Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing

Siri H. Mathisen,[*] Kristoffer Gryte,[†] Thor I. Fossen,[‡] and Tor A. Johansen[‡]

Department of Engineering Cybernetics

Norwegian University of Science and Technology

Trondheim, Norway

This article presents the use of constrained non-linear model predictive control (NMPC) for high-precision deep stall landing of a fixed-wing unmanned aerial vehicle (UAV), through simulations in six degrees of freedom. The control algorithm takes the UAV horizontally in the correct direction of the landing target until the longitudinal line of sight angle from the UAV to the landing target has reached a given limit. Then it descends the UAV directly towards the landing target, keeping a steep flight path and enforcing a deep stall. The performance of the landing scheme has been evaluated with different path angles and wind velocities, where wind measurements from the last time step are assumed known by the NMPC. This showed acceptable landing performance in winds up to 7 m/s, and indicated that steeper flight paths are more susceptible to wind disturbances. To challenge the assumption of fast and accurate wind estimates, a simulation where the gust wind was assumed unknown by the NMPC was performed. This reduces the performance of the landing controller, but the majority of the feasible paths successfully land in a deep stall within 2 meters of the target.

Nomenclature

p_n	Inertial north position of the UAV
p_e	Inertial east position of the UAV
p_d	Inertial down position of the UAV
target_n	Inertial north position of the landing target
target_e	Inertial east position of the landing target
target_d	Inertial down position of the landing target
u	Body frame velocity measured along the i^b axis
v	Body frame velocity measured along the j^b axis
w	Body frame velocity measured along the k^b axis
NED	North-east-down
$\mathbf{R}_b^n$	Rotation matrix from body frame to NED frame
$\mathbf{R}_n^b$	Rotation matrix from NED frame to body frame
$\mathbf{q}$	Unit quaternions, components are denoted as η,ε_1,ε_2,ε_3
p	Roll rate along the body frame x-axis
q	Roll rate along the body frame y-axis
r	Roll rate along the body frame z-axis
M	The mass of the vehicle
Γ_i	Function of the moments and products of inertia
$\mathbf{f}$	External forces acting on the body frame, components denoted as f_x, f_y, and f_z

[*] Ph.D. candidate, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

[†] Ph.D. candidate, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, AIAA Student Member, `kristoffer.gryte@itk.ntnu.no`

[‡] Professor, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, AIAA Member

$\mathbf{m}$	External moment acting on the body frame, components denoted as l, m, and n
J_y	Moment of inertia about the body frame y-axis
$\mathbf{f_g}$	Gravitational forces acting on the body frame
$\mathbf{f_a}$	Aerodynamic forces acting on the body frame
$\mathbf{f_p}$	Propulsional forces acting on the body frame
g	Acceleration of gravity
ρ	Air density
α	Angle of attack
β	Sideslip angle
C_X	Function of C_D, C_L, and α
C_{X_q}	Function of C_{D_q}, C_{L_q}, and α
$C_{X_{\delta_e}}$	Function of $C_{L_{\delta_e}}$, $C_{D_{\delta_e}}$, and α
C_Z	Function of C_D, C_L, and α
C_{Z_q}	Function of C_{D_q}, C_{L_q}, and α
$C_{Z_{\delta_e}}$	Function of $C_{D_{\delta_e}}$, $C_{L_{\delta_e}}$, and α
C_{Y_*}	Aerodynamic force coefficient acting on the body frame along the y-axis
S_{prop}	Area swept by the propeller
C_{prop}	Aerodynamic propeller coefficient
k_{motor}	Motor efficiency coefficient
V_a	Airspeed, components denoted as u_r, v_r, w_r
C_{l*}	Aerodynamic roll moment coefficient acting on the body frame along the x-axis
C_m	Pitch moment coefficient
C_{m*}	Aerodynamic pitch moment coefficient acting on the body frame along the y-axis
C_{n*}	Aerodynamic yaw moment coefficient acting on the body frame along the z-axis
k_{T_p}	Propeller torque coefficient
k_ω	Propeller speed coefficient
S	Wing surface area
δ_a	Aileron deflection in radians
δ_e	Elevator deflection in radians
δ_r	Rudder deflection in radians
δ_t	Control signal for throttle
γ	Path angle
χ	Course angle
LOS	Line of Sight
$\mathbf{V}_w^b$	Wind velocity in the body frame, consists of body frame gust components $(u_{w_g}, v_{w_g}, w_{w_g})$ and NED frame steady components $(w_{n_s}, w_{e_s}, w_{d_s})$

Subscript

i	Variable number, $i = 1..8$
$*$	Coefficient dependency, $* = 0, \beta, \delta_a, \delta_e, \delta_r, p, q, r$

Superscript

$*$	Desired value

I. INTRODUCTION

To recover a small UAV in a small space without a runway, belly landing on a soft surface or an arrest system like a recovery net can be used. It is desirable to ease the impact by minimizing the speed at which the UAV meets the landing target. One method for achieving this is to land the UAV in a deep stall. In this case, the UAVs angle of attack needs to go beyond the stall angle, where the drag coefficient of the UAV increases and the lift coefficient decreases. This makes the UAV loose altitude while at the same time losing speed in the horizontal direction. Since control authority is reduced, it is difficult to control a UAV in a deep stall, and side winds or turbulence might cause the UAV to spin or lose control. It is also important to be able to accurately guide the UAV at the same time as it is being decelerated, to be able to accurately hit the landing target.

Earlier work with net landing is described and tested experimentally several times in the literature: In Ref. 1, a

Skywalker X8 flying wing UAV is successfully guided into a recovery net with 1 meter vertical and lateral accuracy using a low-cost single-frequency RTK GNSS navigation system. In Ref. 2, net recovery is described and simulated with a small fixed-wing UAV using a spiral landing trajectory. The landing is divided in two phases, where the first one consists of either stationary or moving spirals to reduce altitude, and then a pseudo pursuit guidance law guides the UAV into the net with a small impact angle. The method was simulated, but no experiments were described in the article. In Ref. 3, a vision-based net-recovery technique is described and confirmed with simulations and experimental results. The platform used is a radio-controlled airplane called Hobbyking FPV with a camera system mounted in front. The UAV is guided along a direct line towards the recovery net, tracking a desired flight path angle defined as the line-of-sight angle between the UAV and the recovery net. The experimental results show that the net is detected by the vision system and that the UAV is safely guided into the recovery net. In Ref. 4, several autonomous net recovery approaches were evaluated through tests on stationary and moving nets, both on the ground and on water. The UAV, of type Silver Fox, successfully hit the stationary net, the moving net, and the net on water, using Differential Global Positioning System (DGPS). The recovery net folds itself around the UAV to prevent it to fall out of the net after impact, and in the last tests, no damage occurred to neither UAV nor net. The UAV flies nose first into the net, at a relatively high speed.

The previously mentioned studies have different approach methods before impact with the recovery net: In the experiment described in Ref. 1, the UAV flies directly into the net at a constant path angle, Ref. 2 presents a UAV that flies in spirals before initiating the final phase and hitting the net, the UAV in Ref. 3 keeps a constant path angle before hitting the net as well, and the study presented in Ref. 4 also focuses on the actual hit, not the trajectory leading to the net. To reduce the risk of damage on both the UAV and the recovery net, it is preferable to reduce the speed of the UAV before the impact. This article suggests the use of a deep stall to reduce the speed of the UAV. Deep stall is a flight condition where the angle of attack is very high, beyond the stall angle, and usually causes the aircraft to rapidly loose speed and altitude.

