Linear Model Predictive Control for the Encirclement of a Target Using a Quadrotor Aircraft*

Mohamad Iskandarani1, Sidney N. Givigi2, Camille Alain Rabbath3, Alain Beaulieu4

Abstract—Encirclement is a task accomplished by an Unmanned Aerial Vehicle (UAV) in order to maintain awareness and containment of a given target. The aim of the UAV encircling this target is to maintain close proximity at all times. In this paper, the problem of maintaining a circular path around a target is considered and a Linear Model Predictive Control (LMPC) strategy is implemented on a Qball-X4 quadrotor aircraft in order to follow the circular path. A linear model for the two-dimensional movement of the UAV and its respective MP controller has been designed in MATLAB Simulink, simulated in a X-Plane/MATLAB interface and implemented on the actual vehicle in real-time. The results of the LMPC in simulation are compared to those found while implementing the algorithm on a physical platform. The contributions of this paper lay in the implementation of an autonomous Linear MP controller for the encirclement of a stationary target by a Qball-X4 quadrotor.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are becoming increasingly popular in civilian and military fields due to their diverse utility and low cost. Victim search and rescue along side reconnaissance operations are some applications where autonomous UAVs are utilized [1], thus minimizing human resources and risk of injury. Formation algorithms have been developed over many years for autonomous UAV use. [2] explores a swarming algorithm on a group of unmanned vehicles responsible for infrastructure protection. [3] implements a target capturing algorithm in order to follow a sinusoidally moving target. [4] implements a feedback control law in a cooperative hunting behaviour on a mobile robot troop enclosing a target.

Model Predictive Control (MPC) is a control strategy used widely for its robust handling of multiple manipulated variables controlled under strict constraints [5]. The control of autonomous Quadrotor aircraft using MPC has been explored by several authors over the years. [6] maintains a UPATcopter prototype unmanned quadrotor at desired height using MPC. [5] put in place a nonlinear Decentralized MPC (DMPC) policy for the encirclement of a stationary and moving target by a group of UAVs. [7] uses MPC to control a helicopter with three degrees of freedom (3DOF) set on a table using a support arm. In this paper, we have chosen LMPC versus other methods, such as Proportional-Integral-Derivative (PID) or state-feedback controllers, since it is ideal for accomplishing a complex task due to the use of cost function optimization. Thus we are able to use constraints and weights in order to deal with multiple state variables and stringent constraints. Moreover, PID for example is not useful for the objective of encirclement since it necessitates a reference path, while LMPC does not.

Experimentation in the field of MPC applied to UAVs can be seen in multiple papers [6], [7]. [6] maintains the UAV at a desired height which means that the controller’s main objective is to stabilize the vehicle at a given \( x \) and \( y \) coordinate. While the objective of [6] is to stabilize the UAV at a stationary position, our objective is to dynamically follow a circular path. In our case, we use MPC to produce the desired path which is then passed to low-level control, composed of PID controllers. Other papers, such as [7], take a different approach, where MPC is used to control a helicopter with 3DOF. The nonlinear 6th order plant is linearized around an equilibrium point and imported to the controller. In our case, the target vehicle is the Qball-X4, a six degrees of freedom quadrotor [8]. Again, unlike [7] the system is responsible for producing a desired path using MPC and not dealing with its low-level control. Both papers use a linear model for the real-time implementation of MPC.

Nonlinear MPC, as proposed in [5], cannot be implemented in real-time due to computational complexity. In order to accomplish dynamic encirclement with a real-time system, a linear model of the UAV’s movement in \( x \) and \( y \) has been deduced using a least squares algorithm for system identification, to replace this nonlinear model. Furthermore, a linearized plant is beneficial in reducing the computational time of any controller. Thus, the main contribution of this paper is the implementation of LMPC to the encirclement problem, applied to a Quadrotor aircraft which runs in real-time. The results presented show that the optimization problem can be efficiently solved and that the stability of the aircraft is guaranteed if the small-angle approximation is satisfied.

This paper is organized in the following manner. Section II covers the quadrotor’s dynamics. In Section III, we explore the problem of dynamic encirclement and propose the linear four-state model. Section IV proposes the control design necessary to accomplish the task of encirclement. X-Plane flight simulator is used in this paper to test the LMPC
strategy and Section V reports the results achieved. The experiment is discussed in detail in Section VI. Finally, Section VII concludes this work with discussion of the results and possible future work.

