TECHNICAL NOTE 251205

Tuning ORCA™ Motor's PID Controller for Enhanced Position Tracking

Revision 251205
irisdynamics.com | +1 (888) 995 -7050

Tuning ORCA Motor's PID Controller for Enhanced Position Tracking

ORCA™ motors feature an integrated PID position controller that uses position-sensor feedback to track the error between the position sensor and target position, generating a force command that accurately realizes the target position. Its fast internal PID control loop, operating at approximately 3 kHz, minimizes latency compared to motors that rely on external PID control. Low latency results in rapid response times that enhance motion quality, enable finer control, and improve overall system performance. The motor can be commanded to reach position targets in either Position or Kinematic modes. In Position Mode, a stream of targets is fed directly to the motor. In Kinematic Mode, the user defines the endpoints and duration of the motions, and the ORCA defines the motion trajectory.

Definitions

PID Control Loop	Adjusts the control variable based on the error between the target and feedback.
Error	The error between the ORCA's target and sensed position.
Set Point	The desired target.
Measured Process Variable	The position monitored by the ORCA's onboard position sensor.
Controller Output	The force output by the ORCA, which acts as the output that adjusts the process variable.
Disturbance	External force on the motor's shaft, such as user interaction, debris, or machine wear.
Low-Latency	Refers to a delay with minimal duration, with the duration being somewhat dependent on the use case, as the term is used in a number of contexts including networking, audio, hardware, and software [1]. Low latency, in reference to ORCA motors, refers to system response times measured in microseconds.

ORCA's PID Control Loop

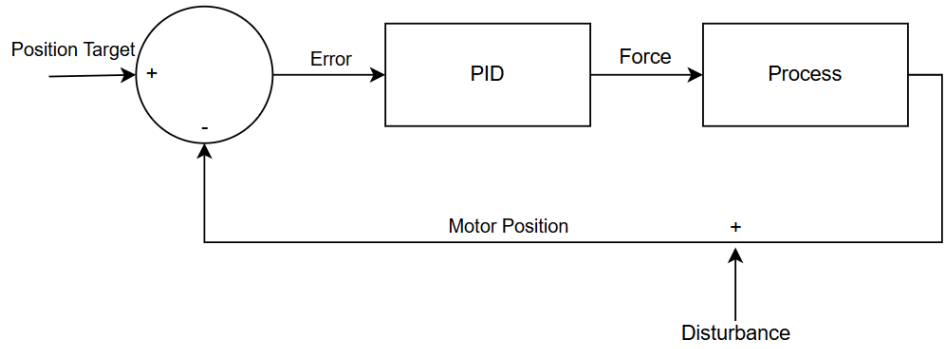


Fig. 1. ORCA Motors' Closed Loop Control.

The ORCA's PID control loop operates through the following and then repeats, correcting for any disturbances as they arise.

1. Measuring the ORCA's sensed position (the measured process variable).
2. Comparing it to the target (set point) to calculate the controller error.
3. Computing the controller's output, the force output by the motor, which adjusts the process variable.

When to Tune Your ORCA Motor's PID Controller

ORCA motors are programmed with default PID tuning, which is suitable for an unloaded application with limited speed or force. In many cases this tuning is sufficient. If the position is not tracking the target position well or if the application requires any of the following, additional tuning may be necessary:

- Vertical mounting
- High-speed motion
- Custom shaft length
- Load applied to the motor
- Presence of system disturbances
- Need to minimize overshoot
- Need to minimize steady-state error
- Human interaction with the device

When Not to Tune your ORCA Motor's PID Controller?

PID tuning is not necessary if you are using Force or Haptic Mode. It is also not necessary if your rate of communication with Modbus is slow. Prior to PID tuning it is best to filter the Modbus input or to increase the rate of communication first.

Tuning Parameters and ORCA Motors

Tuning Parameters

Proportional Gain (P gain): Determines the force applied in proportion to the current error between the sensed and target position. Increasing the proportional gain decreases the rise time at which a target is reached. Increasing this value also increases the overshoot. If this value is set too high, it can result in unstable oscillations.

