

# Nemo V1.1 - User manual

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Defining a motor</b>	<b>1</b>
2.1	Adding a motor to the library . . . . .	3
<b>3</b>	<b>The main window</b>	<b>4</b>
<b>4</b>	<b>The ‘Compare’ Tab</b>	<b>5</b>
<b>5</b>	<b>The ‘Single motor analysis’ Tab</b>	<b>6</b>
<b>6</b>	<b>The ‘Simulation’ Tab</b>	<b>7</b>

## 1 Introduction

**Nemo** is a tool to compare motors and actuators. The notion of motor and actuator is actually merged into one: for the software, a ‘motor’ is a real motor, plus a reduction ratio (only simple transmissions are supported).

Nemo is made of several independant tabs, each with a configuration sidebar and a plot. Each tab is described below.

*General remark:* Because Matplotlib is quite slow, the plot is only updated on demand: whenever the user does a modification, a ‘Update plot’ button appears: this allows the user to do several modifications before asking for a plot update.

## 2 Defining a motor

A motor (actuator) is parametrized in Nemo by 8 parameters:

- $R$ , the per-phase resistor
- $L$ , the per-phase inductance
- $k_e$ , the (single phase) BEMF constant
- $n$ , the number of poles
- $I_{max}$ , the maximum quadrature current
- $I_{nominal}$ , the nominal quadrature current
- $\rho$ , the reduction ratio
- $U$ , the input voltage

Additionally, a motor also has a name, used to create a legend.

These parameters can be set manually through the GUI, in the motor configuration widget shown below.

Motor properties

Name	MyActuator RMD-X6 V3						
R (Ohm)	0.275	−	+	L (mH)	0.090	−	+
Ke (V.s/rad)	0.0900	−	+	N poles	28	−	+
Iq max(A)	10.8	−	+	Iq nom(A)	3.6	−	+
Reduction ratio	8.0	−	+	U (V)	48.0	−	+
Reset from template				▼			
				Helper			

Derived constants

To help you enter these values correctly, clicking on the “Helper” button will bring a dialog which will compute the corresponding per-phase value from datasheet information (e.g. line to line values, delta motors, magnet characterized in terms of  $k_v$ ,  $k_t$  or  $k_e$ ...)

**Nemo - Motor creation helper**
×

### Motor creation helper

**Input**

Connection type ☒ Star (wye) ☐ Delta

Resistance and Inductance

R (Ohm)  − + ☐ Terminal (aka. line to line or phase to phase)

L (mH)  − + ☒ Single phase

Magnetic field strength (choose one)

☒ Ke  
☐ KV  − +  
☐ Kt

☐ V.s/rad, phase to phase  
☒ V.s/rad, single phase  
☐ V/rpm, phase to phase  
☐ V/rpm, single phase

Iq nominal  − + ☒ A ☐ Arms

Iq max  − +

Poles  − + ☒ poles ☐ pairs

Battery voltage (V)  − +

Parameter summary: equivalent star motor

R phase (Ohm)	0.2750	L phase (mH)	0.0900
Ke phase (V.s/rad)	1.0000	N poles	14
Iq nominal (A)	3.60	Iq max (A)	10.80

### Verification

Check that the computed values are reasonably close to the datasheet (differences can exist due to approximations in the datasheet, or because Nemo neglects non-linearities that may affect motor performance).

	Unit	
Km, articular	Nm/√W	2.335
	W/Nm <sup>2</sup>	0.183
Power (non defluxing)	W	400.8
Ke phase-phase	V/rad.s	1.732
	V/krpm	181.380
KV	rpm/V	6
Ktq, articular	Nm/A	1.500
Max RMS current	A	7.6
Max torque, articular	Nm	16.2
No load speed, articular	rad/s	27.7
	rpm	265
Max speed @ max torque	rad/s	24.7
	rpm	236

Cancel
Create

Additionally, a ‘motor library’ provides pre-configured motors for faster, error-free loading. These motor are then available from a drop-down menu in the motor creation widget.

