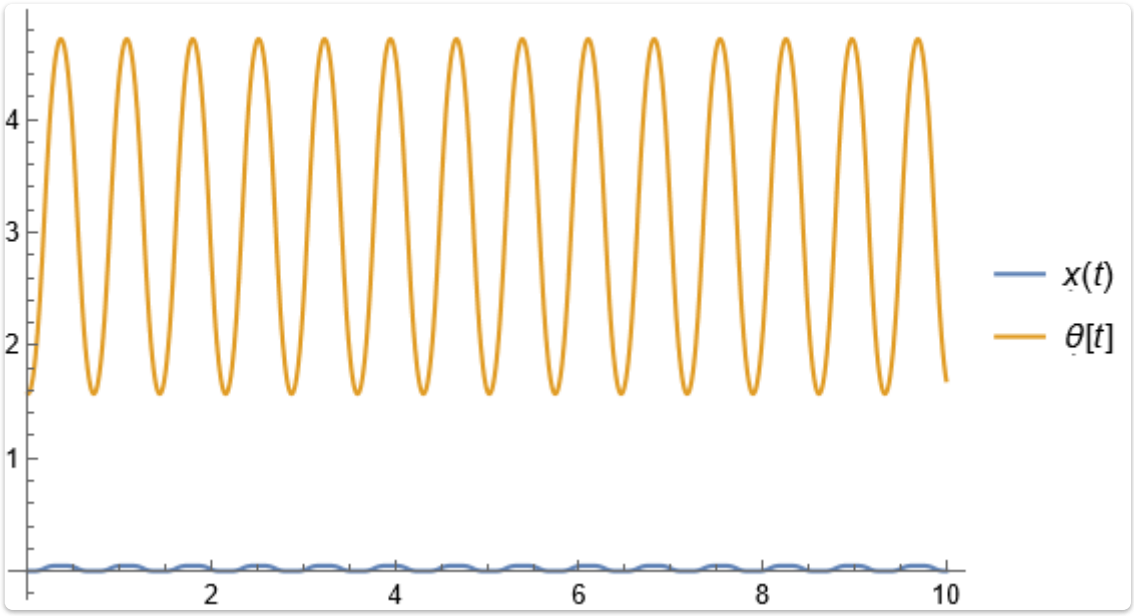
$$-g\,\text{l1cm}\,\text{m1}\cos(\theta[t]) + \frac{1}{2}\left(\text{m1}\left(\text{l1cm}^2\,\theta'(t)^2\sin^2(\theta[t]) + (\text{l1cm}\,\theta'(t)\cos(\theta[t]) + x'(t))^2\right) + M\,x'(t)^2\right) + \frac{1}{2}\,\text{I1}\,\theta'(t)^2$$

Questi sono i parametri.

```
In[ ]:= params = {
          g → 9.8,
         l1 → .15,
          l1cm → .075,
         m1 → .15,
          M → .32,
          I1 → m1 * l1^2 / 12,
          η1 → 0
        };
```

η_1 è l'attrito viscoso del pendolo, Ho visto che alla fine è ininfluente sul risultato.

Per verificare che il sistema sia corretto ho fatto una integrazione numerica. Il risultato sembra soddisfacente. Questa è per il braccio che parte in orizzontale. Ne ho fatte diverse, tutte sembrano a posto.



Qui posso confrontare abbastanza facilmente la simulazione con i dati del sistema reale. Basta salvare i dati dei sensori e fare un plot. Per il doppio pendolo invece non saprei bene come fare, visto che è un sistema caotico e non penso di ottenere lo stesso risultato.

A questo punto linearizzo il sistema attorno al punto di equilibrio. Posso farlo in diversi modi. Io di solito uso il modo spiegato nello Strogatz per cui basta calcolare la jacobiana del sistema in quel punto.
In realtà Mathematica ha una funzione che fa esattamente questo... (Avevo già controllato per l'esame di sistemi complessi che il risultato della funzione è quello che mi aspetto).