Integral Gain (I gain): Determines the force applied in proportion to the accumulated error over time (i.e. the area of the error). Adjusting the integral gain helps to reduce or remove steady-state error. If the position target cannot be reached, the force will continue to ramp up and can result in large continuous forces. I gain is inherently unstable, so if tuning occurs using I-gain, it must be accompanied by the appropriate P gain.

Derivative Gain with Respect to Velocity (DV gain): Determines the force applied proportional to the negative velocity of the motor. It reflects the change in sensed position rather than the change in error. This acts like a damper and is useful in removing overshoot.

Derivative Gain (DE gain): Determines the force applied proportional to the rate of change of the error (slope of the error). The DE gain is useful in removing overshoot.

Control Action: The sum of all of the actions.

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\,\mathrm{d}\tau + K_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t},$$

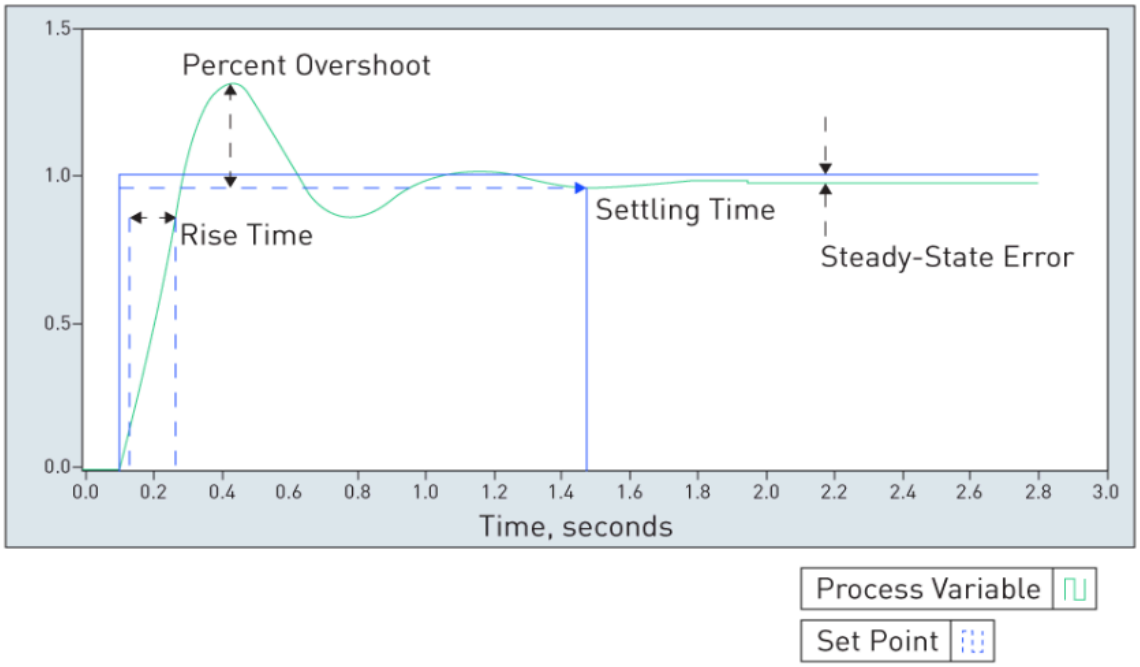


Fig 2. Depiction of rise time, percent overshoot, settling time, and steady-state error [2].

Tuning Through IrisControls

The simplest way to modify the ORCA motor's default PID parameters is through IrisControls. The tuning parameters can be modified by editing the text fields to the right of the arrow.

- After updating the tuning values, click outside the text field or press the Escape key.
- Click **Apply Tuning** to update the modified tuning. The values shown in green are displayed when the motor is in motion.
- To ensure the tuning persists through power cycles, click **Save Configuration**.

