2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Feedback Equivalence Between Curve & Straight Line Tracking for Unmanned Aerial Vehicles

George P. Moustris, Panagiotis Kyriakis and Costas S. Tzafestas

***Abstract*— We present a feedback equivalence transformation for the Dubins airplane model, which maps a monotone reference curve in space, to a straight line. This enables one to reduce path following and trajectory tracking for UAVs to the following and tracking of a straight line. We show that under this equivalence map, the system's equations remain invariant. Thus, in the new transformed space, one can use existing controllers, simplify them for straight lines, and apply them for tracking. We analyse the requirements of the reference curves which allow bijectivity and showcase simulation results using existing controllers.**

I. INTRODUCTION

Reference tracking, either in the form of path following or that of trajectory tracking, is a fundamental behaviour of UAVs, and any autonomous vehicle in general. Although several formulations for the reference exist e.g. waypoints [1], piece-wise linear paths [2], differentiable paths [3], one of the most common and simplest ones is the straight line [4]. Several controllers have been presented in the literature which tackle this problem (see [5] for a recent survey).

This work is the last in a string of papers presented in [6] [7] [8], regarding the reduction of path following and trajectory tracking to straight line tracking. We extend the transformation described in [6], called *Skew Rectification Map*, to allow for a reference path/curve in $3D$ space. Originally, this mapping was applied to the *planar* tracking task in mobile robots, hence the workspace was $2D$. The reference in the so-called *Physical Workspace*, was mapped to a straight line in the *Canonical Workspace*. When the map was applied to the system states and input, it was shown that the transformed dynamics in the Canonical space was *the same*. Effectively, this reduced reference tracking in the Physical space, to straight line tracking in the Canonical space, for the same system, although technical restrictions must apply for the reference path viz. it must be a monotone curve.

In this paper we present the direct extension of the SRM in $3D$ tracking for the dynamic equations of a UAV, along with the paths that allow for bijectivity. Again, a generalized notion of monotonicity is used, which is described in more detail in the sequence. The map enables to transfer the control of the tracking/following task of the UAV in the Canonical space, where the reference is a straight line, while in the actual Physical space it is tracking a (differentiable) curve.

G.P. Moustris, P. Kyriakis and C.S. Tzafestas are with the School of Electrical & Computer Engineering, National Technical University of Athens, Greece, emails: gmoustri@gmail.ntua.gr,pvkiriakis@gmail.com, ktzaf@cs.ntua.gr

II. PRELIMINARY CONCEPTS

In this work, we use the dynamic extension of the *Dubins Airplane* [9], which is a widely used kinematic model for analysis and control of unmanned aerial vehicles (UAVs). We treat the model as a moving vector in the *Physical Workspace* W_P. The position of the vector is denoted by $\boldsymbol{r}_p = (x_p, y_p, z_p)$, and is controlled by the input $\boldsymbol{u_p} = (u_{p,1}, u_{p,2}, u_{p,3})$ describing the model's ground speed v, heading (azimuthal) rate $\dot{\phi}_p$ and inclination (polar) rate $\dot{\theta}_p$ respectively (Fig. 1. The kinematic equations are thus,

$$\dot{\boldsymbol{q}}_p = f(\boldsymbol{q}_p)\boldsymbol{u}_p = \begin{pmatrix} sin\theta_p cos\phi_p \\ sin\theta_p sin\phi_p \\ cos\theta_p \\ 0 \\ 0 \end{pmatrix} u_{p,1} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_{p,2} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_{p,3} \quad (1)$$

where the state vector $\boldsymbol{q}_p = (x_p, y_p, z_p, \phi_p, \theta_p)$ describes the position and orientation of the UAV. The state space $\mathbb{M}_p = \mathbb{W}_p \times \mathbb{S}^1 \times [0, \pi]$ is called the *Physical State Space*. Equation (1) describes a control-linear system which must be augmented by bounding input constraints of the form,

$$\begin{aligned} u_{p,1} &\leq |u_{p,1}^{max}| \\ u_{p,2} &\leq |u_{p,2}^{max}| \\ u_{p,3} &\leq |u_{p,3}^{max}| \end{aligned} \quad (2)$$

to conform to more a realistic behaviour of actual physical systems.

