

An In-depth Look at the Multicopter Control System Architecture



Authors: Anton Erasmus

6 July 2020

PX4 Developer Summit 2020

Introduction

Controllers:

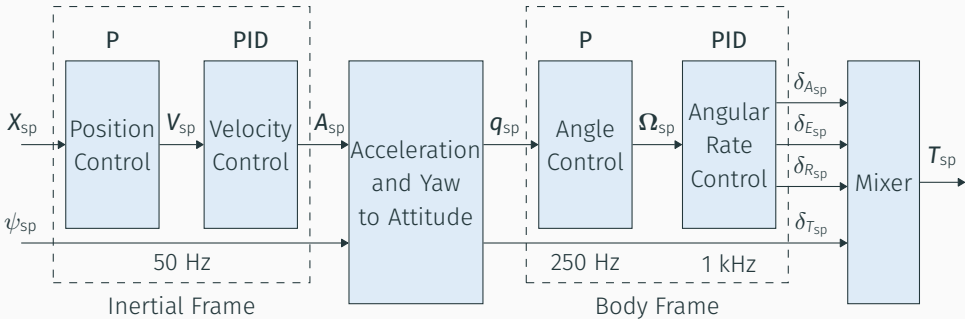
- Architecture
- Parameters
- Modules
- Control Law Functions



Scope: Controller architecture

More Information: [Talk] Overview of Multicopter Control from Sensors to Motors

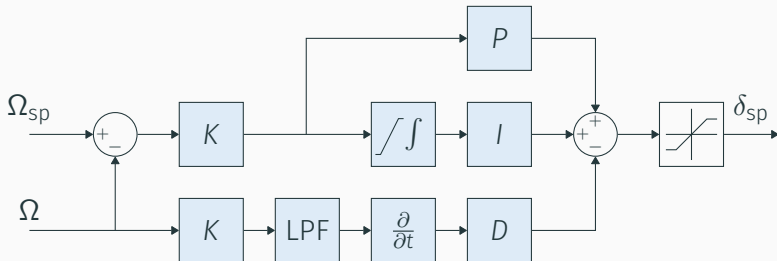
Controller Architecture Overview



Structure:

- Cascaded Control Architecture
- PID Control

Angular Rate Controller



Control Loop:

- KPID: Standard vs Parallel
- Avoid Integral Wind-up
- Derivative Lowpass Filter
- Derivative on measurement: eliminate derivative kick

Angular Rate Controller: Parameters and Modules

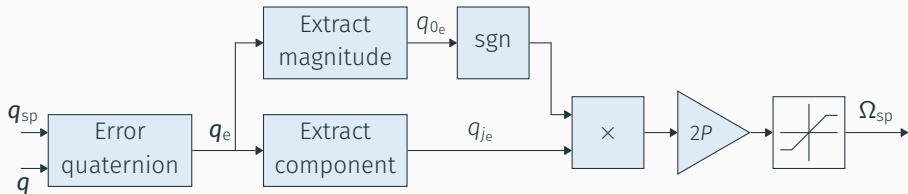
Parameters:

- *MC_ROLLRATE_K*: *K*
- *MC_ROLLRATE_P*: *P*
- *MC_ROLLRATE_I*: *I*
- *MC_ROLLRATE_D*: *D*
- *MC_RR_INT_LIM*: Integrator limit
- *IMU_DGYRO_CUTOFF*: Derivative LPF
- *IMU_GYRO_CUTOFF*: Measurement LPF

mc_rate_control/RateControl/RateControl.cpp

```
Vector3f RateControl::update(Vector3f &rate ,  
                             Vector3f &rate_sp ,  
                             Vector3f &angular_accel ,  
                             float dt ,  
                             bool landed)
```

Angle / Attitude Controller



Control Loop:

- Quaternion attitude representation
- Why? Euler angle singularity
- Paper: Nonlinear Quadcopter Attitude Control (2013) by Dario Brescianini, Markus Hehn and Raffaello D'Andrea
- Limit angular rates

Angle / Attitude Controller: Parameters and Modules

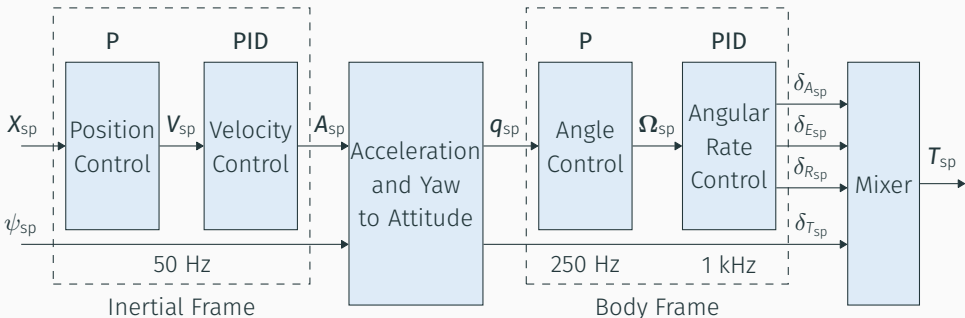
Parameters:

- *MC_ROLL_P*: *P*
- *MC_ROLLRATE_MAX*: Maximum roll rate
- *MC_ACRO_R_MAX*: Maximum roll rate in Acro mode
- *MC_YAW_WEIGHT*: Deprioritize yaw

mc_att_control/AttitudeControl/AttitudeControl.cpp

```
void AttitudeControl::setAttitudeSetpoint(Quatf &qd,  
                                            float yr_sp)  
Vector3f AttitudeControl::update(Quatf q)
```

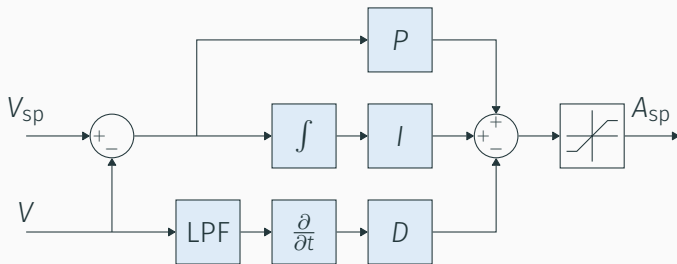
Acceleration and Yaw to Attitude



mc_pos_control/PositionControl

```
void PositionControl::_accelerationControl()  
void PositionControl::getAttitudeSetpoint(...)  
void ControlMath::thrustToAttitude(...)  
void ControlMath::bodyzToAttitude(...)
```


Velocity Controller



Control Loop:

- Prioritise vertical velocity
- Tracking anti-windup (based on paper)

Velocity Controller: Parameters and Modules

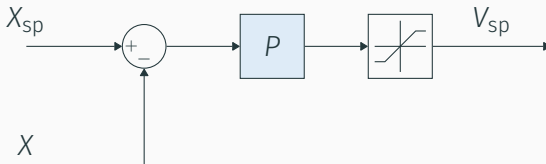
Parameters:

- $MPC_XY_VEL_P_ACC$: P_{xy}
- $MPC_XY_VEL_I_ACC$: I_{xy}
- $MPC_XY_VEL_D_ACC$: D_{xy}
- $MPC_Z_VEL_P_ACC$: P_z
- $MPC_Z_VEL_I_ACC$: I_z
- $MPC_Z_VEL_D_ACC$: D_z
- MPC_VELD_LP : Derivative LPF
- $MPC_TILTMAX_AIR$: Maximum pitch/roll angle
- MPC_THR_MAX : Maximum vertical thrust

`mc_pos_control/PositionControl/PositionControl.cpp`

```
void PositionControl::_velocityControl(const float dt)
```

Position Controller



Control Loop:

- P-control
- Limit velocity setpoints

Position Controller: Parameters and Modules

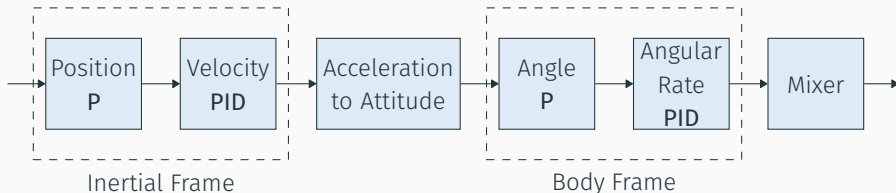
Parameters:

- MPC_XY_P : P_{xy}
- $MPC_XY_VEL_MAX$: Maximum XY velocity
- MPC_Z_P : P_z
- $MPC_Z_VEL_MAX_UP$: Maximum upward Z velocity
- $MPC_Z_VEL_MAX_DN$: Maximum downward Z velocity

mc_pos_control/PositionControl/PositionControl.cpp

```
void PositionControl::setInputSetpoint(...)
bool PositionControl::update(float dt)
void PositionControl::_positionControl()
```

Summary



More Information

- User Guide
 - MC PID Tuning Guide
- Developer Guide
 - Parameter Reference
 - PX4 Architectural Overview: Controller Diagrams (TBA)
- PX4 Dev Summit Talk: Overview of Multicopter Control from Sensors to Motors

Thank you!