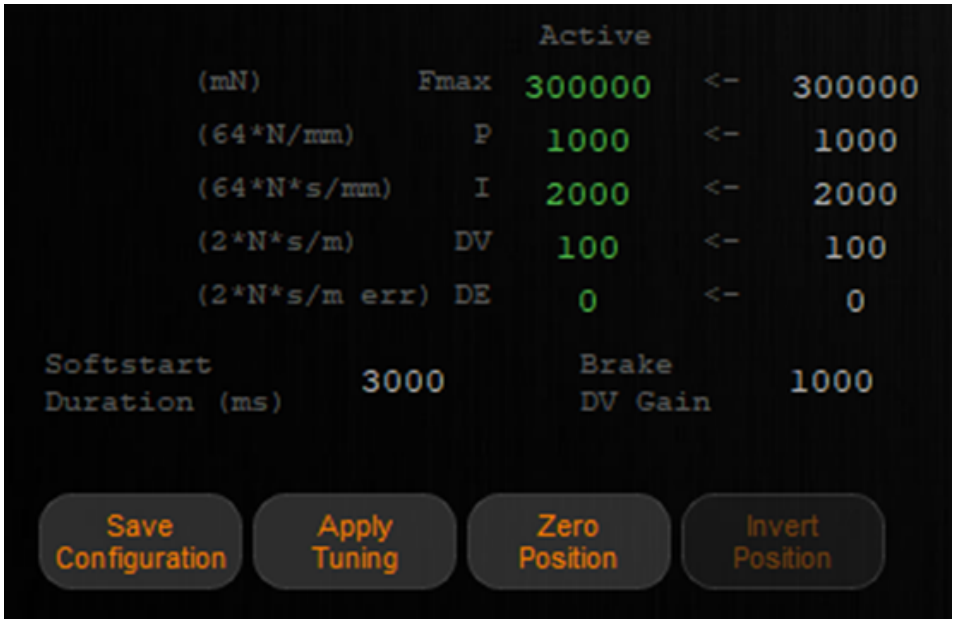


Fig. 3. Set PID Tuning Page's tuning parameters within IrisControls.

PID Tuning Using the SDK

PID tuning can also be executed in code:

PID Function Definitions

```
// C++
tune_position_controller(uint16_t pgain, uint16_t igain, uint16_t dvgain, uint32_t
sat, uint16_t degain=0);

// Python
tune_position_controller(pgain: int, igain: int, dvgain: int, sat: int, dgain:
int=0)
```

Fig. 4. Function definitions for PID tuning in C++ and Python.

A PID tuning example using Python is shown in the sections to follow.

General Tuning Guidelines

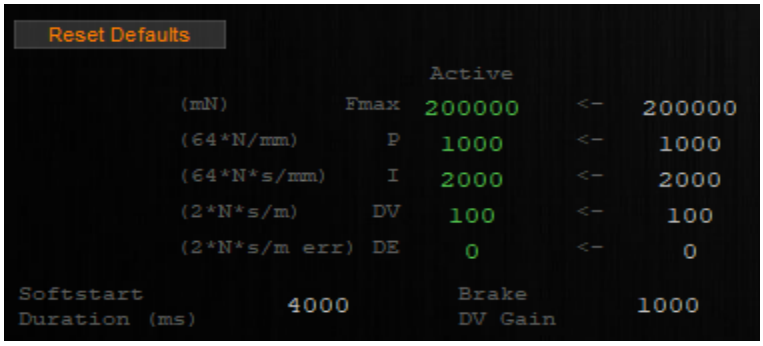
While tuning, ensure the motor is set up in the desired configuration, running movements which replicate real-world conditions. If the motor will operate under load, ensure it is loaded. If the motor is mounted vertically, tune it in a vertical position. Start by adjusting values up or down by up to 20% at a time. Start with small changes, then increase the adjustment if no changes are noticed.

Force Settings

During tuning, the Maximum Force, or **Fmax** can be lowered to protect against unstable responses. **Fmax** is a force limit for the PID controller. If this limit is hit, it will not cause a motor error, but will saturate the controller so that no additional force will be output. If the force saturates during the motion, increase **Fmax** to accurately observe the effects of your tuning.

The **Softstart Duration** sets how long the Position Controller's force output takes to ramp up linearly, and can be modified as needed.

Setting **Fmax** will limit the force without producing an error, while **Max Force** (under Error Threshold) will raise a 128 error: User Max Force Exceeded, along with limiting the force. Setting **Max Force** to 0 will disable this error.