Now consider a reference curve $r_{p,r} \in \mathbb{W}_p$ parametrized by time t, such that,

$$\boldsymbol{r_{p,r}}(t) = (x_{p,r}(t), y_{p,r}(t), z_{p,r}(t)), t \geq 0 \quad (3)$$

The curve satisfies the system equations Eq.(1). For the trajectory tracking task, one seeks to find a control law $\boldsymbol{u}(\boldsymbol{q_p}, \boldsymbol{r_{p,r}}, t)$, which asymptotically sends the system to the reference, i.e.,

$$\lim_{t\to\infty} \|\boldsymbol{r_p}(t) - \boldsymbol{r_{p,r}}(t)\| = 0 \quad (4)$$

For the path following task, we consider only the image of $r_{p,r}$. Specifically, let $\boldsymbol{I}_p = (x_{p,r}(w), y_{p,r}(w), z_{p,r}(w))$ be a path in $\mathbb{W}_p$ parametrized by some variable $w \in \mathbb{R}$. The problem is then to find a feedback law $\boldsymbol{u}(\boldsymbol{q_p}, \boldsymbol{I}_p, t)$, such that,

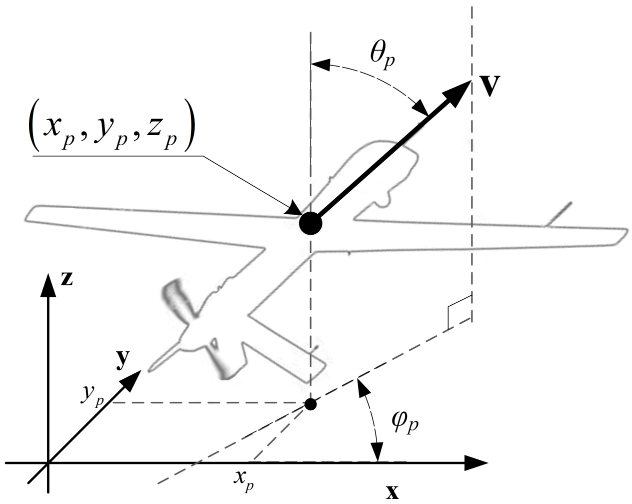

Fig. 1. Illustration of the UAV states in the Physical space

$$\lim_{t\to\infty} \|d(\boldsymbol{r}_p(t), \boldsymbol{I}_p)\| = 0 \tag{5}$$

where $d : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is a suitable distance function.

We introduce a new set of coordinates (x_c, y_c, z_c), parametrizing the so-called *Canonical Workspace* $\mathbb{W}_c$. Given a reference path $\boldsymbol{I}_p(w)$ in the Physical workspace, the two domains are connected by a diffeomorphism $\Phi : \mathbb{W}_c \to \mathbb{W}_p$, sending the canonical coordinates to the physical coordinates. However, Φ has the added property of mapping the reference physical path $\boldsymbol{I}_p(w)$ to a straight line in $\mathbb{W}_c$, i.e.

$$\boldsymbol{I}_c = (x_{c,r}, 0, 0) = \Phi^{-1}(\boldsymbol{I}_p) \tag{6}$$

By extending the mapping to include the states ϕ_p, θ_p, we introduce the transformation $\Psi : \mathbb{M}_c \to \mathbb{M}_p$, sending the *Canonical State Space* to the *Physical State Space*. Thus, if

$$\begin{aligned} \boldsymbol{q}_p &= (x_p, y_p, z_p, \phi_p, \theta_p) \\ \boldsymbol{q}_c &= (x_c, y_c, z_c, \phi_c, \theta_c) \end{aligned} \tag{7}$$

are the physical and canonical states, then,

$$\boldsymbol{q}_p = \Psi(\boldsymbol{q}_c) \tag{8}$$

sends the Canonical State Space to the Physical State Space. Mind that the reference in the Canonical Workspace is now a straight line. Consequently, if $\boldsymbol{u}_c(\boldsymbol{q}_c, \boldsymbol{I}_c, t)$ is a path following controller in the Canonical Workspace, it can be simplified to track only straight lines. However, the image of the UAV in $\mathbb{M}_c$ is mapped to the Physical State Space $\mathbb{M}_p$ through (8), and the they UAV tracks the reference curve in the Physical workspace.