Il mio sistema linearizzato è quindi questo:

$$\left(\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{g\ l1cm^2\ m1^2}{l1cm^2\ M\ m1 + I1\ (M+m1)} & \frac{l1cm\ m1\ \eta 1}{l1cm^2\ M\ m1 + I1\ (M+m1)} & \frac{I1 + l1cm^2\ m1}{l1cm^2\ M\ m1 + I1\ (M+m1)} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{g\ l1cm\ m1\ (M+m1)}{l1cm^2\ M\ m1 + I1\ (M+m1)} & -\frac{(M+m1)\ \eta 1}{l1cm^2\ M\ m1 + I1\ (M+m1)} & -\frac{l1cm\ m1}{l1cm^2\ M\ m1 + I1\ (M+m1)} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right) \quad S$$

e numericamente

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -3.08392 & 0. \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 128.839 & 0. \end{pmatrix}$$

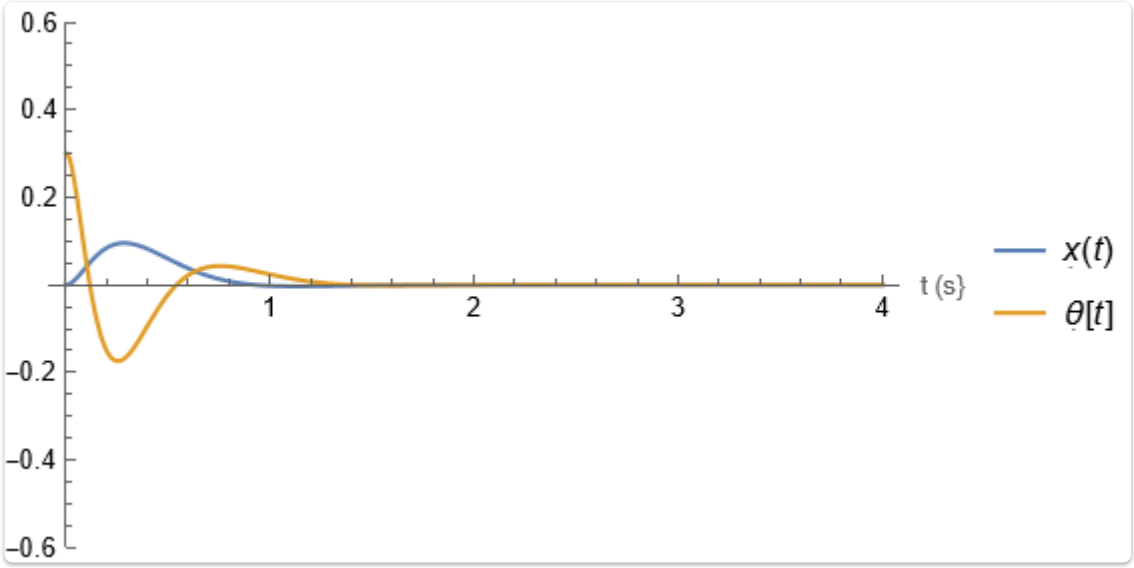
B = MatrixForm=
$$\begin{pmatrix} 0 \\ 2.7972 \\ 0 \\ -27.972 \end{pmatrix}$$

Già qui è interessante andare a guardare gli autovalori e autovettori del sistema e vedere che sono instabili. Nella tesi ho intenzione di dedicare un po' di pagine a spiegare come si studia la stabilità dei sistemi lineari e a mostrare quali sono le condizioni perchè un autovalore sia stabile o meno, a seconda che il sistema sia discreto o continuo.