II. QUADROTOR DYNAMICS

The Qball-X4 Aircraft is a quadrotor designed by Quanser (Fig. 1). Similarly to [6], we assume that the structure of the UAV is rigid and symmetrical. The center of gravity is found in the middle of the design, positioned at equal distance away from the four motors. Since each motor rotates in the opposite direction from its axis counterpart, a motion neutral frame is created [9]. Fig. 2 represents the coordinate system used throughout this work. Notice that the pitch angle $\alpha$ is in the same direction as the $y$ axis while the roll angle $\phi$ is in the same direction as the $x$ axis. The height is perpendicular to both axes but is not part of the LMPC strategy.

[9] highlights an accurate description of the quadrotor’s movement and present the Qball-X4 dynamical equations. The position of the UAV is varied by slightly changing the pitch and roll angles in the appropriate direction. It is important here to highlight that the UAV dynamics are based on small angle approximation, a method used to linearize the quadrotor system of equations.

After linearization through small angle approximation, these equations are put into state-space form for PID design. The PID controllers are responsible for the low-level control of the nonlinear UAV dynamics. A system identification method, based on the least-squares algorithm, is used to approximate the low-level dynamics and their PID control into a second order transfer function. Section III explains this approximation in more detail. The present paper discusses a higher level of control using MPC, the interested reader may consult [9] for a more in depth discussion on PID controller design and its low-level control.

FIGURE 2. UAV motor rotation along side Cartesian coordinate system. The orange strip shows the tail of the aircraft.

This nonlinear system is not implementable with a MPC strategy in real-time on the Qball-X4 due to its computational complexity. As a solution to this problem, a least squares algorithm is used to approximate the low-level PID control along side the UAV nonlinearities to a second order system in $x$ and $y$. This method helps in providing an accurate plant model for the Qball-X4 to be controlled by the LMPC. First, data is collected from the low-level dynamics of the UAV by applying a step input. Second, this data is interpreted using the least-squares algorithm to match to a corresponding second order system. The following transfer functions represent the approximations of the UAV’s movement in the Cartesian coordinates:

\[
X = \frac{1.7455}{s^2 + 2.616s + 1.7116} X_d \quad (3)
\]

\[
Y = \frac{0.4973}{s^2 + 1.1384s + 0.4973} Y_d \quad (4)
\]

Where the inputs to equations (3) and (4), $X_d$ and $Y_d$, assuming that the height and yaw controllers have no influence on the lateral movement of the UAV. [5] give an accurate description of the UAV dynamics while encircling a target:

\[
\begin{bmatrix}
\dot{x}(t) \\
\dot{y}(t) \\
\dot{\theta}(t) \\
\dot{V}(t)
\end{bmatrix} = 
\begin{bmatrix}
x(t) \\
y(t) \\
\theta(t) \\
V(t)
\end{bmatrix}, \quad \begin{bmatrix}
\omega(t) \\
a(t)
\end{bmatrix}
\]

where the vector $\dot{X}(t)$ represents the states of the aircraft while $\dot{U}(t)$ represents the inputs to the system. $x$ and $y$ are the Cartesian coordinates; $\theta$ is the heading of the UAV; $V$ is its speed; $\omega$ is its angular velocity and $a$ is its acceleration. The state-equations that describe the motion of the aircraft are [5]:

\[
\dot{X}(t) = f(\bar{X}(t), \bar{U}(t)) = 
\begin{bmatrix}
V(t) \cos \theta(t) \\
V(t) \sin \theta(t) \\
\omega(t) \\
a(t)
\end{bmatrix}
\]

III. PROBLEM FORMULATION

We consider the problem of one UAV dynamically encircling a stationary target. The task is strictly two dimensional,
are the desired $x$ and $y$ respectively, sent to the Qball-X4. The LMPC is responsible for controlling these models by optimizing a specific cost function that compares the reference trajectory to the current behaviour and adjusts the control signal accordingly. We transform the transfer functions above to state-space:

$$
\dot{X}_1(t) = \begin{bmatrix} -2.6166 & -1.7116 \\ 1 & 0 \end{bmatrix} X_1(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} X_d \quad (5)
$$

$$
\dot{X}_2(t) = \begin{bmatrix} -1.384 & -0.4793 \\ 1 & 0 \end{bmatrix} X_2(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} Y_d \quad (6)
$$

Where $\dot{X}_1(t)$ is a state-vector made up of speed and position in the $x$ direction while $\dot{X}_2(t)$ is a state-vector made up of speed and position in the $y$ direction.

