Model predictive UAV-tool interaction control enhanced by external forces

Basaran Bahadir Kocer, Tegoeh Tjahjowidodo, Gerald Seet

Institutions: Nanyang Technological University

Published on: 01 Apr 2019 - Mechatronics (Pergamon)

Topics: Model predictive control, Control theory, PID controller and Robotic arm

Related papers:
- Constrained Estimation-based Nonlinear Model Predictive Control for UAV-Elastic Tool Interaction
- Centralized predictive ceiling interaction control of quadroto VTOL UAV
- External Forces Resilient Safe Motion Planning for Quadrotor
- Discrete-Time Formulation for Optimal Impact Control in Interaction Tasks
Model Predictive UAV–Tool Interaction Control Enhanced by External Forces

Basaran Bahadir Kocer, Tegoeh Tjahjowidodo, Gerald Gim Lee Seet

School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore

Abstract

One of the major challenges of automated systems is attributed to the interaction task. This process involves external forces which may be dangerous, for example in an unmanned aerial vehicle (UAV) that interacts with unknown environment. There are numerous potential applications in UAVs that require physical interaction with its environment. However, this scenario brings evident challenges to be addressed, i.e., (i) the dedicated parts for the physical interaction (e.g., robotic arm) might change the moment of inertias and the center of gravity of the UAVs; (ii) contact phase might cause chattering effects; (iii) versatile external forces during interaction can degrade the performance; (iv) the needs of the UAVs to respect the bounds on the controller actions as well as the upper limits of the additional sensors equipped with a tool. In order to handle the aforementioned challenges in a systematic way, an optimization–based approach is proposed for use on the control and the estimation design. The translational states and unmeasured forces are estimated by nonlinear moving horizon estimation (NMHE) after each new measurement becomes available. The estimated external forces are then fed into the nonlinear model predictive control (NMPC) which provides the total force and three angular positions. For the total forces, a novel control allocation is designed to maintain the interaction with the ceiling at the desired level. The angular values, three outputs of NMPC, are given to proportional–derivative–integral (PID) controllers to maintain attitude stability. Using the external force information given by the NMHE, the presented interaction controller is able to interact with the environment experimentally in milliseconds.

Keywords: unmanned aerial vehicles, physical interaction, moving horizon estimation, model predictive control, quadrotor, interaction control

1. Introduction

The flourish in computation techniques [1], microelectromechanical sensors [2], hardware technology [3], and involvement of experts furnish a high potential for automated and smart systems, which manifest themselves in intelligent UAVs for real–world applications. Needless to say, the unique flying capabilities that include vertical takeoff and hovering in the air have made UAVs more prevalent [4, 5]. However, the paradigm of using UAVs has shifted from their free–flight utilization to hovering in the air to an inverted pendulum when the UAVs are equipped with a tool on the top side. However, they might exhibit exact opposite behavior in terms of the stability [6]. Even though most of the studies discuss on UAVs with attached additional parts (e.g., robotic arm) on the underside, there exist some preliminary studies in the literature for UAVs with attached upper parts (e.g., inspection tool). For instance, in [7], a UAV system is designed to hook itself to some anchor points on the ceiling. The presented study is promising but the evaluation is restricted only in a simulation environment. Among those limited applications, a bridge inspection operated by a UA V with a robotic arm is presented in [18]. In this paper, a PID controller is used in real time for a case of a UAV with an attached tool on the upper side. However, the PID controller can handle neither the variations in the model nor the change in the operating conditions. A similar case is presented in [19] for the application of UAVs in bridge inspection. In the paper, a comparative study is carried out for a designed backstepping algorithm with integral action and a PID for a quadrotor equipped with an upper side robotic manipulator. The results show a superiority of the backstepping algorithm based on a decoupled approach. Although the results are obtained experimentally, a constraint satisfaction is not fur-
ther elaborated, despite, in most of the cases, many constraints exist for such operations where the controller needs to respect them explicitly [20,21,22].

In this study, the real-time application of NMPC and NMHE for the interaction control of a UAV–tool operation is presented. The system is designed to closely monitor an unknown ceiling by instrumenting an elastic contact tool from the upper side. In this context, a low–cost system is instrumented consisting of a spring–based tool to attach on the UAV which does not need a force sensor. Due to the high computational demand when the nonlinear nature of the problem is addressed along with the constraints within the optimization settings, a tailored solver needs to be employed [23,24,25]. To the best of the authors’ knowledge, nominal optimization–based approaches are considered within the physical interaction control of UAVs so far [26,27,28,29]. However, in this paper, the three external forces on each axis are updated within the optimization–based controller itself. The task of the designed NMHE is to provide the constrained estimations for the unknown forces to be fed into the NMPC. A novel control allocation is also presented wherein the vertical force effect from the unknown ceiling is added to the total controller effort. Under the external uncertainties and noisy measurements, the proposed approach is able to handle arbitrary task–based trajectories with a reasonable accuracy in a millisecond range. In this context, the computational times and Karush–Kuhn–Tucker (KKT) values in the optimization problem are also investigated. In summary, the contributions of this implementation are as follows:

• An interaction control and estimation leveraging whole nonlinear model of the system for a case–specific elastic tool and the UAV are implemented in real–time.