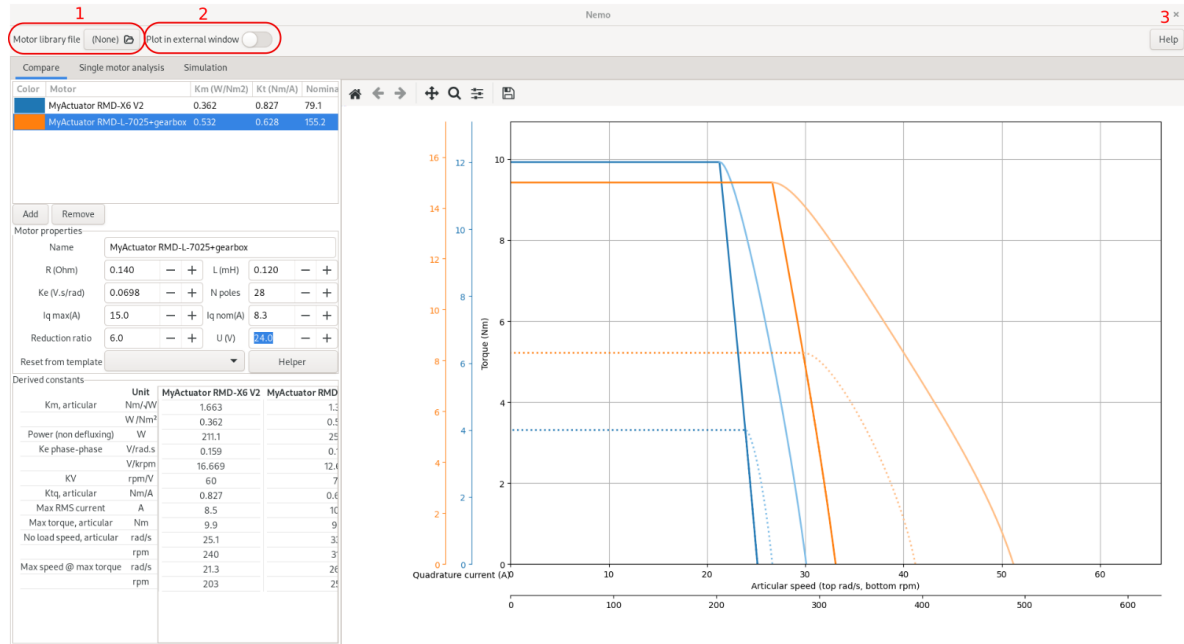
## 2.1 Adding a motor to the library

The motor library is simply a json file describing each motor: the format is given below. A custom json file can be loaded using the “motor library file” button at the top.

```
"MyActuator RMD-X6 V3":
{
  "R": 0.275,
  "L": 0.090,
  "ke": 0.0900,
  "i_quadrature_max": 10.8,
  "i_quadrature_nominal": 3.6,
  "np": 28,
  "U": 48,
  "reduction_ratio": 8
},
```

### 3 The main window

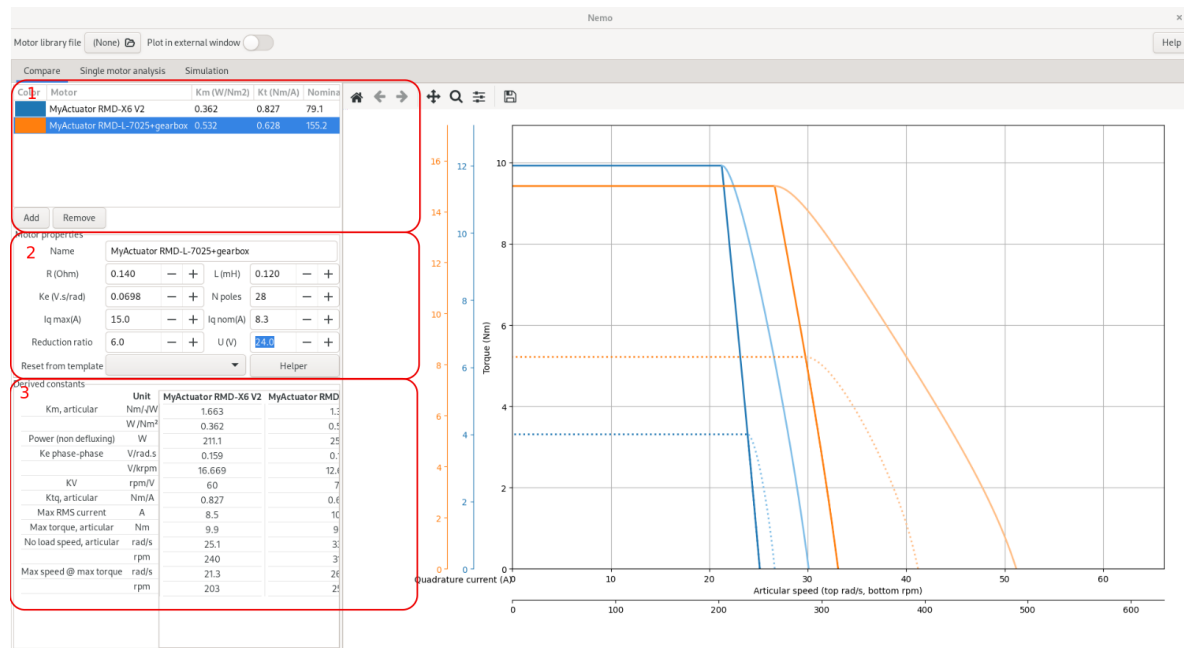
The top of the main window gives access to several functions:



- 1: *library loading button*: use this to replace the default library by another library json file
- 2: *external plot switch*: The main objective of Nemo is to plot graphs. But because part of the screen is taken the GUI, the plot is not that big. To overcome this issue, the plot can be done in its own floating window, by activating this switch. When the 'external plot' switch is on, the plot is done on another window, which can be maximized / move to another screen. To return to the single window mode, you can either deactivate the switch, or minimize the external window (*note that the external window cannot be closed, only minimized*)
- 3: *help* A drop-down menu, with a link to this user manual, and the mathematical documentation explaining the underlying model and computations.

## 4 The ‘Compare’ Tab

**Objective:** to compare the theoretical characteristics of several motors



### GUI

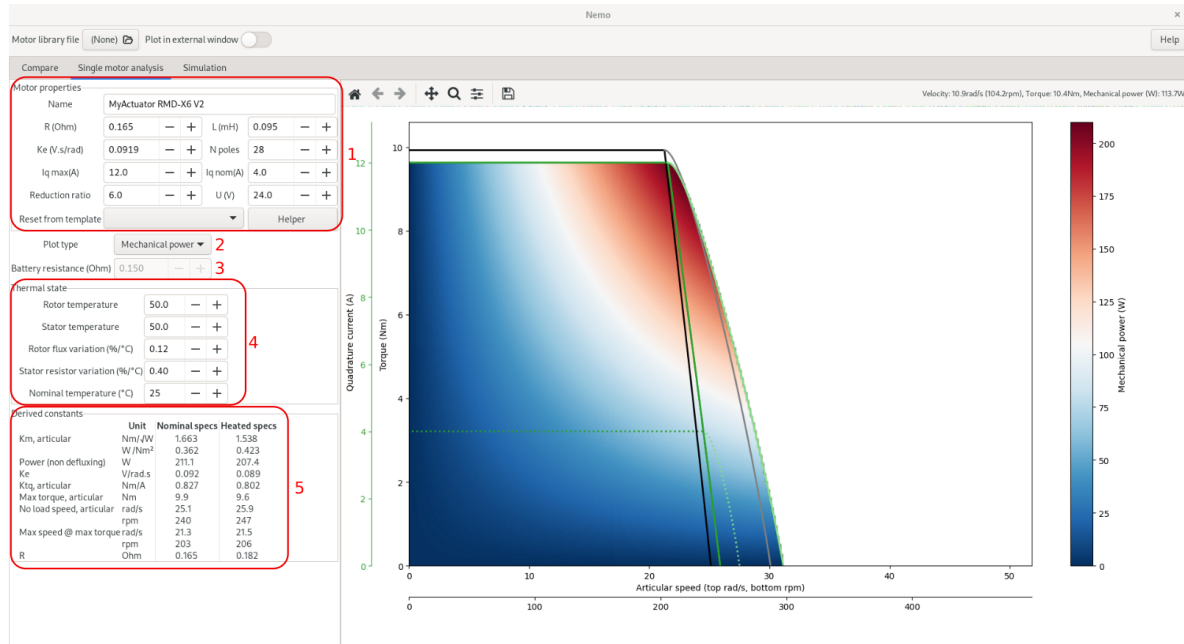
- 1: *motor list* this lists the motors being compared. Elements in the list can be selected ; the selected element can be removed by the ‘Remove’ button, while the ‘Add’ button adds a new motor to the list.
- 2: *motor properties*: this indicates the properties of the selected motor, which can thus be edited and modified.
- 3: *derived constants*: this passive list shows a set of constants derived from the motor parameters. See the mathematical documentation for an explanation of each constant.