III. TRANSFORMATION

To analyse the transformation, let $\boldsymbol{I}_p(x_c) = (x_{p,r}(x_c), y_{p,r}(x_c), z_{p,r}(x_c))$ be a reference path in the Physical workspace. We have identified the parametrizing variable w with x_c. Let $\boldsymbol{u}_y, \boldsymbol{u}_z \in \mathbb{W}_p$ be two *non-parallel* unit vectors. We define Φ as,

$$\Phi : \boldsymbol{r}_p = \boldsymbol{I}_p(x_c) + y_c\boldsymbol{u}_y + z_c\boldsymbol{u}_z \tag{9}$$

We see that the vectors $\boldsymbol{u}_y, \boldsymbol{u}_z$ produce a linear displacement of the reference path towards their direction; hence they will be called the *shifting vectors* of the transformation. The Jacobian matrix of Φ, can be easily calculated as,

$$\boldsymbol{J}_\phi = \begin{pmatrix} \boldsymbol{v}_{p,r} & \boldsymbol{u}_y & \boldsymbol{u}_z \end{pmatrix} \tag{10}$$

and its determinant is the triple product of the column vectors, i.e.,

$$|\boldsymbol{J}_\phi| = \boldsymbol{v}_{p,r} \cdot (\boldsymbol{u}_y \times \ \boldsymbol{u}_z) = \boldsymbol{v}_{p,r} \cdot \boldsymbol{n}_s \tag{11}$$

Here $\boldsymbol{v}_{p,r} = \boldsymbol{I}_p^{'}(x_c)$ is the tangent vector to the reference physical curve, essentially the physical *reference velocity*, and $\boldsymbol{n}_s = \boldsymbol{u}_y \times \ \boldsymbol{u}_z$ is the orthogonal vector to $\boldsymbol{u}_y$,$\boldsymbol{u}_z$. Apparently $\boldsymbol{n}_s$ defines a plane, called the *shifting plane* of the transformation. A necessary condition for the invertibility of the transformation, is that the Jacobian must not be singular; thus by (11), the shifting plane must not be tangent to the curve. However this is not enough. From (9) we see that for each $\boldsymbol{I}_p(x_c)$, the vectors $\boldsymbol{u}_y, \boldsymbol{u}_z$ span the shifting plane at the *specific* x_c. Thus, if the shifting plane cuts the curve at two points, then there are *two* x_c corresponding to that plane. Consequently, the map cannot be inverted at that points. In order for the Φ to be invertible, $\boldsymbol{I}_p(x_c)$ must be *monotone*. Here we use a generalized form of monotonicity [10] i.e. a curve is monotone if there is some coordinate system in which at least one of its coordinate functions is monotonic. For example, a helix is monotone w.r.t. its axis.

To invert Φ, we multiply (9) with $\boldsymbol{n}_s$, cancelling the shifting vectors. We then have,

$$\boldsymbol{n}_s\boldsymbol{r}_p = \boldsymbol{n}_s\boldsymbol{I}_p \tag{12}$$

We define new coordinates $\hat{x}_p = \boldsymbol{n}_s\boldsymbol{r}_p$ and $\hat{x}_{p,r}(x_c) = \boldsymbol{n}_s\boldsymbol{I}_p(x_c)$. We observe that $\partial\hat{x}_{p,r}/\partial x_c = \boldsymbol{v}_{p,r}\boldsymbol{n}_s = |\boldsymbol{J}_\phi| \neq 0$, thus by inverting we get,

$$x_c = \hat{x}_{p,r}^{-1}(\hat{x}_p) \tag{13}$$

Without loss of generality, we consider $\boldsymbol{u}_y, \boldsymbol{u}_z$ to be orthogonal. By dotting (9) with $\boldsymbol{u}_y$ we get,

$$y_c = \boldsymbol{u}_y\big(\boldsymbol{r}_p - \boldsymbol{I}_p(x_c)\big) = \boldsymbol{u}_y\Big(\boldsymbol{r}_p - \boldsymbol{I}_p\big(\hat{x}_{p,r}^{-1}(\hat{x}_p)\big)\Big) \tag{14}$$

In a similar fashion, we calculate the third canonical coordinate. The inverse transformation Φ^{-1} can finally be written as,

$$\Phi^{-1}: \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} \widehat{x}_{p,r}^{-1}(\widehat{x}_p) \\ \boldsymbol{u}_y\Big(\boldsymbol{r}_p - \boldsymbol{I}_p\big(\hat{x}_{p,r}^{-1}(\hat{x}_p)\big)\Big) \\ \boldsymbol{u}_z\Big(\boldsymbol{r}_p - \boldsymbol{I}_p\big(\hat{x}_{p,r}^{-1}(\hat{x}_p)\big)\Big) \end{pmatrix} \tag{15}$$