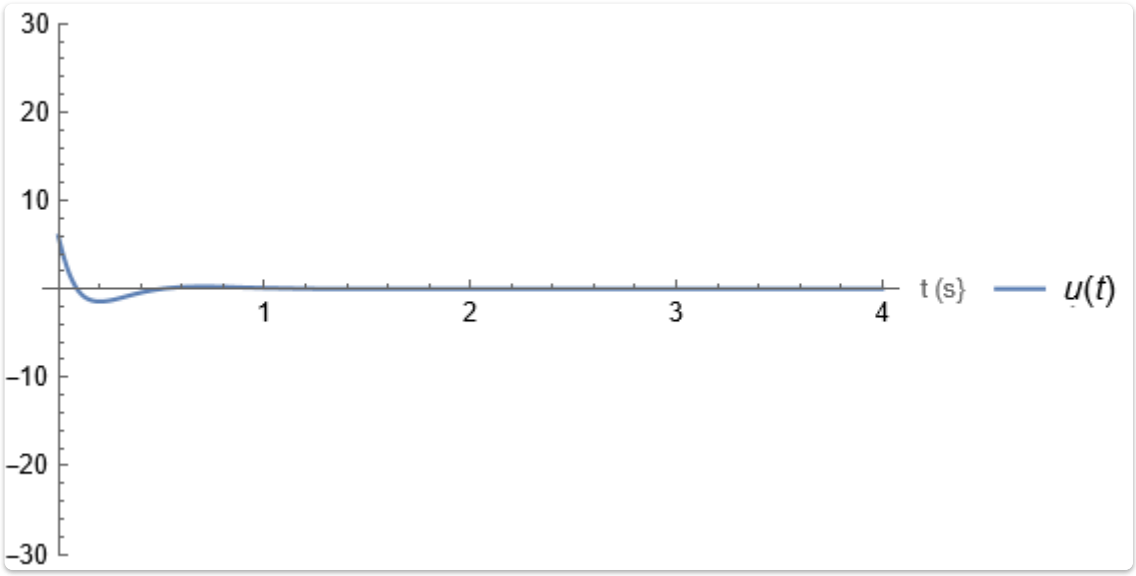
Ora, i coefficienti del controller si calcolano risolvendo un equazione di riccati. Nella tesi volgio spiegare bene come si ottiene questa equazione. Per ora ho usato la funzione LQRegulatorGains già fatta di matematica.
E' necessario specificare una matrice di gain. Lei mi aveva consigliato di usare la matrice metrica, ma non ho capito bene il perchè. Per ora ho usato dei valori essenzialmente a caso.

```
Qmatrix = DiagonalMatrix[{100, .001, .1, .0001}];
Rmatrix = DiagonalMatrix[{.5}];
Klqr = LQRegulatorGains[ssm //. params, {Qmatrix, Rmatrix}];
```

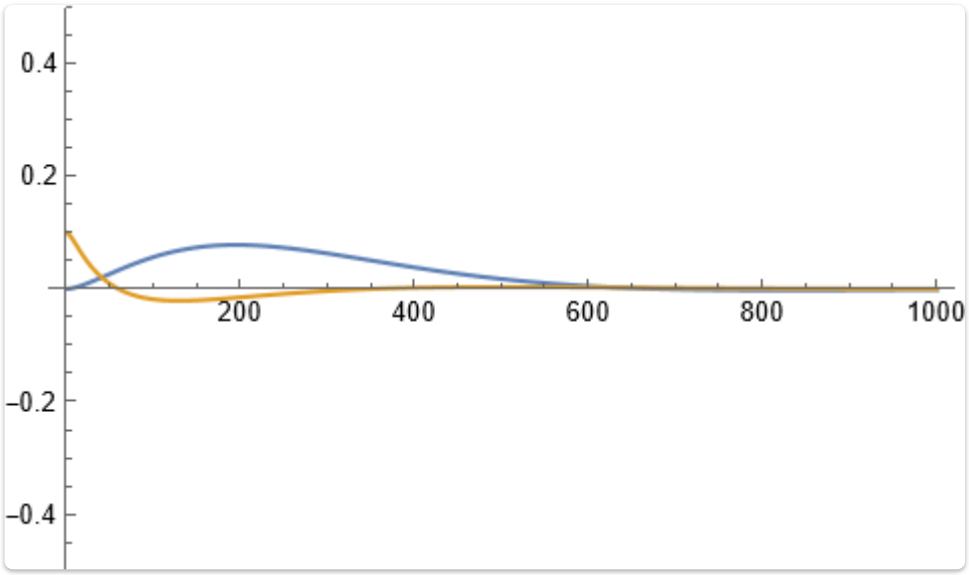
E anche qui ho fatto diverse integrazioni...