In order to optimize for the radius and angular velocity of the UAV, a change of variable is done where equations (5) and (6) are transformed using the following transformation:

$$
\begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ \frac{\omega \sin \theta}{r} & -\frac{\omega \cos \theta}{r} & \frac{\cos \theta}{r} & -\frac{\sin \theta}{r} \\ \frac{\omega \sin \theta}{r} & -\frac{\omega \cos \theta}{r} & \frac{\cos \theta}{r} & -\frac{\sin \theta}{r} \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (7)
$$

We then linearize the above equations around 8 different points on a circular path as seen in Fig. 3.

![Fig. 3. Linearization points around a circular path. The radius is 1 m.](image)

In this paper, we minimize the following cost function based on the weights given in Section IV:

$$
J(\bar{Z}, \Delta u) = \sum_{i=0}^{p-1} \Gamma^T Q \Gamma + \Delta u(k + i|k) \Gamma R \Delta u(k + i|k) \quad (8)
$$

The components of the cost function are:

$$
\Gamma = \bar{Z}(k + i + 1|k) - r(k + i + 1|k) \quad (9)
$$

where $p$ is the prediction horizon; $\bar{Z}(k + i + 1|k)$ are the states, radius and angular velocity, found after transformation of equations (5) and (6), predicted for time $k + i + 1$ at time $k$; $r(k + i + 1|k)$ is the reference sampled for time $k + i + 1$ at time $k$; $\Delta u(k + i|k)$ is the manipulated variables rate calculated for time $k + i$ at time $k$; $Q$ and $R \Delta u$ are positive semi-definite matrices that hold the weights for the output variables and the manipulated variables rate respectively. The same cost function is used for both states rate and manipulated variables rate.

Therefore, equation (8) is a quadratic system that uses the states of equation (5) and (6) for its optimization. This linear state-space is a valid approximation of its nonlinear counterpart in equation (2) when the assumption of small angle approximation is satisfied. This assumption is especially important for the low-level PID control. Considering the cost function in equation (8) and the linear models of equations (5) and (6), our model is computationally faster than the nonlinear one proposed in [5], because it is convex in nature. This increase in algorithmic efficiency makes it possible for the MPC strategy to be implemented on real-time platforms such as the Qball-X4.

IV. Control Design

The goal of this paper is to show that the control design presented in the previous section can be implemented in a real-time platform. As an important step towards that goal, we need to make sure that real-time constraints could be met. Real-time is defined here as the LMPC updating its cost function every 0.5 seconds. This assurance is provided through a simulation. However, we need to guarantee that the simulated environment provides a good approximation of dynamic effects of flight such as the ground effect, while taking into consideration some possible delays in measurements and actuation. The chosen platform is X-Plane [10].

X-Plane is used to simulate the quadrotor aircraft operation while the MPC strategy highlighted in equation (8) is run in real-time in MATLAB/Simulink. This flight simulator offers a testing platform capable of including one’s own UAV. A nonlinear UAV model based on the characteristics of the Qball-X4, was designed for testing of the LMPC strategy [11]. Using such simulators for validation of control strategies is essential to minimizing the risk of injury to personnel operating in the lab environment and reducing the chance of damages to equipment during testing.

Since the dynamics of the quadrotor and the low-level PID controllers are approximated to be the state-space of (5) and (6), and the cost function is quadratic (8), the optimization problem is convex and it may be solved efficiently with a quadratic programming algorithm [12]. The minimization of the cost function $J$ is at the heart of our LMPC strategy and if it may be done in real-time, the control strategy may be able to be ported to a real platform. Notice that the reference $r$ of equation (8) is the ideal radius and speed to be achieved by the UAV during dynamic encirclement. The radius of the circular path is set to 1m while the speed of the UAV is set to 0.1047 rad/s. This angular velocity is found due to the fact that Qball-X4 is encircling its target at a radius of 1m within a limited time frame of 60 seconds. The cost function optimization will try to ensure that the current radius and speed are matched to the desired reference. Fig. 4 shows how the LMPC takes into consideration the reference and
the cost function in the control of the Qball-X4 in dynamic encirclement.

The main controller is used both in X-Plane simulation and for the actual Qball-X4. X-Plane simulation helps us pinpoint errors early and adjust the controller before porting it to the actual UAV. It must be emphasized that the prediction and control horizons in each case are chosen so that the computational effort is minimized at all times and that the same code used in the simulation is used for the experiment.

