Nonlinear MPC for Fixed-wing UAV Trajectory Tracking: Implementation and Flight Experiments

Nonlinear MPC for Fixed-wing UAV Trajectory Tracking: Implementation and Flight Experiments

Thomas Stastny*, Adyasha Dash†, Roland Siegwart‡
Swiss Federal Institue of Technology, (ETH) Zurich
Leonhardstrasse 21, 8092, Zurich, Switzerland

In this work we design a high-level Nonlinear Model Predictive Controller for lateral-directional fixed-wing UAV trajectory tracking in wind. Model identificaiton of closed loop low-level roll channel dynamics is conducted towards representing a low-order equivalent of the low-level autopilot response to high-level commands. We show trajectory tracking with various horizon lengths in high winds in simulation and demonstrate track convergence to sequential Dubins Car segments in flight experiments with a small autonomous unmanned aircraft using the designed algorithm. Discussion on appropriate objective formulation and weighting is given.

I. Introduction

Unmanned aerial robots are becoming ubiquitous in today's society. Whether from impactful public exposure such as demonstrations of large scale three-dimensional aerial modeling projects,[1] propositions of forward-looking postal delivery,[2] or increasing interest in airborne support for applications including disaster relief,[3,4] crop monitoring,[5] and infrastructure inspection,[6] the word "drone" (for better or worse) is now within the common vernacular. Particularly interesting platforms, relative to rotor-craft, for their longer endurance and speed in mapping and sensing applications, are fixed-wing unmanned aerial vehicles (UAVs). Even more advantageous, within the fixed-wing vehicle class, are small scale and hand-launchable platforms for their ease of deployment and minimal system complexity. While a wealth of work on advanced control algorithm design, specifically towards such systems, is present in recent literature, much still remains within the confines of simulation, and needs experimental validation.

One particular avenue within this realm is work on optimization-based approaches towards trajectory tracking. In particular, Nonlinear Model Predictive Control (NMPC) algorithms offer a broad range of possible formulations and applications. For instance, one of the earliest uses of NMPC for trajectory tracking control of unmanned fixed-wing aircraft can be found in the work of Kang et al.[7] Their cost function was designed to minimize the normal distance from a UAV to a desired line segment thereby turning the tracking problem into a regulation problem with an objective to drive the error to zero. The single line tracking cost function is then extended to allow the tracking of multiple line segments with obstacle avoidance. Kang et al. follow this up with stability analysis, and verification using hardware in the loop simulation.[8] Both works involve a kinematic model of an aircraft assuming planar motion and the existence of a low level controller to track the high level NMPC commands. Yang et al. further extend the previous two works formulating an adaptive NMPC for fixed-wing navigation through a cluttered environment, which varies the control horizon according to the path curvature profile for tight tracking.[9] By adding actuator slew limit to the optimization termination requirement in addition to the cost monotonocity, they show that the proposed optimization algorithm removes control input oscillations and tracks the trajectory more accurately than the conventional fixed horizon NMPC.

The path-following problem for fixed-wing UAV in presence of wind disturbances is addressed by Rucco et al.,[10] where the objective of trajectory tracking with minimum control effort is fulfilled for a planar aircraft

*Ph.D Candidate, Autonomous Systems Lab, thomas.stastny@mavt.ethz.ch
†M.Sc Student, Autonomous Systems Lab, adash@student.ethz.ch
‡Professor of Robotics and Intelligent Systems, Autonomous Systems Lab, rsiegwart@ethz.ch

model using a sample-data Model Predictive Control (MPC) approach. Extension to three dimensions is seen in[11] , where Gavilan et al. describe a high-level guidance algorithm based on MPC using a nonlinear 3DoF aircraft model for state prediction. The nonlinear optimization problem is then solved using a iterative scheme which uses a modified robust missile guidance law as hotstart to gaurantee feasibility.

As outlined from the literature, many possibilities exist for the use of NMPC in high-level fixed-wing UAV control. However, real-time implementation has been challenging up until recently, owing to the large computational complexity and time taken by the non-convex optimization problem posed. Efforts by researchers such as Quirynen et al. have focused on exploiting the structure of nonlinear models with linear subsystems using Implicit Runge Kutta methods, resulting in improved speeds of the solver.[12] The tools have since then been made available as a part of the ACADO toolkit,[13] allowing user-friendly implementation of system dynamics and objectives in a MATLAB or C interface, and generating real-time capable code. With improved computational runtimes, an opportunity now exists to more broadly field NMPC algorithms on small fixed-wing UAV platforms.

While many of the NMPC approaches cited tend only to consider vehicle kinematics, assuming lower-level loops will sufficiently track the higher-level commands, the concept of identifying lower level dynamics has promise in enhancing the performance of the controller. However, system identification of fixed-wing UAVs in the classical sense, i.e. open-loop analysis, is often off-putting to researchers as it can be time consuming, tedious, and moreover challenging when e.g. wind tunnel facilities are not available and flight test based identification must be applied. As opposed to open-loop analysis where the system response is obtained with no feedback control, e.g. a direct aileron→roll system model and/or an aileron→roll rate system model, the objective of closed-loop system analysis is to obtain the system response to the low-level controller's command inputs (e.g. stick deflections or roll reference commands from the high-level controller) to actual roll or roll rate output.

Early examples of system identification for control-augmented fixed-wing aircraft can be found in the works of Murphy,[14] and Mitchell et al.[15] Their work concerns gaining an insight into pilot comfort levels with the augmented systems of highly maneuverable fighter aircrafts by reducing the complex high order longitudinal and lateral directional dynamics to Low Order Equivalent Systems (LOES), thereby obtaining a quantitative measurement of pilot ratings, as well as possible explanations of unexpected aircraft behavior. As an example, Murphy uses rudder pedal force and input stick deflection as inputs, with sideslip angle, stability axis, roll rate, stability axis yaw rate, bank angle, and lateral acceleration as system states to identify the closed-loop lateral dynamics. The system is treated as a Multiple Input Multiple Output (MIMO) system and parameter estimation methods in Frequency domain are used. LOES models are typically fixed a priori whose structure can be found in this detailed report.[16] For all these systems, the input delays play a significant role in flight quality performance and pilot experience.

Morelli took a similar approach to identify the closed loop system for both longitudinal and lateral directions using LOES models.[17] However, unlike the previous example, SISO approach is favored here. Different types of maneuvers are executed, and it is found that certain multi-step maneuvers in combination with appropriate identification methods could be used instead of frequency sweeps, which often take longer times and are difficult to execute at high pitch angles.

In recent times, Luo et al.[18] identified the inner closed-loop system of the roll-channel of a small fixed-wing UAV as a first-order plus time delay model in order to design a fractional order $(PI)^{\lambda}$ controller. A fifth order ARX model is calculated first using least squares algorithms which is then approximated to a continuous time first order plus time-delay model.

