AIAC-2019-137

# A Path Planning Algorithm for Loiter Maneuver: A Limit Cycle Approach

Metehan Yayla\* Middle East Technical University Ankara, Turkey Ismail O. Uzunlar<sup>†</sup> TOBB ETU Ankara, Turkey Ali Turker Kutay<sup>‡</sup> Middle East Technical University Ankara, Turkey

# ABSTRACT

In this paper, we propose a path planning algorithm for loitering maneuver using limit cycles. Specifically, actual position of the aircraft is plugged in to nonlinear dynamics of a stable limit cycle. Then, the desired velocity to converge the desired loitering path is generated by the vector field around the limit cycle. Relying on the Jordan-Schnflies Curve Theorem, the method for loitering path planning is extended to any simple closed curve other than exact circular paths. Knowing the transformation between two topological spaces, one can achieve the desired path tracking performance with a user-defined loop frequency and simple closed loitering path. In addition, robustness against noise is achieved by shaping the vector field around the limit cycle without effecting the control performance. Efficacy of the proposed path following algorithm is illustrated through numerical simulations with planar motion of a point mass.

Keywords: Path Planning, Limit Cycle, Loiter, Robust Tracking

#### INTRODUCTION

In the last decades, unmanned air vehicles are widely used for many applications such as surveillance, mapping, area monitoring and imaging. Clearly, those applications need an efficient and robust flying performance. It has great importance to guide the aircraft to position it on its way. Therefore, one of the most crucial part of UAV guidance becomes setting its path efficiently. For these purposes, plenty of path planning algorithms are introduced in the literature including but not limited to carrot following algorithm, vector field based path following, linear quadratic regulator (LQR) path following, pure pursuit and line of sight path following, and non-linear guidance law (NLGL) (see [Sujit et al., 2014], and references therein). Carrot chasing algorithm, which is easy to implement, uses time-varying virtual point with heading information to create desired path for a vehicle ([Lundgren, 2003]). Vector field path following algorithm computes vector field around the desired path. These vectors which are directed towards the path in the field can be used for heading command to unmanned vehicles ([Nelson et al., 2007]). Linear quadratic regulator (LQR) path following algorithm applies  $\mathcal{H}_2$  optimal and robust control theory to the path planning. It produces the minimal controller outputs to minimize cross track error and its derivative ([Ratnoo et al., 2011]). Pure pursuit guidance law

<sup>\*</sup>PhD Student in Department of Aerospace Engineering, METU, Email: myayla@metu.edu.tr

<sup>&</sup>lt;sup>†</sup>MSc Student in Electrical and Electronics Engineering, TOBB ETU, Email: i.uzunlar@etu.edu.tr

<sup>&</sup>lt;sup>‡</sup>Asst. Prof. in Department of Aerospace Engineering, METU, Email: kutay@metu.edu.tr

generates the desired heading through next way-point. Next, LOS guidance law adds an extra level of desired heading information using cross-track error. Since the method consists of a multilevel approach, tuning parameters double in that path planning algorithm ([Kothari and Postlethwaite, 2013]). Non-linear Guidance Path Following method uses pure pursuit guidance law and creates a lateral acceleration to track its desired path. This method comes with a benefit of tracking any circular path in given dynamic limits of a vehicle ([Park et al., 2007]). Table 1 summarizes these path following algorithms' advantages and disadvantages.

	Advantages	Disadvantages
		Low disturbance rejection perfor-
Carrot Chasing	Easy implementation, Simple and	mance, High velocity or non-
	easy to understand.	optimum algorithm parameters may
		cause oscillations.
	Lower Cross-track error, High dis-	Too many parameters to tune, Hard
Vector Field	turbance rejection performance	to implement in 3D environment.
LQR	Minimize control effort, Path infor-	
	mation does not need to known, Dis-	Solving algebraic Riccati equation is
	turbance rejection can be adjusted	computationally expensive.
	by weighting matrices	
PLOS	Less oscillations compared to carrot	Contains two sensitive gains for two
	chasing, Easy implementation	different guidance algorithm.
NLGL	Can be applied different paths,	
	Lower Cross-track error compared	Higher cross-track error compared
	to carrot chasing	to vector field.