The LMPC for the X-Plane simulation and implementation on Qball-X4 uses the following constraints on inputs:

\[
\begin{align*}
&X_d: -1 \text{m} \quad 1 \text{m} \\
&Y_d: -1 \text{m} \quad 1 \text{m}
\end{align*}
\]

where the constraints on the inputs are used in order to make sure that the desired position sent to he Qball-X4 is small. Since the UAV will not have to move long distances in very short periods of time, the low-level control will not force a big roll or pitch angle, thus satisfying the assumption of small angle approximation. Furthermore, the boundaries on UAV position in \(x\) and \(y\) are to make sure that the Qball-X4 does not exit the range of the Optitrack camera array. It must be noticed that this will necessarily create a small steady-state error as the vehicle always tries to avoid reaching the unit circle as well as avoiding overshooting.

A LMPC strategy was successfully implemented on the Qball-X4 for the encirclement of a stationary target, after successful testing in simulation. The positive definite matrices \(Q\) and \(R_{\Delta u}\) for the implementation of the LMPC strategy on the Qball-X4 and in simulation are:

\[
Q = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix};
R_{\Delta u} = \begin{bmatrix}
0.1 & 0 \\
0 & 0.1
\end{bmatrix}
\]

The prediction horizon used is eight while the control horizon is five.

V. SIMULATION RESULTS

A. Setup

As mentioned before, experimental UAV systems are expensive and can be dangerous if used incorrectly in any experimental setup. Many flight simulators are available for commercial use, such as Microsoft’s Flight Simulator and Flight Gear, but X-Plane was chosen because of its stability, high level of support and its realistic flight and atmospheric models [11].

Another advantage of X-Plane is that it also offers an easy-to-use program, called Plane-Maker, for creating one’s own UAV, and a nonlinear model for the Qball-X4 quadrotor could easily be created in Plane-Maker [11]. Furthermore, the quadrotor simulated in X-Plane can receive its control commands from a different program through a UDP port and the control algorithm could be run on MATLAB. Similar to the actual UAV, the simulated model also has a built-in low-level control, which means that the MPC system could be implemented as described in Section IV.

In order to make the simulation more realistic, the simulation setup is comprised of two processing units communicating through TCP/IP UDP protocol [11] so nondeterministic delays could be used to test the LMPC. One processing unit runs the X-Plane simulator while the other runs MATLAB/Simulink. A UDP rate of 50 packets per second is used, along side multiple I/O inputs and outputs such \(x, y, \text{height}\) and \(\text{velocities}\) are being sent and received through the X-Plane/MATLAB interface.

B. Results

The UAV in X-Plane operated for 87 seconds, which means that the LMPC needed to produce one pass of 60 seconds for the aircraft. The height is maintained at 0.5 m at all times. 20 seconds of stabilization time is allotted to the UAV in order for it to converge to the proper height before transitioning to Cartesian movement. In Fig. 5, we observe that the LMPC is able to produce a path that converges to the ideal circular pattern. Fig. 6 shows how the path of the UAV is actually similar to the path produced by the LMPC with minor noise due to the PID controllers and the MATLAB/X-Plane interface. In Fig. 7, the steady-state error is very small which anticipates promising results when implementing this controller in the more demanding experimental environment of the Qball-X4.

VI. EXPERIMENTAL RESULTS

A. Setup

Multiple systems work in concert in order to operate the Qball-X4 aircraft. The experimental setup is comprised of a processing unit running MATLAB/Simulink and QuaRC software, Optitrack camera feedback and the Qball-X4 itself [9]. Low-level PID controllers are responsible for the height, yaw and positional movements of the UAV. The PID and the MPC are designed in MATLAB/Simulink and compiled to C code on the processing unit by the QuaRC software. All controllers are then ported wirelessly to the
UA V and operate in real-time on the UA V’s GUMSTIX microcontroller without any dependence on the ground station other than the position feedback measurements, which is done through a set of sixteen OptiTrack V100:R2 cameras that pick up a unique pattern, called trackable, made of reflective balls glued to the UA V frame. This trackable can be seen in Fig. 1. The Optitrack system acts as a pseudo-GPS that localizes the vehicle at all times and sends this info to the processing unit. The testing arena is a 4x4m square which depends on the camera orientation and the corresponding calibration. Safety systems have been put in place to protect personnel and equipment [9].