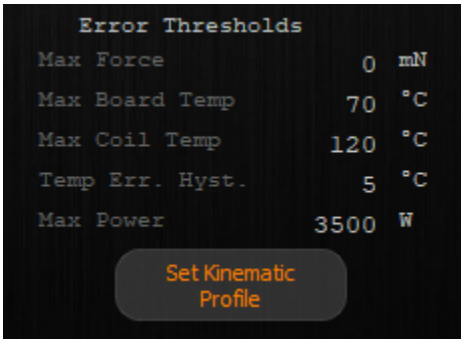


*Fig. 5 & 6. The **Fmax** is reduced to 200000 and the **Softstart Duration** is increased, from the default values. The **Max Force** can be set to 0 in the Error Thresholds panel.*

General PID Tuning Guidelines

Some general guidelines for PID gain adjustments are also provided in the table below.

Desired System Change	Recommended PID Gain Adjustment
Improving target position tracking or movement that is too slow to hit the target position	Increase P gain Decrease DV gain
To reduce steady state error	Increase I gain
To address a large overshoot	Increase the DV gain
To decrease a long settling time or reduce oscillation	Increase DV gain Decrease P gain Increase I gain

If steady state error is not a concern, it's possible to achieve good tuning with only the P gain.

Tuning for Fast Movement with Vertical Mounting

The following parameters were used for tuning for fast movement, with vertical mounting using an ORCA 6-48V.

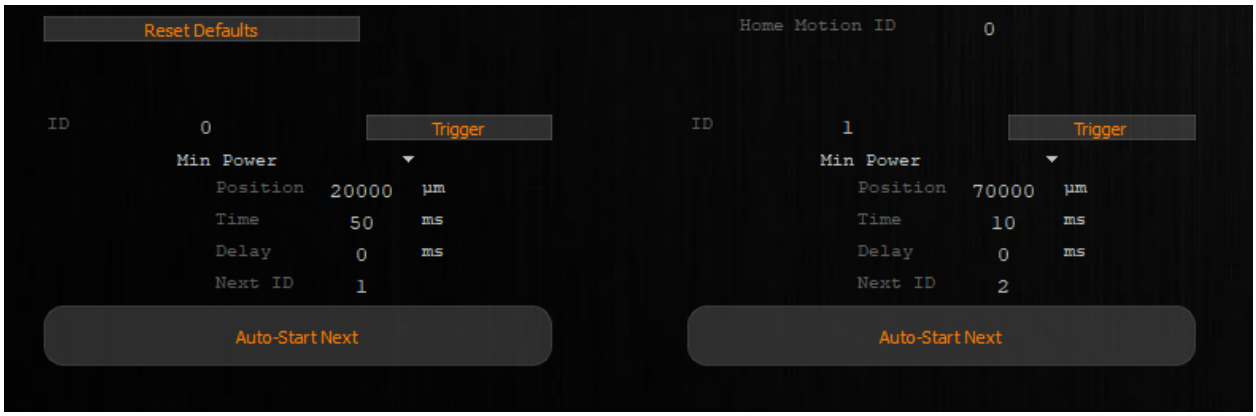


Fig. 7. Motions used while tuning in IrisControls. The motion on the right was executed first, followed by the motion on the left.

To reduce oscillations, it is helpful to begin by increasing DV gain, then adjust the P gain, followed by the I gain. In this example, Fmax was also increased to enable a faster motion.

This value can be increased up to the motor's maximum force output, specified in the datasheet.

When tuning for fast movement using the parameters shown in Figure 7, DV gain was initially increased in increments from 200, to 400, to 800, with a final value of 1000. The I gain was initially removed, but then added back to reduce the steady state error. The P gain was raised to 2000 to reduce the rise time with smooth performance achieved at a value of 3000.

