REWL-30

30mm high performance reaction wheel

For Nano-Satellites

Features

- Performance
  - Momentum of 10 mNms @ 20.000 RPM
  - Torque 2 mNm
  - Control: Momentum, torque, speed or motor voltage
  - Automatic motor flux reduction (FOC)
- Physical
  - 30 x 30 x 21 mm
  - Mass: 59 gram
  - Rotor inertia: 5 x 10^{-6} kg mm^2
- Interface
  - CAN
  - Flying leads w/wo connector
- Power
  - 5 - 6V DC
  - Regenerative braking
- Temperature Range
  - Operating temperature range -40°C to 60°C
- Reliability
  - Long life brushless motor design
  - Radiation total dose tested EEE parts
  - Vibration rated for all launch vehicles
  - 5 years design lifetime
  -

Description

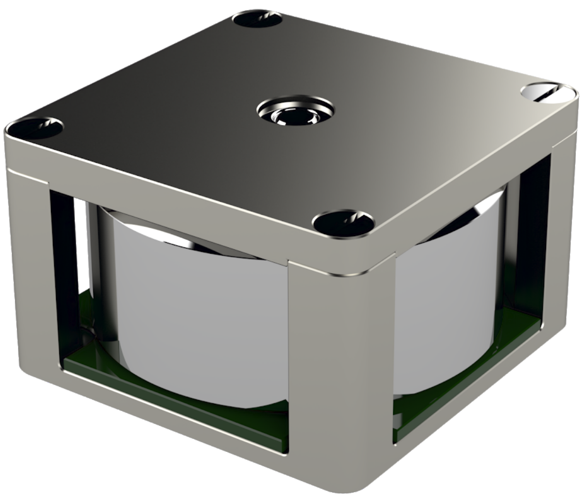
Fully integrated reaction wheel unit for high performance satellite attitude control for Nanosatellite missions with mission lifetime up to 5 years (minimum).

The REWL-30 wheel is an integrated 3-phase outrunner Permanent magnet synchronous motor (PMSM) with 8 rare-earth magnet poles in the rotor and 6 teeth in the stator. Material for the body is Al-7075-T6, rotor is made of ferritic stainless steel while the magnets are Neodymium.

The rotor is axially suspended between two hybrid ceramic high precision bearings chosen for long life and low friction in vacuum conditions. The wheel is commutated by its own internal microcontroller, which runs the control loop to control speed and acceleration upon commands from the ADCS computer.

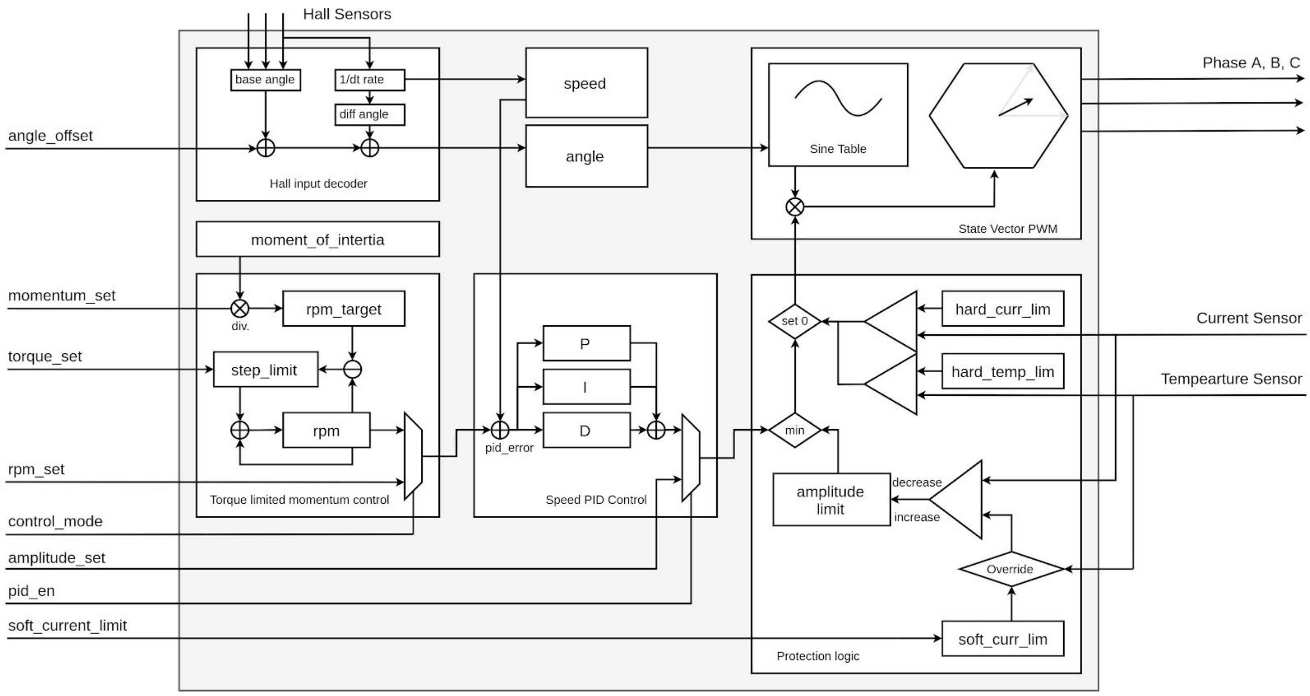
Each wheel has a CAN bus interface with CSP making them accessible to the satellite communication bus. The wheels are fitted with basic telemetry sensors: Temperature, current and speed.