Table 1: Comparison of Path Following Algorithms

Most common and widely used path for loitering is the exact circle. It is a path at which the aircraft follows a circle around an origin point with a dedicated radius. In Fig 1, path following algorithm performance parameters can be seen. There, d is the distance that defines *cross-track error* which is shortest distance between the vehicle and desired path,  $\Theta_c$  is the commanded heading for geometric path following algorithms.



Figure 1: Loiter path and its parameters

In this paper, we propose a path planning algorithm for loitering maneuvers for any simple closed paths using limit cycles. It is known that the frequency and amplitude of a limit cycle are fixed. Motivating from this point, we define a limit cycle as the loitering path where the loop frequency and radius of loitering maneuver are defined by the user. This means that a designer is able to adjust the limit cycle dynamics so that the desired time for a loop and desired radius of a circular path are fulfilled. In

this respect, proposed algorithm has similarities with guidance with Lyapunov vector fields. Authors ([Dale, 2003]) employs a Lyapunov vector field based algorithm for UAV flock coordination. However, in our paper, we generalize these vector field based algorithms to any simple closed curve by giving the necessary theoretical foundations. Hence, our paper significantly improves the outcomes of the work done by ([Dale, 2003]). Furthermore, one can achieve higher robustness than any methods against external disturbances by judiciously tuning the limit cycle parameters in the proposed algorithm.

The organization of the paper is as follows: The first section introduces the proposed path planning algorithm for loitering maneuver. In subsections, brief information on limit cycles are given, the path planning algorithm is proposed for a circular path, the generalization of proposed algorithm from a circular path to any simple closed curve using topological transformations is outlined. For the simulations, a controller is designed using Lyapunov's stability theory. Next, in the following sections, we list the future studies to be completed for final manuscript, give the initial results based on numerical simulations, and conclude the paper.

#### PROPOSED PATH PLANNING ARCHITECTURE FOR LOITER

#### Limit Cycle Review

Limit cycles are unique features of nonlinear systems, and they can be defined as the *isolated* periodic solutions *-with fixed amplitude and fixed period-* of nonlinear systems without external excitation. Consider the following nonlinear autonomous system:

$$\dot{x}_1 = \omega_o x_2 - \beta x_1 \left( x_1^2 + x_2^2 - r_o^2 \right) 
\dot{x}_2 = -\omega_o x_1 - \beta x_2 \left( x_1^2 + x_2^2 - r_o^2 \right)$$
(1)

With the state transformation  $x_1 = r \cos{(\theta)}$  and  $x_2 = r \sin{(\theta)}$ , nonlinear system in Eqn 1 can be expressed as

$$\dot{r} = -\beta r \left( r^2 - r_o^2 \right) \dot{\theta} = -\omega_o$$
<sup>(2)</sup>

Note that  $\dot{r} > 0$  if  $r < r_o$ , also  $\dot{r} < 0$  if  $r > r_o$ . Then, it can be concluded that the state will oscillate around the origin through a circle with a radius of  $r = r_o$ . Hence, such a limit cycle is called *stable limit cycle* since it converges to periodic solution when  $r \neq r_o$ . In addition, the period of oscillation is  $\frac{2\pi}{\omega_o}$  ([Slotine et al., 1991]). Visual interpretation of a stable limit cycle is given in Fig 2.



Figure 2: Stable Limit Cycle Behavior for  $r_o = 1, \omega_o = 1$ 

3 Ankara International Aerospace Conference

#### <u>Remarks</u>:

- 1. For the nonlinear system given in Eqn 1, there exists only one limit cycle.
- 2. Eqn 2 illustrates that the radius of limit cycle can be adjusted with  $r_o$ . In addition, the period of oscillation can be determined with  $\omega_o$ .
- 3. The design parameter  $\beta$  in Eqn 1 controls the damping of the limit cycle. From Eqn 2, we can see if the parameter  $\beta$  increases, the time constant becomes smaller which makes the system more aggressive against deviations from equilibrium.
- 4. One can reverse the direction of rotation by simply negating  $\omega_o$ ; i.e. with  $-\omega_o$ .

#### Loiter Maneuver Design Using Limit Cycle

In this section, we will explain the path following algorithm for loiter maneuver of a fixed-wing aircraft based on limit cycle approach that we explained in previous section. Given a target position, loiter maneuver will form a simple closed curve around the target position. For simplicity, we will assume the curve that the aircraft is desired to track during loitering maneuver is an exact circle. Later in the paper, we will remove this assumption.