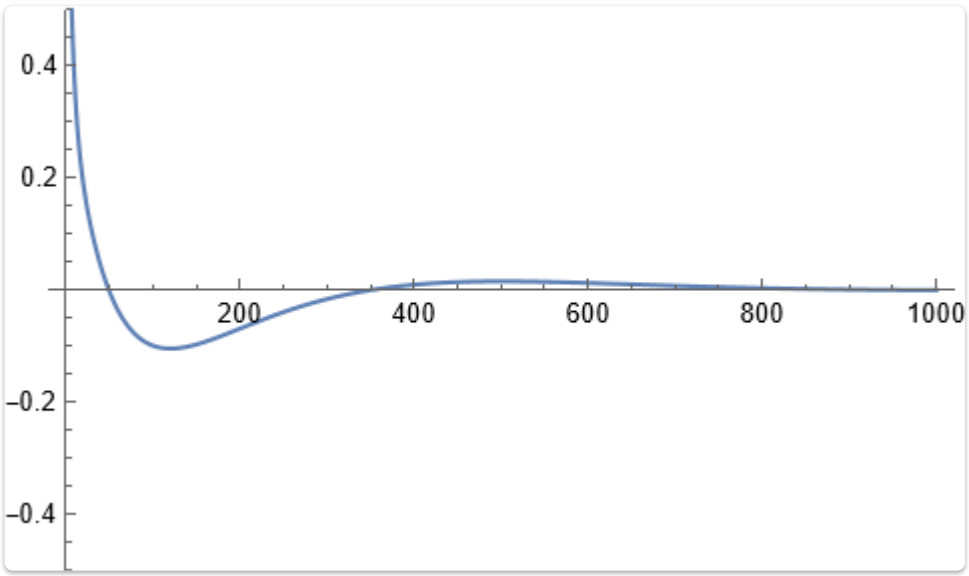


(u è la forza esercitata dal motore).



In queste integrazioni il controllo è continuo.
I risultati con il controllo discreto sono simili...