Our recommendation for complete 3 axis control is to use the four wheels in the classic tetrahedron configuration for the benefit of both redundancy and elimination of zero crossing the wheel speed



REWL-30 Reaction Wheel

Functional Block Diagram



REWL-30 Functional Block Diagram

Functional description

The REWL-30 uses hall-sensor based sinusoidal SVPWM commutation with PID speed control, and a torque controller as input for the PID. The inner loop runs at up to 30 kHz with PWM clock of 60 kHz. The outer speed control loop runs at 100 Hz.

Angular Momentum Control

When control_mode is set to 1, the momentum controller is enabled. This controls the rpm_set output variable, depending on the momentum_set and torque_set input parameters.

The angular momentum is set using the Si unit [mnMs]. The controller uses the rotors moment of inertia to translate the desired input value to an rpm_target parameter. The target is then compared to the current rpm, to determine the difference or the desired change in rpms.

If the torque_set parameter is set to zero (or a very high value) the desired rpm_step will be applied immediately, effectively giving the maximum torque. If the torque_set is not zero, this is translated to a maximum acceleration or rpm/sec and the rpm output variable will be increased at this rate, effectively applying a torque limit.

If control_mode is set to 0, the output variable rpm_set, is not modified and can be controlled directly from an external controller.

PID Speed Control

When the pid_en parameter is 1, the speed controller will adjust its amplitude output variable in order to try and match the speed sensor input to the desired rpm_set variable.

When pid_en is set to 0, the amplitude output parameter will not be touched, and can be controlled directly from an external controller.

The parameters, kp, ki and kd, have been set to good default values, but can be adjusted to the user needs.

Protection Logic

The desired amplitude signal can be either set to zero, or softly decreased based on the protection modes.

The hard overcurrent limit and hard temperature limits are set by factory, and will immediately result in the amplitude being set to zero.

A soft_current_limit parameter can be set by the user in order to limit the current used. This can be used as a simple torque controller, as current and torque are related, but is is not as precise as using the actual torque controller. It should mainly be used as a failsafe, or secondary limit.

The soft current limit will be overridden by the temperature sensor so the maximum available torque falls with higher temperature.

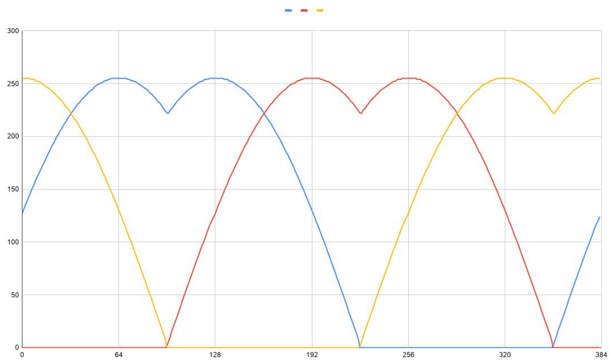
The soft limiter uses an asymmetric limiter, so each time the current is over the soft limit, the amplitude is decreased very rapidly, and then it is slowly increased again. This gives a quick reaction time, but a sawtooth torque ripple while in current limitation.

State Vector PWM

The output amplitude,or control voltage, is fed to the SVPWM module as a duty cycle. The wheel rotor angle, which estimated with a resolution of 0.94 degrees, is used in conjunction with a third harmonic injected SVPWM or THIPWM waveform to decide the three phase output signals. The amplitude or duty cycle is multiplied with the output signals to reduce the speed.