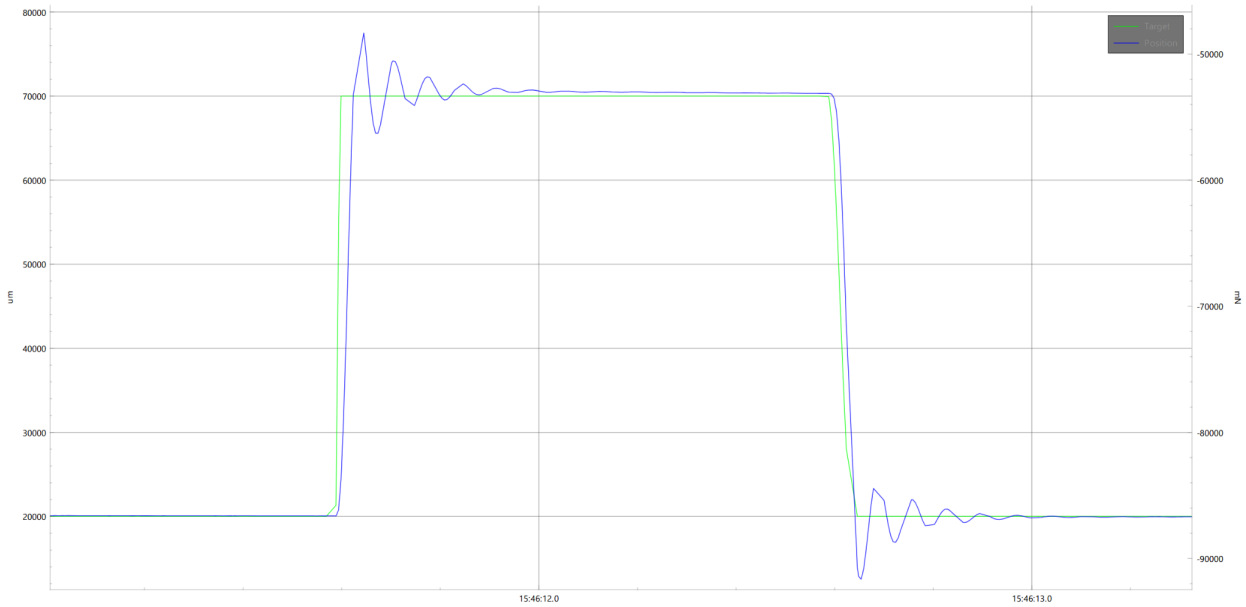


Fig. 8. The ORCA's target and actual position, shown in IrisControls, prior to tuning the motor.