Deep stall landing of small vehicles has not been as thoroughly studied, and even less so for UAVs, but the phenomenon still appears in various publications: In Ref. 5, a study of the flight characteristics of a manned deep stall is presented. Using the Schweizer SGS 1-36 Sprite sailplane, the pilot smoothly entered a deep stall and let the angle of attack stabilize between 30° and 70°. After falling down rapidly, at a speed of around 20 m/s, the pilot lowered the angle of attack to around $25^\circ - 30^\circ$, resulting in an unstable flight, before increasing the speed and returning to a safe flight. It was registered that while in the deep stall phase, though flying stable, a lateral-directional oscillation could be noticed. This was explained as the pilot's lack of lateral-directional control when the angles of attack went beyond 60°. A thorough flight characteristic with both predicted and measured data is appended in the article. Also the article presented in Ref. 6 describes experiences achieved during piloted flight in deep stall in a simulator. Their conclusions focus on how a deep stall can be avoided, as it normally causes the airplane to loose altitude and it may take long time to regain control over the plane.

Experimental results for a deep stall landing in a UAV can be found in Ref. 7. The article presents an analysis with longitudinal dynamics of a deep stall landing, including remarks on how the lift and drag coefficients appear in a deep stall. It describes how the speed is decreased in the deep stall as a consequence of the drag and lift coefficients for high angles of attack, and that a high touch down speed might limit this landing method to small UAVs. In both the simulations and the experiments, an unspecified plane with a horizontal tail plane is used. The article explains that "the only mechanism to get the plane into deep stall is quickly tilt of the horizontal tail plane ". The article concludes that deep stall landing can decrease the speed while maintaining a deep path angle, which was confirmed through trim analysis, flight simulations and flight data from a flight test. In Ref. 8, the post stall landing manoeuvre with a tail plane UAV is modelled and simulated. The longitudinal route of the UAV is divided into an initial relatively steep conventional landing approach where the altitude is reduced, before an extended flare phase causes the UAV to gain altitude, and thus potential energy, reducing the speed, before it stalls to further reduce the speed and then land in post-stall. The article investigates the effect of deflecting the elevator over time periods of various lengths, and graphs are shown with the simulated trajectory and the velocity of the UAV for the different cases.

In Ref. 9, a post-stall landing inspired by bird perching is described. A model of a simple UAV is formulated and linear quadratic regulator (LQR) trees were used to implement the controller. Non-linear model predictive control (NMPC) was also considered as a candidate, but LQR-trees was found to be more efficient, but have worse theoretical performance properties. The theory was supported by experimental flight test results. In Ref. 10, an analysis of a safe deep stall landing is performed. The article only considers longitudinal dynamics, and the model uses a delta wing vehicle by AeroSpy Sence & Avoid Technology. Simulations and experimental deep stall landing manoeuvres are performed and discussed in the article, and a hybrid system that is able to safely abort a deep stall landing under hazardous conditions is presented.

This paper presents an NMPC employed on the 6-DOF dynamics of a fixed-wing UAV, to guide it in a deep stall while at the same time landing in a given location. The model for the fixed-wing UAV is described in Ref. 11 and consists of north-east-down positions relative to the inertial frame, body velocity, quaternions for representation of attitude, and angular velocity in the body frame. The UAV used in this article is controlled by rudder, elevator, aileron, and throttle. The NLP of the NMPC consists of an objective function, which should be minimized over a time horizon of N discrete time intervals, and constraints.

This work is a continuation of Ref. 12, where the NMPC guided deep stall landing of a longitudinal model of a fixed-wing UAV was studied. In this article, the lateral dynamics are also included, considering the side wind influence on the UAV. Through simulations done in Python with the open-source software package CasADi Ref. 13, it is investigated how the landing precision and minimal speed is influenced by wind gusts and crosswinds.

II. Model

The UAV model in this article is the 6-DOF dynamics of a fixed-wing UAV in quaternion representation.[11,14] The specific UAV model used is an Aerosonde UAV.[11]

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \mathbf{R}_b^n(\mathbf{q}) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{M} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}, \tag{2}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} -\varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 \\ \eta & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & \eta & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & \eta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{J_y} m \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix}, \tag{4}$$

$$\mathbf{q} = \begin{bmatrix} \eta & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}^T \tag{5}$$

The rotation matrix $\mathbf{R}_b^n(\mathbf{q})$ and the constants Γ_i in the above equations are defined in Eq. (43) and Eqs. (38)–(42) respectively.

The forces f_x, f_y, f_z and moments l, m, n that act upon the UAV are the aerodynamic, the gravitational, and the propulsion forces and moments. This is seen in Eq. (6) and Eq. (10). The aerodynamic force is highly dependent on the UAVs relative speed, making changes in the wind important to the model.

$$\mathbf{f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \mathbf{f}_g + \mathbf{f}_a + \mathbf{f}_p, \tag{6}$$

$$\mathbf{f_g} = \mathbf{R}_n^b(\mathbf{q}) \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix}, \tag{7}$$

$$\mathbf{f_a} = \frac{1}{2} \rho V_a S \begin{bmatrix} C_X(\alpha) + C_{X_q}(\alpha) \frac{c}{2V_a} q + C_{X_{\delta_e}}(\alpha) \delta_e \\ C_{Y_0} + C_{Y_\beta} \beta + C_{Y_p} \frac{b}{2V_a} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \\ C_Z(\alpha) + C_{Z_q}(\alpha) \frac{c}{2V_a} q + C_{Z_{\delta_e}}(\alpha) \delta_e \end{bmatrix}, \tag{8}$$

$$\mathbf{f_p} = \frac{1}{2}\rho S_{prop} C_{prop} \begin{bmatrix} (k_{motor}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix}, \tag{9}$$

$$\mathbf{m} = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} -k_{T_p}(k_\omega \delta_t)^2 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\rho S \begin{bmatrix} b[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r] \\ c[C_m(\alpha) + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e] \\ b[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r] \end{bmatrix}. \tag{10}$$

$$V_a = \sqrt{u_r^2 + w_r^2 + w_r^2},$$

$$\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix}, \tag{11}$$

$$\mathbf{V}_w^b = \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix} = R_n^b(\mathbf{q}) \begin{bmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{bmatrix} + \begin{bmatrix} u_{w_g} \\ v_{w_g} \\ w_{w_g} \end{bmatrix} \tag{12}$$

$$\alpha = \tan^{-1}\frac{w_r}{u_r}, \;\; \beta = \sin^{-1}\frac{v_r}{V_a} \tag{13}$$

$$\begin{bmatrix} C_{X_*} \\ C_{Z_*} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -C_{D_*} \\ -C_{L_*} \end{bmatrix} \tag{14}$$

All constants for the UAV model and the specific values on the ones used in this article can be found in Ref. 11. The relationship between body-axis coefficients, C_{X_*} and C_{Z_*}, and stability-axis coefficients C_{D_*} and C_{L_*} is given by Eq. (14).