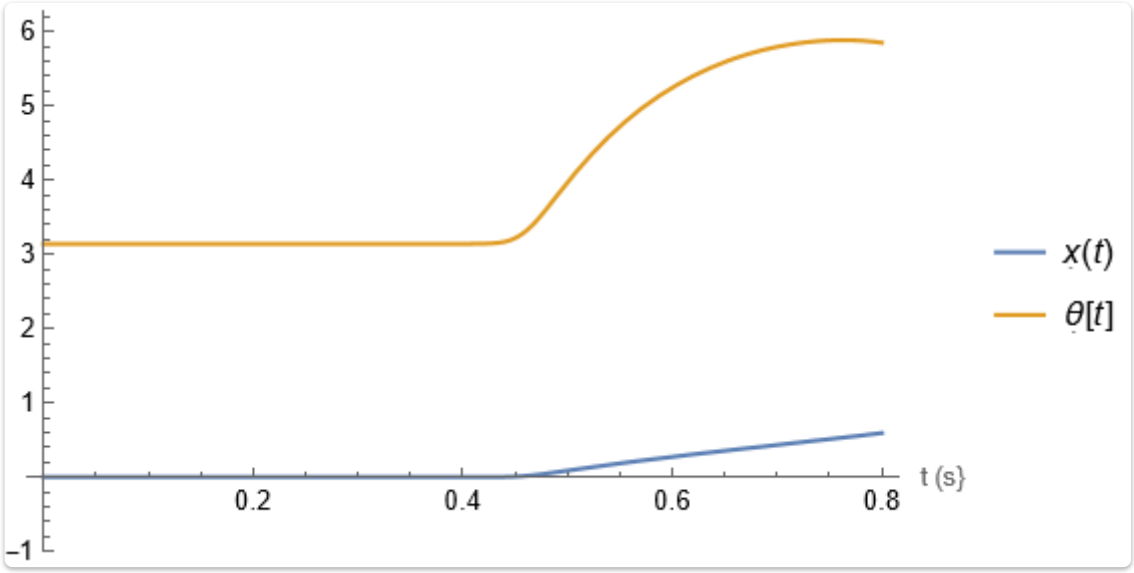


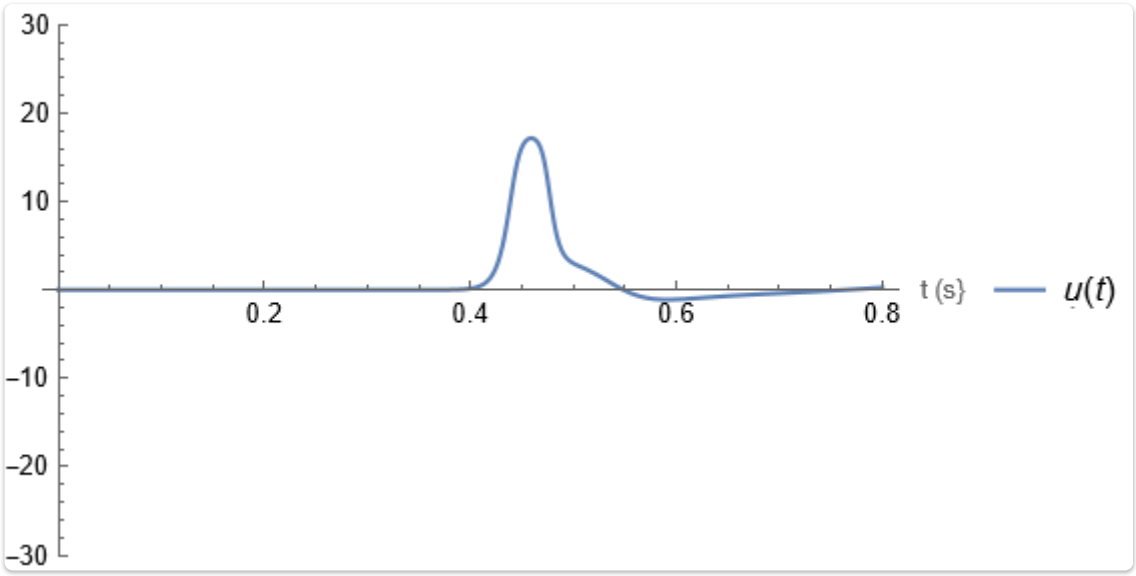
Nella pratica io ho visto che essenzialmente, più sono piccoli gli intervalli, meglio è. Devo però tenere abbastanza tempo in mezzo per permettere ai sensori di calcolare le velocità. Io ho visto che 5ms di tempo funzionano molto bene.