The PWM module runs with a switching speed of up to 60 kHz, and the angle estimator and waveform update performed at half the PWM output frequency.

Double sided, Centre Aligned Synchronous PWM is used for all three phases, in order to reduce switching harmonics. Furthermore Zero Sequence Vector Modulation with bottom clamping is used to reduce switching losses[1]



Third Harmonic Sine, with bottom clamping

Speed Estimation

When the wheel is running, the rising and falling edge of the hall sensors provides timing information that can be used to calculate the speed.

A 48 MHz 24-bit hardware timer is used to accurately time the period between the two falling edges of a Hall sensor. There are 8 magnets in the rotor, which gives 8 slightly different period times, due to their physical alignment and magnetic field variances. The last 8 period counters are stored in the period[8] register in a circular insertion mode. The mean of these 8 are calculated and stored in the period_mean8 register. Using the full rotation cycle mean gives a very high precision and low noise speed estimate.

The lower the speed, the more rarely the estimate is updated. The estimator works down to a speed of 21 RPM, but at this low speed the update rate is lower than the control loop and the wheel cannot run stable in closed loop speed control.

Angle Estimation

The three Hall sensors provide a base angle for the wheel, but the accuracy of this is only 60 degrees and is only accurately known at the point where the hall sensors change state. In order to increase the accuracy, the angle is updated based on linear interpolation using the speed estimate. That means the angle estimate is very accurate at steady state, and either lagging or leading slightly when the wheel is accelerating.

The angle estimator is considering the direction of the wheel and a fixed factory calibrated angle_hyst parameter is used to subtract half the hysteresis angle caused by the hall sensors. The result is accurate angle estimation in both directions.

[1] https://microchipdeveloper.com/mct5001:visualizing-zsm-alternatives

Angle offset

The angle offset parameter is used to allow the controller to work with any hall to stator offset angle. Basically any output angle can be made from any hall sensor input angle. This allows a great deal of freedom in production and design of PCB and hall sensor placement. The accuracy of the offset also allows for fine adjusting to obtain the best possible torque and efficiency.

Angle Advance

The advance angle is used to negate the effects of back EMF on the output waveform. With traditional Field oriented Control theory, this is handled by sampling the phase current, and calculating the in-phase and quadrature currents using the Clark and Park transformations.[2] The best performance is obtained when the stator flux current is zero. At low speed the back EMF is negligible and the current lags the voltage by a precise 90 degrees. At higher speeds the back EMF combines and produces a further lag of the current relative to the voltage. This produces an undesirable in-phase current which is a waste of energy.

At each RPM setting, there will be a different angle offset required to obtain the highest efficiency. A factory checkout measures the effect of the back EMF, and adjusts the angle_gain parameter in order to obtain optimal efficiency across all speeds.

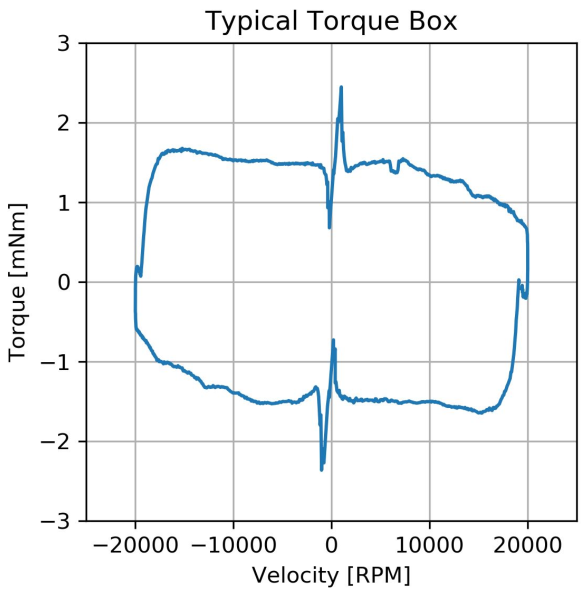
This provides the benefits of Field Oriented Control, without having heavy mathematical transformations in the loop, speeding up the control loop considerably.

The wheel is capable of measuring the current at each speed step, and automatically tuning for the best efficiency. Setting test_en = 2 starts and automatic adjustment of the angle_offset. This is already performed on all wheels during checkout.

