Towards Model-predictive Control for Aerial Pick-and-Place

Gowtham Garimella[1] and Marin Kobilarov[1]

***Abstract*— This paper considers pick-and-place tasks using aerial vehicles equipped with manipulators. The main focus is on the development and experimental validation of a nonlinear model-predictive control methodology to exploit the multi-body system dynamics and achieve optimized performance. At the core of the approach lies a sequential Newton method for unconstrained optimal control and a high-frequency low-level controller tracking the generated optimal reference trajectories. A low cost quadrotor prototype with a simple manipulator extending more than twice the radius of the vehicle is designed and integrated with an on-board vision system for object tracking. Experimental results show the effectiveness of model-predictive control to motivate the future use of real-time optimal control in place of standard ad-hoc gain scheduling techniques.**

I. Introduction

This work develops a nonlinear model-predictive control (NMPC) approach to enable agile pick-and-place capabilities for aerial vehicles equipped with manipulators. Aerial manipulation using vertical take-off and landing (VTOL) vehicles is a relatively new research area with a potential for various novel applications such as coordinated assembly, construction, and repair of structures at high altitudes, or operating in difficult-to-access, remote, or hazardous locations to e.g. install sensors or obtain samples. Autonomous control of such system is challenging primarily due to disturbances from interactions with the environment, due to additional dynamics caused by a moving manipulator, and due to difficulties associated with dexterous manipulation.

Initial work related to aerial manipulation included slung load transportation with helicopters [6], [20], grasping with novel adaptive end-effectors [27], [26], construction using teams of quadcopters [18], or pole balancing tasks [7]. More recently, there has been a focus on autonomous construction and environment interaction, with initial demonstrations in laboratory settings. The Aerial Robotics Cooperative Assembly System (ARCAS) project [2], [12], [8], Mobile Manipulating Unmanned Aerial Vehicle project [25], [16], [17] and Airobots project [1], [31] have demonstrated complex manipulation and assembly tasks using multiple degrees of freedom manipulators. Other important developed capabilities include telemanipulation [22], [11] or avian-inspired agile grasping [30]. In addition to control-related challenges, accurate pose estimation of objects is of central importance and has been considered through image-based visual servoing [29] and marker-based pose computation [2], [7]. Real-time recognition and aerial manipulation of arbitrary

[1]Gowtham Garimella and Marin Kobilarov are with the Department of Mechanical Engineering, Johns Hopkins University, 3400 N Charles Str, Baltimore, MD 21218, USA `ggarime1|marin@jhu.edu`

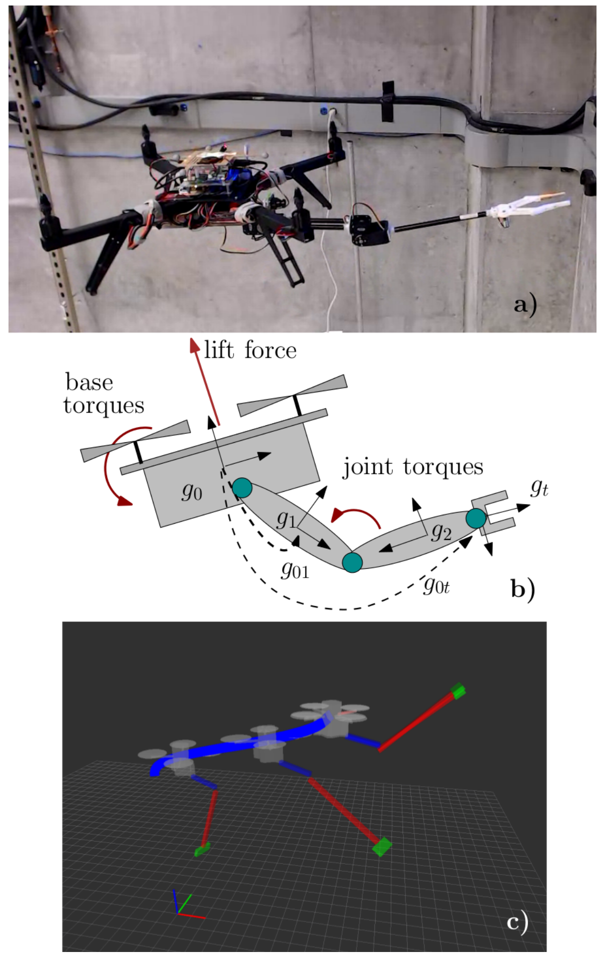

Fig. 1. a) a prototype quadrotor with manipulator, b) schematic model, c) a computed optimal trajectory viewed in Robot Operating System (ROS).

unengineered objects in natural settings remains largely an open problem.

Control strategies for aerial manipulation can be divided into *coupled* which consider the full multi-body system model [19], [14], [24], and *decoupled* based on separate controllers for the base body and manipulator [28]. The key difference is that the decoupled approach treats external forces from the arm or environment as disturbances to be compensated by the vehicle.