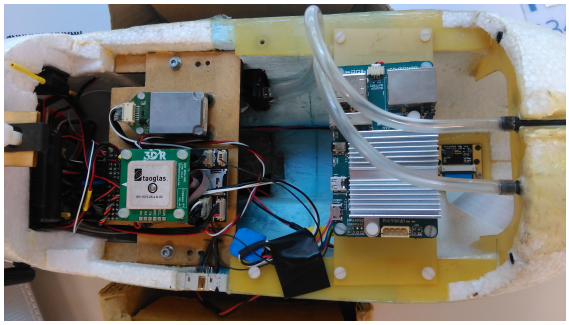
Capitalizing on new methods for auto-generation of fast NMPC code and the experience of previous authors in LOES modeling, we explore in this work the design of a high-level NMPC for real-time implementation on a small UAV, as well as simplified closed-loop identification procedures for modeling low order system equivalents of the low-level autopilot system. The identified model is incorporated into a modular high-level NMPC for general trajectory tracking in wind and verified through simulation and flight experiments.

II. System Overview

System identification, controller design, and flight experiments within this work are peformed on a small, 2.6 m wingspan, light-weight 2.65 kg, low-altitude, and hand-launchable fixed-wing UAV, Techpod, see Fig. 1a. The platform is a standard T-tail configuration, fixed-pitch, pusher propeller. Onboard avionics

including a 10-axis ADIS16448 Inertial Measurement Unit (IMU), u-Blox LEA-6H GPS receiver, and Sensirion SDP600 flow-based differential pressure sensor feed measurments to a Pixhawk Autopilot, an open source/open hardware project started at ETH Zurich.[19] Pixhawk features a 168 MHz Cortex-M4F microcontroller with 192 kB RAM for online state estimation and low-level control. A light-weight, robust Extended Kalman Filter runs onboard fusing sensor measurements to provide state estimates including a local three-dimensional wind vector, modeled statically with slow dynamics.[20] As Techpod flies at a nominal airspeed of $14\,\mathrm{m\,s^{-1}}$, the aircraft is easily susceptible to high winds present in low flight altitudes (below 500 m AGL), motivating explicit consideration of wind estimates within high-level position control.

(a) Techpod Test Platform



(b) Techpod Avionics/Hardware

As processing power on the Pixhawk microcontroller is somewhat limited, an additional onboard ODROID-U3 computer with 1.7 GHz Quad-Core processor and 2 GB RAM, running Robotic Operating System (ROS)[21] is integrated into the platform for experimentation with more computationally taxing algorithms. High-level controllers can be run within ROS node wrappers which communicate with the Pixhawk via UART serial communication; average communication latency was observed $<3\,\mu\mathrm{s}$, see Fig. 1b.

The control architecture implemented on Techpod in the current work can be seen in Fig. 2. The low-level autopilot, all processed onboard the Pixhawk, contains a standard cascaded PID structure with additional compensation for coordinated turns, i.e. a yaw damper signal, $r_r = \frac{g \sin\phi}{V}$. Attitude errors are fed to a PI block followed by rate errors running through a D block (proportional gain on rates), finally generating appropriate actuator commands δ_e, δ_a, δ_r (elevator, aileron, and rudder deflections, respectively). The EKF feeds back appropriate signals to each respective controller resulting in a stabilized closed loop low-level system. For the purposes of this work, we consider only the typical flight regimes employed in the vast majority of UAV missions, where flight is mostly planar, and maneuvers are mostly docile. Within this regime, the described PID control architecture is reasonably fit to track attitude commands for higher level trajectory following objectives.

High-level altitude and airspeed control is achieved with an implementation of the Total Energy Control System (TECS), also onboard the Pixhawk Flight Management Unit (FMU). Airspeed references are mostly tracked utilizing pitch commands, and altitude holds are maintained using throttle inputs δ_T. We assume altitude and airspeed are reliably tracked, and our focus lies with lateral-directional trajectory tracking. Detail into the last remaining blocks in the control architecture will be elaborated in Section IV.

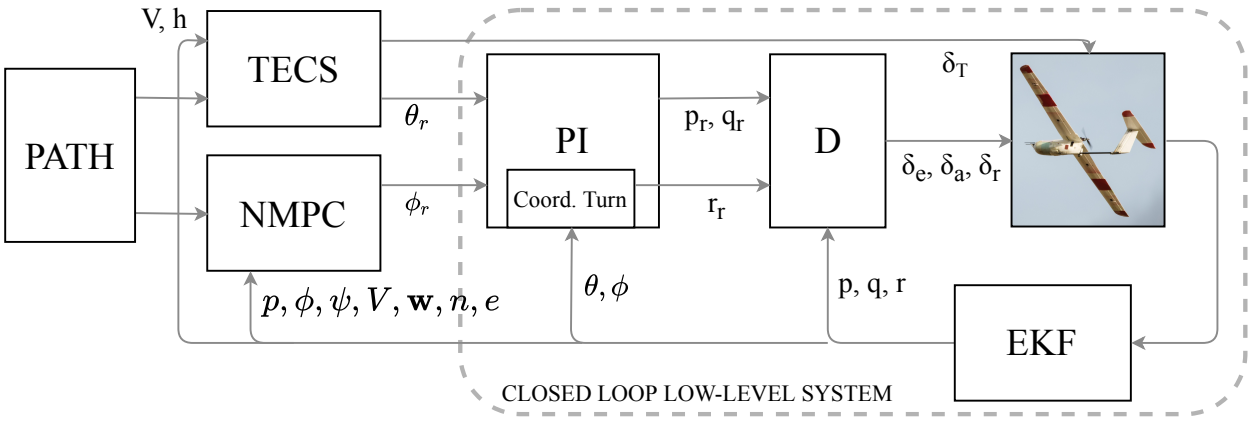


Figure 2: Control architecture

III. Flight Dynamics & Identification

In this section, we consider the lateral-directional kinematics of fixed-wing aircraft, expressed under a coordinated turn assumption, as follows in the inertial frame, $\mathcal{F}_I$:

$$\begin{aligned} \dot{n} &= V\cos\xi + w_n \\ \dot{e} &= V\sin\xi + w_e \\ \dot{\xi} &= \tfrac{g\tan\mu}{V} \end{aligned} \tag{1}$$

where n and e are the Northing and Easting positions, respectively, ξ is the heading, V is the air-mass-relative airspeed, μ is the bank angle, g is the acceleration of gravity, and w_n and w_e are the Northing and Easting components of the wind vector, respectively. Note that bank, μ and heading, ξ angles are defined about the air-mass-relative airspeed vector. This distinction is important when considering flight dynamics in wind, where the ground-relative flight path of the vehicle is defined as the course angle, χ from North, $\hat{n}$ to the ground speed vector, $\mathbf{v}_g$, see Fig. 6.

Making the assumption that the low-level controller is able to adequately regulate sideslip and altitude, i.e. the airspeed vector lies on the body-x axis, we may reasonably make the simplification of equating roll, ϕ and bank, μ angles, which are typically very close in value. This assumption is useful when prescribing attitude references to the low-level controller, which considers estimated body-axis, $\mathcal{F}_B$ defined Euler angles as feedback.