If we extend Φ to include the angles ϕ_p, θ_p, we can compute the map Ψ between the Canonical State Space and the Physical State Space. By applying Ψ to the dynamical system (1), we can get the new transformed system in $\mathbb{M}_c$. The two systems are then said to be *state space equivalent* [11], [12], [13]. To begin with, let

$$\boldsymbol{r}_p = \Phi(\boldsymbol{r}_c) \tag{16}$$

By differentiating (16) we have,

$$\boldsymbol{v}_p = \boldsymbol{J}_\phi \boldsymbol{v}_c \tag{17}$$

Here $\boldsymbol{J}_\phi$ is actually the *pushforward* of Φ in local coordinates, which sends the tangent vector $\boldsymbol{v}_c = \dot{\boldsymbol{r}}_c$, of the tangent space $T_{r_c}\mathbb{M}_c$ at $r_c \in \mathbb{M}_c$, to the tangent vector $\boldsymbol{v}_p = \dot{\boldsymbol{r}}_p$, of the tangent space $T_{r_p}\mathbb{M}_p$ at $r_p \in \mathbb{M}_p$. The tangent vectors can be written involving the unitary tangents $\boldsymbol{c}_p, \boldsymbol{c}_c$, expressed using the direction cosines, viz.,

$$\begin{aligned} \boldsymbol{v}_p = \|\boldsymbol{v}_p\|\boldsymbol{c}_p \quad &, \boldsymbol{c}_p = (cosa_p, cosb_p, cosc_c)^\intercal \\ \boldsymbol{v}_c = \|\boldsymbol{v}_c\|\boldsymbol{c}_c \quad &, \boldsymbol{c}_c = (cosa_c, cosb_c, cosc_c)^\intercal \end{aligned} \tag{18}$$

Henceforth, we will treat standard vectors as *column vectors*. By squaring (17), we get the length ratio of the transformation,

$$\gamma^2 = \frac{\|\boldsymbol{v}_p\|^2}{\|\boldsymbol{v}_c\|^2} = \|\boldsymbol{J}_\phi \boldsymbol{c}_c\|^2 = \boldsymbol{c}_c^\intercal \boldsymbol{g} \boldsymbol{c}_c \tag{19}$$

where $\boldsymbol{g} = \boldsymbol{J}_\phi^\intercal \boldsymbol{J}_\phi$ is the metric tensor in the physical space. Combining (17) and (19), we get the relation between the unit vectors,

$$\boldsymbol{c}_p = \frac{\boldsymbol{J}_\phi}{\gamma}\boldsymbol{c}_c = \frac{\boldsymbol{J}_\phi \boldsymbol{c}_c}{\|\boldsymbol{J}_\phi \boldsymbol{c}_c\|} \tag{20}$$

To calculate ϕ_p, θ_p, we use the relation between the spherical coordinates and the direction cosines given by,

$$\begin{aligned} sin\theta cos\phi &= cosa \\ sin\theta sin\phi &= cosb \\ cos\theta &= cosc \end{aligned} \tag{21}$$

The subscripts have been dropped since (21) holds for both the canonical and physical angles. Dividing the first two equations, we have for ϕ_p,

$$tan\phi_p = \frac{cosb_p}{cosa_p} = \frac{\boldsymbol{e}_2^\intercal \boldsymbol{c}_p}{\boldsymbol{e}_1^\intercal \boldsymbol{c}_p} = \frac{\boldsymbol{e}_2^\intercal \boldsymbol{J}_\phi \boldsymbol{c}_c}{\boldsymbol{e}_1^\intercal \boldsymbol{J}_\phi \boldsymbol{c}_c} \tag{22}$$

where $\boldsymbol{e}_i, i = 1, 2, 3$ are the basis vectors in $\mathbb{M}_p$. To prevent singularity in the division, we apply the following constraint on θ_p,

$$0 < \theta_p < \pi \tag{23}$$

meaning that the UAV cannot go straight up or straight down. For θ_p, in a similar manner we get,

$$cos\theta_p = cosc_p = \boldsymbol{e}_3^\intercal \boldsymbol{c}_p = \frac{\boldsymbol{e}_3^\intercal \boldsymbol{J}_\phi \boldsymbol{c}_c}{\gamma} \tag{24}$$

