Chapter 2
Theoretical Background

2.1 Modeling

The major two methodologies normally employed to compute the dynamic model of a mechanical system are the *Lagrangian* and the *Newton-Euler* formalisms. The two methods are equivalent and obviously lead to the same outcome, but the practical procedure is quite different. Furthermore, they could give different insights about the system and its properties.

The first, the Lagrangian formalism, is a systematic and elegant approach to derive the analytical dynamic equations describing the model of the system, independently from the reference frame. In particular, choosing a proper set of generalized coordinates and simply computing the kinematics and potential energies, the Lagrangian formalism allows to compute the dynamic equations, naturally including system constraints and reaction forces. Nevertheless, notice that it becomes unpractical for complex system with many degrees of freedom.

On the other hand, the Newton-Euler method is an efficient and recursive method, especially suited for manipulators with an open kinematic chain and complex systems. It treats each joint of a robot as an independent part, and then computes the coupling between them using the so called *forward-backward* recursive algorithm. However, a particular attention has to be taken for constrained systems. Indeed one has to explicitly consider reaction forces related to system constraints.

In the following we recall the basis of the two methods, mostly from a practical point of view, and the particular remarks and considerations made during this thesis. Fore more details we refer the interested reader to [2–5].

2.1.1 Lagrange Formalism

The fist step consists on choosing a set of independent coordinates $\mathbf{q} = [q_1 \ \ldots \ q_n]^\top \in \mathbb{R}^n$, called *generalized coordinates*. Those fully describe the configuration of the

M. Tognon and A. Franchi, *Theory and Applications for Control of Aerial Robots in Physical Interaction Through Tethers*, Springer Tracts in Advanced Robotics 140,
https://doi.org/10.1007/978-3-030-48659-4_2

system and its $n \in \mathbb{N}_{>0}$ degrees of freedom. Accordingly to the chosen generalized coordinates, we can then compute the *generalized forces* acting on the system. Consider a set of forces $\mathbf{f} = [\mathbf{f}_1^\top \ \dots \ \mathbf{f}_m^\top]^\top \in \mathbb{R}^{3m}$, where the generic force $\mathbf{f}_i \in \mathbb{R}^3$ is applied on the system at point $\mathbf{r}_i \in \mathbb{R}^3$, with $i = 1, \dots, m$ and $m \in \mathbb{N}_{\geq 0}$. We can then compute the generalized force $\xi_j(\mathbf{f}, \mathbf{q}) \in \mathbb{R}$ w.r.t. the jth generalized coordinate q_j as:

$$\xi_j(\mathbf{f}, \mathbf{q}) = \sum_{i=1}^{m} \mathbf{f}_i^\top \frac{\partial \mathbf{r}_i}{\partial q_j}, \quad j = 1, \dots, n. \tag{2.1}$$

We can now define the *Lagrangian* function, $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$, equal to the difference of total kinetic energy, $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$, and potential energy, $\mathcal{U}(\mathbf{q}, \dot{\mathbf{q}})$, i.e., $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}, \dot{\mathbf{q}})$. Finally, the equation of motions of the system are given by the following Lagrange equations:

$$\frac{d}{dt}\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_j} - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_j} = \xi_j(\mathbf{f}, \mathbf{q}), \quad j = 1, \dots, n. \tag{2.2}$$

For the type of mechanical systems under exam, the potential energy usually corresponds to the sole gravitational potential energy, and the kinematic energy can be computed as a quadratic form, $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^\top \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix of the system. The equations of motion in (2.2) can be then rewritten in the more usual form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\xi}(\mathbf{f}, \mathbf{q}), \tag{2.3}$$

where $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ contains the centrifugal and Coriolis terms, while $\mathbf{g}(\mathbf{q})$ contains the gravitational terms, and $\boldsymbol{\xi}(\mathbf{f}, \mathbf{q}) = [\xi_1(\mathbf{f}, \mathbf{q}) \ \dots \ \xi_n(\mathbf{f}, \mathbf{q})]^\top \in \mathbb{R}^n$.