• For the first time, the NMPC is updated by the NMHE [30] for the horizontal and vertical force effects for an aerial physical interaction task.

• An elastic tool is compressed while the UAV tracks designed circular trajectories by sliding on the surface experimentally in high force values using a novel control allocation by an optimization–based controller.

• A practical approach is proposed to allow practitioners to plan and apply the UAV free–flight/contact trajectories efficiently using a single and open–source software.

The UAV–tool interaction leveraging an optimization–based framework consisting of NMPC and NMHE is one of the pioneering studies by directly estimating the external forces for the physical interaction. This paper is organized as follows: The problem statement for the UAV–tool interaction is presented in Section II. The system and its dynamics are introduced in Section III. The interaction control and estimation framework using NMPC and NMHE are introduced and explained in Section IV. The experiments are demonstrated and discussed in Section V. Finally, conclusions including future works are drawn in Section VI.

2. Problem Statement

Consider a continuous–time, highly nonlinear system:

\[ x(t) = f(x(t), u(t), d(t)), \]  

where \( t \) is time, \( x \) is the system state which is a subset of \( \mathbb{R}^n \), and \( u \in \mathbb{U} \) is the set of admissible control inputs which is a subset of \( \mathbb{R}^m \). The disturbance \( d \in D \) is assumed to be bounded. It is also assumed that \( f : \mathbb{X} \times \mathbb{U} \times D \rightarrow \mathbb{R}^n \) is Lipschitz continuous. Given a reference set \( P \) that projects the real interaction values within \( D \in \mathbb{R}^n \) to the state space,

\[ P = \{ x_r \in \mathbb{R}^n \mid x_r = x'_r, \Delta z \leq 0 \} \]  

\[ x_r = x_r^u, \quad \Delta z > 0 \}, \]  

where \( \Delta z \) represents the displacement of the tool which is attached on the UAV. Therefore, the reference trajectory component \( x'_r \) is given for the free–flight and the \( x_r^u \) part is generated for the contact phase. Even though the first component can be tracked with current approaches, the second part still remains a challenge to be addressed.

Problem 1. In this context, design a system which can achieve the following:

• P1 (Dynamic specification) The system needs to be in contact with the environment without crashing the system while avoiding extreme ceiling effects.

• P2 (Constraint satisfaction) There is a need to respect physical limits, e.g., maximum force values of the rotors not to saturate the controller effort. In a practical application, for a potential inspection scenario, the system has to limit the contact force to protect the sensor.

• P3 (Offset–free interaction) The error due to the reference set given in \( P \) should be minimized.

Note that, besides for the physical interaction task, a similar case is also manifested itself in a path–following control [30].

3. Modeling

In the case of the interaction, where the contact with the environment is critical, the system has to sustain its stability under the reaction forces and complex aerodynamic effects particularly when it flies close to a structure, e.g., at the ceiling. Here, the modeling of the contact phase is highly convoluted with the dynamic of the UAV and its additional effects, which results in a complex system.

Remark 1. The system may need to operate in an uncluttered place. In order to interact with an unknown surface, the attached tool is characterized to provide a compliance behavior on the quadrotor. The front view of this representation is illustrated in Fig. 1. The area between two smooth curves in the figure illustrates unknown contact surface (ceiling), where the interaction force, \( F_{est} \), is lumped through the dynamic tool represented by the pair of spring and damper. The main idea behind this approach is augmenting the system model by external forces.
Assumption 1. It is assumed that the interaction tool on top of the UAV can be modeled using three lumped elements of mass, spring and damper. It is also assumed that UAV body has rigid system characteristics.

Therefore, the force exerted on $z$ axis can be given as

$$\begin{align*}
F_{ext} &= \begin{cases} 
0 & \Delta z \leq 0, \\
(k\Delta z + b\Delta z) & 0 < \Delta z,
\end{cases} 
\tag{3a}
\end{align*}$$

$$\Delta z = l_s(0) - l_s(i), \tag{3b}$$

$$0 < l_s \leq l_{ce}, \tag{3c}$$

where $k$ and $b$ are the stiffness and damping coefficients, respectively. The reaction forces during the interaction are represented by $F_{ext}$ and the interaction level is characterized by $\Delta z$. The subscripts of the external forces are given depending on the interacted orientation and its related axes. In the free–flight case where there is no interaction, the elastic tool length is indicated by $l_s(i)$. Once it is in contact, the changing values are represented by $l_s(i)$. Additionally, $l_{ce}$ is the maximum length of the tool (the subscript ce stands for the ceiling effect). Since the second order system parameters directly impose bounds for the external forces, their values are vital to compensate for impacts

forces and maintain contact forces in a desired bound.