In this paper, we propose an optimal control algorithm for generating reference trajectories to pick an object using aerial vehicle. Experimental verification has been performed using a minimalist low-cost system based on a two-degree of freedom manipulator with a simple gripper. The task is made challenging by using a monocular camera to recognize and track the target object. To facilitate recognition, objects are engineered with LED markers correspoding to known features. A detailed nonlinear model is employed by the optimal control framework to capture the interaction between the arm and quadcopter. Currently due to computational limitations, NMPC operates at 10Hz and is not used for *real-time control*. Hence a high frequency nonlinear controller is coupled with the optimal control framework to track the reference trajectories.

The paper is organized as follows. The dynamical multi-body system modeling and numerical optimal control approach are described in §II and §III, respectively. Then we proceed to describe the experiments conducted to validate the optimal controller in §IV. Finally we provide the results of the experiments conducted and discuss future work in §V.

II. System Modeling

The aerial robot is modeled as a free-flying multi-body system consisting of $n+1$ interconnected rigid bodies arranged in a tree structure. The configuration of body $\#i$ is denoted by $g_i \in SE(3)$ and defined as

$$g_i = \begin{pmatrix} R_i & p_i \\ 0 & 1 \end{pmatrix}, \quad g_i^{-1} = \begin{pmatrix} R_i^T & -R_i^T p_i \\ 0 & 1 \end{pmatrix}.$$

where $p_i \in \mathbb{R}^3$ denotes the position of its center of mass and and $R_i \in \mathrm{SO}(3)$ denotes its orientation. Its body-fixed angular and linear velocities are denoted by $\omega_i \in \mathbb{R}^3$ and $\nu_i \in \mathbb{R}^3$. The pose inertia tensor of each body is denoted by the diagonal matrix $\mathbb{I}_i$ defined by

$$\mathbb{I}_i = \begin{pmatrix} \mathbb{J}_i & 0 \\ 0 & m_i I_3, \end{pmatrix}$$

where $\mathbb{J}_i$ is the rotational inertia tensor, m_i is its mass, and I_n denotes the n-x-n identity matrix. The system has n joints described by parameters $r \in \mathbb{R}^n$. Following standard notation [23], the relative transformation between the base body#0 and body#i is denoted by $g_{0i} : \mathbb{R}^n \to \mathrm{SE}(3)$, i.e.

$$g_i = g_0 g_{0i}(r).$$

The control inputs $u \in U \subset \mathbb{R}^{m=n+4}$ denote the four rotor speeds squared and the n joint torques. More specifically, $u_i = \Omega_i^2$ for $i = 1, \ldots, 4$ where Ω_i is the rotor speed of the i-th rotor, and u_{4+i} denotes the i-th joint torque, for $i = 1, \ldots, n$.

The *configuration* of the system is thus given by $q \triangleq (g, r) \in Q \triangleq SE(3) \times \mathbb{R}^n$, where $g \in SE(3)$ is a chosen reference frame moving with the robot. In this work we take the base body as a moving reference, i.e. $g \equiv g_0$. The *velocity* of the system is given by $v \triangleq (V, \dot{r}) \in \mathbb{R}^{6+n}$, where $V \in \mathbb{R}^6$ denotes the body-fixed velocity of the moving frame g and $\dot{r} \in \mathbb{R}^n$ denotes the joint angle velocities. The base velocity satisfies $\widehat{V} = g^{-1}\dot{g}$ where the "hat" operator $\widehat{V}$ for a given $V = (\omega, \nu)$ is defined by

$$\widehat{V} = \begin{bmatrix} \widehat{\omega} & \nu \\ 0_{1\times 3} & 0 \end{bmatrix}, \quad \widehat{\omega} = \begin{bmatrix} 0 & -w_3 & w_3 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \quad (1)$$

With these definitions, the full *state* of the system is $x \triangleq (q, v) \in X \triangleq Q \times \mathbb{R}^{6+n}$.

Continuous Equations of Motion: The coordinates for our setting are $q = (g, r)$ where the pose $g \in SE(3)$ and r represents joint parameters. For optimal control purposes, it is necessary to avoid Euler angle singularities and, in addition, it is advantageous to avoid unit quaternion constraints. To achieve this, the dynamics is defined directly on state space X as:

$$\dot{g} = g\widehat{V} \quad (2)$$
$$M(r)\dot{v} + b(q, v) = Bu, \quad (3)$$

where the mass matrix $M(r)$, *bias* term $b(q, v)$, and constant control matrix B are computed analogously to standard methods such as the articulated composite body algorithm [5] or using spatial operator theory [10]. With our coordinate-free approach the mass matrix in fact only depends on the shape variables r rather than on q and for tree-structured systems can be computed readily according to