Swing up del pendolo singolo

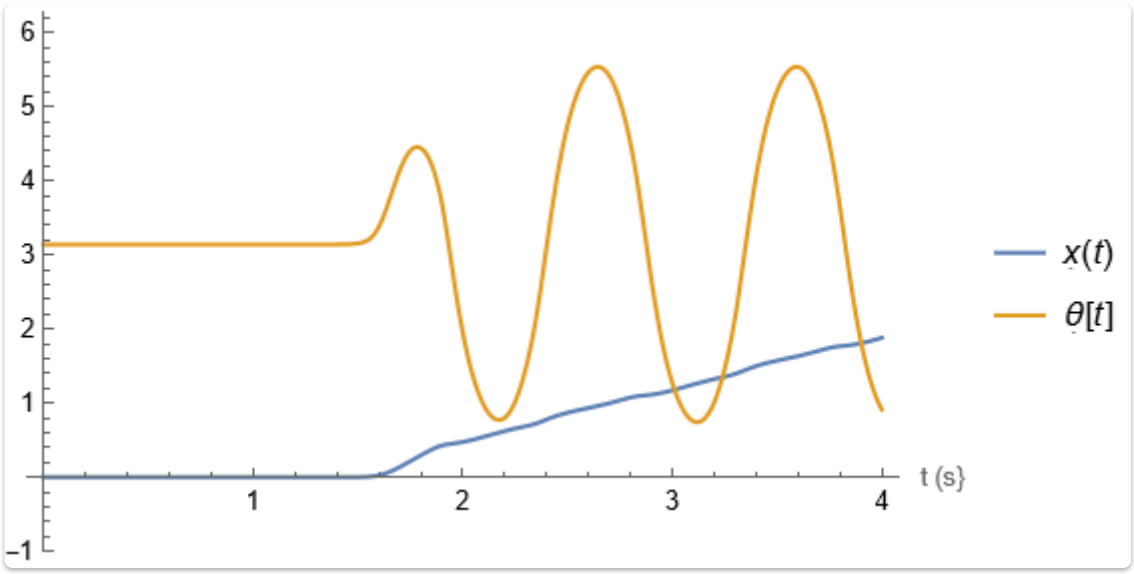
Essenzialmente la strategia è dare un impulso al pendolo in modo da trasferirgli l'esatta energia che lui ha quando si trova nella posizione verticale, fermo. La formula è molto semplice, l'ho presa da questo paper:
https://web.ece.ucsb.edu/~hespanha/ece229/references/AstromFurutaAUTOM00.pdf
la parte interessante del paper è la dimostrazione di come questo metodo porti a far alzare il pendolo, usando il metodo di Lyapunov (paragrafo 3). Per questa cosa di sicuro dedicherò qualche pagina nella tesi.

Questi sono i risultati di un integrazione:





Se il motore è abbastanza potente, il pendolo si alza in una sola mossa. Altrimenti, c'è bisogno di più input...



Ora, il problema con questo metodo è che non stiamo controllando la posizione.
In questo paper, provano a trovare un modo per fare lo swing up tenendo x vicino a zero: https://ieeexplore.ieee.org/document/1383932
in realtà io sospetto che ci sia qualche problema nei calcoli, visto che ho provato a ricreare quello che hanno fatto e ottengo un risultato diverso.
In merito a questo avevo fatto qualche altra simulazione, ma ora non le trovo più. In pratica ho visto che basta semplicemente spegnere il motore se si discosta troppo dall'origine.

Motore

Nel modello sopra, u è una forza. Il motore DC che sto usando in realtà ha in input un voltaggio. Da quello che ho capito io, il motore funziona così:

- La corrente I che usa è proporzionale a voltaggio V e velocità di rotazione ω, secondo una legge del tipo:

  $$I = \frac{V - K_I \omega}{R}$$

  dove R è la resistenza del motore e K_I un coefficiente da determinare.
- La corrente è direttamente proporzionale al momento torcente applicato dal motore:

  $$\tau = K_T I$$

  .

Ricordando che $\tau = rf$ e $\omega = v/r$, posso
ottenere l'equazione del voltaggio che devo applicare per ottenere una certa forza f, in funzione della velocità attuale v:

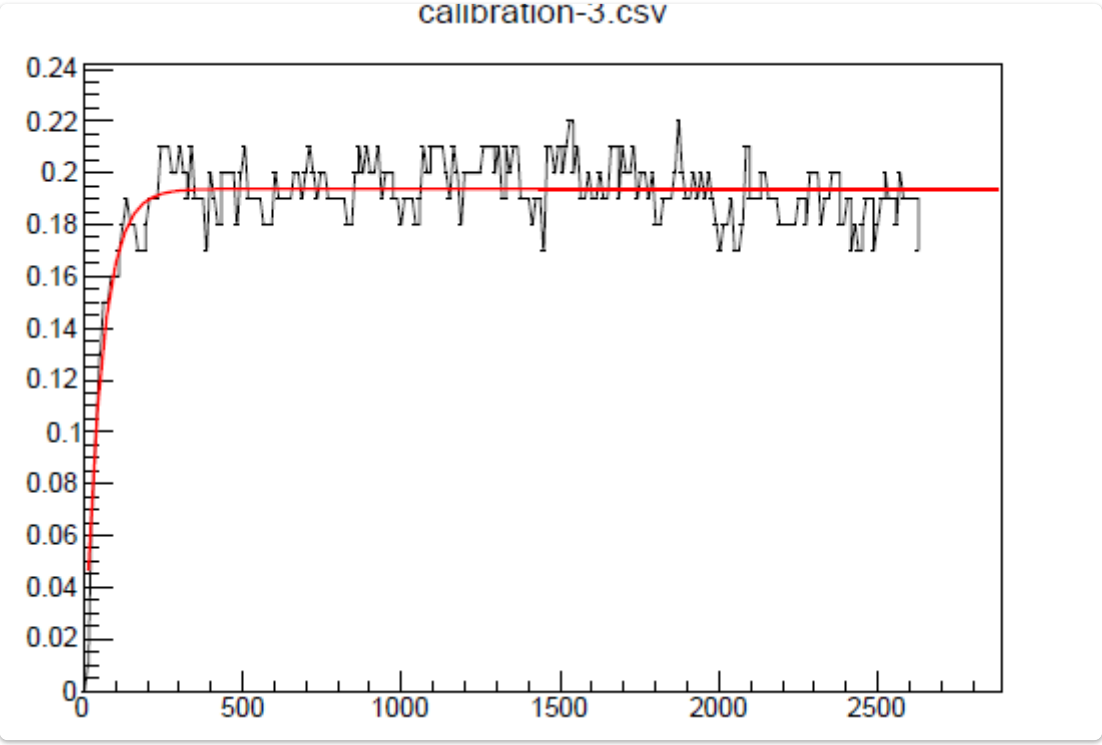
$$f = AV - Bv \quad \rightarrow V = \frac{1}{A} f + \frac{B}{A} v$$

. Io assumo che i parametri A e B siano costanti per il motore e per il sistema. Posso quindi ricavarli sperimentalmente.
Tolgo il pendolo dal sistema e lascio solo il carrello. Posso quindi scrivere che $f = M\dot{v}$.