Consider the limit cycle in Eqn 1. Corresponding vector field for design parameters  $r_o = \omega_o = \beta = 1$  is illustrated in Fig 2. At every time instant, actual position will be placed into this vector field so that the resultant velocity drives x-position and y-position to be on the limit cycle. Under any disturbances that result in deviations from ideal position, the vector field enforces the trajectory to come back to the limit cycle. That's why the proposed algorithm is robust against disturbances. Mathematically, desired velocities  $\dot{x}_d$  and  $\dot{y}_d$  are given as follows:

$$\dot{x}_{d} = \omega_{o}y - \beta x \left(x^{2} + y^{2} - r_{o}^{2}\right) 
\dot{y}_{d} = -\omega_{o}x - \beta y \left(x^{2} + y^{2} - r_{o}^{2}\right)$$
(3)

where x and y are actual positions,  $\dot{x}_d$  and  $\dot{y}_d$  are desired velocities that drive the system to the limit cycle. Additionally, we introduce a first order low-pass filter with cut-off frequency of  $\omega_f$  to estimate the derivative of desired velocity:

$$\dot{V}_{x_d} = \omega_f \left( \dot{x}_d - V_{x_d} \right)$$
  
 $\dot{V}_{y_d} = \omega_f \left( \dot{y}_d - V_{y_d} \right)$ 

For simplicity, let us assume planar motion of a point mass where corresponding dynamics is given by

$$\begin{aligned} \ddot{x} &= u_1 \\ \ddot{y} &= u_2 \end{aligned} \tag{4}$$

with x and y being positions on the horizontal plane,  $u_1$  and  $u_2$  being corresponding control inputs. Since the aim of this paper is to introduce a path planning algorithm, without any loss of generality, we can assume the existence of stabilizing controllers  $u_1$  and  $u_2$ . In the latter sections, we design a simple velocity hold controller, as well.

The block diagram representation of the proposed algorithm for the x-position is illustrated in Fig 3. <u>Remark</u>: The proposed algorithm offers variety of advantages against existing path planning algorithms for loiter maneuver. Specifically,

1. With the proposed architecture, we can physically set the desired time for a loop during a loiter maneuver by adjusting the design parameter,  $\omega_o$ . In aforementioned path planning algorithms, imposing such a performance parameter is not possible.



Figure 3: Block Diagram for Proposed Path Planning Architecture

- 2. Based on the structural limits, radius of loitering maneuver can easily be adjusted with design parameter  $r_o$ . In that respect, the design parameter  $r_o$  offers a significant flexibility during maneuver.
- 3. During a level cruising flight, when the loitering signal is commanded, design parameter  $\beta$  adjusts the rate to align with the loiter course. Another benefit that comes with  $\beta$  is robustness against disturbances such as wind. When the aircraft is deviated from loitering path, design parameter  $\beta$  determines the level of aggressiveness to come back to loitering course. In that respect, with higher  $\beta$ , higher robustness can be achieved against wind.
- 4. Proposed algorithm inherently includes position controller that outputs the desired velocity to be applied in sequential control architecture. The vector field attracts the aircraft to the limit cycle, and the convergence rate can be adjusted using  $\beta$ . Thus, a tunable position controller is included with  $\beta$  being the tuning design parameter.
- 5. Other than practical benefits, the proposed path planning algorithm also offers analytical advantages. Note that the proposed path planning algorithm is a dynamical system. Thus, it can directly be included in the stability analysis of the closed-loop system which is not the case in most of the existing path planning algorithms. In addition, during the loitering maneuver (i.e. during the motion on limit cycle), a quasi-linear system can be obtained for linear analysis tools using describing functions.

# Generalization to Arbitrary Loitering Path

In the previous part, we introduced the path planning algorithm for loitering maneuver in the form of exact circle. However, one may not desire to have a loitering maneuver so. In such a case, we can always find a transformation from a circle to any simple closed curve.

Jordan-Schnflies Curve Theorem ([Pandey, 2007]): For any simple closed curve in the plane, there is a homeomorphism of the plane which takes that curve into the standard circle.

Jordan-Schnflies's theorem states that there always exists a one-to-one mapping between points in two topological spaces that is continuous in both directions.