In this article, wind is modeled as a sum of a steady component in the inertial frame and a gust component in the body frame. While the steady wind components w_{n_s}, w_{e_s} and w_{d_s} are modeled as constant, the gust component is modeled as white noise $w_i(s)$ that is shaped according to the Dryden wind model:[15]

$$u_{w_g}(s) = \quad H_u(s)\, w_1(s) = \frac{\sigma_u \sqrt{\frac{2L_u}{\pi V_a}}}{\frac{L_u}{V_a}s + 1} w_1(s) \tag{15}$$

$$v_{w_g}(s) = \quad H_v(s)\, w_2(s) = \sigma_v \sqrt{\frac{2L_v}{\pi V_a}} \frac{\frac{\sqrt{3}L_v}{V_a}s + 1}{\left(\frac{L_v}{V_a}s + 1\right)^2} w_2(s) \tag{16}$$

$$w_{w_g}(s) = \quad H_w(s)\, w_3(s) = \sigma_w \sqrt{\frac{2L_w}{\pi V_a}} \frac{\frac{\sqrt{3}L_w}{V_a}s + 1}{\left(\frac{L_w}{V_a}s + 1\right)^2} w_3(s) \tag{17}$$

The turbulence intensities and turbulence scale lengths are given by Eqs. (18)–(23).

$$\sigma_w = 0.328 W_{20} \tag{18}$$

$$\sigma_v = \frac{\sigma_w}{(0.177 + 0.00270 \cdot h)^{0.4}} \tag{19}$$

$$\sigma_u = \frac{\sigma_w}{(0.177 + 0.00270 \cdot h)^{0.4}} \tag{20}$$

$$L_u = \frac{3.28h}{(0.177 + 0.00270 \cdot h)^{1.2}} \tag{21}$$

$$L_v = \frac{3.28h}{(0.177 + 0.00270 \cdot h)^{1.2}} \tag{22}$$

$$L_w = 3.28h \tag{23}$$

where W_{20} is the steady wind speed at 20 feet, and where metric units have been used. All this is in agreement with Ref. 15, however to avoid singularities in the simulation as $h \to 0$ the height parameter h in the Dryden model is set equal to 50 meters.

The non-linear lift and drag coefficients $C_D(\alpha)$ and $C_L(\alpha)$, and the pitch moment $C_m(\alpha)$, are given by one 6^{th} and two 12^{th} degree polynomial equations. The UAV model in use in this article is the one of an Aerosonde, but a NACA 4415 wing profile with an aspect ratio of 15.24 has been chosen to represent its lift, drag and pitch moment coefficients. Although the wing profile of an Aerosonde differs from a NACA 4415 profile, it is still used as it is more realistic than the aerodynamic models given in Ref. 11, since the data is given for angles of attack above the stall angle. The angle of attack dependent lift, drag and pitch moment coefficients are from figures A.5 (a), (b) and (c) in Ref. 16. They are only valid when $-10^\circ \leq \alpha \leq 110^\circ$, and are results from wind tunnel tests on airfoils. The relationships between the angle of attack, α, and the polynomial functions for the drag, lift and pitch moment coefficients are shown in figure 1. In addition to these angle of attack dependent coefficients, there is an effect on the drag force, lift force, and pitching moment caused by the pitch rate q and the elevator deflection δ_e. As can be seen in figure 1, the UAV has a stall angle at around 18° ($C_L = 1.42$) and a secondary lift peak around 50°. The drag coefficient is at its maximum around 90°. The pitch moment coefficient will be zero depending on the elevator deflection angle, but is shown in the figure with zero elevator deflection angle.

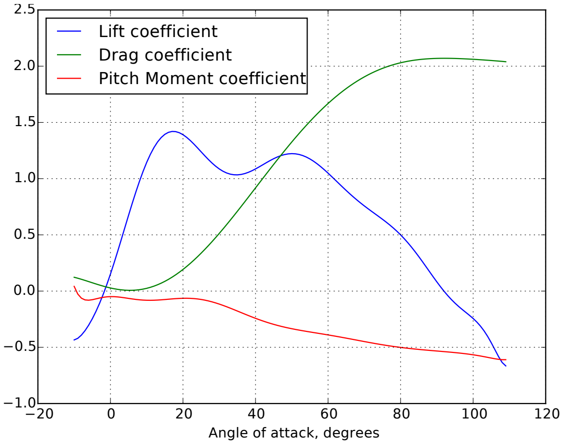


Figure 1: $C_L(\alpha)$, $C_D(\alpha)$, and $C_m(\alpha)$

III. Methods

III.A. Deep Stall Landing Concepts

When the UAV's angle of attack increases, the boundary layer of the air flow eventually separates from the airfoil of the wing and creates a turbulent wake behind the wing. This causes a massive reduction in the lift on the wing and an increase of drag. The angle of attack when this occurs, is called the *stall angle*. It can be recognized in a lift coefficient characteristic as the angle of attack that gives the maximum lift.[11,17–21] The term *deep stall* refers to angles of attack higher than the stall angle, from which it might be difficult to recover.[6] Ref. 18 defines the deep stall as the condition where the angle of attack gives a stable trim point beyond the stall angle. There are many slightly different definitions of the trim point, as the condition where the sum of moments is zero,[17,22] or where there is zero moment *and* zero resultant forces around the center of gravity.[18] In this article, the definition by Ref. 22 is used. It argues that as an aerial vehicle normally is naturally lateral-directionally stable because of symmetry, longitudinal trim is usually enough to find. The conditions for stable longitudinal trim are thus:

$$C_m = 0, \quad \frac{dC_m}{d\alpha} \leq 0, \tag{24}$$

The stability condition ensures that the aircraft will return to the trim position after a minor disturbance.[18] The pitch moment, given by Eq. (10), contains one component given by the elevator deflection, one by the angle of attack, and one by the pitch rate. In trim, the pitch rate will be zero as a consequence of no moment around the body y-axis of the UAV. With zero elevator deflection, the graph for the pitch moment is as shown in figure 1. When the elevator deflection is non-zero, the graph will be vertically shifted, which means that with a positive elevator deflection component of the moment, the trim condition where the pitch moment is zero will be moved to a higher angle of attack. With a non-linear pitch moment coefficient, there can be multiple trim points. When the angle of attack is higher than the stall angle, and in addition the gradient of the moment coefficient is negative, a stable trim point for a deep stall is given.

III.B. Model Predictive Control

Instead of using linear controllers like a PID controller or an LQR, a non-linear model predictive controller (NMPC) is used to control the UAV in these simulations. This is chosen both because it gives an optimal solution, and because the non-linearities and constraints in the model demands a more flexible controller than linear controllers. NMPC is a control method that tries to optimize an objective function constrained by the systems dynamics, as well as physical and operational constraints. This is done by optimizing a sequence of control actions over a time horizon based on the predicted state of the system, and then applying the first control action to the system. Then the optimization is repeated, using the measurement of the response of the system as an initial state for the optimization. The prediction horizon is shifted one time step further, and the process is warm started, using the time shifted optimal values from the previous optimization as an initial guess for the next optimization. The 6 degrees of freedom (6-DOF) dynamics of a UAV operating at high angle of attack is highly non-linear, which benefits from a non-linear MPC to control it. While a linear MPC defines an optimization problem that can be solved with convex quadratic programming, the NMPC is usually solved by transforming the control problem into a non-linear program (NLP) and then optimize it. Although an NMPC is a time and resource consuming control method, it gives advantages as it handles non-linearities and constraints well.[23]

The NMPC problem contains an objective function, which should be minimized, and constraints, which define the area wherein the minimum should be found. Eq. (25) show the objective function which is a function of the state and control trajectories **x** and **u**. Eqs. (26)–(27) shows the constraint for the initial state of the system, where the prediction horizon starts, and the differential equations representing the model. Eqs. (28)–(29) show the constraints on the control variables and on the states.

$$\min_{\mathbf{x}(\cdot),\mathbf{u}(\cdot)} \int_{t_0}^{t_0+T} F(\mathbf{x}(t),\mathbf{u}(t))\mathrm{d}t + E(\mathbf{x}(T)) \tag{25}$$

$$\text{subject to}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{26}$$

$$\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) = \mathbf{0}, \tag{27}$$

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}, \tag{28}$$

$$\mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max}, \forall t \in [t_0, t_0+T]. \tag{29}$$

In the implementation, NMPC uses direct multiple shooting to discretize the model.[24] This way of creating a non-linear program first divides the control horizon into N discrete control intervals, and then solves one optimization problem for each control interval simultaneously.