The UAV with elastic tool equations are defined in a compact form with rigid body assumptions,

$$\begin{align*}
\dot{m} \ddot{\xi} &= RF - mgz, \tag{4a}
J \ddot{\Theta} &= \tau - \Theta \times J \dot{\Theta}, \tag{4b}
\end{align*}$$

where $\xi = [x, y, z]^T$ is the vector of translational states in inertial frame which is fixed to earth, $\Theta = [p, q, r]^T$ is the vector of angular rates and $J$ is the matrix of inertial moments. Additionally, $z$ is the axis on the inertial frame through the gravity direction. Here, $R$ is the transformation matrix between the body and inertial frames belongs to a special orthogonal group SO(3) and satisfies the condition of $R^{-1} = R^T$.

$$R = \begin{bmatrix}
C_\theta C_\phi & C_\theta S_\phi & -S_\theta \\
S_\theta C_\phi & S_\theta S_\phi & C_\phi \\
S_\phi & -C_\phi & 0
\end{bmatrix}, \tag{5}$$

where $\eta = [\phi, \theta, \psi]^T$ is the vector of Euler angles with abbreviations $c, \cos, s, \sin$. The external forces and moments are introduced to the system as

$$\begin{align*}
F &= \begin{bmatrix} 0 & 0 & F_z \end{bmatrix}^T + \begin{bmatrix} F_{ext} & F_{ext} & F_{ext} \end{bmatrix}^T, \tag{6a}
\tau &= \begin{bmatrix} L & M & N \end{bmatrix}^T + \begin{bmatrix} \tau_{ext} & \tau_{ext} & \tau_{ext} \end{bmatrix}^T. \tag{6b}
\end{align*}$$

Therefore, the attitude dynamics of the system can be given by

$$\dot{\Theta} = R \dot{\eta}, \tag{7}$$

where $R \in \mathbb{R}^3$ is the Euler rotation matrix,

$$R = \begin{bmatrix} 1 & c_\phi & c_\phi \sin \theta \\
0 & c_\phi & -s_\phi \\
0 & s_\phi/c_\theta & c_\theta
\end{bmatrix}. \tag{8}$$

In order to drive the system, the following control allocation matrix is defined.

$$\begin{bmatrix}
F_z \\
L \\
M \\
N
\end{bmatrix} = \begin{bmatrix}
c_\gamma & c_\gamma & c_\gamma & c_\gamma \\
-lc_\gamma & lc_\gamma & lc_\gamma & -lc_\gamma \\
lc_\gamma & lc_\gamma & -lc_\gamma & -lc_\gamma \\
-c_\theta & -c_\theta & c_\theta & -c_\theta
\end{bmatrix} \begin{bmatrix}
\Omega_x \\
\Omega_y \\
\Omega_z \\
\Omega_w
\end{bmatrix}, \tag{9}$$

which specifies the achievable limits of the system due to the maximum rotor speed ($\Omega_{max}$). The physical first limit stands out at actuation level which is the achievable maximum value for the rotor speeds. Since the maximum value of the rotors directly specifies a bound, the control allocation gives the upper bound for the achievable control limits by $F_{zmax}, L_{max}, M_{max}$ and $N_{max}$ (max. total thrust and moments on $x, y$ and $z$ axes, respectively).

The schematic representation in Fig. 1 shows the system on the $z$ axis and the following equations can be rewritten for the upper bound case,

$$\dot{m} \dot{w} = c_\phi \ddot{b} F_{zmax} - mg - F_{extmax}, \tag{10}$$

where $F_{extmax}$ is the tolerable maximum interaction force by the control actions.

The effect of the external force arises after the initial contact with the ceiling as given in (3a). It has to be highlighted here the vertical force for the contact sensor needs to be constrained to avoid a possible crash. In this context, the bound on the external force is redefined for the upper limit strength of the related sensor at the top:

$$F_{zextmax} = F_{kmax} + F_{dmax}, \tag{11}$$

In order to address the first subproblem P1, the aerodynamic analysis proposed in [32] is used. To avoid extreme ceiling effects, a lightweight elastic tool (77 gr) is 3D printed and mounted on the quadrotor. Even if the length of the arm is 30 cm, its deflection is limited to 12.5 cm. At the same time, the compensation mechanism is designed after a set of simulations for the desired interaction level. The specifications of the system including second order system parameters are given in Table 1.
which is the first component of the stage cost. The matrix $Q$ is positive-semidefinite and assigned to penalize the differences between the reference and the current state. In addition, the optimal control sequences for the inputs are given by

$$C_2 = (u_n(t) - u(t))^T R (u_n(t) - u(t)),$$

where the limits on the controls are given by $u_{\text{min}}$ and $u_{\text{max}}$. The integral starts from the current instant $t_f$ and takes the values over the prediction horizon. The terminal instant approximates the infinite horizon. In summary, the differentially flat states and controls are given as follows:

$$x(\cdot) = [x, y, z, u, v, w]^T,$$
$$u(\cdot) = [F_z, \phi, \theta, \psi]^T.$$

The specified parameters for the controller are given in Table 2.