As a first step, we define the loitering path as a circle as we did in the previous section. Next, using a continuous transformation, which always exists if the desired path is simple closed curve, we can map the transform of the output of path planning algorithm to the new space.

<u>Example</u>: Consider a circle C that is defined as  $C = \{(x, y) : x^2 + y^2 = r^2, x, y \in \Re\}$ . The objective is to obtain a rounded square S to be the desired loitering path. Then, one-to-one and continuous transformation is given by

$$\bar{x} = \frac{1}{2}\sqrt{2 + x^2 - y^2 + 2x\sqrt{2}} - \frac{1}{2}\sqrt{2 + x^2 - y^2 - 2x\sqrt{2}}$$
$$\bar{y} = \frac{1}{2}\sqrt{2 - x^2 + y^2 + 2x\sqrt{2}} - \frac{1}{2}\sqrt{2 - x^2 + y^2 - 2x\sqrt{2}}$$

where the inverse transformation is given by

$$x = \bar{x}\sqrt{1 - \frac{1}{2}\bar{y}^2}, \qquad y = \bar{y}\sqrt{1 - \frac{1}{2}\bar{x}^2}$$

with  $(\bar{x}, \bar{y})$  being coordinates of rounded square. Note that the transformations in both directions are continuous and one-to-one as stated by Jordan-Schnflies's curve theorem. In order to have a rounded square, the radius of the circle should be r < 1. Figure 4 and Fig 5 illustrate this transformation.



Figure 4: Circles to be Transformed Figure 5: Transformed Rounded Squares

Overall block diagram for the proposed path planning algorithm is given in Fig 6.



Figure 6: Complete Block Diagram for Proposed Path Planning Architecture

#### Velocity Hold Controller Design

Having generated the desired velocity profile using limit cycles, we design a velocity hold controller using a Lyapunov design method. Recall the dynamics for the point mass:

$$\ddot{x} = u_1$$
$$\ddot{y} = u_2$$

For simplicity, let us assume circular loitering path. Then, our velocity tracking objective is to drive the system velocities to desired ones; i.e.  $\dot{x} \rightarrow \dot{x}_d$  and  $\dot{y} \rightarrow \dot{y}_d$ . Define velocity errors as  $e_x \triangleq \dot{x} - \dot{x}_d$  and  $e_y \triangleq \dot{y} - \dot{y}_d$ . Choose the Lyapunov candidate as

$$\mathcal{V} = \frac{1}{2} \left( e_x^2 + e_y^2 \right) > 0.$$

Then, time derivative of the Lyapunov candidate along the system trajectories becomes

$$\dot{\mathcal{V}} = e_x \left( \ddot{x} - \ddot{x}_d \right) + e_y \left( \ddot{y} - \ddot{y}_d \right) \\ = e_x \left( u_1 - \ddot{x}_d \right) + e_y \left( u_2 - \ddot{y}_d \right) \\ 6$$

Ankara International Aerospace Conference

Let the controllers  $u_1$  and  $u_2$  be designed as

$$u_{1} = \ddot{x}_{d} - k_{1}e_{x} = \ddot{x}_{d} - k_{1}\left(\dot{x} - \dot{x}_{d}\right)$$
$$u_{2} = \ddot{y}_{d} - k_{2}e_{y} = \ddot{y}_{d} - k_{2}\left(\dot{y} - \dot{y}_{d}\right)$$

where  $k_1, k_2 > 0$  are positive constants,  $\dot{x}, \dot{y}$  are available for feedback, and  $\dot{x}_d, \dot{y}_d, \ddot{x}_d, \ddot{y}_d$  are determined with limit cycle. Then, time derivative of Lyapunov function becomes

$$\dot{\mathcal{V}} = -k_1 e_x^2 - k_2 e_y^2 < 0$$

Hence, controllers  $u_1$  and  $u_2$  exponentially stabilizes the closed loop system. <u>Remarks.</u>

- 1. Note that desired velocities are obtained such that the vector field around the limit cycle enforces the trajectory to converge to the limit cycle. Hence, exponential tracking of desired velocities ensures the ideal tracking of the loitering path.
- 2. The proposed path planning algorithm can be considered analogous to *model following control* which is known to be a robust controller. Note that in model following control, a reference model generates the desired dynamics to be tracked by the system. Similarly, in the proposed path planning algorithm, limit cycle mimics the behavior of the reference model in model following control. Hence, we can conclude the proposed architecture reflects the robustness of model following control into path planning.