The kinematics described in equation (1) can be further enhanced with knowledge of the underlying low-level closed-loop plant dynamics. I.e. identifying a representable transfer function, $\phi/\phi_r(s)$ from the roll input, ϕ_r received by the low-level autopilot, to the resulting roll angle, ϕ measured in stabilized flight.

The objective of closed-loop system identification is to capture the dynamics of the aircraft with the low-level controller in the loop, which guarantees the flight stability of the UAV under various maneuvers. The low-level PID controller should be appropriately tuned before the system identification experiments, though to what degree the low-level loops perform is not necessarily important (outside of instability), as the identification should capture whatever dynamic is present for use in the high-level controller.

For identification of the roll channel, a series of multi-step inputs called 2-1-1 maneuvers were chosen. A 2-1-1 maneuver is a modified doublet input, which ideally consists of alternating pulses with pulse widths in the ratio 2-1-1. As demonstrated by Morelli,[17] flight time required for the 2-1-1 maneuver is approximately one-sixth of the time required for the standard frequency sweep maneuver, thus enabling one to gather more data in the same flight time, which is often limited by battery capacity. At the same time, concatenated 2-1-1 maneuvers make for suitable identification inputs for both frequency and time domain system identification approaches, at par with frequency sweeps. For our purposes, a unit pulse width of 1 s was chosen. Amplitude of these maneuvers were varied in the range allowed by the low-level controller, i.e., $|\phi| \in [0°, 30°]$.

Ordinarily, once in stabilized mode, the low-level controller receives its pitch and roll command references from a high-level controller, such as $\mathcal{L}_1$-Navigation (also known as Nonlinear Guidance Law, NLGL)[22] for lateral-directional path following and TECS (Total Energy Control System) for airspeed and altitude control. For system identification experiments, the autopilot source code was modified such that, once the system

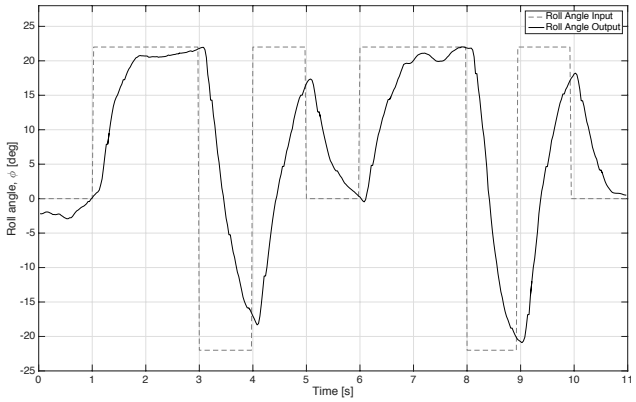


Figure 3: Sample system ID maneuver with respective roll angle output.

identification switch on the RC transmitter was on, pitch and roll commands from high level controller are over-written by commands for system identification maneuvers. For roll channel identification, pitch reference command to the low-level controller was kept at zero, while the roll reference command was varied as per the 2-1-1 maneuvers. An example can be seen in Fig. 3. With appropriate settling time added before and after the maneuvers, a set of two 2-1-1 maneuvers could be performed at a stretch, which facilitates concatenation of data later for system identification. Several different data sets are collected in this manner, including some with different roll amplitudes.

For system identification analysis, simple ARX models up to fifth order were chosen. Models beyond that order were likely to overfit the data, and additionally, would add too many additional states to the high-level controller, increasing computational complexity and size. For every data-set, 20 ARX models with different numbers of pole-zero combination were created. Additionally, each of the 20 models was also evaluated with and without a set of delays, resulting in 10 variations of every ARX model.

For system identification, time domain based Instrument Variables (IV) method is used considering the ease of use facilitated by ready to use computation tools, and guaranteed good results[18 17].

Once the individual ARX models with and without delays for a particular order model were evaluated, the best among them was chosen. This was then compared with similarly chosen ARX models of different combinations of zeros and poles. With every data-set, fifth order models typically gave best individual fits to the data sets, however second and third order models tended to generalize more and give better validation fits. Further, the addition of delay did not have any significant effect on improving the model fits to validation data. Since our NMPC is formulated in continuous time (see more information in Section IV), and modeling of a delay there is more difficult, inclusion of delay for the sake of an insignificant increase in model fit is not justified. The models of a particular order with the best fits among the different models of the same order are shown in Fig. 5

Considering all the factors, namely ease of augmentation to the system dynamics and the size of computational complexity thereby, and good model fits to validation data-sets, a second order model of the following form was chosen.

$$\frac{b_0}{s^2 + a_1 s + a_0} \tag{2}$$

From this model, the kinematics in equation (1) are augmented in the time domaim with two differential equations describing the identified second order roll channel dynamics. Note, again, that we assume $\mu \approx \phi$.

$$\begin{pmatrix} \dot{\mu} \\ \dot{\mu} \end{pmatrix} = \begin{pmatrix} \dot{\mu} \\ b_0 \mu_r - a_1 \dot{\mu} - a_0 \mu \end{pmatrix} \tag{3}$$