By inverting (22) and (24), the map between the Canonical and Physical state spaces can be written as,

$$\Psi: \begin{pmatrix} \boldsymbol{r}_p \\ \phi_p \\ \theta_p \end{pmatrix} = \begin{pmatrix} \Phi(\boldsymbol{r}_c) \\ tan^{-1}(\frac{\boldsymbol{e}_2^\intercal \boldsymbol{J}_\phi \boldsymbol{c}_c}{\boldsymbol{e}_1^\intercal \boldsymbol{J}_\phi \boldsymbol{c}_c}) \\ cos^{-1}(\frac{\boldsymbol{e}_3^\intercal \boldsymbol{J}_\phi \boldsymbol{c}_c}{\gamma}) \end{pmatrix} \tag{25}$$

To investigate the invertibility of Ψ, we compute its Jacobian, viz.,

$$\boldsymbol{J}_\Psi = \begin{pmatrix} \boldsymbol{J}_\phi & \boldsymbol{0}_{3\times 2} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix} \tag{26}$$

and by known properties of block matrices, the determinant is,

$$|\boldsymbol{J}_\Psi| = |\boldsymbol{J}_\phi||\boldsymbol{D}| \tag{27}$$

Since $|\boldsymbol{J}_\phi| \neq 0$, the invertibility of of Ψ depends on the non-singularity of $\boldsymbol{D}$, which is described by,

$$\boldsymbol{D} = \begin{pmatrix} \frac{\partial\phi_p}{\partial\phi_c} & \frac{\partial\phi_p}{\partial\theta_c} \\ \frac{\partial\theta_p}{\partial\phi_c} & \frac{\partial\theta_p}{\partial\theta_c} \end{pmatrix} \tag{28}$$

This matrix is essentially the Jacobian matrix of the following system,

$$\begin{aligned} \phi_p &= \phi_p(\phi_c, \theta_c) \\ \theta_p &= \theta_p(\phi_c, \theta_c) \end{aligned} \tag{29}$$

which by (20), has a solution. It can be proven that, (the proof is rather extensive and shall be presented elsewhere)

$$|\boldsymbol{D}| = \frac{|\boldsymbol{J}_\Phi|}{\gamma^2}(\frac{\boldsymbol{c}_c^\intercal(\boldsymbol{I}_3 - \boldsymbol{e}_{33})\boldsymbol{c}_c}{\boldsymbol{c}_c^\intercal \boldsymbol{J}_\Phi^\intercal(\boldsymbol{I}_3 - \boldsymbol{e}_{33})\boldsymbol{J}_\phi \boldsymbol{c}_c})^{1/2} = \frac{|\boldsymbol{J}_\Phi|}{\gamma^3}\frac{sin\theta_c}{sin\theta_p} \tag{30}$$

where $\boldsymbol{e}_{ij} = \boldsymbol{e}_i \boldsymbol{e}_j^\intercal$ denote the basis tensors of the 2^{nd} order Cartesian tensor. In order to avoid singularity, the following constraint is also implemented,

$$0 < \theta_c < \pi \tag{31}$$

This is juxtaposed to the constraint in (23), and forbids the UAV to go straight up/down in the Canonical workspace.

IV. FEEDBACK EQUIVALENCE

Since Ψ acts on the state variables and transforms them, it also affects the kinematic equations of the UAV in the Physical State space, i.e. Eq. (1). We observe that,

$$\left.\begin{aligned}\dot{\boldsymbol{q}}_p &= f(\boldsymbol{q}_p)\boldsymbol{u}_p \\ \dot{\boldsymbol{q}}_p &= J_\Psi \dot{\boldsymbol{q}}_c\end{aligned}\right\} \Rightarrow \dot{\boldsymbol{q}}_c = \boldsymbol{J}_\Psi^{-1} f(\Psi(\boldsymbol{q}_c))\boldsymbol{u}_p \tag{32}$$

Using block-wise inversion for J_Ψ, its inverse is,

$$\boldsymbol{J}_\Psi^{-1} = \begin{pmatrix} \boldsymbol{J}_\Phi^{-1} & \boldsymbol{0}_{3\times 2} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{J}_\Phi^{-1} & \boldsymbol{D}^{-1} \end{pmatrix} \tag{33}$$

