Field Oriented Control

A Software Kernel for Servo Drives and Frequency Converters

zub machine control AG
Buzibachstrasse 31, CH-6023 Rothenburg, Switzerland
Phone +41 41 54150-40 · Fax +41 41 54150-49
info@zub.ch · www.zub.ch
zub machine control AG
focused on motor & motion control

Motion control is our daily business

zub AG is focused on motion control solutions for the machine, tool, and device building industry. We are the preferred motion control partner of many global plant and machine manufacturers. Even one of the major frequency converter brands relies on our engineering experience. Our customers maximize their profit by using our very specialized knowledge in the range of modern motor regulation algorithms and highly sophisticated motion control tasks like CAM profiling and drive synchronization.

Our product portfolio

We offer ready-to-run products like motion control units and servo amplifiers, but we also offer control software modules which can be adapted to different types of target hardware. Since all of our products and software modules were developed inhouse without any standard libraries, we can offer customizations to almost anything to support application requirements for functionality or hardware platform. Our software is used on 8 bit, 16 bit, 32 bit microprocessors, DSPs, and multi-processor platforms running with and without an operating system.

Consulting and Engineering Services

A strong knowledge of the motion control business and the experience of hundreds of applications allow us to offer the most efficient consulting and engineering services. We are independent and are not linked to any motor, frequency converter, servo amplifier, or PLC company. Our aim is to provide you with cost-optimized and high-functionality solutions aligned to your requirements. This gives the desired competitive edge to your company and to your products.
Drive control:
The software makes the difference

Highly sophisticated control algorithms improve motor performance

Even apparently simple applications can be very demanding for the motor control. A crane is a good example of that. The full load comes to bear abruptly as soon as the slack rope tightens. The motor control must cope with such a sudden load change from 0 to 100 % at all speeds. This is a demanding task for any frequency converter and demonstrates the quality of the control algorithm in use.

Motor control algorithms for asynchronous motors

zub machine control AG developed the motor control software for the frequency converter of an Asian manufacturer of dockside cranes. The most up-to-date control algorithms were evaluated, tested, and implemented to ensure stable and quick reaction, even at full load change, of sensor and sensorless asynchronous motors.

Motor control algorithms for brushless motors

Modern motor control algorithms in combination with sinusoidal commutation, give the best results for brushless motors over the full range of speed. The brushless motor can run with almost no torque ripple even at low speed. A typical application is a pump with constant, non-pulsing flow.

Motor control algorithms for stepper motors

A vector oriented control algorithm eliminates the well-known noise of stepper motors and allows smooth and very accurate movements at reduced costs compared to brushless motors.

zub machine control AG provides the software

The motor control software for all examples above was implemented by zub machine control AG and can be adapted to different hardware platforms.
FOC software kernel for frequency converters and asynchronous motors

Operating Modes
- Velocity Mode
- Torque Mode
- Positioning and Synchronisation Mode (in combination with Motion Control kernel)
- Speed and torque limited operation
- Sensor and sensorless motor operation
- Field weakening above nominal velocity

Control Algorithms & Techniques
- FOC = Field oriented control
  Field Oriented Control is a math-intensive technique for controlling AC induction motors with respect to motor speed and improved dynamic behavior. This allows optimal motor operation and reduces motor size, cost, and power consumption. FOC offers the best the pre-conditions for smooth and precise motor operation and quick reaction on load variations.
- Luenberger state observer
  The Luenberger observer models a real system in order to provide an estimate of its internal characteristics by observing indirect effects to the system output. The FOC implementation of zub machine control AG uses a Luenberger observer to estimate torque and flux.
- MRAS velocity estimation in sensorless mode
  MRAS stands for “model reference adaptive systems” and is an adaptive control which is able to adjust its parameters to the process. FOC uses MRAS in the case of sensorless operation to determine motor speed.
- Voltage limitation and field weakening
  D- and Q-path current controllers independently identify the required voltage. However, the geometric addition of the two components must not exceed the available voltage. For this reason, the controller enters field weakening mode above nominal speed. With this technique, up to twice nominal velocity can be reached. Under certain circumstances, the voltage limit might still be reached. The controller must then reduce the D or the Q-voltage component.