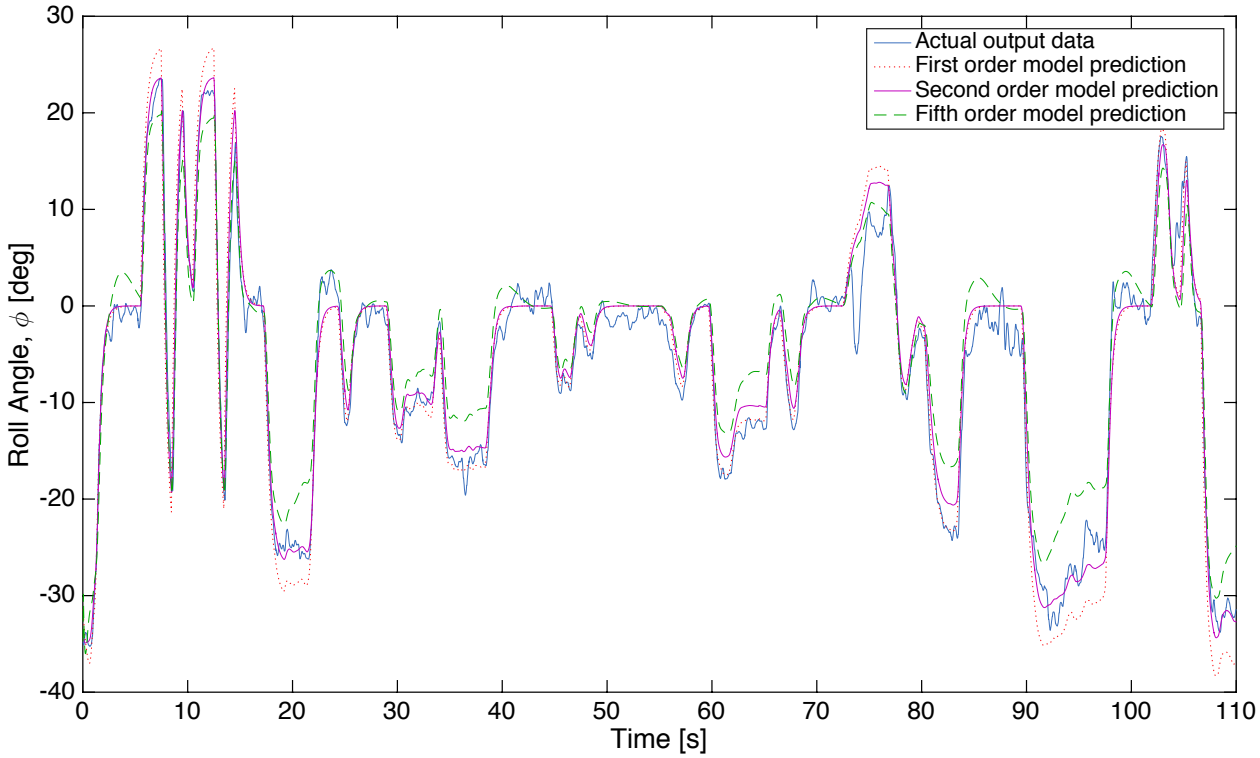


Figure 4: Actual roll angle output vs. time response of various models.

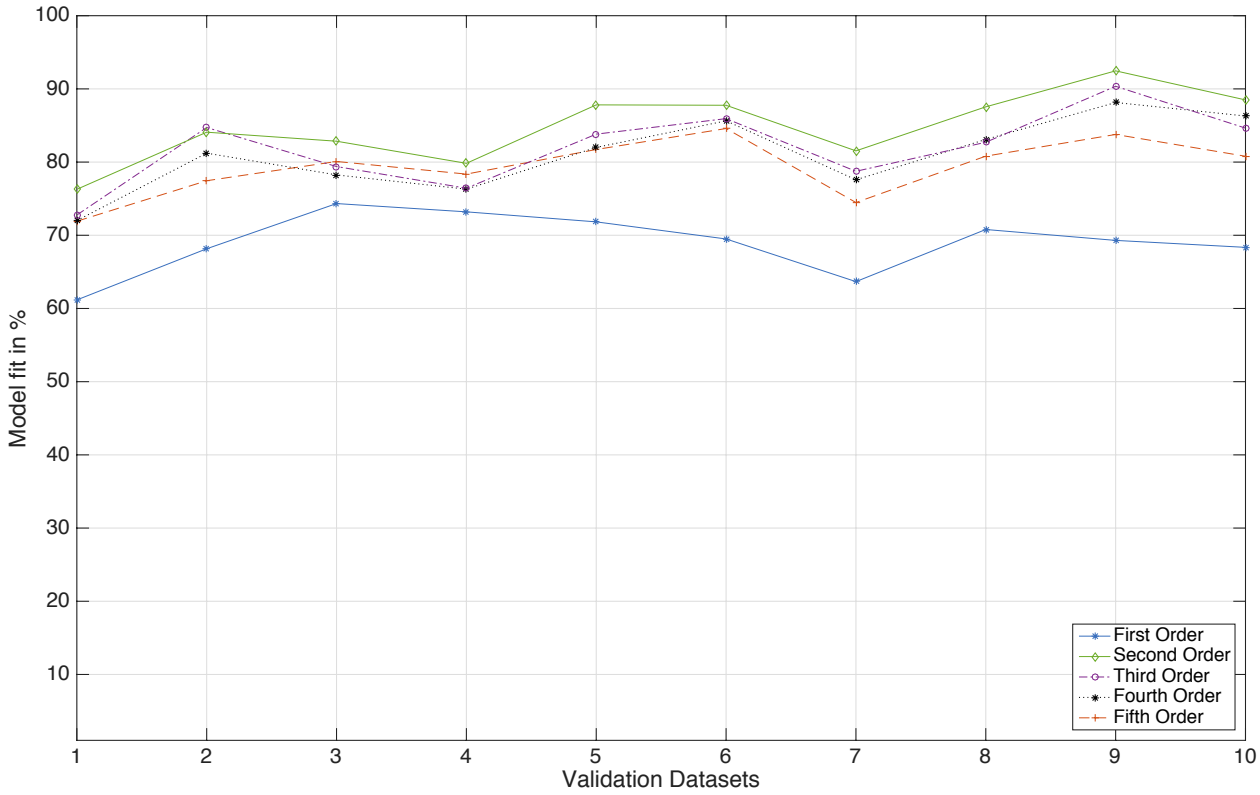


Figure 5: Validation fits on different data sets with different order models.

IV. Nonlinear Model Predictive Control

In this section, we formulate a high-level lateral-directional trajectory tracking controller in a Nonlinear Model Predictive Control (NMPC) scheme. A general objective is constructed for minimizing the position error to a given track,

$$e_t = (\mathbf{d} - \mathbf{p}) \times \bar{\mathbf{T}}_d \tag{4}$$

where $\bar{\mathbf{T}}_d$ is the unit tanget vector at the closest point $\mathbf{d}$ from the UAV position $\mathbf{p}$ to the current path, while also aligning the vehicle course with the desired trajectory direction, i.e. minimize

$$e_\chi = \chi_d - \chi \tag{5}$$

where $\chi_d = \text{atan2}\left(\bar{T}_{d_e}, \bar{T}_{d_n}\right) \in [-\pi, \pi]$. Here, we use the atan2 function from the standard C math library. See also Fig. 6. Use of this general objective formulation allows inputing any path shape, so long as the nearest point from the UAV position can be calculated and a direction of motion along the path (i.e. the path tangent) is given for minimization throughout the horizon. In this paper, we limit our discussion *circle* and *line* following, geometry for which finding the closest point in the two-dimensional plane has a simple analytic form, the calculation of which will be omitted for brevity.

