

Cogging torque measurement rig

Cogging torque measurement rig

A project in IN5590 by Abdulrahman Alsaleh Almughir, Said Hemaz, E-mail: abdua@ifi.uio.no, said-hem@ifi.uio.no

1 Intro

Cogging torque is a position-dependent torque in permanent-magnet motors that causes vibration and noise. Our goals were to: build a low-cost test rig for cogging torque that uses an Arduino UNO R4 + Dynamixel AX-18A to drive the motor, and develop analysis tools to compare control settings. The rig supports a master's thesis on cogging-torque estimation and compensation.

2 Tools and hardware

Mechanical and electronics

- Dynamixel AX-18A with Dynamixel shield
- Arduino UNO R4 Minima
- 2× 10 kg load cells with 2× HX711
- External encoder E6A2-CW3C (partly damaged)
- 12 V PSU and 3D-printed PLA rig on acrylic base

Software and tools

- Arduino IDE 2.x with Dynamixel2Arduino, HX711
- Python (numpy, pandas, matplotlib) for analyse_cogging.py
- SolidWorks + 3D printing for mechanical design

3 Optimization of control

We optimised the **Dynamixel speed** between two positions (POS_A, POS_B). A good speed should: give clear repeatable cogging peaks, avoid oscillations and strange dynamics and still allow efficient data collection.

Firmware

The Arduino program drives the Dynamixel between POS_A and POS_B, then reads both HX711 boards (L1, L2) and the encoder and at last applies a simple low-pass filter $x_{smooth} = 0.8x_{smooth} + 0.2x_{raw}$ and streams CSV at 115200 baud as `time_ms,pos,L1,L2,encoder`.

4 Results and reality gap

Ideally, cogging torque depends only on rotor position. In practice:

- L1 was relatively stable; L2 had large offsets and drift.
- The same position did not always give the same torque, especially at high speed.
- The external encoder was damaged, so we relied on Dynamixel position and load cells.
- Misalignment, friction, flexibility and limited HX711 bandwidth all influenced the measurements.

We therefore used baseline-subtracted net forces, simple smoothing, and focused on *qualitative* cogging-torque profiles.

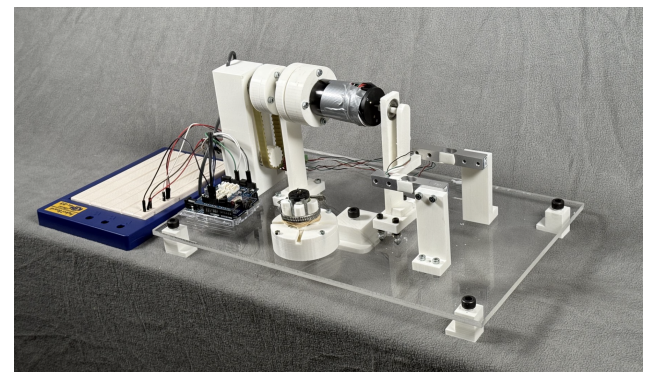
5 Conclusion and future work

We now have a working cogging-torque test rig with Arduino, Dynamixel and dual load cells, plus firmware and Python tools for repeatable experiments and visualisation.

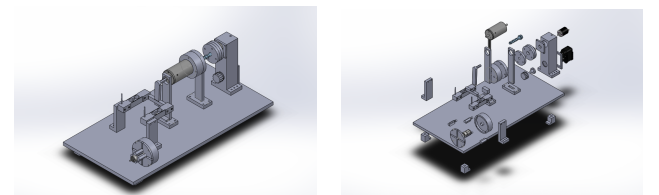
Future work:

- lighter and stiffer rig with improved motor holder, coupling and load-cell mounting,
- more systematic speed profiles and better cogging-compensation models, possibly with faster sensors.

Extra figures



Full cogging-torque test rig with Dynamixel, test motor and dual load cells.



CAD model of the rig (left) and exploded view showing modular parts.



Early motor-holder concepts: cylinder design and heavier 4-jaw chuck.



First load-cell holder that blocked sensor bending; later re-designed.



UNIVERSITY
OF OSLO

GitHub repo →