IV. Problem Definition

The aim of the research described in this article is to develop an algorithm to deep stall land a small, fixed wing UAV in a specific location. To do this, a double objective is formulated: The UAV should track the pre-decided landing target precisely along a given flight path angle and course angle, and land there in a deep stall. This is done in two steps, as described in Algorithm 1 and shown in figures 2a and 2b: When the landing is initiated, the UAV follows the landing target by tracking its lateral line of sight (LOS) angle, but keeping the same altitude. When the UAV's longitudinal LOS angle to the landing target is smaller (or greater in magnitude, since the landing path angle is negative) than a given limit, γ_0, it starts descending with a path angle that follows the longitudinal LOS angle. While descending, the UAV also tracks the desired deep stall angle of attack.

The LOS angles from Algorithm 1 are given by:

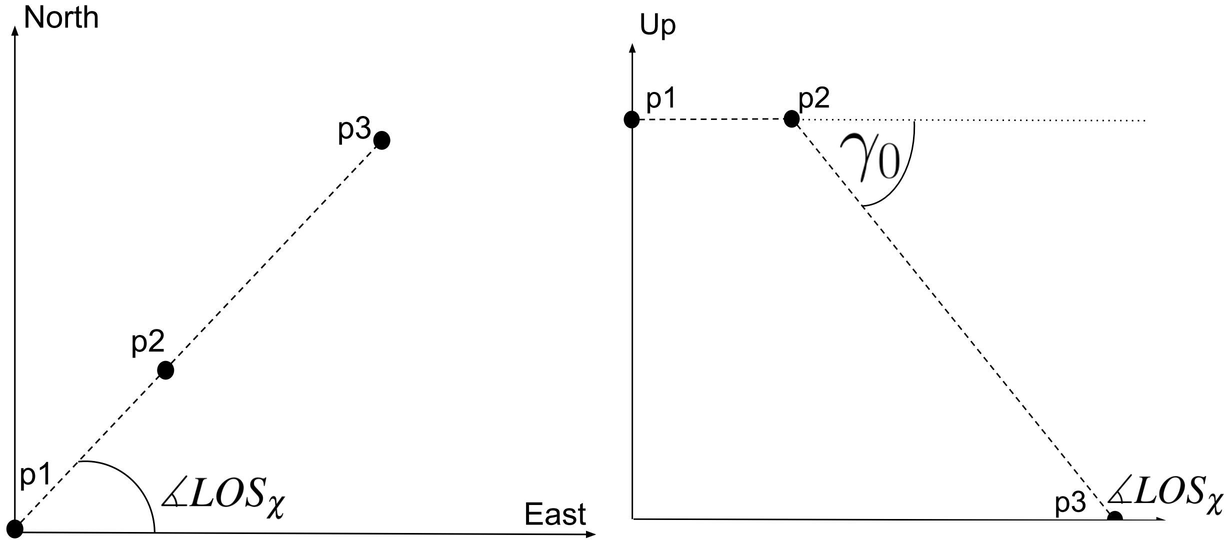
$$\measuredangle LOS_{\gamma} = -\arctan\frac{-p_d - \text{target}_d}{\sqrt{(p_n - \text{target}_n)^2 + (p_e - \text{target}_e)^2}} \tag{30}$$

$$\measuredangle LOS_{\chi} = atan2(\text{target}_e - p_e, \text{target}_n - p_n) \tag{31}$$

Algorithm 1 Landing procedure

```
Require:
  while $\measuredangle LOS_\gamma > \gamma_0$ do
      $\gamma^* = 0$
      $\chi^* = \measuredangle LOS_\chi$
  end while
  while $\measuredangle LOS_\gamma \leq \gamma_0$ do
      $\gamma^* = \measuredangle LOS_\gamma$
      $\chi^* = \measuredangle LOS_\chi$
      $\alpha^* = \measuredangle DeepStall$
  end while
```

where $atan2(y,x)$ is the angle in radians between the positive x-axis of a plane and the point given by the coordinates (x, y) on it.



(a) Horizontal course for the UAV

(b) Flight path for the UAV

Figure 2: Proposed trajectories for the UAV

To descend in deep stall to the landing target, the difference between the path angle and the longitudinal LOS angle is minimized while the course angle is still kept at the lateral LOS angle to the landing target. To keep a deep stall, the pitch moment is set to track a zero reference at the same time as the speed is minimized. This stabilizes the UAV at the pitch moment equilibrium point where the drag is the largest. In addition to achieving zero pitch moment, the roll and yaw moment are minimized to prevent oscillations, and the change in control actions is minimized. The optimization problem can be defined as (32)-(34), with penalty parameters from (35).

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \sum_{k=0}^{N-1} (\chi_k - \chi^*)^2 Q_1^2 + (\gamma_k - \gamma^*)^2 Q_2^2 \\
+Q_3^2(u_k^2 + v_k^2 + w_k^2) + Q_4^2 m_k^2 + Q_5^2 n_k^2 + Q_6^2 l_k^2 \\
+(\mathbf{u}_k - \mathbf{u}_{k-1})^T R^T R(\mathbf{u}_k - \mathbf{u}_{k-1}) \\
+(\chi_N - \chi^*)^2 Q_7^2 + (\gamma_N - \gamma^*)^2 Q_8^2 \\
+Q_9^2(u_N^2 + v_N^2 + w_N^2) + Q_{10}^2 m_N^2 + Q_{11}^2 n_N^2 + Q_{12}^2 l_N^2
\end{aligned}
\tag{32}
$$

$$
\text{subject to} \quad \mathbf{x}_{k+1} = \mathbf{f_d}(\mathbf{x}_k, \mathbf{u}_k) \tag{33}
$$

$$
\text{given} \quad \mathbf{x}_0 \tag{34}
$$

$$
\begin{aligned}
Q_1 = Q_7 = 2000, Q_2 = Q_8 = 2300, Q_3 = Q_9 = 2 \\
Q_4 = Q_{10} = 1800, Q_5 = Q_6 = Q_{11} = Q_{12} = 300 \\
R = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 35 & 0 \\ 0 & 0 & 0 & 22 \end{bmatrix}
\end{aligned}
\tag{35}
$$

Both the states of the UAV model and the control variables are included as optimization variables in the NLP. Constraints are placed on the state variables u, v, w to stay within a maximum speed, on α since the lift, drag, and pitch moment curves are only valid in this area, and on the control inputs, since they have physical limitations.

$$
\begin{aligned}
-25m/s \leq u, v, w \leq 25m/s \\
-10^\circ \leq \alpha \leq 110^\circ \\
-30^\circ \leq \delta_a, \delta_e, \delta_r \leq 30^\circ \\
0 \leq \delta_t \leq 1
\end{aligned}
\tag{36}
$$

The transition between the path angles in these two path legs is smoothed by a cubic spline. To make the UAV keep a path angle approximately equal to γ_0, the spline has to be initiated before the UAV reaches that LOS angle.