$$M(r) = \left[\begin{array}{c|c} \mathbb{I}_0 + \sum_{i=1}^{n} A_i^T \mathbb{I}_i A_i & \sum_{i=1}^n A_i^T \mathbb{I}_i J_i \\ \hline \sum_{i=1}^n J_i^T \mathbb{I}_i A_i & \sum_{i=1}^n J_i^T \mathbb{I}_i J_i \end{array}\right] \quad (4)$$

using the adjoint notation $A_i := \mathrm{Ad}_{g_{0i}^{-1}(r)}$, and Jacobian $J_i := \sum_{j=1}^n [g_{0i}^{-1}(r)\partial_{r_j} g_{0i}(r)]^\vee$,where $g_{0i}(r)$ is the relative transformation from the base body to body $\#i$ and $\mathbb{I}_i$ is the inertia tensor of body $\#i$ [23].

The bias term $b(q, v)$ encodes all Coriolis, centripetal, gravity, and external forces. Finally, for a quadrotor model the constant control matrix B has the form

$$B = \left[\begin{array}{cccc|c} 0 & -lk_t & 0 & lk_t & \\ -lk_t & 0 & lk_t & 0 & \\ k_m & -k_m & k_m & -k_m & \\ 0 & 0 & 0 & 0 & 0_{6\times\ell} \\ 0 & 0 & 0 & 0 & \\ k_t & k_t & k_t & k_t & \\ \hline & 0_{\ell\times 6} & & & I_{\ell\times\ell} \end{array}\right],$$

where l, k_t, k_m are known constants. This can be easily extended to other multi-rotor configurations.

Discrete Dynamics: For computational purposes we employ discrete-time state trajectories $x_{0:N} \triangleq \{x_0, \ldots, x_N\}$ at equally spaced times $t_0, \ldots, t_N \equiv t_f$ with time step $\Delta t = \frac{t_f - t_0}{N}$. The discrete state at index k approximates the continuous state at time $t_k = t_0 + k\Delta t$, i.e. $x_k \approx x(t_k)$ and is defined by $x_k = (g_k, r_k, V_k, \Delta r_k)$, where Δr_k denotes the joint velocities at k-th stage. A simple discrete-time version of the continuous dynamics (2)–(3) is then employed:

$$g_{k+1} = g_k \,\mathrm{cay}\left(\Delta t V_{k+1}\right), \quad (5)$$
$$r_{k+1} = r_k + \Delta t \Delta r_{k+1}, \quad (6)$$
$$M(r_k)\frac{v_{k+1} - v_k}{\Delta t} + b(q_k, v_k) = Bu_k. \quad (7)$$

This is a first-order semi-implicit method since one first updates the velocity v_{k+1} using the dynamics (7) and then updates the configuration using the kinematics (5)–(6). The method requires small time-steps to ensure stability ($\Delta t \leq$

100ms is sufficient for the aerial systems considered), higher-order methods are also possible [15], [13].

Note that the base pose update (5) is performed using the Cayley map cay : $\mathbb{R}^6 \to SE(3)$ defined (see e.g. [15]) by

$$\text{cay}(V) = \begin{bmatrix} I_3 + \frac{4}{4+\|\omega\|^2}\left(\widehat{\omega} + \frac{\widehat{\omega}^2}{2}\right) & \frac{2}{4+\|\omega\|^2}(2I_3 + \widehat{\omega})\,\nu \\ 0 & 1 \end{bmatrix}, \quad (8)$$

instead of the more standard exponential map on $SE(3)$ [23], [4] since it is an accurate and efficient approximation, i.e. $\text{cay}(V) = \exp(V) + O(\|V\|^3)$, it preserves the group structure, and has particularly simple to compute derivatives. Its inverse is denoted by $\text{cay}^{-1} : SE(3) \to \mathbb{R}^6$ and is defined for a given $g = (R, p)$, with $R \neq -I$, by

$$\text{cay}^{-1}(g) = \begin{bmatrix} [-2(I+R)^{-1}(I-R)]^\vee \\ (I+R)^{-1}p \end{bmatrix}.$$

III. Model Predictive Control

To achieve agile pick-and-place motions we employ model-predictive control to optimize future trajectories over the interval $[t_0, t_f]$ where t_0 is the current time and t_f is a specified moving horizon. A typical horizon $t_f - t_0$ for the considered aerial maneuvers is between 2 and 5 seconds. Two methods for unconstrained optimal control are considered in view of their capacity for near real-time performance: a simple Gauss-Newton shooting method and a Stagewise Newton sweep method.