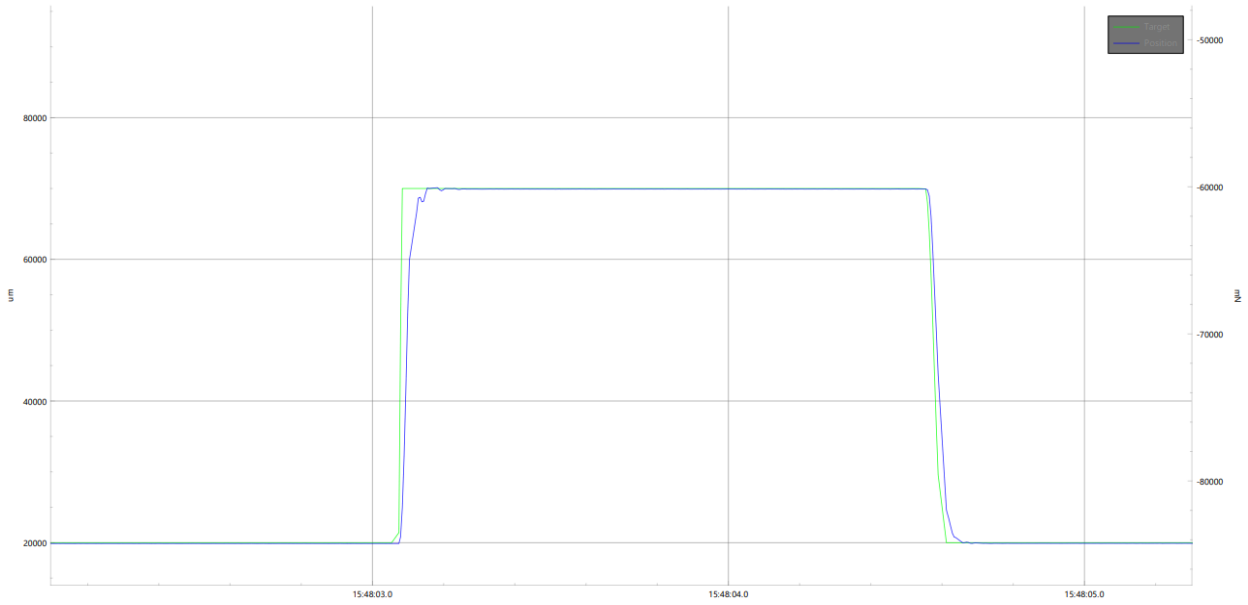


Fig. 9. The ORCA's target and actual position, following PID tuning.

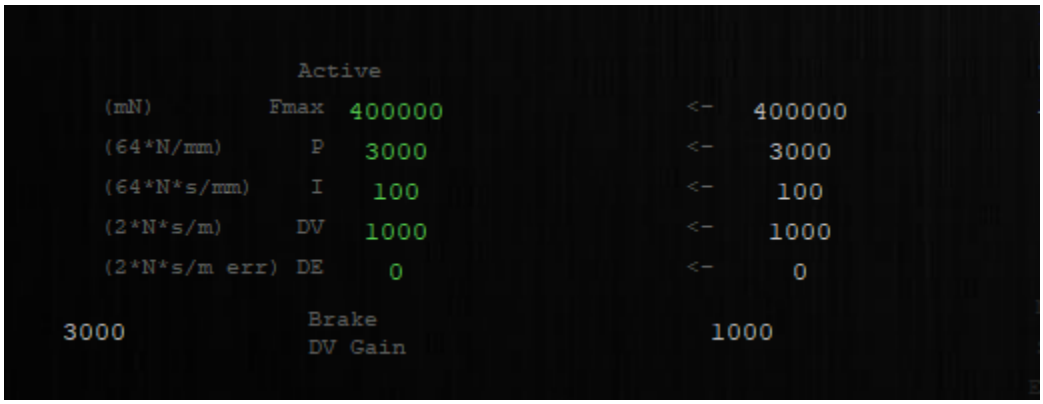


Fig. 10. The final tuning parameters of Fmax: 400000, P: 3000, I: 2000, and DV: 1000.

Tuning for Loaded Movement with Vertical Mounting

For tuning a vertically mounted motor with a 20 pound load, the I gain was removed and the DV gain was increased to reduce oscillations. DV gain was adjusted first from 200 to 400, up to 800. The P gain was increased to 3000 to improve position following. The I gain was added back in at 1500 to improve steady state error.

Fig. 11. Initial motion parameters using IrisControls and final tuning parameters of ***Fmax:*** *4000, P: 4000, I: 1500, and DV: 800.*