**Plot:** each motor is plotted in the specified color:

- the full line plot is the characteristic, at the specified max torque & battery voltage
- the dotted line indicates the nominal current (thus nominal torque) limit
- the lighter-colored curve is the defluxing limit of the motor.

## 5 The ‘Single motor analysis’ Tab

**Objective:** study more in details the behavior of a single motor - including efficiency thermal variation, battery power draw...



### GUI

- 1: *motor properties*: the properties of the plotted motor, at the nominal temperature
- 2: *plot type*: the type of information being plotted in the color plot:
  - the mechanical power
  - the thermal power
  - the total (mechanical + thermal) power
  - the efficiency (mechanical / total power)
  - the battery state (drawn battery current, predicted battery voltage, obtained by modeling the battery as a constant voltage source + serie resistor)
- 3: when plotting the battery state, the battery's internal resistance
- 4: *temperature information*: coefficient of the simple linear thermal model applied:
  - the stator's resistance increases linearly with temperature, with a given coefficient (0.4%/°C is the nominal value for copper)
  - the rotor's magnetic flux decreases linearly with temperature, with a given coefficient (-0.12%/°C is a typical value for Nd magnet)
- 5: *derived constants*: several derived constants, with their variation between nominal and heated value.

### Plot

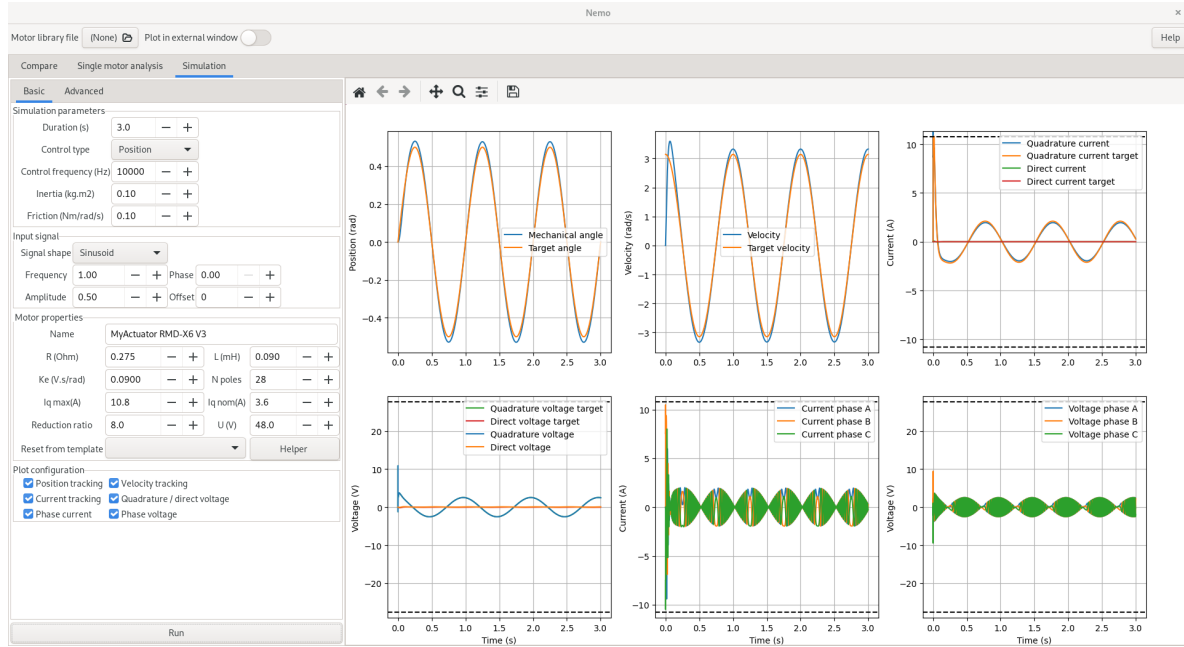
- In black, the characteristic of the motor at nominal temperature
- In green, the heated motor characteristic
- The color map depends on the plot type