A. Optimal Control Formulation

The NMPC problem can be generally formulated as the minimization of:

$$J(x_{0:N}, u_{0:N-1}) \triangleq L_N(x_N) + \sum_{k=0}^{N-1} L_k(x_k, u_k), \quad (9)$$

$$\text{subject to:} \quad x_{k+1} = f_k(x_k, u_i), \quad u_k \in U \quad (10)$$

where f_k encodes the integrator (5)–(7) and U defines the admissible control set. The stage-wise cost penalizes deviation from a desired nominal state x_d and controls u_d and is given by

$$L_k(x_k, u_k) = \frac{1}{2}\|x_k - x_d\|^2_{Q_k} + \frac{1}{2}\|u_k - u_d\|^2_{R_k}, \quad (11)$$

while the terminal cost is defined by

$$L_N(x_N) = \frac{1}{2}\|x_N - x_f\|^2_{Q_f}, \quad (12)$$

where $Q_k \geq 0, Q_f > 0, R_k > 0$ are appropriately chosen diagonal matrices to tune the vehicle behavior while reaching a desired final state x_f. In the aerial robot application the matrix Q_k contains non-zero terms corresponding to a desired velocity only.

B. Overloading $\pm$ operator on the group SE(3)

Numerical optimal control is based on vector calculus which is not directly applicable to states $x = (g, r, v)$ containing matrix elements $g \in SE(3)$. Hence, we use vector operators with analogous "retract" and "lift" operators on $SE(3)$.

The *lift* operator on $SE(3)$ is equivalent to operator *minus* $(\cdot) - (\cdot) : SE(3) \times SE(3) \to \mathbb{R}^6$

$$g_b - g_a = V \iff \text{cay}^{-1}(g_a^{-1}g_b) = V,$$

$$g_b - g_a \triangleq \text{cay}^{-1}(g_a^{-1}g_b) = V \in \mathbb{R}^6, \quad (13)$$

or practically speaking the differences between two poses approximately equals the constant body-fixed velocity V with which g_a moves to align with g_b after one unit of time. The *retract* operator on $SE(3)$ is equivalent to *plus* or *minus* $(\cdot) \pm (\cdot) : SE(3) \times \mathbb{R}^6 \to SE(3)$ according to

$$g_a \pm V \triangleq g_a\text{cay}(\pm V) = g_b \in SE(3), \quad (14)$$

i.e. adding/subtracting a vector V to/from the matrix g_a is interpreted as shifting g_a using a unit-time transformation with constant body-fixed velocity V. These operations are literally overloaded in our C++ implementation when one attempts to subtract two SE(3) matrices, or add/subtract a vector V to/from a matrix g_a. With these definitions, the errors $x_i - x_d$ and $x_N - x_f$ appearing in the costs (11),(12) are defined using the lift operator (13) so that, e.g. the latter with $x_f = (g_f, r_f, v_f)$ should be understood as

$$x_N - x_f \equiv \begin{bmatrix} \text{cay}^{-1}(g_f^{-1}g_N) \\ r_N - r_f \\ v_N - v_f \end{bmatrix}.$$

C. Gauss-Newton shooting method

One of the simplest, often overlooked, but surprisingly effective methods for solving the optimal control problem (9)–(10) is a shooting method exploiting the least-squares nature of the costs (11)–(12). It is formulated by parametrizing the discrete control trajectory $u_{0:N-1}$ using a vector $\xi \in \mathbb{R}^{\ell \leq Nm}$, encoded through the functions $u_k = \phi_k(\xi)$ for each $k = 0, \ldots, N-1$. For instance, ξ could contain the knots of a B-spline from which each u_k is extracted. The simplest parametrization is to simply set $\xi = u_{0:N-1}$. Using the dynamics each state can be expressed as a function of ξ which is encoded through the functions $x_k = \psi_k(\xi)$ for $k = 0, \ldots, N$. The cost is then expressed as $J(\xi) = \frac{1}{2}h(\xi)^T h(\xi)$, where $h : \mathbb{R}^\ell \to \mathbb{R}^{N(m+n+6)}$ is given by

$$h(\xi) = \begin{bmatrix} \sqrt{R_0}\,(\phi_0(\xi) - u_d) \\ \sqrt{Q_1}\,(\psi_1(\xi) - x_d) \\ \sqrt{R_1}\,(\phi_1(\xi) - u_d) \\ \vdots \\ \sqrt{Q_{N-1}}\,(\psi_{N-1}(\xi) - x_d) \\ \sqrt{R_{N-1}}\,(\phi_{N-1}(\xi) - u_d) \\ \sqrt{Q_f}\,(\psi_N(\xi) - x_f) \end{bmatrix}.$$

Since $R_i > 0$ the Jacobian $\partial h(\xi)$ is guaranteed to be full rank and one can apply a Gauss-Newton iterative method directly to update $\xi \to \xi + \delta\xi$ where $\delta\xi = -(\partial g^T \partial g)^{-1}\partial g^T g$. In addition, the Jacobian has a lower-triangular structure that can be exploited in the Cholesky GN solution. The complexity of this method is still $O(\ell^3)$ which is only acceptable for small ℓ, e.g. $\ell \leq 100$ in order to achieve real-time performance. The key advantage of the GN approach is its simplicity and robustness by employing standard regularization and line-search techniques [3].