Auto-Setup Capabilities
- Estimation of the required motor and hardware parameters
- Automatic adaptation of the PI speed and current control parameters (by bandwidth factors)
FOC in use by frequency converters of a crane

Demanding requirements of a crane application

- Support of sensor and sensorless mode
- Up to 2 x nominal torque
- High velocity accuracy
  
  approx. 0.004%, ±0.001 Hz
- Flying changeover between different modes
  
  e.g. from torque to velocity mode and vice versa
- Speed / torque limitation, even in generator mode
- Holding up to 2 x nominal load at 0 Hz in sensor mode and close to 0 Hz even in sensorless mode!
- Minimum velocity drop on abrupt load changes, even at low speed, in both sensor and sensorless mode!
- Extremely smooth acceleration and deceleration with defined maximum jerk.
  
  (Jerk limited ramps are part of the optional motion control software kernel.)

Software / Hardware Interface

- Sampling and analog/digital conversion of currents and DC-voltage
- PWM generation for the inverter / IGBT control
- Quadrature encoder interface for sensor feedback
- CPU in use: C28xxx (Delfino DSP series by Texas Instruments)
- Minimum requirements for a target system:
  
  CPU / DSP (32 bit) with FPU (= floating point unit), "C" compiler
FOC proven in different test series

The performance of the FOC software kernel was tested and proven in different test series at a university in Germany, a test bench in Switzerland, and finally by a crane manufacturer in Asia.

Schematic diagram of the testing environment

The load is simulated by a 200 kW motor and a high-end frequency converter in torque mode. The torque measurement is done directly on the motor shaft with an independent sensor. The brake is used to apply the load abruptly to the 37 kW test motor controlled by zub’s FOC.

General remarks about measured and calculated torque

The torque experienced by the FOC test motor consists of three components …

… the load torque (simulated by the 200 kW “Load Motor” in torque mode)

… the torque required to compensate for the inertia of the “Load Motor”

… the torque required to compensate for the inertia of the “Test Motor”

When accelerating or decelerating the system, the torque estimated by FOC's observer is not equal to the measurement at the motor shaft which covers the first two components only. The torque estimated by FOC, and effectively experienced by the “Test Motor”, includes all components.
FOC Test case: Slack rope

Test description
The nominal load (= 240 Nm motor shaft) is lifted from the ground with a rope that is loose in the beginning and tightens during the lifting process. The opposite happens when the load is moved downwards and touches the ground, i.e. the load seems to “disappear” abruptly.

The test forces torque and velocity overshoots by abrupt load changes. Low overshoots and fast settling times prove the quality of the control loops and the Luenberger observer.

Characteristics
Full step response, 25 Hz, ±240 Nm abrupt load change, 40 ms rise time.

1. Start of acceleration
2. End of acceleration ⇒ Motor is running with a constant speed of 25 Hz with no load
3. Load applied: approx. +240 Nm
4. Load inverted: approx. -240 Nm
5. Load released
6. Start of deceleration
7. End of deceleration

Scalings
Left / Right: Velocity [1/1000 Hz], 25000 = 25 Hz / Torque [1/1000 Nm], 240000 = 240 Nm

Curves
Red: Reference velocity [Hz] (mainly covered by the green curve)
Green: Actual velocity [Hz]
Black: Measured torque [Nm] at the motor shaft [Nm]
Orange: Internally calculated torque [Nm] by the observer
FOC Test case: Slack rope

Velocity mode with sensor feedback

Sensorless velocity mode
FOC Test case: Hoisted rope (blocked by a brake)

Test description
A load suspended on a rope is blocked by a brake. The frequency converter generates a torque against the brake. The brake is opened when the torque reaches a configurable level. The motor then accelerates to 25 Hz under nominal load condition (= 240 Nm).