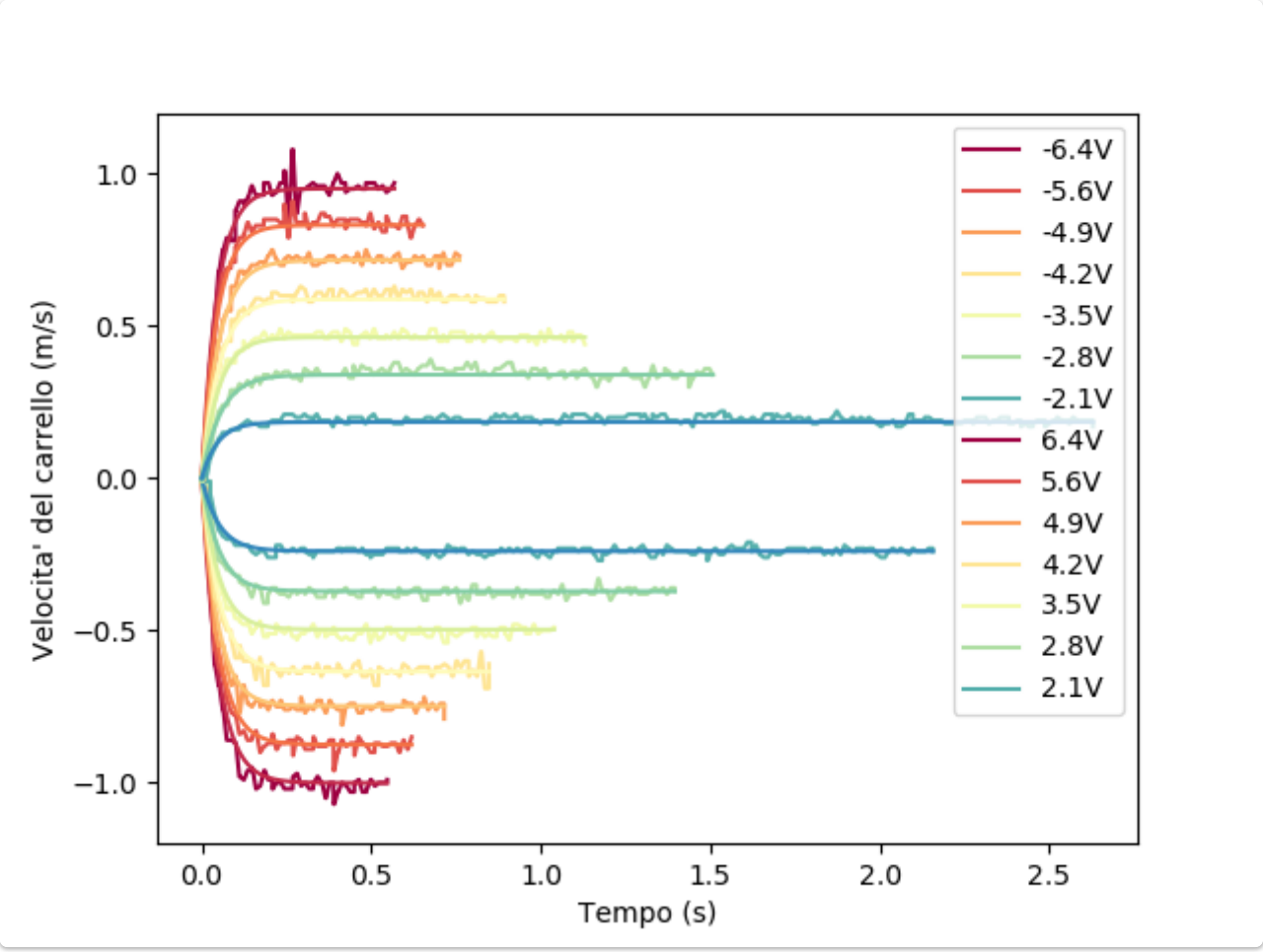
Risolvo l'equazione del moto e trovo, per un voltaggio fissato, la velocità in funzione del tempo:

$$v(t) = \frac{A}{B}V(1 - \exp(-\frac{B}{m}t))$$

Effettivamente, se vado a vedere come si comporta il carrello applicando un voltaggio costante, trovo proprio questo comportamento. In questo modo, posso ricavare A e B con un fit.



Ho provato vari voltaggi e il risultato è sempre lo stesso.



Ora, il mio problema è che dopo aver ricavato A e B, se li inserisco nel codice e provo a muovere il carrello applicando una forza costante, trovo in realtà che il carrello avanza con velocità che aumenta esponenzialmente, per poi fermarsi a una certa soglia.

Alla fine i coefficienti che ho trovato dal fit non andavano bene per stabilizzare il pendolo. Li ho cambiati leggermente "a mano" e alla fine sono riuscito a farlo funzionare abbastanza bene per stabilizzare il pendolo.

Doppio pendolo

Qui ho sempre fatto i calcoli con mathematica. Sono tutti in questo notebook, con anche le simulazioni:
https://www.wolframcloud.com/obj/matteobonna/Published/Doppio%20pendolo.nb
I parametri che ho messo sono verosimili, anche se è difficile calcolarli in modo preciso visto che le parti sono stampate in 3d (non sono omogenee). Ho visto però che influiscono poco sui coefficienti del controller.
Variando i parametri del sistema, i rapporti dei coefficienti del controller rimangono simili tra di loro. Quello che cambia è un fattore moltiplicativo davanti a tutti...