Remark The *inverse dynamics* problem consists into computing the generalized forces $\boldsymbol{\xi}(\mathbf{f}, \mathbf{q})$ given a certain motion expressed in terms of $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}$ and $\mathbf{q}$. Considering the generalized forces as inputs and the motion as output, this problem is equivalent to the control problem, i.e., compute certain inputs to obtain certain desired outputs. Given the analytic expression of the dynamic model (2.3), the Lagrangian formalism is often used to solve the *inverse dynamics* problem, and thus the control problem. □

2.1.2 Newton-Euler Formalism

The Newton-Euler formalism is based on two recursive steps: *(i)* forward recursion, and *(ii)* backward recursion.

The first *forward recursion* is done to propagate the links velocities and accelerations from the first link to the final one. The translational and rotational velocities and acceleration of the ith link are computed based on the one of the previous $(i-1)$th link and on the ith joint, according to its type (either prismatic or revolute). The method is repeated for all the links starting from the base link, of which we know velocities and accelerations, up to the last one.

The second *backward recursion* propagates forces and moments from the last link to the first one. Knowing the force and moment applied to the $(i+1)$th link, we compute the one applied to the ith link resolving the Newton-Euler equations. Defining $\mathbf{f}_i \in \mathbb{R}^3$ and $\boldsymbol{\tau}_i \in \mathbb{R}^3$ the force and moment acting on the ith link at position $\mathbf{r}_i \in \mathbb{R}^3$ (analogously for the $(i+1)$th link), we have to solve the balance equations of forces and moments at the i-link w.r.t. the ith link frame:

$$\mathbf{f}_i = \mathbf{f}_{i+1} + m_i \mathbf{a}_i + m_i \mathbf{g}_i \tag{2.4a}$$

$$\boldsymbol{\tau}_i = \boldsymbol{\tau}_i - \mathbf{f}_i \times \mathbf{r}_i + \mathbf{f}_{i+1} \times \mathbf{r}_{i+1} + \mathbf{J}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \mathbf{J}_i \boldsymbol{\omega}_i, \tag{2.4b}$$

where $m_i \in \mathbb{R}_{>0}$ and $\mathbf{J}_i \in \mathbb{R}^{3\times 3}_{>0}$ are the mass and inertia[1] of the ith link, $\mathbf{a}_i \in \mathbb{R}^3$ is its linear acceleration, $\boldsymbol{\omega}_i \in \mathbb{R}^3$ and $\dot{\boldsymbol{\omega}}_i \in \mathbb{R}^3$ are its angular velocity and acceleration, respectively, and $\mathbf{g}_i \in \mathbb{R}^3$ is the gravity vector. Notice that all the previous quantities are defined w.r.t. the ith link frame. The method is repeated for all the links starting from the final one, whose external forces and moments are known, back to the first one.

Finally one could retrieve a closed form dynamic model, like the one in (2.3), resolving all together the forward and backward equations. However, doing it analytically might not be an easy task. We skip the detailed equations because of their complexity. Nevertheless, we refer the interested reader to the well known books [2–5].

2.1.3 Rigid Body Dynamics

In view of the fact that an aerial vehicle is often modeled as a rigid body, it is convenient here to review the dynamic model of such a basic element. A free rigid body, i.e., not subjected to constraints, has six degrees of freedom: three translational and three rotational. Let us assign an inertial *word frame*, $\mathcal{F}_W$ with arbitrary center O_W and axes $\{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$, and a *body frame*, $\mathcal{F}_B$, rigidly attached to the object, with center O_B centered on the body center of mass (CoM), and axes $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$. It is useful to notice here that $\mathbf{x}_W^W = \mathbf{e}_1 = [1\ 0\ 0]^\top$, $\mathbf{y}_W^W = \mathbf{e}_2 = [0\ 1\ 0]^\top$ and[2] $\mathbf{z}_W^W = \mathbf{e}_3 = [0\ 0\ 1]^\top$. The three translational degrees of freedom are described by the position of O_B with respect to $\mathcal{F}_W$, in turn described by the vector[3] $\mathbf{p}_B^W \in \mathbb{R}^3$. The description of the remaining

[1] The notation $\mathbb{R}^{n\times n}_{>0}$ denotes the set of positive-definite real matrices, i.e., $\mathbb{R}^{n\times n}_{>0} = \{\mathbf{A} \in \mathbb{R}^{n\times n} \mid \mathbf{x}^\top \mathbf{A}\mathbf{x} > 0\ \forall\, \mathbf{x} \in \mathbb{R}^n\}$.