*Example:* in this picture, the rotor and stator are both heated by 25°C from their nominal temperature. As a result, the torque decreases (due to a lower  $k_e$ ). The increase in  $R$  reduces the max velocity at max torque, while the reduction of  $k_e$  decreases the back EMF, hence a larger no-load speed.

## 6 The ‘Simulation’ Tab

**Objective:** simulate motor motion using a FOC controller. More specifically:

- the motor driver follows the classical cascade control loop structure: a position PI feeds a velocity PI which feeds a current PI. This outputs direct and quadrature voltage targets, which are then executed using space vector modulation PWM.
- the mechanical equation reads  $I\ddot{\theta} = -\nu\dot{\theta} + \tau_m - \tau_l$ , with  $I$  the inertia,  $\nu$  the viscous friction coefficient,  $\tau_m$  the torque produced by the motor, and  $\tau_l$  an extra resistive torque (load torque).



### GUI

The configuration of the simulation has many parameters, and thus is done in two separate tabs shown here side by side:

The image displays two side-by-side screenshots of a motor control simulation software interface, showing the 'Basic' and 'Advanced' tabs.

**Basic Tab (Left Window):**

- Simulation parameters (1):** Duration (s) 3.0, Control type Position, Control frequency (Hz) 10000, Inertia (kg.m<sup>2</sup>) 0.10, Friction (Nm/rad/s) 0.10.
- Input signal (2):** Signal shape Sinusoid, Frequency 1.00, Phase 0.00, Amplitude 0.50, Offset 0.
- Motor properties (3):** Name MyActuator RMD-X6 V3, R (Ohm) 0.275, L (mH) 0.090, Ke (V.s/rad) 0.0900, N poles 28, Iq max(A) 10.8, Iq nom(A) 3.6, Reduction ratio 8.0, U (V) 48.0.
- Plot configuration (4):** Position tracking, Velocity tracking, Current tracking, Quadrature / direct voltage, Phase current, Phase voltage.
- Run button (10):** Run

**Advanced Tab (Right Window):**

- Direct current target value (5):** Signal shape Constant, Frequency 1.00, Phase 0.00, Amplitude 1.00, Offset 0.
- Additional load torque (6):** Signal shape Constant, Frequency 1.00, Phase 0.00, Amplitude 1.00, Offset 0.
- Current loop PI gains (7):** Kp (V/A) 1.00, Ki (Hz) 100.00, Maximum integral correction (V/A) 10.00.
- Velocity loop PI gains (8):** Kp (A/(rad/s)) 5.00, Ki (Hz) 5.00, Maximum integral correction (A/(rad/s)) 10.00.
- Position loop PI gains (9):** Kp ((rad/s)/rad) 10.00, Ki (Hz) 0.50, Maximum integral correction ((rad/s)/rad) 10.00.
- Run button (10):** Run

- Basic:
  - 1: Simulation global parameters
    - \* Simulation duration
    - \* The type of control: position, velocity or current (i.e. torque)
    - \* Control loop frequency
    - \* The system's mechanical inertia
    - \* The system's viscous friction
  - 2: Configuration of the input signal. This signal is either a position, velocity or current reference, depending on the type of simulation selected.
  - 3: The motor being simulated. Note that the whole simulation is done at the articular level, i.e. the position and velocity plotted, or the feedback gains, refer to the output position (after the reduction ratio).
  - 4: Configuration of the plot - select the plots that you want to appear.
- Advanced
  - 5: Direct current target: this is an exogenous signal, which can be used e.g. for defluxing
  - 6: Load torque (exogenous signal)
  - 7: Current loop PI gains and anti-windup limit value.
  - 8: Velocity loop PI gains and anti-windup limit value.
  - 9: Position loop PI gains and anti-windup limit value.
- 10: Button to launch the simulation