5

Nonlinear Control in Human–Like Biomechanics

In this Chapter we develop the basics of nonlinear control theory as is used in modern human–like biomechanics. It includes control variations on the central theme of our *covariant force law*, $F_i = mg_{ij}a^j$, and its associated *covariant force functor* $\mathcal{F}_* : TT^*M \longrightarrow TTM$ (see section 2.7 above).

5.1 The Basics of Classical Control and Stability

In this section we present the basics of classical control and stability theory, to be used in the subsequent sections.

5.1.1 Brief Introduction into Feedback Control

The *basic formula of feedback control* reads

$$Sensing + Computation + Actuation = Feedback\,Control \tag{5.1}$$

The formula (5.1) implies the *basic premise of control engineering*:

- Given a *system* to be controlled and the specifications of its *desired behavior*, construct a *feedback control law* to make the closed–loop system display its desired behavior.

  The three basic goals of feedback control are (see [Mur97]):

1. *Stability*, which states that bounded inputs produce bounded outputs;
2. *Performance*, which defines how to achieve desired response; and
3. *Robustness*, which balances stability versus performance in the presence of unknown dynamics.

For example, consider the popular problem of stabilization of an *inverted pendulum* (see Figure 5.1), in which dynamics is governed by the Newtonian–like equation

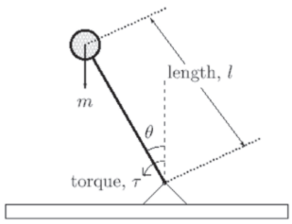

Fig. 5.1. An inverted pendulum (see text for explanation).

$$J\ddot{\theta} - mgl\sin\theta = \tau,$$

and we want to start from a large angle, say $\theta(0) = 60°$ and move to the vertical upright position, $\theta = 0$.

One choice of a stabilizer is (see [Wil00])

$$\tau = -k_d\dot{\theta} - k_p\theta - mgl\sin\theta,$$

where τ is the stabilizing torque, while k_d and k_p are positive constants. In this case closed loop dynamics is given by

$$J\ddot{\theta} + k_d\dot{\theta} + k_p\theta = 0,$$

which is globally stable and linear.

An alternative controller is given by

$$\tau = -k_d\dot{\theta} - 2mgl\sin\theta,$$

leading to the globally stable nonlinear closed–loop dynamics

$$J\ddot{\theta} + k_d\dot{\theta} + mgl\sin\theta = 0.$$

This example shows how the feedback and feedforward control amounts to modifying the dynamics of the plant into a desired form. It is further expanded as a difficult nonholonomic problem of a unicycle (5.2.2) below.

To summarize, the basic components of a feedback control system are (see Figure 5.2):

1. Plant, including (bio)physical system, actuation and sensing;
2. Controller, including state estimator and regulator; and
3. Feedback, including interconnection between plant output and controller input.

Control systems are usually represented using:

- Linear or nonlinear ODEs; and
- Block diagrams with transfer functions (Laplace transform based).

Historically, four periods can be distinguished in control theory (see [Mur97]):

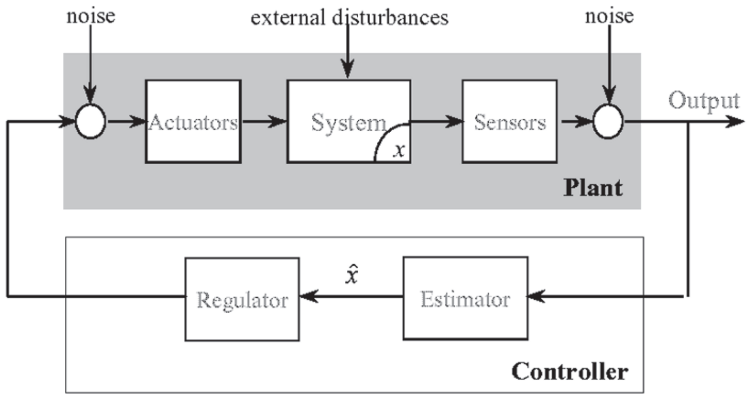

Fig. 5.2. The basic components of a feedback control system (see text for explanation).

1. Classical control (1940–1960). This period is characterized by:
   - Frequency domain based tools; stability via gain and phase margins;
   - Mainly useful for single–input, single–output (SISO) systems;
   - Control is one of the main tools for the practicing engineer.
2. Modern control (1940–1960). In this period:
   - The so–called *state–space approach* was developed for linear control theory;
   - It works both for SISO and multi–input, multi–output (MIMO) systems;
   - Performance and robustness measures are often not made explicit.
3. Post–modern control (1940–1960). This period:
   - Generalizes ideas in classical control to MIMO context; and
   - Uses operator theory at its core, but can be easily interpreted in frequency domain.
4. Nonlinear control (1990–). This period is characterized by specialized techniques for control of nonlinear plants.