[2] More in general, $\mathbf{e}_i \in \mathbb{R}^3$ is the canonical vector with 1 in position ith and zero otherwise.

[3] In this thesis, the superscript is used to indicate the frame of references. When not present, $\mathcal{F}_W$ has to be intended as the reference frame, if not otherwise specified.

three orientation degrees of freedom is a bit more delicate because there are several possible representations [3, 4, 6]. The most popular and used are:

- The *exponential coordinates* are a minimal three-parameter representation of rotations which define an axis of rotation and the corresponding angle of rotation. However, combinations of rotations is not straightforward and the axis of rotation is undetermined when the angle of rotation goes to zero.
- The *Euler-angles* is another minimal three-parameter representation of rotations. It is also very intuitive, since it is based on three successive rotations about the main axes of the body frame. One of the most popular convention in the aeronautic field consists in successive rotations along the moving axes $\mathbf{z}_B$, $\mathbf{y}_B$ and $\mathbf{x}_B$ (in this order) about the angles ψ, θ and ϕ (*Yaw-Pitch-Roll*) respectively.[4] However, this representation has a singularity. To avoid singularities at the control level, we will use this convention only to represent rotations in plots.
- The *rotation matrix*,[5] $\mathbf{R}_B^W \in \mathsf{SO}(3)$, unequivocally describes the rotation of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$. Although this representation has no singularities, it is actually redundant since nine elements describe only three degrees of freedom. Nevertheless, it eases the operations to rotate vectors and to combine rotations. These facts together with the absence of singularities make this representation the preferable for the design of controllers for aerial vehicles. This is why, in this thesis, we will always describes rotations by rotation matrices.
- The *quaternions* represent rotations by a normalized four-dimensional vector, i.e., four variables subjected to one constraint. In this way, the quaternion parametrization does not have singularities. This parametrization is also very popular for it efficiency in terms of computational cost. However, in this thesis we still prefer rotation matrices for their simplicity. This will clearly appear in Chap. 4.

Choosing $\mathbf{p}_B^W$ and $\mathbf{R}_B^W$ to describe the rigid body configuration, we can write the dynamics as in (2.4), using the Newton-Euler approach:

$$m\ddot{\mathbf{p}}_B^W = -mg\mathbf{e}_3 + \mathbf{f} \tag{2.5a}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}}_B^B = -\boldsymbol{\omega}_B^B \times \mathbf{J}\boldsymbol{\omega}_B^B + \boldsymbol{\tau}, \tag{2.5b}$$

where $m \in \mathbb{R}_{>0}$ and $\mathbf{J} \in \mathbb{R}_{>0}^{3\times3}$ are the mass and inertia of the rigid body w.r.t. $\mathcal{F}_B$, $\ddot{\mathbf{p}}_B^W \in \mathbb{R}^3$ is its linear acceleration, $\boldsymbol{\omega}_B^B \in \mathbb{R}^3$ and $\dot{\boldsymbol{\omega}}_B^B \in \mathbb{R}^3$ are its angular velocity and acceleration w.r.t. $\mathcal{F}_W$ expressed in $\mathcal{F}_B$, respectively, $g \approx 9.81$ is the gravitational constant, $\mathbf{f} \in \mathbb{R}^3$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ are the sum of forces and moments applied to the body CoM, respectively. Furthermore we recall the differential kinematic relation $\dot{\mathbf{R}}_B^W = \mathbf{R}_B^W \boldsymbol{\Omega}_B^B$, where $\boldsymbol{\Omega}_\star$ is the skew symmetric matrix associated to $\boldsymbol{\omega}_\star$.