### 4. Interaction Control

The system described in (1) can be reformulated as

$$\dot{x}(t) = f(x(t), u(t)) + d(t),$$

(12)

where the system is represented as the sum of a priori model and a disturbance model. This approach allows us to treat the disturbance separately, however, it still requires precise parameters estimation in real-time applications for a priori model [33, 34]. Nevertheless, the main motivation behind this representation is to update the system to represent the differences between the exact plant and the mathematical model while interacting with an unknown environment. To form all variables in a vector-matrix format, it is assumed to have constant disturbance effect, wherein $d = 0$. Therefore, external forces are lumped in the system as

$$d(\cdot) = [F_{\text{ext}}, F_{\text{ext}}, F_{\text{ext}}]^T.$$

(13)

To sense the external forces, it is possible to use an off-the-shelf sensor. However, force sensors come with a cost which might contradict to our main motivation to provide a low-cost system. Additionally, the sensor technology significantly developed over the past few decades but it may not be possible to measure all the states in a precise manner. Therefore, an estimation approach is required to be used for real-time applications. With respect to the need of having an interaction task, it is essential to provide constraints of a “safety jacketing” feature. In this context, the external force is required to be estimated in predefined bounds to update the controller. Simultaneously, it is also expected that the controller does not result in a high interaction force command to avoid hard contact on the sensor. Consequently, an optimization-based estimation and controller framework is required to address the subproblem P2 in a systematic way [35].

### 4.1. NMPC Formulation

NMPC executes an open-loop optimal control approach. However, for the given problem, the control action needs to be iteratively computed at each time step with an update given by the feedback for the current state in the following form

$$C_1 = (x_n(t) - x(t))^T Q (x_n(t) - x(t)),$$

(14)

which is the first component of the stage cost. The matrix $Q$ is positive-semidefinite and assigned to penalize the differences between the reference and the current state. In addition, the optimal control sequences for the inputs are given by

$$C_2 = (u_n(t) - u(t))^T R (u_n(t) - u(t)),$$

(15)

which is the second part of the stage cost. The nominal input values are represented by $u_n$. The positive–definite matrix $R$ penalizes the differences between the nominal input values and the current control actions. Another component of the objective function is the terminal cost to act at the end of the prediction horizon

$$J_{\text{e}} = (x_n(t_N) - x(t_N))^T Q_N (x_n(t_N) - x(t_N)).$$

(16)

Similar to the $Q$, the weight matrix $Q_N$ penalizes the error value at the last instant of the horizon, which is $t_N$, to reach within the finite time. Here, $t_N = t_f + t_h$, where the prediction horizon is $t_h$. Therefore, NMPC problem formulation ($V_1$) can be expressed as follows [36]:

$$V_1(\cdot) = \int_{t_f}^{t_N} (C_1 + C_2) dt + J_{\text{e}},$$

(17a)

subject to:  
$$x = f(t, x, u, d), \forall t \in [t_f, t_N],$$
$$u_{\text{min}} \leq u(t) \leq u_{\text{max}},$$
$$(17b)$$
$$x(t_f) = \hat{x}(t_f),$$
$$x(t) \in X, u(t) \in U.$$  

(17c)

(17d)

(17e)

where the limits on the controls are given by $u_{\text{min}}$ and $u_{\text{max}}$. The integral starts from the current instant $t_f$ and takes the values over the prediction horizon. The terminal instant approximates the infinite horizon. In summary, the differentially flat states and controls are given as follows:

$$x(\cdot) = [x, y, z, u, v, w]^T,$$
$$u(\cdot) = [F_z, \phi, \theta, \psi]^T.$$

(18a)

(18b)