B. Results

Similarly to the X-Plane simulation, the reference radius and speed are 1m and 0.1047 rad/s respectively. The testing runs for approximately 85 seconds. Again, 20 seconds are allotted to the UA V as stabilization time in order to improve the results. This stabilization time consists of the UA V taking off and hovering at the origin. Without the delay, the UA V would have lag time and a slight initial error before converging to the desired reference. The height of the UA V during implementation is maintained at 0.6 m.

Fig. 8 shows in red the dynamic encirclement of the Qball-X4 around a stationary target. The result converges to the ideal path and respects the constraints highlighted in Section IV. Steady-state error is present in Fig. 8 because of the prediction horizon chosen, however, this choice of horizon still maintains algorithmic efficiency. This is a necessary
compromise in order to allow real-time implementation of the control scheme. Furthermore, the noise from the PID controllers used and the OptiTrack sensors contribute to the steady-state error. The protective layer on the Qball, as seen in Fig. 1, influences flight dynamics slightly, which adds to the error. The control horizon remains five during all testing to ensure stability of the UAV. Fig. 9 shows the actual UAV position, broken down to its \( x \) and \( y \) components. We can clearly see that the UAV positions converges to the desired path while maintaining the proper height.

The LMPC cost, shown in Fig. 10, shows the performance of the LMPC in tracking the reference signal. The lower the MPC cost, the better the performance. This figure shows how the cost in \( x \) is decreasing with time, improving the controller performance as the testing continues. This means that with time, the lowering of the LMPC cost signals convergence to the ideal path and therefore better performance. The cost in \( y \) is small to begin with, since the UAV is at desired \( y \) at the start of testing. A small spike is present at 60 seconds due to small disturbance of the UAV during testing. This does not affect performance since the cost at 60 seconds is 0.036 which is very small.

Finally, Fig. 11 summarizes the objective of the LMP controller. We can clearly see that the radius is maintained very close to the ideal radius of 1m. Furthermore, the speed of the UAV is maintained at an average of 0.1139 rad/s, very close to the required 0.1047 rad/s. This is found by calculating the slopes of the angular position \( \theta \) found in Fig. 11. The results presented respect the constraints set in Section IV (between -1 m and 1 m) which means that the assumption of small angle approximation holds true. The angular velocity of the UAV is almost the required one of 0.1m/s. Thus, these two assumptions being satisfied, the state-space representation of equation (5) and (6) is a valid approximation of the nonlinear system of equation (2). The Qball-X4 successfully encircled the stationary target while maintaining the required radius and angular velocity.

VII. CONCLUSION

In this paper, we have shown that LMPC may be used for the dynamic encirclement of a stationary target using a quadrotor aircraft. The MPC has been successfully implemented in X-Plane simulation and on the Qball-X4. When implementing on real platforms, it is necessary to consider a stabilization time, where the UAV achieves a set height before following the desired path. The nonlinear model proposed by [5] has been replaced by the linear two-state model and used successfully in our experiments.

Furthermore, the results show that simulation can be used as a good predictor for the behaviour of the aircraft, but noise in actual sensors create another layer of complications that need to be dealt with. However, due to the implicit ability of MPC to deal with noise, the problem can be overcome and very good fitting with the desired behaviour may be achieved as depicted in Fig. 11. Also, the limitation of computation power in actual platforms, which can be seen as constraints on the use of MPC for fast real-time systems was overcome.

![Fig. 8. Path completed by the Qball-X4 quadrotor for the encirclement of a stationary target. The vehicle takes off from a stationary position at \((0m, 0m)\) (represented by the orange 'x'). The blue line consists of the ideal circular path, while the red line is the UAV's actual path.](image)

![Fig. 9. Height of Qball-X4 along side actual UAV x and y positions. The height during testing is maintained at 0.6m. The green line in the height plot shows the actual height of the UAV. The red lines in the position plots represent the x and y paths completed by the UAV. The blue lines represent the path produced as a result of the chosen radius and angular velocity.](image)
in this work. Our results point to the possibility of effectively using MPC for the high level control of UAVs in challenging environments.

Future work may include the implementation of LMPC on multi-UAV systems for the encirclement of stationary and moving targets as well as the implementation and switching of tactics in order to achieve more flexibility of behaviours in real life applications.

ACKNOWLEDGEMENTS

This research was supported in part by DRDC Valcartier and Director of Technical Airworthiness Engineering and Support 6.

REFERENCES