A more efficient method with complexity $O(N(n+m))$ that exploits the recursive optimal control problem structure is presented next.

D. Stagewise Newton and Differential Dynamic Programming

The second optimal control method used in this work is based on a coordinate-free recursive NMPC formulation [13], [15] for optimization on state spaces with Lie group structure such as SE(3). The particular method we employ is Stagewise Newton (SN) [3] which is also closely related to Differential dynamic programming (DDP) [9].

Stagewise methods explicitly require the linearization of the cost and of the dynamics. On non-Euclidean manifolds X such linearization is achieved using *trivialized* variations and gradients [13]. In particular, for the class of systems considered in this work, the linearized discrete dynamics takes the form

$$dx_{k+1} = A_k dx_k + B_k \delta u_k, \tag{15}$$

with $dx_k \triangleq (dg_k, \delta r_k, \delta v_k)$ where $dg \triangleq (g^{-1}\delta g)^\vee \equiv ((R^T \delta R)^\vee, R^T \delta p) \in \mathbb{R}^6$ is the *trivialized variation* on SE(3). Similarly, the trivialized gradient $d_g L \in \mathbb{R}^6$ of a function $L : SE(3) \to \mathbb{R}$ is defined by

$$d_g L \triangleq \nabla_V \Big|_{V=0} L(g\,\mathrm{cay}(V)), \tag{16}$$

for some $V \in \mathbb{R}^6$. With these definitions, any standard iterative optimization method such as SQP, SN, or DDP can be applied by replacing the standard gradients $\nabla_g L$, $\nabla_g^2 L$ and variations δg, with the trivialized gradients $d_g L$, $d_g^2 L$ and trivialized variations dg.

Finite-difference linearization of the dynamics.: Since the resulting multi-body dynamics (5)–(7) has a complex nonlinear form, we employ finite differences for computing the Jacobians A_k and B_k. The default choice is central differences:

$$A_k^i \approx \frac{f(x_k + \epsilon e_i, u_k) - f(x_k - \epsilon e_i, u_k)}{2\epsilon},$$
$$B_k^j \approx \frac{f(x_k, u_k + \epsilon e_j) - f(x_k, u_k - \epsilon e_j)}{2\epsilon},$$

for $i = 1, \ldots, n+6$, and $j = 1, \ldots, m$, where each e_i is a standard basis unit vector with only one non-zero element at its i-th component. We again emphasize that the $+$ and $-$ signs above should be interpreted as the overloaded operators (13),(14) whenever elements of SE(3) are involved.

Closed-form cost gradients.: The trivialized gradient and Hessian of L_i are straightforward to compute and only require an extra term to account for the Cayley map. They are given by:

$$dL_k = \begin{bmatrix} \mathrm{dcay}^{-1}(-\Delta_k) & 0 \\ 0 & I \end{bmatrix}^T Q_k (x_k - x_d), \tag{17}$$

$$d^2 L_k \approx \begin{bmatrix} \mathrm{dcay}^{-1}(-\Delta_k) & 0 \\ 0 & I \end{bmatrix}^T Q_k \begin{bmatrix} \mathrm{dcay}^{-1}(-\Delta_k) & 0 \\ 0 & I \end{bmatrix}, \tag{18}$$

where $\Delta_k = \mathrm{cay}^{-1}(g_d^{-1} g_k)$ for each $k = 0, \ldots, N-1$. Equivalent expressions also hold for the gradients of L_N, with g_d replaced by g_f, and Q_k with Q_f. Note that for simplicity the Hessian was approximated by ignoring the second derivative of cay. The trivialized Cayley derivative denoted by $\mathrm{dcay}(V)$ for some $V = (\omega, \nu) \in \mathbb{R}^6$ is defined (see e.g. [15]) as

$$\mathrm{dcay}(V) = \begin{bmatrix} \frac{2}{4+\|\omega\|^2}(2I_3 + \widehat{\omega}) & 0_3 \\ \frac{1}{4+\|\omega\|^2}\widehat{\nu}(2I_3 + \widehat{\omega}) & I_3 + \frac{1}{4+\|\omega\|^2}(2\widehat{\omega} + \widehat{\omega}^2) \end{bmatrix}, \tag{19}$$

it is invertible and its inverse has the simple form

$$\mathrm{dcay}^{-1}(V) = \begin{bmatrix} I_3 - \frac{1}{2}\widehat{\omega} + \frac{1}{4}\omega\omega^T & 0_3 \\ -\frac{1}{2}\left(I_3 - \frac{1}{2}\widehat{\omega}\right)\widehat{\nu} & I_3 - \frac{1}{2}\widehat{\omega} \end{bmatrix}. \tag{20}$$

The linearized dynamics (15), cost gradients (17) and Hessians (18) can now be used as the ingredients of a standard Stagewise Newton algorithm [3] as detailed in [13].