A fast settling time proves the dynamics of the Flux estimation and the quality of the velocity control loop over all.

Characteristics
Full step response, 25 Hz, 240 Nm

1. Motor generates constant torque (in torque mode) to a load blocked by a brake
2. Brake released at a torque level of 200 Nm and frequency converter switched to velocity mode → Acceleration of the 240 Nm load starts
3. End of acceleration → Motor is running at 25 Hz with 240 Nm load applied
4. Start of deceleration
5. Motor is in standstill generating approx. 220 Nm torque against the brake
6. Drive is disabled

Scalings
Left / Right: Velocity [1/1000 Hz], 25000 = 25 Hz / Torque [1/1000 Nm], 240000 = 240 Nm

Curves
Red: Reference velocity [Hz] (mainly covered by the green curve)
Green: Actual velocity [Hz]
Black: Measured torque [Nm] at the motor shaft [Nm]
Orange: Internally calculated torque [Nm] by the observer
FOC Test case: Hoisted rope (blocked by a brake)

Velocity mode with sensor feedback

Sensorless velocity mode
FOC Test case: 2 x Nominal load at low speed

Test description
Twice the nominal load (i.e. $2 \times 240 \text{ Nm} = 480 \text{ Nm}$) is applied to motor shaft and must be moved. The frequency converter is commanded to run the motor at a very low constant speed of $0.5 \text{ Hz}$. The test proves that the FOC of the frequency converter is able to run a motor at low speed under overload conditions. This is extremely demanding in sensorless mode and proves the quality of the MRAS velocity estimation. Many competitive frequency converters are not able to start or stably run a sensorless motor under this condition!

Characteristics

\begin{align*}
0.5 \text{ Hz, } 480 \text{ Nm} &= 2 \times \text{nominal load} \\
\text{1} & \quad \text{Start of movement} \quad \Rightarrow \quad \text{Motor is running at constant speed of } 0.5 \text{ Hz with } 480 \text{ Nm load even in sensorless mode!} \\
\text{2} & \quad \text{End of movement} \quad \Rightarrow \quad \text{Motor holds the twice the nominal load even in standstill!} \\
\text{3} & \quad \text{Drive is disabled} \\
\end{align*}

Scalings

Left / Right: Velocity [1/1000 Hz], 500 = 0.5 Hz / Torque [1/1000 Nm], 480000 = 480 Nm

Curves

- Red: Reference velocity [Hz]
- Green: Actual velocity [Hz]
- Black: Measured torque [Nm] at the motor shaft [Nm]
- Orange: Internally calculated torque [Nm] by the observer

Remarks

- The observed velocity ripple of approx $\pm 0.07 \text{ Hz}$ in sensor mode is due to the encoder resolution and 1 ms counting cycles.
- The velocity drop resulting from the load change to 480 Nm is compensated within approx. 1 second in sensor and sensorless mode.
FOC Test case: 2 x Nominal load at low speed

Velocity mode with sensor feedback

Sensorless velocity mode
**FOC Test case: Velocity change at nominal load**

**Test description**
The frequency converter is commanded to increase the velocity stepwise under constant nominal load condition. This test proves the quality of the Luenberger observer implementation by showing the accuracy of the calculated torque.

Low velocity overshoots and fast settling time validate the velocity estimation in sensorless mode and the quality of the control loop over all.