V. Simulation

The simulations are performed in Python, using the open-source software package Casadi[13] for numerical optimization. Each optimization uses a prediction horizon of $T = 3.0$ seconds, divided into $N = 30$ equidistant control intervals. The optimization variables are given by $\mathbf{X} = [\mathbf{x_0}\ \alpha_0\ \beta_0\ \gamma_0\ \chi_0\ \mathbf{u_0}\ \mathbf{x_1}\ \alpha_1\ \beta_1\ \gamma_1\ \chi_1\ \mathbf{u_1} \ldots \mathbf{x_N}\ \alpha_N\ \beta_N\ \gamma_N\ \chi_N]$. The initial state, is given by $\mathbf{x}_0 = [0,\ 0,\ -120,\ 25,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T$ for all of the following simulations. To confine the time it takes to solve each optimization problem, the maximum number of iterations for each optimization is set to 600. When the optimization is completed, $\mathbf{u_0}$ is the first occurrence of the control actions that are needed to produce an optimal series of states. This optimal control action is used to control the UAV, and the output state will be the start state of the next optimization.

As seen by Eq. (12), the wind is composed by a steady component and a gust component, which follows the Dryden turbulence model.[15] The steady component is constant, and is assumed known to the model predictive controller since it can be estimated from GNSS and airspeed measurements.[25] However, the gust component is changed in each time step. To simulate time-delayed estimates of the wind gust, the last time steps gust measurement is used in the controller. The gust therefore introduces a partly unknown time varying disturbance, testing the robustness of the controller. This means that the simulations will never be perfect with respect to the wind, but lag one time step (100 ms).

What we want to investigate is how well the performance of the constrained NMPC for automatic deep stall landing is. The performance can be divided into what the velocity is at the time of landing, what the deviation is from the desired landing target and what the attitude is at the time of landing.

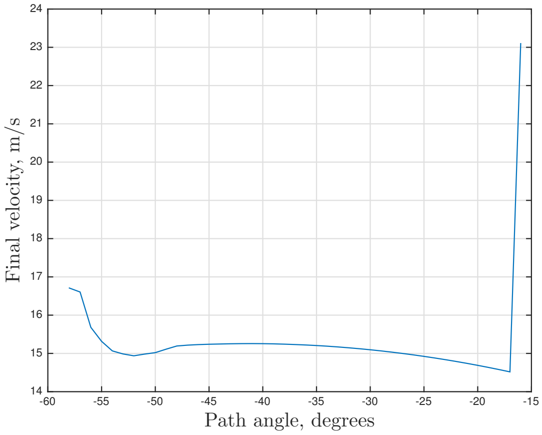


Figure 3: Speed and path angle dependency

V.A. Effect of varying path angle

First, it is investigated if some path angles in the descending phase gives a lower end velocity than others. To focus the analysis on the effect of different path angles, statistical deviations are avoided by assuming zero wind. The outcome is shown in figure 3. When the path angle is greater than $-17°$, the angle of attack is below the stall angle and the UAV is not in deep stall. The speed is therefore higher than when the path angle is $-17°$ or below. When the path angle is steeper than $-45°$, large oscillations arise in the angle of attack trajectory, which is undesirable. In the range $-45° \leq \gamma \leq -17°$, the landing speed is barely changing.

V.B. Simulations with varying gust strength

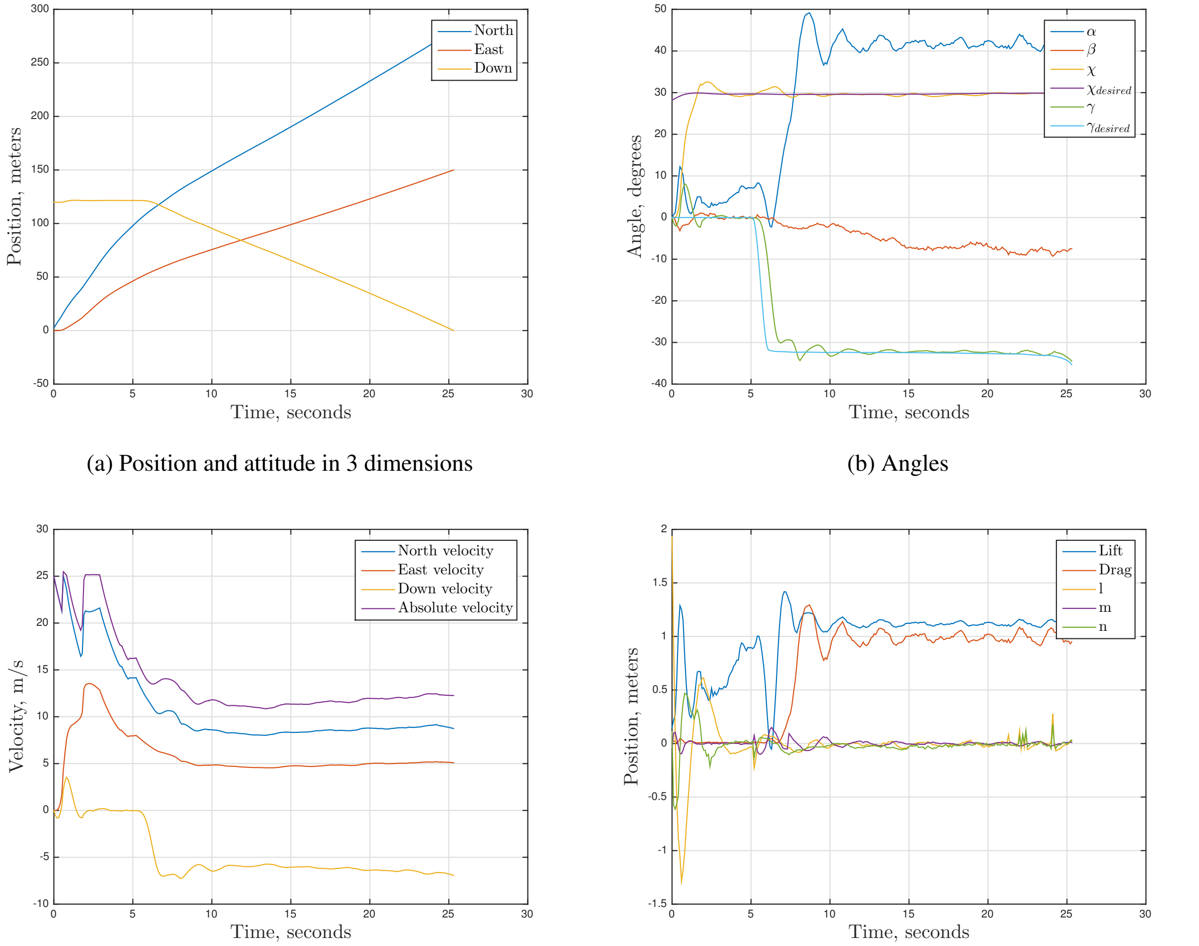
The gust is difficult to predict, as it does not follow any deterministic model. Because of this, its effect on the control can be more unpredictable than the effect of the steady wind, which can be taken into account in the optimization model. Therefore, this simulation setup studies the effect of varying wind conditions. Due to the stochastic property of the wind, this is done through Monte Carlo simulations with 101 trials, and considering the statistical mean and standard deviation. The target position is set to

$$\text{target} = \begin{bmatrix} \text{target}_n \\ \text{target}_e \\ \text{target}_d \end{bmatrix} = \begin{bmatrix} 280 \\ 150 \\ 0 \end{bmatrix}, \tag{37}$$

which gives a desired course angle χ^* that is deliberately set different from zero. To compare the influence from the wind on varying path angles, the simulation considers $\gamma_0 \in \{-20°, -30°, -40°\}$. For simplicity, the steady wind component in the east and down directions are set to zero, while w_{n_s} is simulated with the different values $\{0 \text{ m/s}, -2 \text{ m/s}, -4 \text{ m/s}, -7 \text{ m/s}, -14 \text{ m/s}\}$. This yields a steady wind from the north direction with a magnitude from calm to a moderate gale wind. The gust components are all initialized to zero. As is seen from Eqs. (18)–(20), a large steady wind will also give a large variance in the gust component. Both these factors contribute to making the landing increasingly more difficult with increasing wind conditions.