Now, combing (1) and (20), we get the middle term,

$$f(\boldsymbol{q}_p) = f(\Psi(\boldsymbol{q}_c)) = \begin{pmatrix} \frac{1}{\gamma}\boldsymbol{J}_\Phi \boldsymbol{c}_c & \boldsymbol{0}_{3\times 2} \\ \boldsymbol{0}_{2\times 1} & \boldsymbol{I}_2 \end{pmatrix} \tag{34}$$

and by multiplying (34) with (33),

$$\dot{\boldsymbol{q}}_c = \begin{pmatrix} \frac{1}{\gamma}\boldsymbol{c}_c & \boldsymbol{0}_{3\times 2} \\ -\frac{1}{\gamma}\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{c}_c & \boldsymbol{D}^{-1} \end{pmatrix} \boldsymbol{u}_p \tag{35}$$

Equation (35) expresses the UAV kinematics in the Canonical Space. Notice that the input is still in the Physical space. Since it expresses the physical velocity and the heading/inclination derivatives, they can also be transformed into the Canonical Space. To that end, we note that $\boldsymbol{u}_p = (\|\boldsymbol{v}_p\|, \dot{\phi}_p, \dot{\theta}_p)$. Using (19) we get $\boldsymbol{u}_p = (\gamma\|\boldsymbol{v}_c\|, \dot{\phi}_p, \dot{\theta}_p)$. Now, taking the lower part of (26) and using (32), we get,

$$\begin{aligned}\begin{pmatrix}\dot{\phi}_p \\ \dot{\theta}_p\end{pmatrix} &= \begin{pmatrix}\boldsymbol{C} & \boldsymbol{D}\end{pmatrix}\dot{\boldsymbol{q}}_c = \begin{pmatrix}\boldsymbol{C} & \boldsymbol{D}\end{pmatrix}\begin{pmatrix}\boldsymbol{v}_c \\ \dot{\phi}_c \\ \dot{\theta}_c\end{pmatrix} = \boldsymbol{C}\boldsymbol{v}_c + \boldsymbol{D}\begin{pmatrix}\dot{\phi}_c \\ \dot{\theta}_c\end{pmatrix} \\ &= \|\boldsymbol{v}_c\|\boldsymbol{C}\boldsymbol{c}_c + \boldsymbol{D}\begin{pmatrix}\dot{\phi}_c \\ \dot{\theta}_c\end{pmatrix} = \begin{pmatrix}\boldsymbol{C}\boldsymbol{c}_c & \boldsymbol{D}\end{pmatrix}\begin{pmatrix}\|\boldsymbol{v}_c\| \\ \dot{\phi}_c \\ \dot{\theta}_c\end{pmatrix}\end{aligned} \tag{36}$$

Denoting $\boldsymbol{u}_c = (\|\boldsymbol{v}_c\|, \dot{\phi}_c, \dot{\theta}_c) \equiv (u_{c,1}, u_{c,2}, u_{c,3})$ as the *Canonical Input*, and using (18),

$$\boldsymbol{u}_p = \boldsymbol{M}(\boldsymbol{q}_c)\boldsymbol{u}_c = \begin{pmatrix}\gamma & \boldsymbol{0}_{1\times 2} \\ \boldsymbol{C}\boldsymbol{c}_c & \boldsymbol{D}\end{pmatrix}\boldsymbol{u}_c \tag{37}$$

where $M : \mathbb{U}_c \to \mathbb{U}_p$ is the map which sends the the *Canonical Input Space* to the *Physical Input Space*. Now, applying (37) to (35), we get the *canonical kinematics* of the UAV, i.e.,

$$\dot{\boldsymbol{q}}_c = \begin{pmatrix} sin\theta_c cos\phi_c \\ sin\theta_c sin\phi_c \\ cos\theta_c \\ 0 \\ 0 \end{pmatrix} u_{c,1} + \begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} u_{c,2} + \begin{pmatrix}0\\0\\0\\0\\1\end{pmatrix} u_{c,3} \tag{38}$$