**Characteristics**
Velocity steps 0.4 / 1.0 / 2.5 / 5.0 / 10 Hz, 240 Nm

1. Acceleration up to 0.4 Hz with 240 Nm load applied
2. Acceleration up to 1.0 Hz with 240 Nm load applied
3. Acceleration up to 2.5 Hz with 240 Nm load applied
4. Acceleration up to 5.0 Hz with 240 Nm load applied
5. Acceleration up to 10 Hz with 240 Nm load applied
6. Deceleration down to standstill

**Scalings**
Left / Right: Velocity [1/1000 Hz], 1000 = 10 Hz / Torque [1/1000 Nm], 24000 = 240 Nm

**Curves**
Red: Reference velocity [Hz] (mainly covered by the green curve)
Green: Actual velocity [Hz]
Black: Measured torque [Nm] at the motor shaft [Nm]
Orange: Internally calculated torque [Nm] by the observer

**Remarks**
- The velocity overshoot is very small (approx. 0.1 Hz) and compensated within approx. 100 ms in sensor mode and approx. 400 ms in sensor less mode.
- The torque is very stable and only increased during acceleration phases due to inertia.
FOC Test case: Velocity change at nominal load

Velocity mode with sensor feedback

Sensorless velocity mode
FOC software kernel for servo amplifiers of brushless motors

Operating Modes
- Velocity Mode with adjustable torque limitation
- Torque Mode with adjustable velocity limitation
- Positioning and Synchronization Mode (in combination with Motion Control kernel)

FOC gets the most out of the brushless motor

FOC performs high precision and most effective sinusoidal current control. This results in …

- Almost no torque ripple
  - Smooth motor operation over the entire speed range, even at very low speeds
- Reduced noise emissions
  - A must for medical devices close to the patient
- Reduced current consumption and motor heating
  - Reduced energy costs and best energy efficiency of the motor
FOC software kernel for power stages of stepper motors

Operating Modes
- Velocity Mode
- Torque Mode
- Positioning and Synchronization Mode (in combination with Motion Control kernel)

FOC = Make a stepper behave like a brushless motor

FOC gives to stepper motors all the advantages of brushless motors:
- Reduced noise emissions
  - Stepper motors will produce less annoying noise when using FOC
- Robust on load changes and on overload
  - Keeps the motion profile accurate without losing steps

The stepper motor plus field oriented control is the most competitive solution especially for small handling robots or X/Y/Z positioning tables, e.g. in use by analytical apparatus. Even inexpensive high volume stepper motors with encoders or linear scale feedback can cope with applications that required an expensive brushless motor in the past.
Going one level higher: Fully featured motion control software kernel

Aim
zub machine control AG offers a fully featured motion control software kernel for integration on single or multi-processor systems. The motion control kernel can be used on top of the FOC control software and offers all required functionality for drive positioning and synchronization.

Intelligent Power Stages
The motion control kernel enables frequency converters and servo amplifiers to process sophisticated positioning and synchronization tasks on their own. The drive behaves like an intelligent motion control slave with an integrated PLC. Process control steps related to motion control tasks can be fully delegated to the drive.

Integrated Motion Control Features
- Single or multi-axis positioning
- Speed synchronization
- Position synchronization
- Marker synchronization
- CAM profiling
- Optional: Customizable DS402 implementation (for commanding by CANopen or EtherCAT)

Advanced Process Control Software & Programming Tools
- APOSS programming language and development environment (by zub machine control AG)
- Optional: IEC61131 programming environment
Telling the drive what to do: APOSS - Programming language

**Aim**

The APOSS programming language and the development environment offer maximum flexibility for drives with the motion control kernel installed.

**APOSS Programming Language**

The APOSS programming language is a high level language close to structured text or C. APOSS offers high level commands and parameter sets focused on efficient motion control programming:

- Drive positioning and synchronization
- Local and peripheral digital & analog I/O control
- High-level communication for common interfaces: RS232, RS485, USB, CANopen-Master & -Slave, EtherCAT-Master & Slave
  - On request: Powerlink, Profibus, Profinet
- Process control and monitoring functions
- Conditional and unconditional branches
- Interrupt functions, e.g. on input, bus, position, timer, error state
- Function calls and local variables

**APOSS IDE = Integrated Development Environment**

The APOSS IDE is a fully integrated environment for the configuration, optimization, programming, and debugging of all types of controllers or drives using the motion control kernel by zub machine control AG. The APOSS IDE offers the following tools:

- Modern Editor and Online-Help System
- Online-Debugger
- Smart Oscilloscope
- Motion Visualization
- Graphical CAM-Editor
APOSS sample: Positioning

```c
// Example code for Positioning depending on input signals
// Inp.1: Move 5 motor turns forward / Inp.2: Move 5 motor turns backwards
// Initialize base control settings linked to the motor in use
gosub InitDrive

// Program variables
G_ReIDist = 59 * 1 * (get encoder x(1)) // Distance to move for 5 motor turns

// Define origin at the current position and switch regulation on
deforigia x(1)
setmotor x(1) // Activate regulation of axis 1

// MAIN LOOP *****************************************************
endless:
if ((in 1) or (in 2)) then // If signal on input 1 or 2, ...
    print "Position move from " cpos x(1), " to ";
    if (in 1) then
        move G_ReIDist // ... move 5 motor turns forward
    elseif (in 2) then
        move -G_ReIDist // otherwise move 5 motor turns backwards
    endif
    print cpos x(1)
    wait 1 off // Debounce input 1
    wait 2 off // Debounce input 2
    delay 50 // Wait 50 milliseconds
endif
goto endless
```

![Graphs showing position and velocity changes](image-url)
APOSS sample: Synchronization

```c
// APOSS sample code for master / slave synchronization (= electronic gearbox)

// Initialize base control settings linked to the motor in use
gosub InitDrive

// Define origin position for master and slave
deforigin // Master axis
deforigin x(1) // Slave axis
motor on x(1) // Activate regulation of axis 1

// Define synchronization factors (= gear ratio)
set syncfactors 2 // Move 2 slave position steps ...
set syncfactors 1 // ... per 1 master axis step

// Activate automatic synchronization as a background task
syncp

// MAIN LOOP ******************************************************
endless:
    print 'Master axis: ',mpos,' / Slave axis: ',cpos x(1)
    // Wait 5 seconds, so that print out messages just come up in this cycle
delay 5000
    goto endless
```

Slave position = 2 x Master position

Master position
Choice of modules

- Bus Interface / Bus-Control
- GUI / Panel-Control
- DS402 Interface
- Programmable Logic Control APOSS or IEC61131
- Motion-Control Positioning + Synchronization
- FOC control software
- Low-level Hardware-Interface
- Operating System Interface

On the way to your successful product

1. FOC features?
   - Initial 2-day presentation and discussion:
     Presentation and discussion of the key features of FOC.
     Presentation of additional software add-ons and modules.
     Discussion of how FOC and add-ons can be integrated in a product.

2. Application requirements & FOC?
   - 2-day application workshop:
     Discussion of the base requirements of the application.
     Discussion of the existing or planned hardware / software platform.

3. Concept
   - Initial consulting and engineering task:
     Evaluation of an integration concept.
     Presentation, discussion and verification of the concept.
4 Engineering task & workshops:
Definition of functional and performance characteristics.
Linking the requirements to measurable technical data for final testing.
Finalize the development specification.
Finalize the test specification, which will be used to judge the project's final result.

5 Integration
Engineering and software implementation:
Software development.
Adaptation of existing modules to the platform and special requirements.
Integration on the target platform.

6 Testing & Optimization
Engineering and workshops:
Testing according to the test specification.
Discussion of the test results.
Verification of the project's final results.

7 Release
Building the release version.

8 Going public!
Enjoy the success of your product on the market!
Keep an eye on your competitor's products.

9 Maintenance
Improving features and keeping the product attractive.

10 Going one step ahead!
Start the next innovation cycle.
P.S.: Our “Field Oriented Control” software plus jerk-limited motion profiles, provide an unmatched comfort gain for elevators. Acceleration and deceleration are so smooth that it is hard to believe that the elevator actually moved.

Perhaps motion control software by zub machine control AG is the reason why people like to put their office directly into the elevator cabin.