The end velocity and the final attitude of the UAV, as well as geometric deviation from the target location have been recorded from the simulations. The results are shown in table 1. The results are given as the mean ± the standard deviation, with the maximum absolute value included in parenthesis to better illustrate the outliers. This is based on 101 simulations, where the number of successful trials are indicated in the last column. Infeasible results from the optimization have been removed to better show the performance in the cases where it works. The results show that the success rate is highly depending on the wind speed; the extreme case with $w_{n_s} = -14$ m/s only succeeds in 12% of the trials, with unacceptably large deviations in all trials.

It is also seen that the steepest approach with $\gamma_0 = -40°$ is more susceptible to wind disturbances, as it has a lower success rate. An explanation to this is that as the path angles approach $-40°$, the vertical component of the velocity increase while the horizontal component decrease. At the same time $\theta \approx 0$, so the body z-axis is parallel to



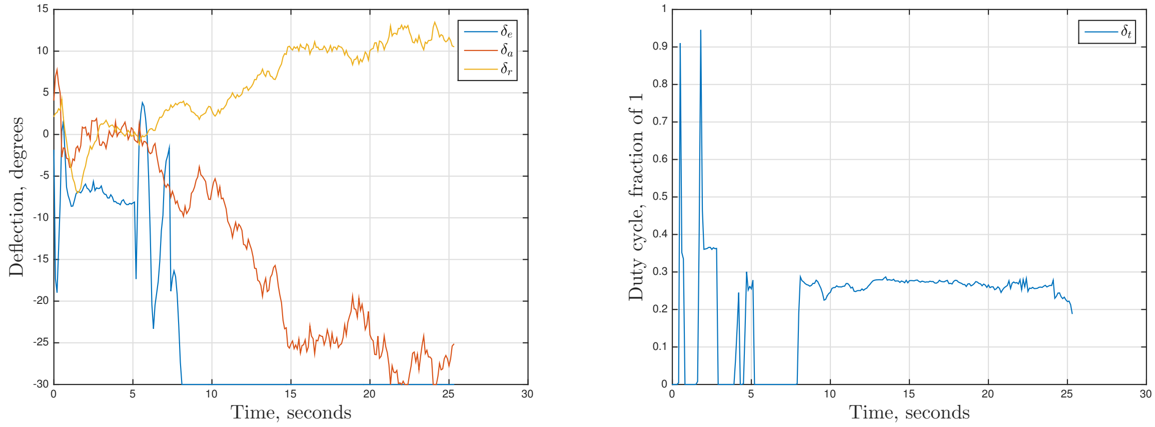
(a) Position and attitude in 3 dimensions

(b) Angles

(c) Inertial velocity

(d) Drag and lift coefficients and roll, pitch and yaw moments

Figure 4: State trajectory from simulation with $\gamma_0 = -30°$ and $w_{n_s} = -4$ m/s



(a) Control surface deflection in degrees

(b) Throttle control in used share of duty cycle

Figure 5: Control trajectory from simulation with $\gamma_0 = -30°$ and $w_{n_s} = -4$ m/s

the NED down-axis. Seeing this in connection with the definition of the body axis gust, Eqs. (15)–(23), this indicates that the gust in the horizontal plane has a larger variance than the vertical component. The larger horizontal gust

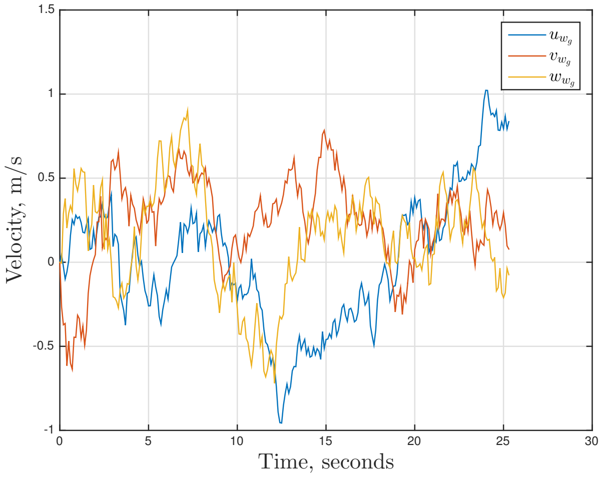


Figure 6: Dryden turbulence from simulation with $\gamma_0 = -30°$ and $w_{n_s} = -4$ m/s

variance, combined with the decreased horizontal velocity component, makes the path of the UAV less predictable and more difficult to control.

With the above exceptions, it can be generally stated that the final landing speed is more dependent on the wind speed than the flight path angle; the landing speeds for the same wind speed has a maximal standard deviation of 1.5 m/s, whereas keeping the path angles constant and looking at how the different wind speeds affect the landing speed result in a standard deviation of 5.1 m/s. From table 1 it is also clear that the standard deviation in the final speeds increase with the magnitude of the wind speed. This is natural as more steady wind also means a larger variance in the gust wind, which directly affects the final speed.

Finally, the results also show that the final attitude is within reasonable values. The roll angle φ is close to zero, with the exception of a few outliers. Naturally the pitch angle θ is larger when the path angle is a smaller negative number, to keep the same α. ψ decreases with the wind speed, since the wind is coming from a different angle than the course angle. This forces the UAV to increase the sideslip to achieve the same course angle, thus the heading has to decrease.

The plots of the state- and control trajectories for one of the simulations with $\gamma_0 = -30°$ and $w_{n_s} = -4$ m/s can be seen in figures 4 and 5. Similarly the gust component of the wind is plotted in figure 6. The angle of attack holds a value of about 42°, which is close to the second peak of the lift curve in figure 1, which gives high values for both the lift and drag foces. This can also be seen from figure 4d. From the plots of the desired path angle in figure 4b it is seen that the controller switches from flying straight to descending almost instantaneously, since the angle of the LOS vector is close to $-30°$ at the start of the simulation. Once the limit is crossed, the desired path angle is adjusted so that the target will be reached. However, due to the delay in the dynamics of the UAV, this has to enforce a slightly larger path angle. Figures 5a and 5b show that the control inputs to the UAV are within the physical limitations of the actuators.

V.C. Simulations with varying side wind strength

Although the landing should ideally be against the wind, it is unrealistic to hope to achieve no disturbing side wind at all. Therefore, the effect side wind has on the performance of the algorithm is investigated. For simplicity, the target location is set to $[200, 0, 0]^T$, enabling the adjustment of side wind to be an adjustment of w_{e_s}. Here, γ_0 was set to $-40°$, as this was showed to be more sensitive to wind in section V.B. Again a Monte Carlo simulation with 100 trials is performed so that the effects of the varying gust component are averaged out in the results, as showed in table 2 using the same notation as in table 1. The simulations are only performed for $w_{e_s} \leq 0$, since the deterministic response of the model is symmetric with respect to the steady east wind component. As is expected the final landing speed decreases, while the heading angle increases for higher magnitude winds. This indicates that the UAV will not hit nose-first, forcing one of the wings to take some of the impact from the net.