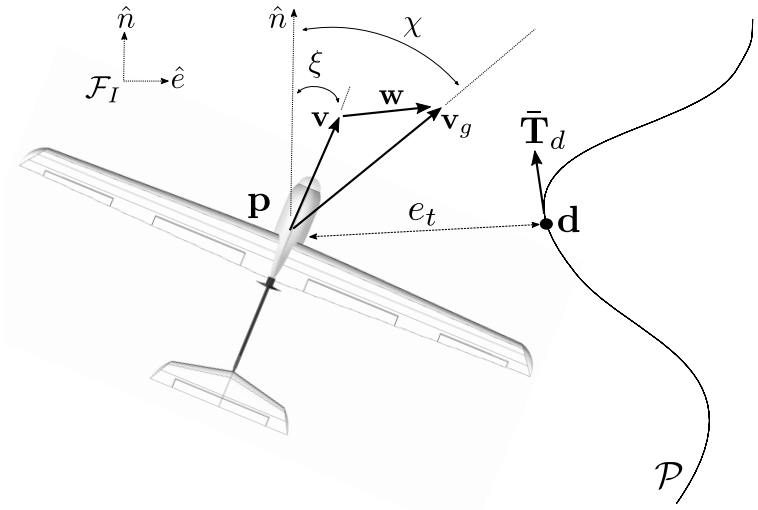


Figure 6: Inertial Frame Definitions

We define the state vector $\mathbf{x} = [n, e, \mu, \xi, \dot{\mu}, x_{sw}]^T$, and control input $\mathbf{u} = \mu_r$, where the augmented state x_{sw} is a switch state used within the horizon in the case that desired trajectories are piece-wise continuously, or generally discretely, defined. The switch variable has no dynamic until a switch condition is detected within the horizon, at which point an arbitrary differential is applied for the remainder of the horizon calculations. I.e.

$$\dot{x}_{sw} = \begin{cases} \alpha & \text{switch condition met} \quad \| \quad x_{sw} > \text{threshold} \\ 0 & \text{else} \end{cases} \tag{6}$$

The switch state, then, either has a value of zero (i.e. the aircraft has not met the switching condition), or some value greater than zero, at which point the internal model within the controller will switch to tracking the next path throughout the remainder of the control horizon. When a track switch has been fully achieved (i.e. with respect to the actual current aircraft position/velocity), the value of the switch state is reset to zero throughout the horizon.

A relevant example of such a case for fixed-wing vehicles is that of Dubins Car or Dubins Aircraft, in the three-dimensional case, path following, see[23] . Dubins paths can be used to describe the majority of desired flight maneuvers in a typical fixed-wing UAV mission. Further, using continuous curves such as arcs and lines allow time-invariant trajectory tracking, as oppose to desired positions in time, a useful quality when

only spatial proximity to the track is desired and timing is less important; for instance, if energy conservation is required and a single low airspeed reference is given to be tracked. For the remainder of the paper, we will consider Dubins segments as path inputs to the high-level controller, though it should be noted that the objective formulation is not limited to these.

We use the ACADO Toolkit[13] for automatic generation of fast, embedded C-code implementations of nonlinear solvers and integration methods. Though formulated in continuous time, a direct multiple shooting technique is used to solve the optimal control problem (OCP), where dynamics, control action, and inequality constraints are discretized over a time grid of a given horizon length N. A boundary value problem is solved within each interval (or shooting node) and additional continuity constraints are imposed. Sequential Quadratic Programming (SQP) is used to solve the individual Quadratic Programs using the active set method implemented in the qpOASES solver.[24] A Gauss-Newton based real-time iteration scheme is used which iteratively improves the current online solution during each step of the process during runtime.[25] Note that the ACADO framework does not explicitly support non-smooth functions. Here, we exploit an option to use externally defined C-based model and objective functions. Numerical jacobians are implemented using a finite difference. Both control and numerical stability properties for this implementation are not guaranteed; however, extensive simulation studies and flight experimentation have to-date shown no ill effects, provided control input constraints are set. A more robust investigation into possible instabilities is to be conducted in future work.

The OCP takes the following continuous form:

$$\begin{aligned}
\min_{U,X} \quad & \int_{t=0}^{T} \Big((\mathbf{y}(t) - \mathbf{y}_{ref}(t))^T \, \mathbf{Q} \, (\mathbf{y}(t) - \mathbf{y}_{ref}(t)) + (\mathbf{u}(t) - \mathbf{u}_{ref}(t))^T \, \mathbf{R} \, (\mathbf{u}(t) - \mathbf{u}_{ref}(t)) \Big) \, dt \\
& + (\mathbf{y}(T) - \mathbf{y}_{ref}(T))^T \, \mathbf{P} \, (\mathbf{y}(T) - \mathbf{y}_{ref}(T)) \\
\text{subject to} \quad & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad \text{(equations (1) \& (3) \& (6))} \\
& \mathbf{y} = h(\mathbf{x}, \mathbf{u}) \\
& \mathbf{u}(t) \in \mathbb{U} \\
& \mathbf{x}(0) = \mathbf{x}\,(t_0)\,.
\end{aligned} \tag{7}$$

where $\mathbf{y} = [e_t, e_\chi, \mu, \dot{\mu}, \mu_r]^T$ and $\mathbb{U}$: $\mu_{r_{min}} \leq \mu_r \leq \mu_{r_{max}}$. Here, μ_r is included in the objective function doubly; once within the standard control penalty (i.e. $(\mathbf{u}(t) - \mathbf{u}_{ref}(t))^T \, \mathbf{R} \, (\mathbf{u}(t) - \mathbf{u}_{ref}(t))$), and again within $\mathbf{y}$, allowing the formulation of a slew rate $\Delta\mu_r(t) = \mu_r(t) - \mu_{r_{k-1}}(t)$, i.e. the devation from the previous horizon control solution, which may be penalized by the weight component $Q_{\Delta\mu_r}$. The previous control horizon $\mu_{r_{k-1}}$ is stored after the last NMPC iteration step and input as a reference value during the next. Note, this is not identical to the typical slew rate penalty often utilized in discrete MPC formulations, but actually a comparison at each shooting node to the previous control solution at that same node. The difference between each subsequent time step within the horizon in a given optimization step is not considered. This penalty is implemented to penalize bang-bang control action caused by unregulated action within the first shooting node between each NMPC iteration. In particular, we are interested in relatively long horizons for considering optimal path convergence in various windy scenarios where the ground speed may grow faster than the chosen horizon allows adequate reaction time. Longer horizons can be achieved without overly increasing the horizon length, and thus the dimensionality of the problem, by the use of larger discretization steps, i.e. we use T_{step}=0.1 s. This, however, in turn exacerbates the mentioned issue of bang-bang control action, as the next NMPC iteration step measurement can possibly deviate enough to induce a large control step. An example of the issue can be seen in Section V. As the next action in the control horizon is applied to the vehicle at each iteration, the deviation in the first few shooting nodes should be penalized, but not the latter steps, as this would result in an overall sluggish control performance. To remedy this discrepancy, a decreasing quadratic function is defined in the control deviation weight horizon, so that latter nodes are not penalized, and earlier nodes are.

Other online parameters augmented to the model and held constant through the horizon are the current airspeed V, current wind vector $\mathbf{w}$, and the current and next sets of Dubins path parameters $\mathcal{P}_{cur}$, $\mathcal{P}_{next}$, where line parameters include $\mathcal{P} = \{type = 0, \mathbf{a}, \mathbf{b}\}$, $\mathbf{a}$ and $\mathbf{b}$ are two waypoints defining a straight segment, and arc parameters include $\mathcal{P} = \{type = 1, \mathbf{c}, R, dir, \xi_0, \Delta\xi\}$, $\mathbf{c}$ is the center point of the arc, R is the radius, dir is the loiter direction, and ξ_0, is the heading pointing towards the entrance point on the arc, and $\Delta\xi$ is the arclength traveled. The path segments are managed and rotated based on an acceptance radius and heading direction criteria.