[4]Notice that this representation is equivalent to the classical Roll-Pitch-Yaw representation. The latter consists in successive rotations along the fixed axes $\mathbf{x}_B$, $\mathbf{y}_B$ and $\mathbf{z}_B$ (in this order) about the angles ϕ, θ and ψ respectively.

[5]$\mathsf{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}^\top\mathbf{R} = \mathbf{I}_3\}$ where $\mathbf{I}_n$ is the identity matrix of dimension n. $\mathsf{SO}(3)$ is also called *special orthogonal group*.

As seen before, when it comes to model a floating vehicle, the use of rotation matrix representation and Newton-Euler method is really convenient. The Lagrange method would have instead required the use of minimal representation for the orientation. Nevertheless, as previously said, Newton-Euler method is not favorable in the presence of constraints and reaction forces. Therefore, the approach employed in this thesis tries to exploit the good features of both Lagrangian and Newton-Euler methods. In particular, in order to model a tethered aerial vehicle, in Sect. 4.3 we firstly use the Lagrangian formalism to identify the most convenient generalized coordinates describing the translational dynamics of the vehicle subjected to the constraint given by the link. We instead used a rotation matrix for the description of the attitude. Afterwards, we applied the Newton-Euler method to retrieve the dynamics of the system and the analytical expression of the internal force. Since one of the control objectives is the precise control of the internal force, the analytical expression will be useful to design a tracking controller based on dynamic feedback linearization.

2.2 Differential Flatness

For the analysis of nonlinear dynamic systems, one important property to verify is the *differential flatness*. This property was firstly introduced by Michel Fliess in the late 1980s, and then exploited in many other works for the control of nonlinear systems [7–9]. The formal definition of a differentially flat systems follows:

Definition A system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ with state vector $\mathbf{x} \in \mathbb{R}^n$ and input vector $\mathbf{u} \in \mathbb{R}^m$, where $\mathbf{f}$ is a smooth vector field, is *differentially flat* if it exists an output vector $\mathbf{y} \in \mathbb{R}^m$, called *flat output*, in the form:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ldots, \mathbf{u}^{(q)}) \tag{2.6}$$

such that

$$\mathbf{x} = \mathbf{g}_{\mathbf{x}}(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(r)}) \tag{2.7}$$

$$\mathbf{u} = \mathbf{g}_{\mathbf{u}}(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(r)}) \tag{2.8}$$

where[6] $\mathbf{h}$, $\mathbf{g}_{\mathbf{x}}$ and $\mathbf{g}_{\mathbf{u}}$ are smooth functions, for some finite $r \in \mathbb{N}_{\geq 0}$. □

The previous definition means that for a differentially flat system, we can express the state and the input vectors as an algebraic function of the flat output vector and its derivatives, up to a finite order.

The implications of differential flatness are favorable for both motion planning and control. Thanks to differential flatness, one can simplify trajectory planning problems both from a theoretical and practical point of view [10–12]. The capacity to

[6]The notation $\mathbf{x}^{(r)}$ represents the rth time derivative of $\mathbf{x}$, i.e., $\mathbf{x}^{(r)} = d^r\mathbf{x}/dt^r$.

obtain the nominal state and input from the output (and its derivatives) allows to plan directly for the flat output, using simple algebraic methods and efficient algorithm. Indeed, the flat output equations of motion are simpler, and in the case of bounds and constraints on the state or input, those can be transformed into constraints on the flat outputs and its derivatives. Although this might produce complex nonlinear constraints on the flat output, one can approximate them with simpler functions with the cost of obtaining a sub-optimal solution, but solving the planning problem in a more efficient way. For example, this method has been successfully employed for the design of a kinodynamic motion planner for an unidirectional-thrust aerial vehicle in a cluttered environment [13].

Furthermore, the knowledge of the nominal state and control input required to follow a certain desired flat output trajectory, can be exploited to design robust controllers [9, 14]. For example, this approach was also used to design a decentralized controller for an aerial manipulator [15].