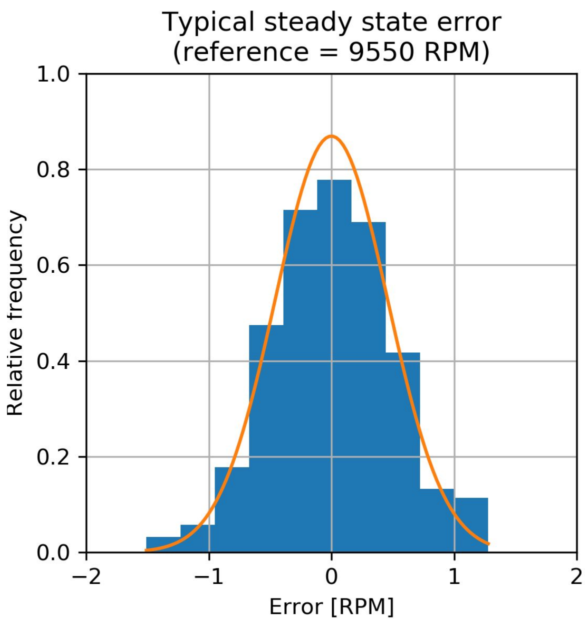
Performance

While the reaction wheel is capable of providing up to 2 mNm torque, is is recommended to limit it to 1.5 mNm. This is because the area of linearity is greatly extended by limiting the torque to 1.5 mNm. Below are a typical torque box measured in a lab setup (e.i. Not in vacuum). It shows that the reaction wheel is capable of providing a torque of 1.5 mNm until around 10.000 RPM, where the torque begins to trail off due to the amount of work that goes into keeping the reaction wheel at a constant velocity.

The nonlinearities around 0 RPM is also very evident on the torque box



The steady state error of the implemented controller has also been evaluated. It was found to have a very small jitter - 0.01 mNm (1σ) and it stays at the commanded RPM within 0.5 RPM (1σ).



2 https://en.wikipedia.org/wiki/Vector_control_(motor)

Exposed Parameters

The following table shows all the exposed parameters of the reaction wheel. Each parameter contains a mask value, this is used as a filter for different types of parameters, e.g. to filter for configuration or telemetry parameters. These flags can be conjoined such that a parameter could be filtered as both a configuration and a telemetry parameter.

Some parameters are used to represent some physical value, and the unit of the value are also denoted in the parameter system.

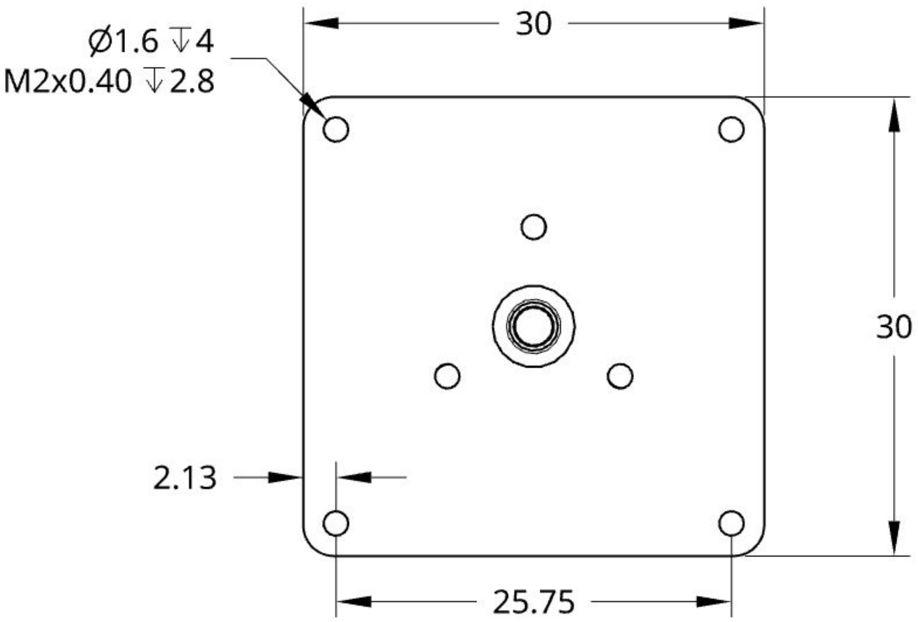
For more information about the parameter system, please see [spaceinventor/libpara]m