V. Simulations & Flight Experiments

In this section, we present simulation results as well as real world flight experiments with the Techpod test platform using the designed NMPC with the identified low-level model.

Before testing on the platform, extensive two-dimensional simluation of the kinematics with augmented second order bank dynamics was carried out to obtain a rough tuning of objective weights as well as experiment with various magnitudes of wind. A sample of the simulation findings is shown. Displayed simluation results were obtained using objective weights $\mathbf{Q}_{diag} = [0.01, 1, 0.1, 0.01, 100]$, $\mathbf{R}_{diag} = 10$ and end term objective weights $\mathbf{P}_{diag} = [0.1, 10, 0, 0.01, 0]$. The discretization time step within the horizon is T_{step}=0.1 s, and the NMPC is iterated every 0.05 s. In Fig. 7, two simulations, one with a control horizon length of $N = 40$, set as a minimum to ensure capturing an entire 90° turn at maximum bank, and the other with $N = 80$ are initialized at the same position and orientation and commanded to track a circle in high wind. The vehilces in the simulation were set to an airspeed of $14\,\mathrm{m\,s^{-1}}$. Further, the desired loiter radius is smaller than the minimum trackable radius at the given flight speed plus added wind component. Thus, as shown, the high winds cause a large deviation from the track when the UAVs fly down wind. The UAV with the longer horizon is able to forsee the future deviation, and plans an adverse control action earlier in the loiter to enable less deviation over the remainder of the horizon. Despite the windy conditions, both horizons, however, are able to converge initially to the circle when feasible. Further, bank angle rates are within acceptable limits, and control input constraints are respected.

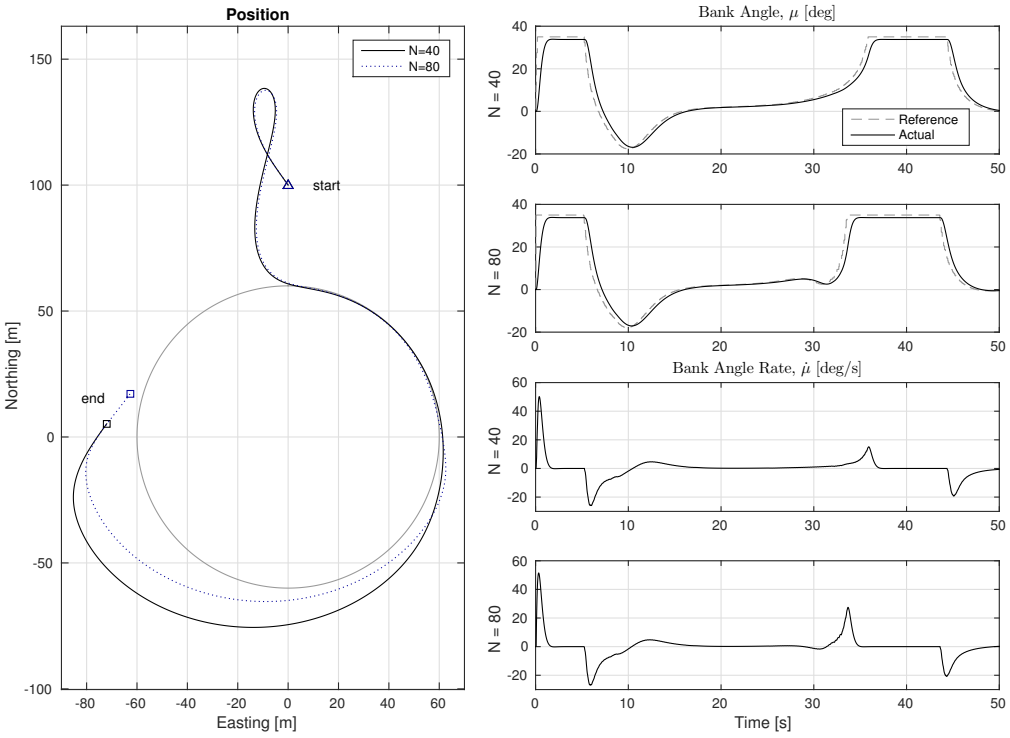


Figure 7: $N = 40$ and $N = 80$ length horizons for circle tracking in $w_e = -10\,\mathrm{m\,s^{-1}}$.

Fig. 8 demonstates the possible bang-bang effect when the deviation of the control horizon from the previous solution is not penalized. The displayed control solutions are take from the same simulation shown in Fig. 7.

After testing the controller in simulation. Two flight experiments were conducted to demonstrate various trajectory following peformance. A horizon length of $N = 40$ was used with objective weights and end term objective weights set to $\mathbf{Q}_{diag} = \mathbf{P}_{diag} = [0.01, 10, 0.1, 0.01, 100]$, $\mathbf{R} = 10$. The discretization time step within the horizon is T_{step}=0.1 s, and the NMPC is iterated every 0.05 s. Solve times for the NMPC running on the ODROID-U3 vary, depending on the type of path $\mathcal{P}_{cur}$ the aircraft is following, as well

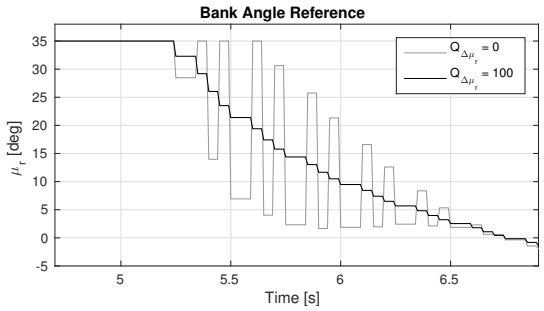


Figure 8: Effect of horizon slew weight on control action.

as the next path $\mathcal{P}_{next}$ in the trajectory, if there are switches within the given horizon length. Table 1 shows the mean, standard deviation, and maximum computation times for the controller running during the Dubins path following experiment (see Figure 11). Note these times include the solver as well as required message handling, path management, and data conversion executed within each iteration of the ROS node. Computation times are taken from portions of flight data wherein the aircraft is only tracking a line or circle, respectively, throughout the control horizon. Switching within the horizon results in increases or decreases in computations depending on which path is being tracked and for what portion of the horizon (e.g. circle tracking times will decrease as a switch to a line path is observed within the control horizon). Both experiments took place during very calm conditions, and the wind speed was negligible.

Table 1: NMPC computational load on ODROID-U3.

	Mean [ms]	Standard Dev. [ms]	Maximum [ms]
Line following	9.96	0.250	10.5
Circle following	13.5	0.439	15.1

In Fig. 9, Techpod is commanded towards a box pattern until returning to a loiter circle. Minimal overshoot is observed, considering the set acceptance radius of 35 m, and convergence within less than 1 m of position error is observed for each line segment and the final loiter circle. Figure 10 shows the commanded and actual roll angles as well as the roll rate, which are both kept within acceptable bounds.