2.3 Dynamic Feedback Linearizing Control

One very common control method for nonlinear systems to solve tracking control problem is the *feedback linearization* [16–18]. The concept of this method consists on finding a particular output, called *linearizing output* and a control law that linearizes the input-output relation, providing a linear system equivalent to the original one. A standard linear controller can be then applied to the latter equivalent linear system in order to track the desired output trajectory. In the following we shall briefly recall how to practically apply this control method. For more details we refer the reader to more specific books on nonlinear systems as [19–21].

Let us consider the nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \tag{2.9a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}), \tag{2.9b}$$

with state vector $\mathbf{x} \in \mathbb{R}^n$, input vector $\mathbf{u} \in \mathbb{R}^m$, output vector $\mathbf{y} = [y_1 \ \dots \ y_m]^\top \in \mathbb{R}^m$, where $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ are smooth functions. From a practical point of view, in order to feedback linearize the system, one has to differentiate every entry of the output until the input appears, i.e., until we can write

$$\begin{bmatrix} y_1^{(r_1)} & \dots & y_m^{(r_m)} \end{bmatrix}^\top = \mathbf{b}(\mathbf{x}) + \mathbf{E}(\mathbf{x})\mathbf{u}, \tag{2.10}$$

where $\mathbf{b}(\mathbf{x}) \in \mathbb{R}^m$ collects all the terms that do not depend on the input, and $\mathbf{E} \in \mathbb{R}^{m\times m}$ is called *decoupling matrix*. If the decoupling matrix is invertible over a certain region, the control law

$$\mathbf{u} = \mathbf{E}(\mathbf{x})^{-1}(-\mathbf{b}(\mathbf{x}) + \mathbf{v}), \tag{2.11}$$

where $\mathbf{v} \in \mathbb{R}^m$ is a new *virtual input*, yields to the simpler linear system

$$\begin{bmatrix} y_1^{(r_1)} & \dots & y_m^{(r_m)} \end{bmatrix}^\top = \mathbf{v}. \tag{2.12}$$

r_i is called the *relative degree* of the ith output entry, and we define $r = \sum_{i=1}^m r_i$ as the *total relative degree*. If the total relative degree is equal to the dimension of the system, i.e., $r = n$, then the system is *exactly feedback linearizable*, i.e., (2.12) is equivalent to the original nonlinear system (2.9) and there is no internal dynamics.

Without loss of generality, let us assume that $\mathbf{E}$ is always not invertible because some of its columns are zero.[7] In particular, let the jth column of $\mathbf{E}$ equal to zero. In other words, this means that the input u_j appears in none output entry. In these cases, in order to make u_j appear, one can apply a *dynamic extension* to the other inputs to delay their appearance in the output derivatives. In details, one can consider the new control input $\bar{\mathbf{u}} \in \mathbb{R}^m$ such that $\bar{u}_i = \dot{u}_i$ if $i \neq j$, and $\bar{u}_i = u_j$ for $i = j$. Now the output has to be differentiated one more time to see the input appear:

$$\begin{bmatrix} y_1^{(r_1+1)} & \dots & y_m^{(r_m+1)} \end{bmatrix}^\top = \bar{\mathbf{b}}(\mathbf{x}) + \bar{\mathbf{E}}(\mathbf{x})\bar{\mathbf{u}}. \tag{2.13}$$

If the new decoupling matrix is invertible and the total relative degree is equal to the system dimension plus the new controller states, then the system is said *dynamic feedback linearizable*. If so, $\bar{\mathbf{u}}$ can be designed similarly to (2.11) to obtain an equivalent linear dynamics as in (2.12). The original inputs u_i can be obtained by integration of $\bar{u}_i$, for $i \neq j$. Notice that the presence of the integrals makes the controller "dynamic".

The tracking of any given desired trajectory, $y_i^d(t)$ for $i = 1, \dots, m$ can be achieved applying any linear control technique to the equivalent linear system (2.12). E.g., it is sufficient to use as outer loop a simple controller based on the pole placing technique. Setting the virtual control inputs as

$$v_i = y_i^{d(r_i)} + \sum_{j=0}^{r_i-1} k_{ij}\xi_{ij}, \tag{2.14}$$

where $\xi_{ij} = y_i^{d(j)} - y_i^{(j)}$. One can set the poles of the error dynamics through the gains $k_{ij} \in \mathbb{R}_{>0}$ and for $j = 0, \dots, r_i$ and $i = 1, \dots, m$ to obtain a sufficiently fast exponentially tracking of the desired trajectories. Notice that an explicit measurement of the output and its derivatives is *not* needed at all, since they are algebraic functions of the state and input.