Table 1: Simulations of different wind conditions and path angles

w_{n_s}	γ_c	Final speed [m/s]	Target error [m]	φ_f [°]	θ_f [°]	ψ_f [°]	Successful trials
0	-20	14.69±0.00 (14.69)	0.68±0.00 (0.68)	-2.72±0.00 (2.72)	18.21±0.00 (18.21)	35.21±0.00 (35.21)	101
	-30	15.23±0.00 (15.23)	0.91±0.00 (0.91)	1.91±0.00 (1.91)	8.73±0.00 (8.73)	30.23±0.00 (30.23)	101
	-40	15.45±0.00 (15.45)	0.48±0.00 (0.48)	2.02±0.00 (2.02)	-0.71±0.00 (0.71)	27.79±0.00 (27.79)	101
-2	-20	12.99±0.34 (14.08)	0.34±0.20 (0.98)	-1.35±2.78 (5.54)	20.49±3.79 (31.26)	29.87±3.60 (35.25)	101
	-30	13.46±0.27 (14.16)	0.37±0.21 (0.82)	0.59±3.14 (4.78)	11.99±1.53 (15.89)	26.46±4.30 (32.72)	101
	-40	13.85±0.30 (14.60)	0.46±0.27 (0.95)	-0.74±2.71 (4.70)	2.05±2.14 (5.99)	28.05±3.95 (32.17)	66
-4	-20	11.05±0.58 (12.29)	0.35±0.22 (0.81)	0.54±3.70 (8.76)	24.03±2.45 (29.56)	22.63±3.88 (28.15)	101
	-30	11.76±0.57 (13.08)	0.35±0.19 (0.82)	0.39±3.63 (7.59)	16.03±2.09 (22.62)	22.41±3.66 (29.13)	101
	-40	12.27±0.55 (13.53)	0.38±0.24 (0.82)	-0.30±3.86 (7.65)	6.57±2.26 (13.52)	21.61±3.93 (27.32)	32
-7	-20	8.19±1.17 (11.35)	0.41±0.22 (1.02)	0.08±4.10 (10.95)	27.98±5.40 (43.33)	15.61±3.84 (25.76)	101
	-30	9.56±1.36 (15.80)	0.36±0.36 (2.82)	1.04±5.09 (23.27)	13.65±12.47 (33.85)	16.88±4.83 (30.02)	100
	-40	10.09±2.24 (20.43)	0.32±0.40 (2.72)	-1.69±6.56 (5.49)	10.16±12.89 (23.43)	14.97±5.33 (36.23)	46
-14	-20	1.68±0.61 (2.80)	178.83±33.79 (220.88)	2.02±0.77 (3.07)	31.39±8.53 (41.74)	-1.36±5.10 (6.07)	6
	-30	4.17±4.07 (14.01)	71.33±38.89 (113.10)	-1.56±3.74 (2.62)	22.96±12.21 (36.66)	6.92±12.38 (30.08)	9
	-40	4.43±3.59 (15.17)	10.46±18.45 (64.46)	-0.73±5.39 (10.79)	21.31±16.67 (41.39)	6.89±9.40 (38.12)	25

Table 2: Simulation of different side winds

w_{e_s}	Final speed [m/s]	Target error [m]	φ_f [°]	θ_f [°]	ψ_f [°]	Successful trials
0	15.54±0.00 (15.54)	0.72±0.00 (0.72)	0.00±0.00 (0.00)	-4.63±0.00 (4.63)	0.00±0.00 (0.00)	100
-1	15.53±0.14 (15.93)	0.51±0.26 (1.13)	-1.61±2.23 (2.54)	-5.09±0.51 (3.85)	4.12±2.20 (8.87)	100
-2	15.38±0.30 (16.10)	0.56±0.32 (1.79)	-0.67±2.08 (2.77)	-4.81±1.52 (2.73)	6.63±3.13 (15.01)	100
-3	15.22±0.46 (16.58)	0.58±0.26 (1.28)	-2.07±2.82 (6.32)	-4.37±1.52 (0.97)	11.48±3.59 (19.90)	100
-4	14.83±0.56 (16.59)	0.56±0.22 (1.15)	-2.50±3.00 (7.82)	-3.07±2.09 (4.32)	17.50±3.22 (25.45)	98
-5	14.39±0.72 (16.54)	0.61±0.24 (1.34)	-2.90±3.40 (5.75)	2.10±3.28 (9.15)	21.58±3.69 (28.52)	100
-6	13.70±0.99 (16.18)	0.56±0.26 (1.54)	-2.45±4.06 (10.12)	4.02±3.96 (16.29)	28.49±4.59 (39.09)	100
-7	13.45±1.10 (16.43)	0.62±0.35 (1.94)	-3.23±5.12 (14.07)	4.69±4.52 (19.09)	33.97±6.24 (55.18)	100
-8	12.60±1.23 (15.73)	0.54±0.29 (1.76)	-3.02±4.73 (8.46)	6.37±5.19 (19.14)	40.36±6.07 (57.11)	99
-9	11.60±1.63 (14.73)	0.57±0.33 (1.62)	-2.68±5.58 (9.76)	9.67±5.87 (26.64)	45.34±7.85 (62.78)	100
-10	10.56±2.35 (16.10)	0.56±0.36 (1.79)	-2.84±6.46 (16.97)	12.84±8.89 (46.85)	51.53±10.44 (84.58)	99
-11	10.03±2.50 (16.82)	0.51±0.39 (2.13)	-2.42±6.19 (16.59)	14.58±8.69 (36.36)	55.12±11.38 (84.53)	99

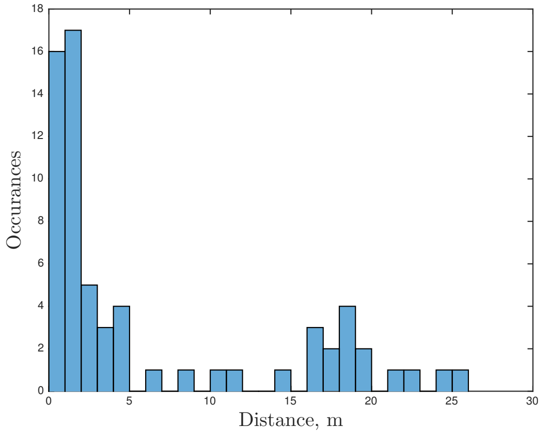


Figure 7: Histogram showing deviation from landing target when no knowledge of the gust wind is assumed

V.D. Simulations with limited wind knowledge

One of the limitations with the previously presented landing scheme is that it assumes perfect knowledge of the wind, up to the previous time step. Despite recent advances within wind estimation,[25] this might be a too optimistic assumption. Therefore a simulation study with a more limited knowledge of the wind is performed. In this case the NMPC only knows the steady wind component, while the gust is assumed unknown. To limit the analysis, this study only looks at the case when $\gamma_0 = -30^\circ$ and $w_{n_s} = -4m/s$. The limited knowledge of the wind makes it even more important for the UAV to not deviate from the path, as a sudden change in wind might make the return to the path infeasible. Since the variance of the Dryden gust wind is larger in x- and y-direction than in the z-direction, Q_1 was increased to 3500 in this simulation. The rest of the setup is the same as in section V.B, to allow for easy comparison of the results. Figure 7 shows a histogram with unit bin widths of the deviations to the landing target. Out of a total 100 landing attempts, 33 of the landings are within 2 meters from the target, while 15 are 17 meters or more from the target. In addition to the 65 successful landings showed in the plot, 35 landing attempts were found to be infeasible. Compared to the results in table 1, the average landing speed is slightly higher in this case, with a much larger standard deviation; 14.79 ± 3.59 m/s. However if we look at the landing speed for the 33 attempts that land within 2 meters from the target, the average is 12.06 m/s with a standard deviation of 1.00 m/s.