### 4.2. NMHE Formulation

The estimation is required to be solved at each time step to provide the current states for the controller. In order to eliminate noise from the measurement, the following expression is introduced,

$$L_1 = (y_m(t) - h(\cdot))^T H (y_m(t) - h(\cdot)),$$

(19a)

$$h(\cdot) = h(\hat{x}(t), \hat{u}(t), d(t)),$$

(19b)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor force-thrust constant ($c_F$)</td>
<td>$4.20 \times 10^{-4} \text{N}\cdot\text{s}^2$</td>
</tr>
<tr>
<td>Rotor torque-drag constant ($c_D$)</td>
<td>$7.50 \times 10^{-6} \text{Nm}^2$</td>
</tr>
<tr>
<td>Mass of the quadrotor ($m$)</td>
<td>$1.40 \text{kg}$</td>
</tr>
<tr>
<td>Arm length ($L$)</td>
<td>$0.30 \text{m}$</td>
</tr>
<tr>
<td>Moment of Inertia around x-axis ($I_{xx}$)</td>
<td>$0.0156 \text{kgm}^2$</td>
</tr>
<tr>
<td>Moment of Inertia around y-axis ($I_{yy}$)</td>
<td>$0.0168 \text{kgm}^2$</td>
</tr>
<tr>
<td>Moment of Inertia around z-axis ($I_{zz}$)</td>
<td>$0.0201 \text{kgm}^2$</td>
</tr>
<tr>
<td>Spring coefficient ($k$)</td>
<td>$160 \text{N/m}$</td>
</tr>
<tr>
<td>Damping coefficient ($b$)</td>
<td>$10 \text{kgms}^{-1}$</td>
</tr>
<tr>
<td>Gravitational acceleration ($g$)</td>
<td>$9.81 \text{ms}^{-2}$</td>
</tr>
</tbody>
</table>

Table 1: Specifications of the System

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time step ($t_f$)</td>
<td>0.02 s</td>
</tr>
<tr>
<td>Prediction and control horizon ($t_h$)</td>
<td>40 s</td>
</tr>
<tr>
<td>Stage cost weight $Q$</td>
<td>diag(30, 30, 10, 1, 2, 5)</td>
</tr>
<tr>
<td>Input weight $R$</td>
<td>diag(30, 30, 80, 4 x 10^{-2})</td>
</tr>
<tr>
<td>Terminal cost weight $Q_N$</td>
<td>diag(60, 60, 20, 2, 2, 5)</td>
</tr>
<tr>
<td>Constraints</td>
<td>$0.5\text{mg} \leq F_z \leq 1.5\text{mg}$</td>
</tr>
</tbody>
</table>

Table 2: Specification of NMPC Interaction
270 where \( \gamma \) gives the measurements for each instant. The
275 measurement function \( h() \) includes the estimated states, controls
276 and disturbances. The matrix \( H \) is chosen upon the measurement
280 errors covariance matrix.

Considering any potential modeling errors and mismatches,
285 a balance function is provided to maintain precise estimations
290 for the level of modeling accuracy with
\[
L_2 = (\omega(t))^T W(\omega(t)), \quad (20)
\]
where \( \omega(t) \) is the process noise and \( W \) is the corresponding
295 positive–definite matrix to mitigate its effect from the estima-
300 tions.

Lastly, for estimation reliability, an additional term called \( \ell \)
305 arrival cost is included in the algorithm. This term is responsible
310 for the past data from the beginning of the estimation till the first
315 instant of the estimation window
\[
\ell_{EW} = (\hat{x}(t_{EW}) - \bar{x}(t_{EW}))^T S(\hat{x}(t_{EW}) - \bar{x}(t_{EW})), \quad (21)
\]
where \( \hat{x} \) is the estimated states by the stage cost and the \( \bar{x} \) is
325 the estimated states by the arrival cost. The matrix \( S \) is the
330 estimation error’s covariance and it is updated throughout the
335 implementation. Readers interested in the computation of the
340 arrival cost can refer to [37]. Therefore, NMHE formulation
345 \((V_2)\) can be given as follows [36]
\[
V_2(t) = \int_{t_{EW}}^{t} (L_1 + L_2)dt + \ell_{EW}, \quad (22a)
\]
subject to:
\[
\dot{x} = f(t,x,u,d), \forall t \in [t_{EW}, t], \quad (22b)
\]
\[
d_{\min} \leq d(t) \leq d_{\max}, \quad (22c)
\]
\[
\hat{x}(t) \in X, \, \bar{x}(t) \in U, \quad (22d)
\]
where the limits on the disturbances are given by \( d_{\min} \) and \( d_{\max} \).
350 The integral starts from \( t_{EW} = t_0 - t_0 \) and takes the values over the
355 estimation window \( t_0 \) until \( t_1 \). The arrival cost term summarizes
360 the past estimation. The measurements are given by
\[
y_m(t) = [x, y, z, u, v, w, \phi, \theta, \psi]^T. \quad (23)
\]