IV. EXPERIMENTAL SETUP

In this section the hardware and software architecture required for running the manipulation experiments is outlined. Later, the experiments conducted and the NMPC based reference trajectory approach are described.

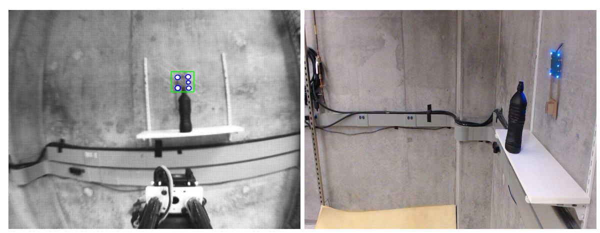

Fig. 2. a) Experimental arena showing the object to grab (black bottle) and led markers b) Led markers as seen from onboard camera

A. Hardware

Our prototype platform is based on the 3DRobotics quadcopter capable of lifting a payload of 1Kg, the Pixhawk autopilot board [21] for low-level attitude and thrust control, and the Odroid XU+E bare board computer for running various control algorithms. The NaturalPoint OptiTrack Motion Capture System has been used for estimating the attitude and position of the quadcopter in the world frame. A lightweight camera (PointGrey Firefly model) is installed onboard for providing the relative position of the target object in the reference frame of the quadcopter. A custom manufactured lightweight arm along with a 3D printed gripper has been installed on the quadcopter to grasp the object.

B. Experimental Scenario

The experimental scenario in Figure 2 shows the object of interest and LED markers which are used by the onboard camera to detect and track the object. The experiment requires the quadcopter to detect the marker, fly to a specified location in front of it, and retrieve the object placed on a stand. This is a challenging task, since the quadcopter has to extend the arm farther beyond it's envelope to grab the target object.

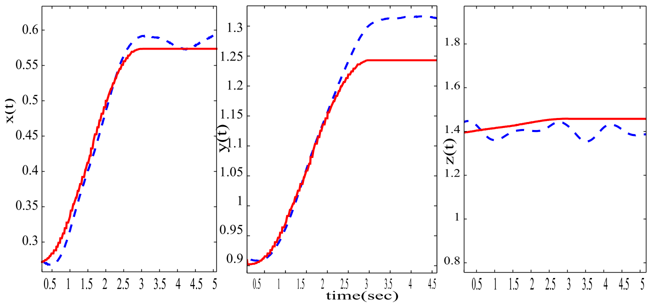

Fig. 3. Optimal control trajectory followed by the quadcopter. The solid line represents the desired trajectory and dashed line represents the actual trajectory followed

C. NMPC-based reference trajectory generation

In this approach, we compute a reference trajectory for the combined system of the quadcopter and arm using Stagewise Newton method described in § III-D. The optimal controller is used in open-loop to compute a reference trajectory for the quadcopter and the arm. The object position is frozen once the quadcopter starts executing the reference trajectory. The computed trajectory contains the full state (position, orientation and body fixed velocities) of the quadcopter, full state of the arm (joint positions and velocities) and the controls needed to achieve them. The desired position and velocity of the quadcopter is fed into the feedback linearization based controller and the desired joint angles and velocities for the arm are achieved through the PID controller on the servo motors. The quadcopter is able to track the trajectory closely as shown in Figure 3. The optimal control approach allows for faster actuation of the arm without losing accuracy. Since both the arm and the quadcopter execute their respective trajectories simultaneously, the object is retrieved in a shorter time interval.

V. Results and Discussion

The series of pictures in Figure 4 show the quadcopter flying to the marker and retrieving the object. The upper half represents the approach to grasp the object and the lower half shows the retrieval of the object (See attached video for more detail). The average time for retrieval using manual reference trajectories (Fig[5]) is around 15 seconds. Using NMPC-based trajectory tracking reduces the time taken to grasp the object to 5 seconds. This confirms that using optimal reference trajectories is superior to manual reference trajectories.

The tip positions for various starting positions converging to the target location (red cuboid) have been plotted in Figure 5. Since the non-optimal manual reference trajectory does not account for the dynamics of the arm, the tip positions are not smooth and take longer time to converge to the grasping location. On the other hand, the tip position for the case using optimal control (denoted by black dashed line) is smoother and still converges with the same accuracy. Following the optimal control based reference trajectory has enabled us to actuate both the quadcopter and the arm simultaneously to grasp the object quickly.