We notice that (38) is identical to (1). The two systems are related by a state and input transformation and are called *feedback equivalent* [11] [12]. We denote this extended transformation as $\Omega = (\Psi, M) : (\mathbb{M}_c \times \mathbb{U}_c) \to (\mathbb{M}_p \times \mathbb{U}_p)$. The two systems present *form invariance* under Ω. This allows to reduce path following and trajectory tracking into line following/tracking for the exact same system. As a consequence, a general following/tracking controller can be simplified to consider only straight lines. Conversely, one can develop controllers for straight line following/tracking, apply them in the canonical domain and elevate them into (monotone) curve controllers in the physical domain. Analytically, if $\boldsymbol{I}_p$ is a reference path in the physical workspace, $\boldsymbol{I}_c = \Phi^{-1}(\boldsymbol{I}_p)$ its projection into the canonical workspace, $\boldsymbol{q}_p = \Psi(\boldsymbol{q}_c)$ the physical/canonical states respectively, and $\boldsymbol{u}_c(\boldsymbol{q}_c, \boldsymbol{I}_c)$ a controller of the canonical system, then its physical counterpart is,

$$\boldsymbol{u}_p = \boldsymbol{M}(\boldsymbol{q}_c)\boldsymbol{u}_c(\Psi^{-1}(\boldsymbol{q}_p), \Phi^{-1}(\boldsymbol{I}_p)) \tag{39}$$

The block diagram of the control loop can be seen in Fig.2.

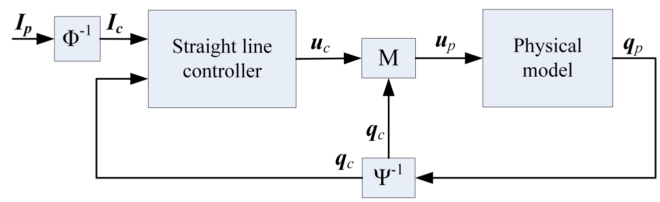

Fig. 2. Block diagram of the control loop

V. SIMULATION EXPERIMENTS

We have implemented our proposed control scheme in MATLAB, simulating the path following task for a helical path. A *straight line* controller, found at [14] , has been utilized. The path is described by,

$$\boldsymbol{I}_p(x_c) = (10e^{0.1x_c}, sinx_c, cosx_c) \tag{40}$$

The shifting vectors are $\boldsymbol{u}_y = \boldsymbol{e}_2, \boldsymbol{u}_z = \boldsymbol{e}_3$ respectively. Applying (9) we have,

$$\Phi : \boldsymbol{r}_p = (10e^{0.1x_c}, sinx_c + y_c, cosx_c + z_c) \tag{41}$$

and for the inverse,

$$\Phi^{-1} : \boldsymbol{r}_c = (10log(0.1x_p), y_p - sinx_c, z_p - cosx_c) \tag{42}$$

where $x_c = 10log(0.1x_p)$ is substituted into the trigonometric functions in (42). The control is applied to a smaller subsystem as a showcase. The subsystem comprises the first three states of the UAV, where the input is the polar and yaw angles, viz.

$$\boldsymbol{v_p} = \dot{\boldsymbol{q}}_p = \begin{pmatrix} sin\theta_p cos\phi_p \\ sin\theta_p sin\phi_p \\ cos\theta_p \end{pmatrix} \tag{43}$$

Using the line controller in [14], the desired UAV velocity vector is given by,

$$\boldsymbol{v_p} = K1(\boldsymbol{r}_c - (x_c + 0.2, 0, 0)) + K2(1, 0, 0); \tag{44}$$

where $K1 = -2$ and $K2 = 1$. The controlled angles are then calculated as,

$$\begin{aligned} \phi_p &= atan2(\boldsymbol{v_{p,y}}, \boldsymbol{v_{p,x}}) \\ \theta_p &= acos(\boldsymbol{v_{p,z}}/\|\boldsymbol{v_p}\|)) \end{aligned} \tag{45}$$

The simulation results are presented in Figs 3 and 4 .

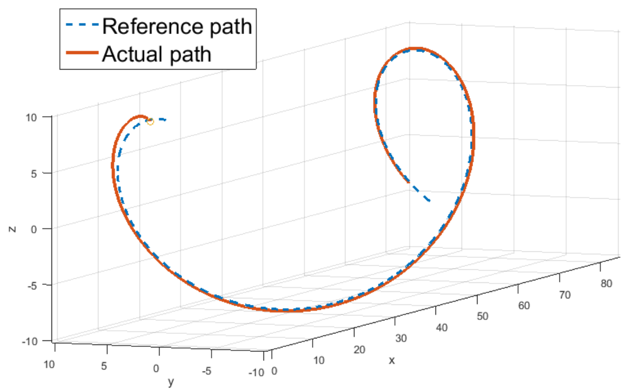

Fig. 3. Simulation trajectories in the physical space, for a helical reference path.