The specified parameters for the estimations are given in Table
365 It is noted that the states and controls are considered in the
370 stage cost weight. However, the states and external forces are
375 weighted in the process noise and the arrival cost.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time step ( t_s )</td>
<td>0.02 s</td>
</tr>
<tr>
<td>Estimation window ( t_0 )</td>
<td>40</td>
</tr>
<tr>
<td>Stage cost weight ( H )</td>
<td>diag(0.0025, 0.0025, 0.0025, 0.0025, 0.0025, 0.0025, 0.0025, 0.0025)</td>
</tr>
<tr>
<td>Process noise weight ( W )</td>
<td>diag(1/30, 1/30, 1/30, 1/30, 1/30, 1/30, 1/30, 1/30)</td>
</tr>
<tr>
<td>Arrival cost weight ( S(0) )</td>
<td>diag(0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01)</td>
</tr>
<tr>
<td>Constraints</td>
<td>[ \bar{a} \leq F, s, \leq 2N ]</td>
</tr>
</tbody>
</table>

Table 3: Specification of NMHE

4.3. Tuning of the Interactive Predictive Controller for UAV Tool Operations

As being a model–based controller, the proposed approach
400 utilizes an optimization algorithm in order to compute the op-
405 timum input sequence, considering the future behavior of the
405 model. Similar to the human decision process, it is an intuitive
410 controller that can also handle multivariable control prob-
415 lems successfully. Another advantage of using an optimization–
420 based controller as compared to conventional controllers, such as
425 PID, it can handle the constraints both on the inputs and states.
Considering these advantages, the predictive controller gains
430 widespread acceptance from industry and control practitioners
435 [38]. In its objective function, the terminal cost and the set of
440 constraints are imposed to keep the advantages of the infinite
445 horizon approximation [39]. Therefore, the stability, robustness
450 and the recursive feasibility can be guaranteed, where the system
455 states stay in an initial feasible set in which there is an admis-
460 sible control action ensures the reachability through prediction
462 horizon. Without loss of generality, wider terminal sets and/or
465 longer prediction horizons result in a larger initial feasible set.
To guarantee the stability, one can impose a terminal (in)equality
470 constraint and/or a corresponding terminal cost [40]. A possible
475 extension can be based on a locally stabilizing control as an ad-
480 dition of the prediction horizon [41]. Similarly, when the state is
485 steered into the terminal set, one can switch to a Lyapunov stable
490 controller [42]. Furthermore, the prediction horizon can be kept
495 sufficiently long [43]. In this implementation, we adapted to im-
500 pose terminal cost with sufficiently long horizons. Additionally,
505 the external force information is bounded in the algorithm.

The first stage of the tuning starts with assigning a proper
510 sampling frequency. In this context, there is a dilemma between
515 the computation time and the closed loop cost based on the
520 objective function. Since the system runs in faster cycles as
525 compared to the industrial process models, the sampling needs
to be held tight. However, once the sampling chosen in smaller
530 ranges, it increases the computation time. On the other hand, the
535 larger sampling times increase the cost of the objective function.
Since it is our aim, where the system needs to be more responsive
540 for the aerial physical interaction, the system is discretized with
545 a grid size of 20 ms using multiple shooting method. In this
550 implementation, Gauss–Legendre integrator of order 4 is utilized
555 with 2 steps per shooting interval.

The second stage of the tuning is the selection of the predic-
560 tion horizon for the predictive control and the estimation window.
Without loss of generality, generic solvers do not scale the com-
565 plexity linearly for the number of horizons and/or estimation
570 windows. However, it can be scaled linearly using a tailored
algorithm such as Riccati recursion-based solver [44]. Unfortu-
575 nately, the dimension of the states might increase the compu-
tation time cubically in Riccati recursion-based approach [45].
In this implementation, we adapted to use active set–method
580 where the computation time still increases more than a linear
585 case [46]. Although the system performance is also acceptable
590 in shorter horizons, the horizon is assigned as 40 to maintain
595 the stability both on the estimation and the control. Before
600 the real–time implementation, the system is tested in Gazebo
environment for different number of horizons and estimation
windows and the computation time analysis is illustrated in Fig. 2. In this context, one can expect a higher computational load for the interactive controller case. In generic rigid body equations, the system is actuated by \([F_x, F_y, F_z]\). In the nominal and NMPC case, first two components of the actuation are replaced by \([F_{x\text{est}}, F_{y\text{est}}]\) pair and it is defined as intermediate state, where \(F_{x\text{est}} = F_{y\text{est}} = 0\). The last component \(F_z\) is defined as a control input. However, in the interactive controller case, the external forces \([F_{x\text{est}}, F_{y\text{est}}, F_{z\text{est}}]\) are defined as online data, where they get update from NMHE. Similar to the nominal case, \(F_z\) is the control action for the vertical force. Even though the intermediate states are constant in nominal controller case, they increase the computation time, particularly for the higher prediction horizons. On the other hand, when the external forces are defined as online data, the computation time is affected relatively less by the higher number of horizons. Since the Gazebo simulation brings a heavy load on the computer, the real time computation values may be varied due to having a different real time scale. The computations become sparse in some instants as it can be seen as outliers in the computation times.