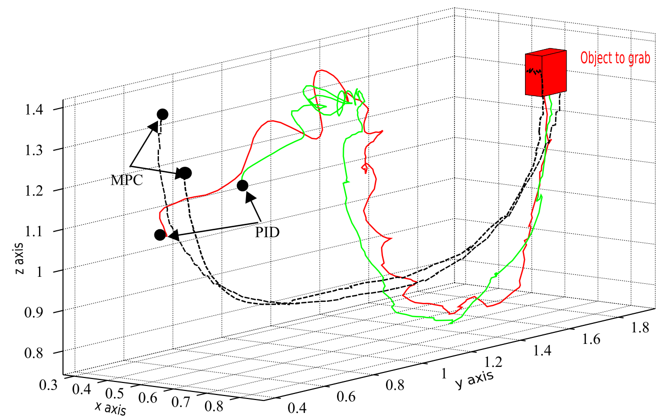

Fig. 5. Comparison of end-effector tip position trajectories for PID and MPC starting from from different initial positions marked with black dots.

There are many challenges faced during the manipulation tasks described above. The quadcopter position can be estimated based on the pose of the markers from the onboard camera. But this estimate turned out to be noisy and is dependent on the distance between the camera and markers. Thus a motion capture system is used to provide a reliable estimate of the quadcopter state. Since we are using a feedback linearization based controller for the quadcopter, we are not using optimal control to its full capacity of directly commanding the quadcopter motors. This explains the slight discrepancies between the actual and the desired quadcopter trajectories shown in Figure 3.

Future work will focus on improving the optimal control algorithm to achieve a real-time performance. By running a fast optimal control algorithm, we expect to bypass the feedback linearizing control and directly use the optimal controller to send the low-level motor commands. To enable markerless object grasping in an outdoor environment, we will rely on robust feature based object recognition and tracking. Combining real-time optimal control with arbitrary feature-based object recognition and tracking will be the subject of future work.

Acknowledgment

The authors take this opportunity to acknowledge Subhransu Mishra for the design and manufacture of the arm.

References

[1] Airobots. http://airobots.ing.unibo.it.
[2] ARCAS. http://www.arcas-project.eu.
[3] D. P. Bertsekas. *Nonlinear Programming, 2nd ed.* Athena Scientific, Belmont, MA, 2003.
[4] Francesco Bullo and Andrew Lewis. *Geometric Control of Mechanical Systems*. Springer, 2004.
[5] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2008.
[6] D. Fusata, G. Guglieri, and R. Celi. Flight dynamics of an articulated rotor helicopter with an external slung load. *Journal of the American Helicopter Society*, 46(1):3–14, 2001.

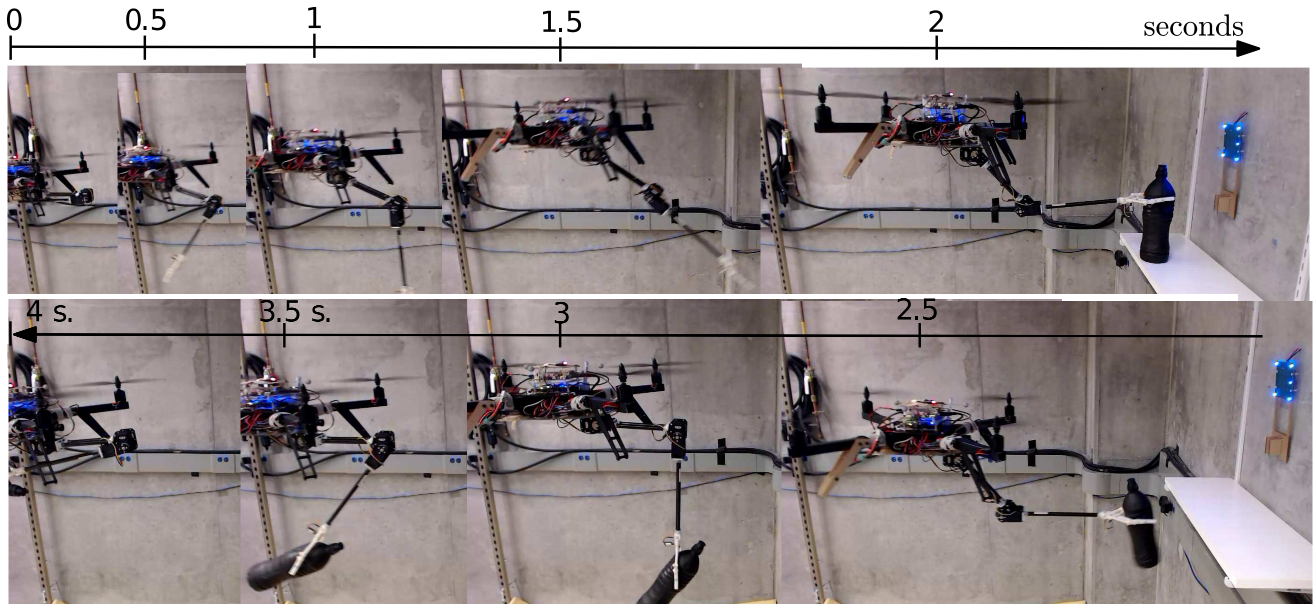

Fig. 4. Series of pictures showing approach, grasping, and retrieval of the object under NMPC-based reference trajectory setting

[7] M. Hehn and R. D'Andrea. A flying inverted pendulum. In *IEEE International Conference on Robotics and Automation (ICRA), 2011*, pages 763 –770, May 2011.