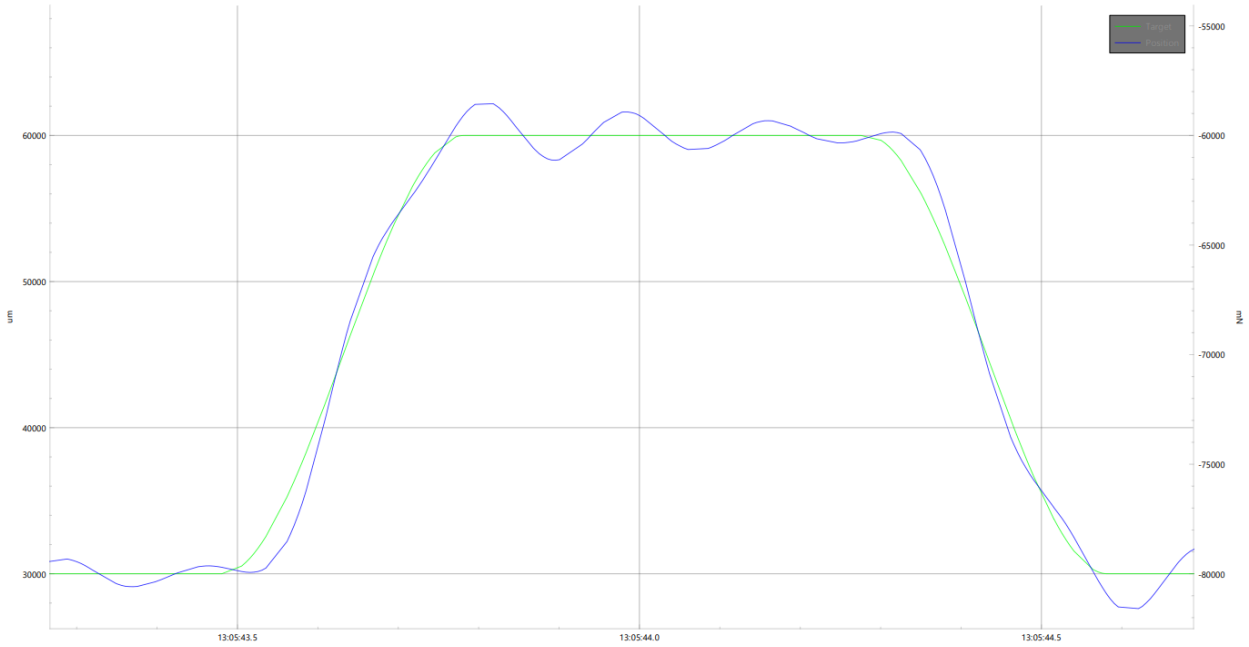


Fig. 12. The ORCA's target and actual position, shown in IrisControls, prior to tuning the motor.

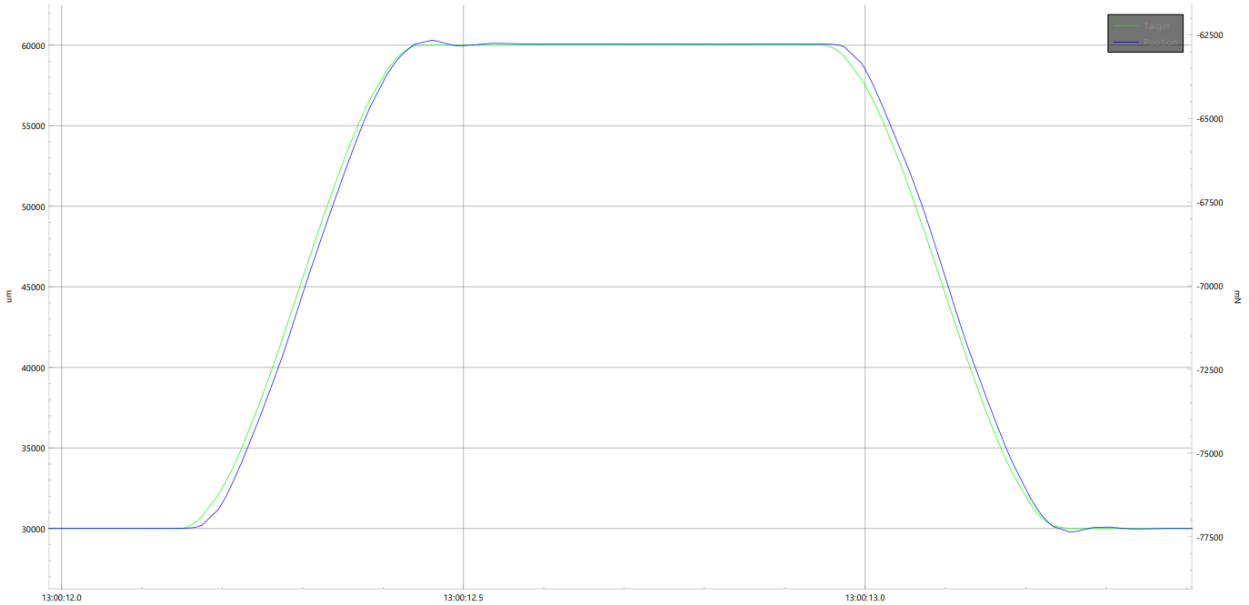


Fig. 13. The ORCA's target and actual position, shown in IrisControls, following tuning.