![Figure 2: Computation times for different number horizons and estimation windows](image)

The last stage of the tuning is the selection of the weight matrices. Readers interested in the predictive controller weight matrices selection can refer to [42]. For the estimations, the stage cost weight \(H\) is assigned based on standard deviations of the measurements. There is also a dilemma in the selection of the process noise variances. While the low values contribute to more precise estimations, this causes a time lag. Accordingly, the process noise values are selected in a balanced manner. Another significant factor is the arrival cost. Once it is set to low values, it results in a more reflexive estimator. However, the extremely low values can lead to the overfitting problem. On the other hand, the higher values cause sluggish response where the system may not capture the external forces in aerial physical interactions. Therefore, the arrival cost variances are set to comparably low values.

### 4.4. Closing the Loop

The proposed approach is illustrated in Fig. 3. Due to the similarity of the optimization settings, this dual problem can be solved by the same algorithm. The system modeled in ACADO (automatic control and dynamics optimization) that creates tailored C codes for a designed optimization problem [47]. The solution is performed using multiple shooting method throughout the horizon where the nonlinear model is discretized wherein Gauss–Legendre integrator is used. This approach is based on parameterizing an infinite time continuous optimal control problem into a finite dimensional nonlinear program. In the solution part, real–time iteration scheme is used for each single sequential quadratic programming (SQP) iteration. In each sequential step, quadratic approximations are used for the cost function and the constraints. Afterwards, the obtained quadratic programming (QP) problem is solved in qpOASES which is one of the most powerful tools for QP–type problems. This process is repeated while the KKT values are minimized for each step. In the solution algorithm, the shifting technique is used for subsequent problems based on the prior information. Therefore, one SQP step is performed based on the previous control and state variables. Readers interested in the stability of the real–time iteration scheme can refer to [48, 49].

![Figure 3: A general block diagram: interactive controller and estimator](image)

As can be seen from Fig. 3, there are two main components in terms of the control. Considering the faster cycles in the rotational channel compared to the translational dynamics, a cascaded architecture is preferred in real–time implementation [50]. The stabilization of the attitude angles and their computations are executed on the embedded platform, Pixhawk, which is based on an STM32 Cortex M4 processor (168 MHz). In order to control the attitude angles, Pixhawk executes two main loops based on C++ for each channel. The high–level controller commands are utilized on Raspberry Pi3 which is serially connected to Pixhawk via FTDI. By defining task–based trajectories from the off–board computer, the system leverages NMHE and NMPC problems simultaneously.

In summary, the subproblem \(P3\) is addressed by updating the model with the current information about the external forces within the constrained optimization. Additionally, the control
allocation is redefined for the total forces:

\[ F_{z_{i-NMPC}} = F_z - F_{z\text{ext}}, \]  
(24)

where \( F_{z_{i-NMPC}} \) is fed into the system as a throttle command for the interacted NMPC case. On the other hand, in the nominal case, \( F_z \) is given to the system to be applied. The closed loop control algorithm, coded in C/C++, is communicated with ROS environment through generated C codes. The interaction control algorithm is executed on a computer with Intel i7-6500U CPU and 16GB of RAM memory. The system is marked to be experimented in a motion capture system by communicating via ROS–Kinetic. The vision system OptiTrack consists of eight cameras to localize the system at 240 Hz over the wifi network.

5. Experiments

The experiments are conducted in two phases: (i) free flight and push recovery operation; (ii) tool sliding on the ceiling. While the system simply approaches the ceiling and presses the contact tool in the former experiment, sliding on the surface while maintaining the contact force is conducted in the latter one. The snapshots from the experiments can be seen in Fig. 4.

5.1. Free–flight and tool interaction

Firstly, an idle point is specified to initialize the trajectories. Fig. 5 shows the trajectory on x and y axes to reach the identified point below the ceiling. Since there is no explicit disturbance in the free–flight phase, the performance of the NMPC and the interacted NMPC are similar to each other. In these experiments, interacted NMPC is updated by \( F_{z\text{ext}} \) to achieve desired interaction levels with the ceiling.

After the UAV reaches the specified point below the ceiling, a sequential ramp signal is given to observe the behavior of the system. In this case, the trajectory on the z axis includes the free–flight below the ceiling and the contact phases. In order to make it visible, the interaction with the ceiling instants are given in green color as can be seen in Fig. 6. Even if the NMPC compresses the tool a few centimeters, it could not follow given trajectories for the higher compression values which require significant force actions. However, the interacted NMPC achieves to track even under significant force values. The corresponding controller efforts (\( F_z \)) can be seen in Fig. 7.

In order to achieve such level of interaction, the controller is updated by the vertical force information \( F_{z\text{ext}} \). In each time step, NMHE provided external force values to update the interacted NMPC. In order to protect the system, the estimation values lie
in a predefined constraint threshold (-6 N) as can be seen in Fig. 8.