We remark that this method is strongly model based. For this reason, according to the specific system, it might result not robust to model uncertainties. Nevertheless,

[7] If $\mathbf{E}$ is not invertible one can always apply an invertible, state-dependent, input transformation that zeroes the maximum number of columns in $\mathbf{E}$.

the additional linear controller helps in reducing those negative effects. Furthermore, for some complex systems, the control law (2.11) might result very complicate to implement due to the inversion of $\bar{\mathbf{E}}$ and the possible presence of dynamic extensions.

2.4 High Gain Observer

As shown in the previous section, in order to implement the control action, the knowledge of the state of the system is needed. However, measuring the whole state $\mathbf{x}$ using many sensors is often practically unfeasible due to, e.g., the costs and payload limitations, in particular for aerial robots. Furthermore, possible sensor failures call for the ability to still control the platform with a forcedly limited number of sensors.

In order to solve nonlinear observation problems there are mainly two classes of methods: *approximate nonlinear observers* and *exact nonlinear observers*. The first class relies on approximating the nonlinearities with linear or almost-linear maps around the current estimate, the main disadvantage being the local approximative nature of the methods. The second class of methods consists in nonlinear systems whose state is analytically proven to converge to the real state of the original system. Designing such observers is in general much more difficult since it is often hard to prove the asymptotic stability of a nonlinear system. However the observers of this class may guarantee (almost) global convergence. This is why in this thesis we decided to search for an observer in the second class.

In the literature of exact nonlinear observers an important role is played by a particular class of systems known as *in the canonical form*. This is the class of nonlinear systems (2.9) that can be transformed in a triangular form, as:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & \cdots\cdots & 0 \\ 0 & 0 & 1 \ \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots\cdots & 0 & 1 \\ 0 & \cdots\cdots & \cdots & 0 \end{bmatrix}}_{\mathbf{A}} \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{B}} \phi(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}(\mathbf{u}) \tag{2.15a}$$

$$w = \underbrace{\begin{bmatrix} 1 & 0 & \cdots\cdots & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x}, \tag{2.15b}$$

where $w \in \mathbb{R}$ is the measurement and $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $\boldsymbol{\lambda} : \mathbb{R}^m \to \mathbb{R}^n$ are any nonlinear map. For this class of nonlinear systems, in order to estimate the state one can use the nonlinear *High Gain Observer* (HGO) [20]:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\phi(\hat{\mathbf{x}}, \mathbf{u}) + \boldsymbol{\lambda}(\mathbf{u}) + \mathbf{H}(w - \mathbf{C}\hat{\mathbf{x}}), \tag{2.16}$$

with $\mathbf{H} = [\frac{\alpha_1}{\epsilon} \ \frac{\alpha_2}{\epsilon^2} \ \dots \ \frac{\alpha_n}{\epsilon^n}]^\top$ and $\epsilon \in \mathbb{R}_{>0}$. If $\alpha_i \in \mathbb{R}_{>0}$ are set such that the roots of $p^n + \alpha_1 p^{n-1} + \cdots + \alpha_{n-1} p + \alpha_n$ have negative real part, then (2.16) ensures almost global convergence of the estimated state to the real one.

Furthermore, let us assume that an output feedback controller $\mathbf{u} = \boldsymbol{\Gamma}(\mathbf{x}, \mathbf{v})$ (as (2.11) or its dynamic version) is applied to the system. Then one can show that there exists $\epsilon^\star > 0$ such that, for every $0 < \epsilon < \epsilon^\star$, the closed loop system with controller $\mathbf{u} = \boldsymbol{\Gamma}(\hat{\mathbf{x}}, \mathbf{v})$ and observer (2.16) is exponentially convergent.