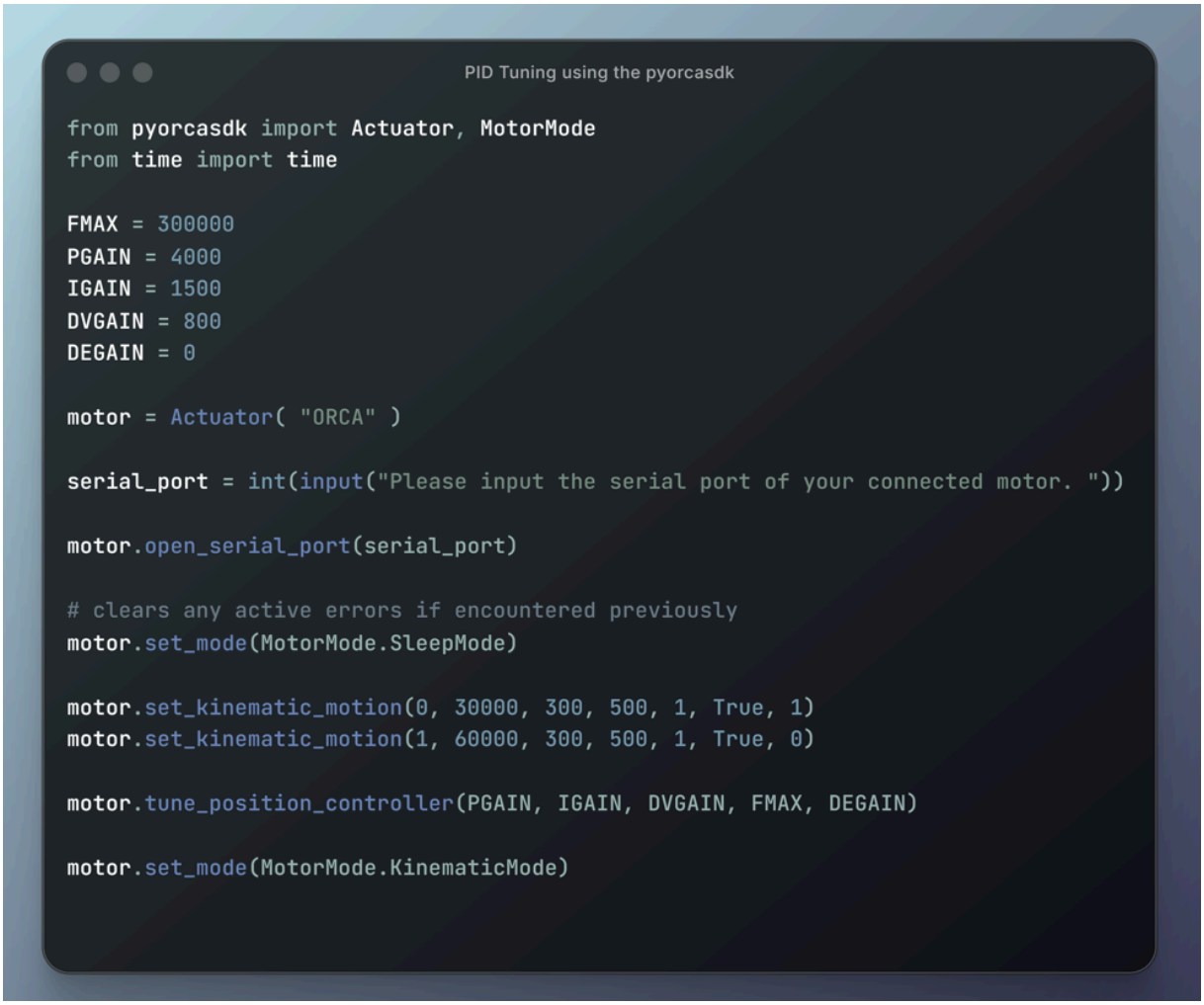


Fig. 12. The same tuning parameters are replicated here using the pyorcasdk.

Tuning for Human Interaction

To create a tactile response based on a human pushing on the stator, the motor's **Fmax** was first decreased to 100, 000 for safety. The P gain was then decreased to 100, and DV gain was increased to 500 to provide a damping effect.