The designed controller generates reference trajectories for the PIDs to control the attitude angles. The generated trajectories by the NMPC as well as the interacted NMPC and the responses of the PIDs are sufficient to maintain the interaction control.

One of the significant aspects of the proposed approach is the computational burden. The system’s computation times are analyzed for both cases. Fig. 9 visualizes the computations for each time instant. In some parts, the computations violate the sampling time range. However, in overall, the system achieved in following the given trajectories and maintaining the stability during the experiments.

Due to having one SQP iteration in the algorithm, the KKT values throughout the experiments are observed. The KKT values are given in boxplots for each flight mode and illustrated in Fig. 10. To highlight the main distribution, the outliers are not given in the figure. The differences between the desired interaction levels and the NMPC reactions resulted in higher KKT values. Thanks to the vertical force information given by NMHE, the KKT values are decreased in the interacted NMPC case. In an ideal case, the KKT values are expected to be zero. However, the nonlinearities, external disturbances and unmatched uncertainties in the problem cause higher values for the KKT. In the free-flight case, where the physical interaction does not exist, the performance of the controllers is close to each other. Since the controller gets an update for the external forces from the NMHE for each time step, it behaves more responsive. In some cases in the experiments, it is observed that the system response becomes slightly aggressive as compared to the nominal case. This can cause small oscillations in real time, which may result in higher KKT values in some instants (e.g., FF-2). However, in overall, it is reduced in interacted NMPC case, particularly for the physical interaction task.

The computation times and the KKT values are summarized in Table 4. Hypothesized force–based predictive approach implements constrained optimization method by explicitly considering the external forces in the algorithm and UAV with tool configuration to solve the contact–based interaction operation in the air. The results have shown that the disturbance and offset caused by the tool effects are suppressed on the vertical axis. Since there is a direct actuation on the vertical channel, it is achieved to mitigate the external forces (in overall 70.90%).

5.2 Sliding on the surface

In these experiments, circular trajectories are generated for two cases: (i) first interaction level (mild vertical force); (ii) second interaction level (high vertical force). In the first interaction level, the elastic tool is compressed in the range of
The number of violation of the sampling time
The percentage of the solution below the sampling time (%)

<table>
<thead>
<tr>
<th></th>
<th>NMPC</th>
<th>Interacted NMPC</th>
<th>NMHE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum (ms)</td>
<td>1.0640</td>
<td>0.6034</td>
<td>2.8690</td>
</tr>
<tr>
<td>Average (ms)</td>
<td>2.7236</td>
<td>2.4443</td>
<td>3.9993</td>
</tr>
<tr>
<td>Maximum (ms)</td>
<td>30.3657</td>
<td>33.1764</td>
<td>17.3490</td>
</tr>
<tr>
<td>Minimum KKT</td>
<td>$1.5171 \times 10^{-7}$</td>
<td>$6.3791 \times 10^{-8}$</td>
<td>$2.2378 \times 10^{-7}$</td>
</tr>
<tr>
<td>Average KKT</td>
<td>0.14123</td>
<td>0.0970</td>
<td>3.7602 $\times 10^{-5}$</td>
</tr>
<tr>
<td>Maximum KKT</td>
<td>2.38194</td>
<td>2.0781</td>
<td>0.0048</td>
</tr>
</tbody>
</table>

The statistical results are given in Fig. [15] In the first interaction level, the mean error is decreased 15.10% and 30.94% on the x and y axes, respectively. In the second interaction level, the mean error is decreased 24.10% and 45.42% on the same axes. As can be seen, in the second interaction level, there is a higher need to get the external force information due to working with higher vertical and horizontal force values.

6. Conclusions

This paper presents a force estimation–based interactive control approach for the UAV–tool contact problems. After defining the associated challenges for the UAV interaction, the
subproblems were identified to be addressed in the proposed approach. In this context, a complete mathematical model was derived through the body and inertial axes systems. Thereafter, an optimization–based framework consisting of NMHE and NMPC was introduced for the proposed interactive control approach. In order to achieve desired interaction levels, a novel control allocation was provided for the total forces. The experiments showed that the system performance under the nominal NMPC can be recovered by the instantaneous force update. The design and implementation procedure of the proposed algorithm is very practical as it does not require a precise model, external force measurement, and exact information about the environment. For the real–time feasibility, the computation times and

---

Figure 12: Controller actions (attitude angles) for tool sliding case: NMPC and interacted NMPC

Figure 13: External force estimation on planar axes
will follow plays an important role. For instance, if we have a short range sensor that can give only limited information about the surroundings of the UAV or target, the system will have an extremely cautious behavior in the approach phase of the mission. In this sense, a trade-off between remote sensing tools and interactive tools will be made in order to obtain optimum path regarding the mission operation time and battery lifetime.

Acknowledgements

The authors acknowledge the support of and wish to thank to the SINGA scholarship (SING-2014-2-0256).

References