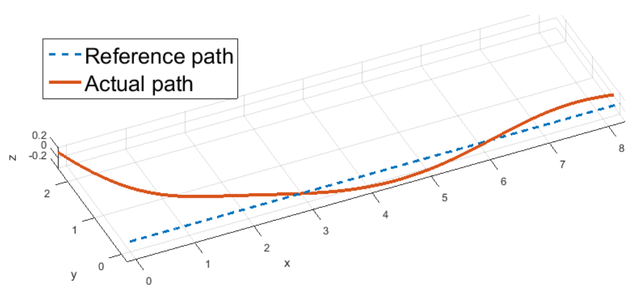

Fig. 4. Simulation trajectories in the canonical space. The helical reference path in the physical space is projected to the canonical $x - axis$.

Notice that the control takes place in the canonical space, where the helical path is now a straight line (the $x - axis$). The controls are computed in this domain, and are projected back to the physical space through (39). This signal is then fed to the actual UAV in the physical domain.

VI. CONCLUSIONS

We have presented a feedback equivalence transformation which keeps the kinematic equations of the extended Dubins Airplane invariant, while mapping the reference path in the original Physical space to a straight line in the canonical domain. This enables the reduction of a class of path tracking problems, namely the ones where the paths are monotone, to the simple task of tracking a straight line. Proof-of-concept simulations have shown the validity of our approach. Future directions include the investigation of higher dimensional references e.g. surfaces for terrain tracking, as well as the identification of more models which present this invariance under our transformation.

VII. ACKNOWLEDGMENT

This research work has been partially supported by the EU-funded Project BabyRobot (H2020-ICT-24-2015, grant agreement no. 687831).

REFERENCES

[1] J. Osborne and R. Rysdyk, "Waypoint guidance for small uavs in wind," *AIAA Infotech@ Aerospace*, vol. 193, no. 1-4, pp. 1–12, 2005.

[2] S. Hota and D. Ghose, *Rectilinear Path Following in 3D Space*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 210–217.

[3] S. Zhao, X. Wang, D. Zhang, and L. Shen, "Curved path following control for fixed-wing unmanned aerial vehicles with control constraint," *Journal of Intelligent & Robotic Systems*, Jan 2017.

[4] R. W. Beard, J. Ferrin, and J. Humpherys, "Fixed wing uav path following in wind with input constraints," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2103–2117, Nov 2014.

[5] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, Feb 2014.

[6] G. P. Moustris and C. S. Tzafestas, "Simplifying mobile robot tracking control through feedback equivalence," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, June 2016, pp. 919–924.

[7] G. Moustris and S. G. Tzafestas, "Reducing a class of polygonal path tracking to straight line tracking via nonlinear strip-wise affine transformation," *Mathematics and Computers in Simulation*, vol. 79, no. 2, pp. 133 – 148, 2008.

[8] G. Moustris, K. Deliparaschos, and S. Tzafestas, "Feedback equivalence and control of mobile robots through a scalable fpga architecture," in *Recent Advances in Mobile Robotics*. InTech, 2011.

[9] H. Chitsaz and S. LaValle, *Time-optimal paths for a Dubins airplane*, Dec 2007.

[10] F. Preparata and K. Supowit, "Testing a simple polygon for monotonicity," *Information Processing Letters*, vol. 12, no. 4, pp. 161 – 164, 1981.

[11] R. B. Gardner and W. F. Shadwick, "Feedback equivalence of control systems," *Systems & Control Letters*, vol. 8, no. 5, pp. 463–465, May 1987.

[12] ——, "Feedback equivalence for general control systems," *Systems & Control Letters*, vol. 15, no. 1, pp. 15–23, July 1990.

[13] W. Respondek, *Geometry of static and dynamic feedback*, ser. Lectures given at the Summer Schools on Mathematical Control Theory Trieste, Italy, September 2001 and Bedlewo-Warsaw, Poland, September 2002, 2001.

[14] M. Owen, R. W. Beard, and T. W. McLain, *Implementing Dubins Airplane Paths on Fixed-Wing UAVs**. Dordrecht: Springer Netherlands, 2015, pp. 1677–1701.