In Fig. 11, an arbitrary sequence of Dubins segments were given to the high-level NMPC. Again, good convergence to the path is seen, with acceptable state responses shown in Fig. 12. Steady-state position error remained within 1 m after convergence to the path. Note this is without the inclusion of integral action, and either model uncertainties, or variable wind conditions perhaps not properly estimated, or gusts could cause larger track errors. The end of the shown flight path is stopped just before converging to the final loiter due to rain fall starting during the flight experiment and manual take-over of the aircraft for landing.

It should be noted that each flight experiment shown here was also flown separately with $\mathcal{L}_1$ guidance for roll command generation, achieving similar performance in these non-windy conditions. Our focus in this work, however, is not on comparing methods, but verfiying the feasibility of the given NMPC formulation in real flights. In higher wind scenarios such as those shown in simulation, similar control performance with $\mathcal{L}_1$ loops would require some form of wind-dependent gain scheduling. Similar scheduling would also likely be required for vector-field based approahes, notorious for being somewhat difficult to tune.[23,26,27]

VI. Conclusions & Future Work

In this paper, we outlined an approach for low order equivalent system modeling and identification of control-augmented low-level roll channel dynamics for a small fixed-wing UAV and, further, demonstrated the importance of their inclusion within the model of a high-level Nonlinear Model Predictive Controller. The control-augmented model identification process was observed to significantly decrease identifying flight time, as well as simplify the resulting model structure, when compared with open-loop, low-level aerodynamic identification. Open-loop simulation of the identified dynamics also demonstrated predictable behavior, even within long horizons, due to the stabilized dynamics; a useful trait for high-level controllers.

An NMPC was designed for Dubins car path following in the two-dimensional lateral-directional plane,

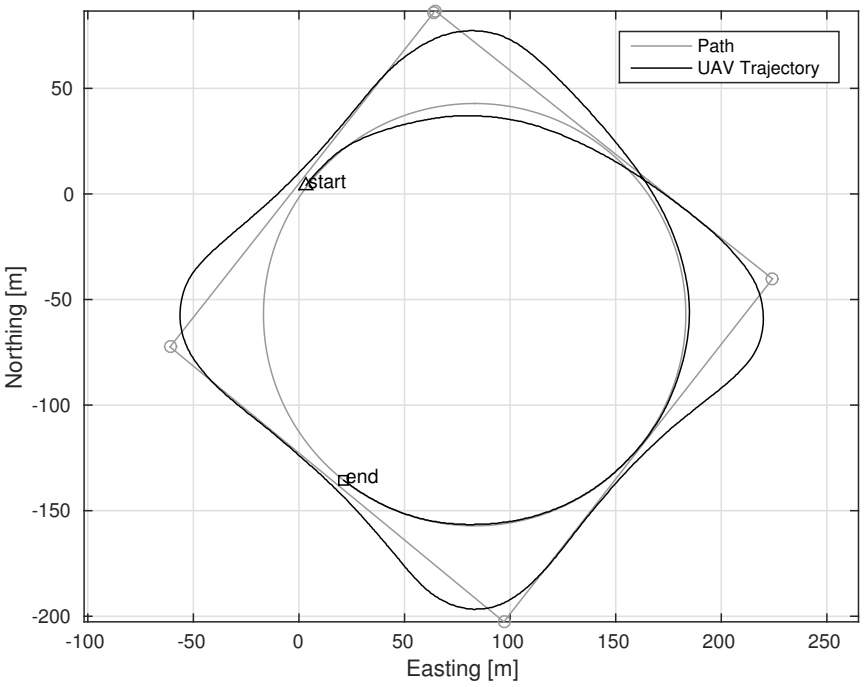


Figure 9: Flight experiment: box tracking.

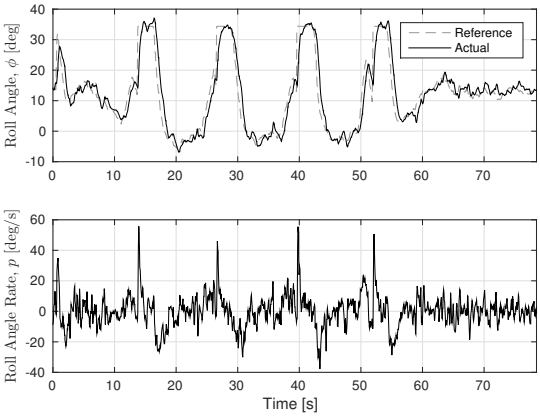


Figure 10: Flight experiment: box tracking attitude and rates.

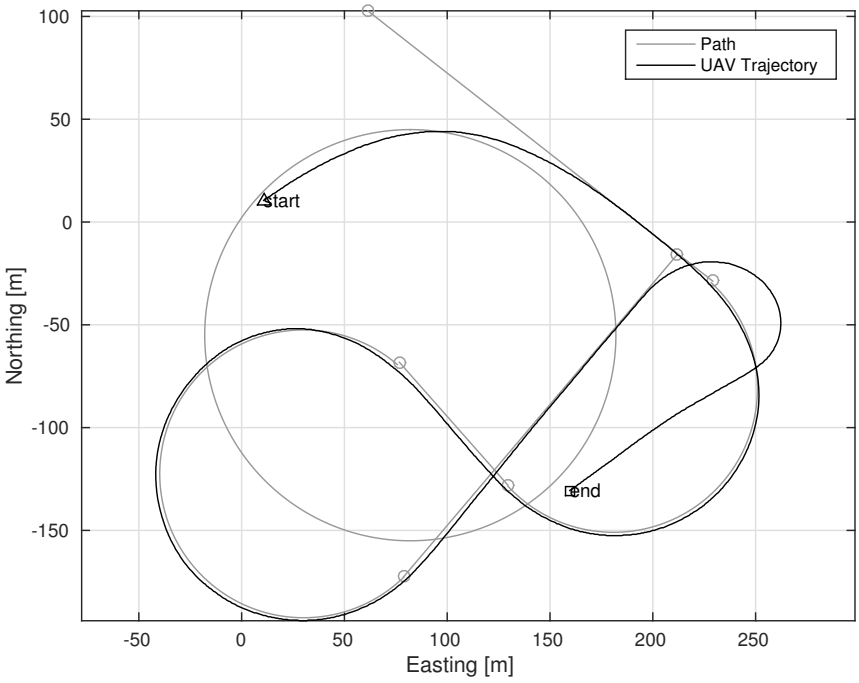


Figure 11: Flight experiment: Dubins tracking.