However, we recall that, due to the possibly high values of the gains, this observer might suffer from peaking phenomenon during the transient and noise sensitivity. To mitigate those problems, many common practical solutions have been presented in the literature, see e.g., [20, 22]. For example, to overcome the peaking phenomenon, it is sufficient to saturate the estimated state on a bounded region that defines the operative state space bounds for the system in exam. In the presence of measurement noise, the use of a switching-gain approach can guarantee a quick convergence to the real state during the first phase while reducing the noise effects at steady state.

References

1. Isidori, A., Moog, C.H., De Luca, A.: A sufficient condition for full linearization via dynamic state feedback. In: 25th IEEE Conference on Decision and Control, vol. 25, pp. 203–208. IEEE (1986)
2. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: Modelling, Planning and Control. Springer (2009)
3. Siciliano, B., Khatib, O.: Handbook of Robotics. Springer (2008)
4. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control, vol. 3. Wiley, New York (2006)
5. Lynch, K.M., Park, F.C.: Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press (2017)
6. Corke, P.: Robotics, Vision and Control: Fundamental Algorithms in MATLAB® Second, Completely Revised, vol. 118. Springer (2017)
7. Fliess, M., Lévine, J., Martin, P., Rouchon, P.: Flatness and defect of nonlinear systems: introductory theory and examples. Int. J. Control **61**(6), 1327–1361 (1995)
8. Murray, R.M., Rathinam, M., Sluis, W.: Differential flatness of mechanical control systems: a catalog of prototype systems. In: ASME International Mechanical Engineering Congress and Exposition, San Francisco, CA, Nov (1995)
9. Rigatos, G.G.: Nonlinear Control and Filtering Using Differential Flatness Approaches: Applications to Electromechanical Systems, vol. 25. Springer (2015)
10. Rouchon, P., Fliess, M., Levine, J., Martin, P.: Flatness and motion planning: the car with n trailers. In: 1993 European Control Conference, pp. 1518–1522, Groningen (1993)
11. Chamseddine, A., Zhang, Y., Rabbath, C.A., Join, C., Theilliol, D.: Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle. IEEE Trans. Aerosp. Electron. Syst. **48**(4), 2832–2848 (2012)
12. De Luca, A., Oriolo, G.: Trajectory planning and control for planar robots with passive last joint. Int. J. Robot. Res. **21**(5–6), 575–590 (2002)
13. Boeuf, A., Cortés, J., Alami, R., Siméon, T.: Enhancing sampling-based kinodynamic motion planning for quadrotors. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2447–2452, Hamburg, Germany, Sept (2015)

14. Tang, C.P., Miller, P.T., Krovi, V.N., Ryu, J., Agrawal, S.K.: Differential-flatness-based planning and control of a wheeled mobile manipulator—theory and experiment. IEEE/ASME Trans. Mechatron. **16**(4), 768–773 (2011)
15. Tognon, M., Yüksel, B., Buondonno, G., Franchi, A.: Dynamic decentralized control for protocentric aerial manipulators. In: 2017 IEEE International Conference on Robotics and Automation, pp. 6375–6380, Singapore, May (2017)
16. Oriolo, G., De Luca, A., Vendittelli, M.: WMR control via dynamic feedback linearization: design, implementation, and experimental validation. IEEE Trans. Control Syst. Technol. **10**(6), 835–852 (2002)
17. De Luca, A.: Decoupling and feedback linearization of robots with mixed rigid/elastic joints. Int. J. Robust Nonlinear Control: IFAC-Affil. J. **8**(11), 965–977 (1998)
18. Martin, P., Devasia, S., Paden, B.: A different look at output tracking: control of a VTOL aircraft. In: Proceedings of the 33rd IEEE Conference on Decision and Control, vol. 3, pp. 2376–2381. IEEE (1994)
19. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice Hall (1991)
20. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall (2001)
21. Isidori, A.: Nonlinear Control Systems, 3rd edn. Springer (1995)
22. Ahrens, J.H., Khalil, H.K.: High-gain observers in the presence of measurement noise: a switched-gain approach. Automatica **45**(4), 936–943 (2009)