VI. Conclusion

The algorithm presented in this article is able to guide a fixed-wing UAV in a precise deep stall landing. The UAV is controlled with an NMPC, which first guides the UAV to a waypoint whose line-of-sight angle to the landing target equals a given angle, with a course angle directed towards the landing target. From that point, the NMPC combines tracking of the necessary path and course angles to the landing target, while keeping the UAV in a deep stall. The deep stall is maintained by keeping the pitch moment at a trim point where the pitch moment is zero, and the pitch moment gradient with respect to the angle of attack is negative. When the UAV is deep stall landing, the drag is increased and the speed is reduced. A small simulation study was performed to analyze which path angles gave the lowest landing speeds, concluding that $-45 \leq \gamma_0 \leq -17$ will land the plane in a deep stall. A larger Monte Carlo simulation with 101 trials was performed to consider the effects of the stochastic wind gust. This showed that the controller was able to land the UAV in steady winds of up to 7 m/s, with an acceptable success rate. However the number of successful trials depend largely on the wind magnitude. Also the results indicate that a steeper flight path is more susceptible to wind disturbances. Finally, the controllers dependence on knowing recent values of the gust wind component was challenged. This showed that the majority of the feasible paths land within 2 meters from the target. However it also showed that more research has to be performed in order to recover the UAV from paths that are infeasible or destined to miss the target. Future work on the field will also include a real-time implementation and experimental testing.

Appendix

The constants Γ_i are functions of the UAVs products and moments of inertia, J_{xz}, J_x, J_y, J_z, and are stated in Eqs. (38)–(42) for completeness.[11]

$$\Gamma_1 = \frac{J_{xz}\left(J_x - J_y + J_z\right)}{J_x J_z - J_{xz}^2} \qquad \Gamma_2 = \frac{J_z\left(J_z - J_y\right) + J_{xz}^2}{J_x J_z - J_{xz}^2} \tag{38}$$

$$\Gamma_3 = \frac{J_z}{J_x J_z - J_{xz}^2} \qquad \Gamma_4 = \frac{J_{xz}}{J_x J_z - J_{xz}^2} \tag{39}$$

$$\Gamma_5 = \frac{\left(J_z - J_x\right)}{J_y} \qquad \Gamma_6 = \frac{J_{xz}}{J_y} \tag{40}$$

$$\Gamma_7 = \frac{J_{xz}^2 + J_x\left(J_x - J_y\right)}{J_x J_z - J_{xz}^2} \qquad \Gamma_8 = \frac{J_x}{J_x J_z - J_{xz}^2} \tag{41}$$

$$\tag{42}$$

The rotation matrix $\mathbf{R}_b^n(\mathbf{q})$ transforms a vector from the body frame into the NED frame. This can be parameterized as Eq. (43), using unit quaternions.[14]

$$\mathbf{R}_b^n(\mathbf{q}) = \begin{bmatrix} 1-2(\varepsilon_2^2+\varepsilon_3^2) & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\eta) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\eta) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\eta) & 1-2(\varepsilon_1^2+\varepsilon_3^2) & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\eta) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\eta) & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\eta) & 1-2(\varepsilon_1^2+\varepsilon_2^2) \end{bmatrix} \tag{43}$$

$$\mathbf{R}_n^b(\mathbf{q}) = \mathbf{R}_b^n(\mathbf{q})^T \tag{44}$$

Acknowledgments

This work has been carried out at the Centre for Autonomous Marine Operations and Systems (AMOS), supported by the Research Council of Norway through the Centres of Excellence funding scheme, grant number 223254. The Norwegian Research Council is acknowledged as the main sponsor of AMOS.

References

[1] Skulstad, R., Syversen, C., Merz, M., Sokolova, N., Fossen, T., and Johansen, T., "Autonomous net recovery of fixed-wing UAV with single-frequency carrier-phase differential GNSS," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 30, No. 5, 2015, pp. 18–27, cited By 0.

[2] Yoon, S., Kim, H., and Kim, Y., "Spiral landing guidance law design for unmanned aerial vehicle net-recovery," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 224, No. 10, 2010, pp. 1081–1096.

[3] Kim, H., Kim, M., Lim, H., Park, C., Yoon, S., Lee, D., Choi, H., Oh, G., Park, J., and Kim, Y., "Fully autonomous vision-based net-recovery landing system for a fixed-wing UAV," *IEEE/ASME Transactions on Mechatronics*, Vol. 18, No. 4, 2013, pp. 1320–1333.

[4] Mulligan, A., Osbrink, A. M., and Patterson, M. C. L., "Precision Recovery Capability for Small UAS," 23rd *Bristol UAV Systems Conference*, 2008.

[5] Sim, A. G., "Flight Characteristics of a Manned, Low-Speed, Controlled Deep Stall Vehicle," *NASA Technical Memorandum*, 1984.

[6] White, M. D. and Cooper, G. E., "Simulator studies of the deep stall," *NASA Conference on Aircraft Operating Problems*, 1965.

[7] Taniguchi, H., "Analysis of deepstall landing for UAV," 26th *International Congress of the Aeronautical Sciences*, Vol. 3, 2008, pp. 2498–2503.

[8] Crowther, W. and Prassas, K., "Post-stall landing for field retrieval of unmanned air vehicles," 14th *Bristol Interanational Unmanned air Vehicle Systems Conference*, 1999.

[9] Moore, J., Cory, R., and Tedrake, R., "Robust post-stall perching with a simple fixed-wing glider using LQR-Trees," *Bioinspiration and Biomimetics*, Vol. 9, No. 2, 2014.

[10] Pointner, W., Kotsis, G., Langthaler, P., and Naderhirn, M., "Using formal methods to verify safe deep stall landing of a MAV," *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, 2011, pp. 5D11–5D110.

[11] Beard, R. and McLain, T., *Small unmanned aircraft: Theory and practice*, Princeton University Press, 2012.

[12] Mathisen, S. H., Fossen, T. I., and Johansen, T. A., "Non-linear Model Predictive Control for Guidance of a Fixed-Wing UAV in Precision Deep Stall Landing," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.

[13] Andersson, J. and Gillis, J., "CasADi," .

[14] Fossen, T. I., *Handbook of Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons, 2011.

[15] MathWorks, "Dryden Wind Turbulence Model (Continuous)," Following Military Specification MIL-F-8785C.

[16] Ostowari, C. and Naik, D., *Post-Stall Wind Tunnel Data for NACA 44XX Series Airfoil Sections*, Solar Energy Research Institute, January 1985.

[17] Kermode, A. C., *Mechanics of Flight*, Pearson Education Limited, 2006.

[18] Phillips, W. F., *Mechanics of Flight*, John Wiley & Sons, 2004.

[19] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill, Inc., 2nd ed., 1991.

[20] Bertin, J. J. and Cummings, R. M., *Aerodynamics for Engineers*, Pearson Education Limited, sixth ed., 2014.

[21] Abbott, I. H. and von Doenhoff, A. E., *Theory of Wing Sections*, Dover Publications Inc., 1959.

[22] Cook, M. V., *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability*, Butterworth-Heinemann, 1997.

[23] Kufoalor, D. and Johansen, T., "Reconfigurable fault tolerant flight control based on Nonlinear Model Predictive Control," *American Control Conference (ACC), 2013*, June 2013, pp. 5128–5133.

[24] Diehl, M., Ferreau, H. J., and Haverbeke, N., *Nonlinear Model Predictive Control - Towards New Challenging Applications*, chap. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, Springer Berlin Heidelberg, 2009, pp. 391–417.

[25] Johansen, T., Cristofaro, A., Sorensen, K., Hansen, J., and Fossen, T., "On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors," *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 510–519.