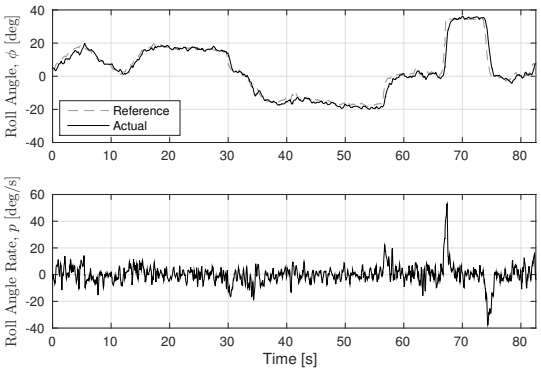


Figure 12: Flight experiment: Dubins tracking attitude and rates.

and was shown capable of good tracking performance, even in high wind conditions (through simulation experiments), and arbitrary path combinations, shown in flight experiments. Design of the objective function was elaborated towards avoiding bang-bang control action, and including track switching behavior within the control horizon. Computation times onboard the ODROID-U3 were observed to be well within feasible limits for online solutions of adequate high-level command generation (or guidance). Horizon lengths up to eight seconds were investigated in simulation, and four seconds within flight experiments, showing utility in the determination of optimal flight path convergence to a given track in strong wind.

Though promising results, alternative objective functions could easily be designed and tested as off-shoots from the given basic formulation, e.g. inclusion of obstacle avoidance, or more complex paths. The true benefit of such an online optimization-based approach to control is the modularity of the cost function, as well as the inherent online adaptability to changing environmental conditions. It is this point where more simple analytic approaches, e.g. $\mathcal{L}_1$ navigation or vector field based approaches, often require gain scheduling per condition, a process which takes time to properly tune.

Further future work will focus on extending the approach to three dimensions, with longitudinal dynamics also included into the formulation. Implications of a three-dimensional controller on low-level model identification would likely necessitate MIMO Linear Time Invariant (LTI) plants, or possibly nonlinear model structures, as longitudinal aircraft dynamics, even when stabilized, typically vary with airspeed and angle of attack. Last, future work should include stability analysis, both in the control algorithm, as well as the numerical methods used to solve the sometimes non-smooth optimization problem posed.

Acknowledgements

This work was supported by the European Commission project SHERPA (#600958) under the 7th Framework Programme.

References

[1]Pix4D, 2015, http://pix4d.com/.

[2]The Local CH, 2015, http://www.thelocal.ch/20150419/swisspost-launches-tests-for-drone-deliveries.

[3]Falcon UAV, 2015, http://www.falconunmanned.com/.

[4]SHERPA Project, 2015, http://www.sherpa-project.eu/.

[5]Hunt, E. R., Hively, W. D., Fujikawa, S. J., Linden, D. S., Daughtry, C. S. T., and McCarty, G. W., "Acquisition of NIR-Green-Blue Digital Photographs from Unmanned Aircraft for Crop Monitoring," *Remote Sensing*, Vol. 2, No. 1, 2010, pp. 290–305.

[6]Metni, N. and Hamel, T., "A UAV for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, Vol. 17, No. 1, November 2007, pp. 3–10.

[7]Kang, Y. and Hedrick, J. K., "Design of nonlinear model predictive controller for a small fixed-wing unmanned aerial vehicle," *AIAA Guidance, Navigation, and Control Conference*, 2006.

[8]Kang, Y. and Hedrick, J. K., "Linear tracking for a fixed-wing UAV using nonlinear model predictive control," *Control Systems Technology, IEEE Transactions on*, Vol. 17, No. 5, 2009, pp. 1202–1210.

[9]Yang, K., Sukkarieh, S., and Kang, Y., "Adaptive nonlinear model predictive path tracking control for a fixed-wing unmanned aerial vehicle," *AIAA Guidance, Navigation & Control Conference*, 2009.

[10]Rucco, A., Aguiar, A. P., Pereira, F. L., and de Sousa, J. B., "A Predictive Path-Following Approach for Fixed-Wing Unmanned Aerial Vehicles in Presence of Wind Disturbances," *Robot 2015: Second Iberian Robotics Conference*, Springer, 2016, pp. 623–634.

[11]Gavilan, F., Vazquez, R., and Camacho, E. F., "A High-level model predictive control guidance law for unmanned aerial vehicles," *Control Conference (ECC), 2015 European*, IEEE, 2015, pp. 1362–1369.

[12]Quirynen, R., Gros, S., and Diehl, M., "Efficient NMPC for nonlinear models with linear subsystems," *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, IEEE, 2013, pp. 5101–5106.

[13]Houska, B., Ferreau, H., and Diehl, M., "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, Vol. 32, No. 3, 2011, pp. 298–312.

[14]Murphy, P. C., "Closed-loop system identification experience for flight control law and flying qualities evaluation of a high performance fighter aircraft," 1996.

[15]Mitchell, D. G. and Hoh, R. H., "Low-order approaches to high-order systems: Problems and promises," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, 1982, pp. 482–489.

[16]Morelli, E. A., "Piloted Parameter Identification Flight Test Maneuvers for Closed Loop Modeling of the F-18 High Alpha Research Vehicle (HARV)," 1996.

[17]Morelli, E. A., "Low-order equivalent system identification for the Tu-144LL supersonic transport aircraft," *Journal of guidance, control, and dynamics*, Vol. 26, No. 2, 2003, pp. 354–362.

[18]Luo, Y., Chao, H., Di, L., and Chen, Y., "Lateral directional fractional order (PI) π control of a small fixed-wing

unmanned aerial vehicles: controller designs and flight tests," *Control Theory & Applications, IET*, Vol. 5, No. 18, 2011, pp. 2156–2167.

[19]Pixhawk Autopilot, 2015, http://pixhawk.org/.

[20]Leutenegger, S., Melzer, A., Alexis, K., and Siegwart, R., "Robust State Estimation for Small Unmanned Airplanes," *IEEE Multi-conference on Systems and Control*, 2014.

[21]Robotic Operating System, 2016, http://ros.org/.

[22]S. Park, J. Deyst and J. P. How, "Performance and Lyapunov stability of a nonlinear path following guidance method," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007, pp. 1718–1728.

[23]Beard, R. W. and McLain, T. W., "Implementing dubins airplane paths on fixed-wing UAVs," *Contributed Chapter to the Springer Handbook for Unmanned Aerial Vehicles*, 2013.

[24]Ferreau, H., Potschka, A., and Kirches, C., "qpOASES webpage," http://www.qpOASES.org/, 2007-2015.

[25]Houska, B., Ferreau, H. J., and Diehl, M., "An auto-generated real-time iteration algorithm for nonlinear {MPC} in the microsecond range," *Automatica*, Vol. 47, No. 10, 2011, pp. 2279 – 2285.

[26]Stastny, T., Garcia, G., and Keshmiri, S., "Robust Three-Dimensional Collision Avoidance for Fixed-Wing Unmanned Aerial Systems," *AIAA Guidance, Navigation, and Control Conference (GNC), AIAA SciTech Forum*, 2015.

[27]Stastny, T., Garcia, G., and Keshmiri, S., "Collision and Obstacle Avoidance in Unmanned Aerial Systems Using Morphing Potential Field Navigation and Nonlinear Model Predictive Control," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 137, No. 1, 2014.