#### **FUTURE WORKS**

- Proposed algorithm will be implemented on 6-DOF simulation of a fixed-wing aircraft.
- Structural limitations during maneuver will be included in the path planning algorithm so that limit cycle design parameters  $r_o$  and  $\omega_o$  result in feasible and safe solution in every loitering scenario.

# CONCLUSIONS

This paper briefly introduces a new path planning algorithm for loitering maneuver using limit cycles. Assuming first the loitering path is an exact circle, outlines of the proposed method is sketched. Then, introduced path planning algorithm is extended to loitering path in the form of any simple closed curve. With the proposed method, not only satisfactory tracking is obtained, but also robustness against external disturbances is achieved. By tuning the design parameters of the limit cycle, time constant of inherent position controller can be adjusted. Thus, we can obtain higher bandwidth for that inherent controller which allows better disturbance rejection performance. Note that the robustness of the proposed algorithm is due to nature of the limit cycle, not because of the velocity controller. Hence, the control effort is not affected by increased inherent time constant. In every case, the velocity tracking is achieved successfully, but the desired velocity is changed by changing the autonomous behavior of the limit cycle which is fulfilled by  $\beta$ . Results in Figures 7-11 verifies this fact. Additionally, the control input remains similar in all cases without being affected by the convergence rate design parameter  $\beta$  as shown in the same figures.

# **APPENDIX - RESULTS**

The significantly large disturbance signal acting on the system in the interval of  $t \in [30, 50]$  is:

$$d_x = 0.3\sin(0.4t) - 0.3\sin(2t) + 1.2\cos(4t), \qquad t \in [30, 50]$$
  
$$d_y = -0.7\sin(1.4t) - 0.9\cos(2t) - 0.2\cos(0.4t)$$

such that

$$\ddot{x} = u_1 + d_x$$
$$\ddot{y} = u_2 + d_y$$

Simulation results are given in Figures 7-11.



Path Following

Control Inputs

Figure 7: Results without Disturbance  $(r_o = 0.9, \omega_o = 1, \beta = 3, k_1 = k_2 = 3)$ 



Figure 8: Results with Disturbance on  $t \in [30, 50]$   $(r_o = 0.9, \omega_o = 1, \beta = 1, k_1 = k_2 = 3)$ 



Figure 9: Results with Disturbance on  $t \in [30, 50]$   $(r_o = 0.9, \omega_o = 1, \beta = 3, k_1 = k_2 = 3)$ 



Path Following

Control Inputs

Figure 10: Results with Disturbance on  $t \in [30, 50]$   $(r_o = 0.9, \omega_o = 1, \beta = 7, k_1 = k_2 = 3)$ 



Figure 11: Results with Disturbance on  $t \in [30, 50]$   $(r_o = 0.9, \omega_o = 1, \beta = 10, k_1 = k_2 = 3)$ 

As seen from the results, increasing the design parameter  $\beta$ , which is equivalent to time constant of a position controller in a sequential loop closure architecture, results in better tracking error performance without affecting the control effort. Increasing  $\beta$  increases the bandwidth of corresponding position controller which improves the disturbance rejection performance.

# References

- Dale, L. (2003). Lyapunov vector fields for uav flock coordination. In presented at the 2nd AIAA Unmanned Unlimited Conf, and Workshop & Exhibit.
- Kothari, M. and Postlethwaite, I. (2013). A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees. Journal of Intelligent & Robotic Systems, 71(2):231-253.
- Lundgren, M. (2003). Path tracking and obstacle avoidance for a miniature robot. Umeå University, Umeå, Master Thesis.
- Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W. (2007). Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529.
- Pandey, R. K. (2007). Theory of Complex Variable, volume 4. Discovery Publishing House.
- Park, S., Deyst, J., and How, J. P. (2007). Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, 30(6):1718–1728.
- Ratnoo, A., Sujit, P., and Kothari, M. (2011). Adaptive optimal path following for high wind flights. *IFAC Proceedings Volumes*, 44(1):12985–12990.
- Slotine, J.-J. E., Li, W., et al. (1991). *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ.
- Sujit, P., Saripalli, S., and Sousa, J. B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *IEEE Control Systems Magazine*, 34(1):42–59.