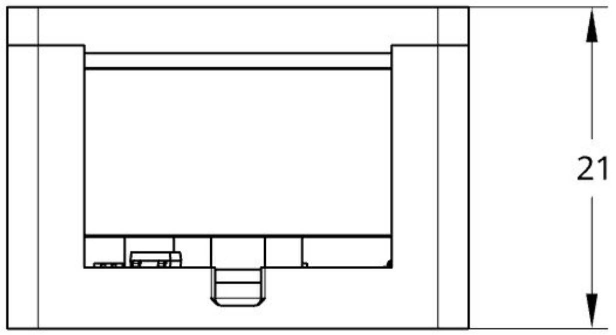
Name	**Mask**	**Unit**	**Description**
temp	0x0009	1/100 deg C	CPU temperature.
csp_node	0x0080		CSP node (address).
csp_rtable	0x0080		CSP routing table.
csp_debug	0x0200		Enable/Disable CSP debugging information.
gndwdt	0x0140	s	Watchdog, if not poked the reaction wheel will reset and load it's backup configuration.
boot_err	0x0060		Used to signal if a error occured during the boot sequence.
boot_cnt	0x0040		Accumulated number of reboots.
boot_cur	0x0040		Current boot image.
boot_img0	0x00c0		Indicates where boot image 0 begins.
boot_img1	0x00c0		Indicates where boot image 1 begins.
time_factor	0x0080		Calibration value related to oscillator uncertainties.
stdbuf_out	0x0008		Captures stdout to a ringbuffer. This ring buffer can be read from the ground station emulating stdio.
stdbuf_in	0x0008		stdin ring buffer. Works in the same way as the above.
period_mean8	0x0008		Mean rotation period over all the the 8 hall sensors.
period	0x0008		Rotation period as seen from each of the hall sensors.
freq_ema	0x0004		Filter constant used in a low pass filter of the hall sensors.
adc_ch11	0x0200		Raw value from ADC ch11.
adc_ch9	0x0200		Raw value from ADC ch9.
temp_brd	0x0008	deg C	Temperature of the board.
current	0x0008	mA	Internal current measurement.
hall_cnt	0x0200		Hall sensor counter.
hall_dir	0x0200		Direction as seen from the hall sensor.

hall_nibble	0x0200		Intermediate value. Used to determine hall_dir.
angle_gain	0x0200		Intermediate value. Used in the Angle Advance algorithm.
angle_hyst	0x0200		Tuning parameter for angle estimation.
sine	0x0200		Output to each coil when using the sinusoidal waveform.
diff_angle	0x0200		See the Angle Offset section.
angle_advance	0x0008		See the section on Angle Advance.
angle	0x0200		Estimated angle of the rotor.
pid_en	0x0004		Enable/Disable PID controller.
kd	0x000c		Gain for the PID.
ki	0x000c		Gain for the PID.
kp	0x000c		Gain for the PID.
rpm	0x000c		Estimated RPM.
rpm_err_sum	0x000c		Integrated error.
rpm_err	0x000c		Apparent error.
rpm_set	0x000c		Reference RPM. Can either be set manually, or used as part of the momentum control (this value is updated automatically).
ramp_en	0x0004		Enable/Disable rate limited RMP. Limited based on torque_set, and the final rpm based on momentum_set.
torque_set	0x000c	mNm	Reference torque..
momentum_set	0x000c	mNms	Reference momentum.
moment_of_inertia	0x0004	gm^2	Inertia of the swing mass.
pwm_period	0x0204		PWM duty cycle period.
pwm_isr_inter	0x0200		PWM interrupt service routine interval.
pwm_isr_ticks	0x0200		Measure of the CPU time spend in the PWM interrupt service routine.
amplitude_max_step	0x0004		Maximum step size in amplitude per update.
amplitude	0x000c		Amplitude of the PWM output.
cooldown	0x0200		Seconds to wait when hitting a hard protection.
test_en	0x0004		See Angle Advance.
curlim_soft	0x0004	mA	Soft current limit. The RW will slowly reduce the torque while in this limit.
curlim_hard	0x0004	mA	Hard current limit. The RW will stop torquing, and free float for a period.
templim_hard	0x0004	1/100 deg C	Hard temperature limit. The RW will stop torquing, and free float for a period.

auto_resume	0x0004		Resume operations on reboot.
rpm_target	0x000c		If ramp_en is set to 1, this value is updated based in the torque profile.
rpm_per_sec	0x000c		If ramp_en is set to 1, this value is updated based in the torque profile.
angle_offset	0x0004		See Angle offset.
waveform	0x0200		Select between a sine wave of square function as input for the electromagnets .
odometer	0x0008	krev	Accumulated number of revolutions of the reaction wheel.

Mechanical Drawings





Mass

The mass of the REWL-30 is 59 gram.

Pinout

The REWL-30 comes with flying leads with the following "pinout":

RED: Power
BLACK: Power ground
GREEN: CAN High
BLUE: CAN Low