[8] G. Heredia, A.E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J.A. Acosta, and A. Ollero. Control of a multirotor outdoor aerial manipulator. In *International Conference on Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ*, pages 3417–3422, Sept 2014.

[9] David H. Jacobson and David Q. Mayne. *Differential dynamic programming*. Modern analytic and computational methods in science and mathematics. Elsevier, New York, 1970.

[10] Abhinandan Jain. *Robot and Multibody Dynamics: Analysis and Algorithms*. Springer, 2011.

[11] SH Jeong and S Jung. Bilateral teleoperation control of a quadrotor system with a haptic device: Experimental studies. 2014.

[12] A.E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano. Control of an aerial robot with multi-link arm for assembly tasks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4916–4921, May 2013.

[13] M. Kobilarov. Discrete optimal control on Lie groups and applications to robotic vehicles. In *IEEE International Conference on Robotics and Automation*, pages 5523–5529, 2014.

[14] M. Kobilarov. Nonlinear trajectory control of multi-body aerial manipulators. *Journal of Intelligent & Robotic Systems*, 73(1-4):679–692, 2014.

[15] M. Kobilarov and J.E. Marsden. Discrete geometric optimal control on Lie groups. *IEEE Transactions on Robotics,*, 27(4):641–655, August 2011.

[16] C. Korpela, M. Orsag, T. Danko, B. Kobe, C. McNeil, R. Pisch, and P. Oh. Flight stability in aerial redundant manipulators. In *IEEE International Conference on Robotics and Automation (ICRA), 2012*, pages 3529 –3530, may 2012.

[17] C.M. Korpela, T.W. Danko, and P.Y. Oh. Designing a system for mobile manipulation from an unmanned aerial vehicle. In *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, pages 109 –114, april 2011.

[18] Quentin J. Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. *Robotics: Science and Systems*, June 2011.

[19] Vincenzo Lippiello and Fabio Ruggiero. Exploiting redundancy in cartesian impedance control of uavs equipped with a robotic arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*, pages 3768–3773. IEEE, 2012.

[20] I. Maza, K. Kondak, M. Bernard, and A. Ollero. Multi-uav cooperation and control for load transportation and deployment. *Journal of Intelligent and Robotic Systems*, 57(1-4):417–449, 2010.

[21] Lorenz Meier. Pixhawk autopilot - px4 autopilot platform, 2014.

[22] A.Y. Mersha, S. Stramigioli, and R. Carloni. Bilateral teleoperation of underactuated unmanned aerial vehicles: The virtual slave concept. In *IEEE International Conference on Robotics and Automation(ICRA)*, pages 4614–4620, 2012.

[23] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC, 1994.

[24] M. Orsag, C. Korpela, M. Pekala, and P. Oh. Stability control in aerial manipulation. In *American Control Conference (ACC), 2013*, pages 5581–5586, 2013.

[25] Matko Orsag, Christopher Korpela, and Paul Oh. Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent and Robotic Systems*, 69(1-4):227–240, 2013.

[26] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar. Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control. *Autonomous Robots*, 33(1-2):129–142, 2012.

[27] P.E.I. Pounds, D.R. Bersak, and A.M. Dollar. The yale aerial manipulator: Grasping in flight. In *IEEE International Conference on Robotics and Automation (ICRA), 2011*, pages 2974 –2975, May 2011.

[28] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello. Impedance control of vtol uavs with a momentum-based external generalized forces estimator. In *IEEE International Conference on Robotics and Automation (ICRA), 2014*, pages 2093–2099, May 2014.

[29] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar. Toward image based visual servoing for aerial grasping and perching. In *IEEE International Conference on Robotics and Automation (ICRA), 2014*, pages 2113–2118, May 2014.

[30] Justin Thomas, Joe Polin, Koushil Sreenath, and Vijay Kumar. Avian-inspired grasping for quadrotor micro uavs. In *American Society of Mechanical Engineers(ASME) 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V06AT07A014–V06AT07A014. American Society of Mechanical Engineers, 2013.

[31] A. Torre, D. Mengoli, R. Naldi, F. Forte, A. Macchelli, and L. Marconi. A prototype of aerial manipulator. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2653–2654, 2012.