Now, as already stated, the *goal of a control system* is to enhance automation within a system while providing improved performance and robustness. For instance, we may develop a *cruise* control system for an automobile to release drivers from the tedious task of speed regulation while they are on long trips. In this case, the *output* of the *plant* is the sensed vehicle speed, y, and the *input* to the plant is the throttle angle, u. Typically, control systems are designed so that the *plant output follows some reference input* (the driver–specified speed in the case of our cruise control example) while achieving some level of *disturbance rejection*. For the cruise control problem, a disturbance would be a road grade variation or wind. Clearly we would want our cruise controller to reduce the effects of such disturbances on the quality of the speed regulation that is achieved [SMO02].

In the area of *robust control* the focus is on the development of controllers that can maintain good performance even if we only have a poor model of the plant or if there are some plant parameter variations. In the area, of *adaptive control*, to reduce the effects of plant parameter variations, robustness is achieved by adjusting (i.e., adapting) the controller on–line. For instance, an adaptive controller for the cruise control problem would seek to achieve good speed tracking performance even if we do not have a good model of the vehicle and engine dynamics, or if the vehicle dynamics change over time (e.g., via a weight change that results from the addition of cargo, or due to engine degradation over time). At the same time it would try to achieve good disturbance rejection. Clearly, the performance of a good cruise controller should not degrade significantly as your automobile ages or if there are reasonable changes in the load the vehicle is carrying [SMO02].

We use *adaptive mechanisms* within the control laws when certain *parameters* within the plant dynamics are *unknown*. An adaptive controller is used to improve the closed–loop system robustness while meeting a set of performance objectives. If the plant uncertainty cannot be expressed in terms of unknown parameters, one may be able to reformulate the problem by expressing the uncertainty in terms of a *fuzzy system*, *neural network*, or some other *parameterized nonlinear system*, like an *adaptive Lie–derivative controller*. The uncertainty then becomes recast in terms of a new set of unknown parameters that may be adjusted using adaptive techniques.

When developing a *robust control design*, the focus is on *maintaining stability* even in the presence of *unmodelled plant dynamics* or *external disturbances*. The approach in robust control is to accept *a'priori* that there will be model uncertainty, and try to cope with it.

The issue of robustness has been studied extensively in the control literature [SMO02]. When working with linear systems, one may define phase and gain margins which quantify the range of uncertainty a closed–loop system may withstand before becoming unstable. In the world of *nonlinear control design*, we often investigate the stability of a closed–loop system by studying the behavior of a *Lyapunov function candidate*. The Lyapunov function candidate is a mathematical function designed to provide a simplified measure of the control objectives allowing complex nonlinear systems to be analyzed using a scalar differential equation. When a controller is designed that drives the Lyapunov function to zero, the control objectives are met. If some system uncertainty tends to drive the Lyapunov candidate away from zero, we often simply add an additional stabilizing term to the control algorithm that dominates the effect of the uncertainty, thereby making the closed–loop system more robust.

Now, by adding a static term in the control law that simply dominates the plant uncertainty, it is often easy to simply stabilize an uncertain plant, however, driving the system error to zero may be difficult if not impossible.

Consider the case when the plant is defined by [SMO02]

$$\dot{x} = \theta x + u, \tag{5.2}$$

where $x \in \mathbb{R}$ is the plant state that we wish to drive to the point $x = 1$, $u \in \mathbb{R}$ is the plant input, and θ is an unknown constant. Since θ is unknown, one may not define a static controller that causes $x = 1$ to be a stable equilibrium point. In order for $x = 1$ to be a stable equilibrium point, it is necessary that $\dot{x} = 0$ when $x = 1$, so $u(x) = -\theta$ when $x = 1$. Since θ is unknown, however, we may not define such a controller. In this case, the best that a static nonlinear controller may do is to keep x bounded in some region around $x = 1$. If dynamics are included in the nonlinear controller, then it turns out that one may define a control system that does drive $x \to 1$ even if θ is unknown.

On the other hand, an adaptive controller can be designed so that it estimates some uncertainty within the system, then automatically designs a controller for the estimated plant uncertainty. In this way the control system uses information gathered on–line to reduce the model uncertainty, that is, to figure out exactly what the plant is at the current time so that good control can be achieved. Considering the system defined by (A.19), an adaptive controller may be defined so that an *estimate* of θ is generated, which we denote by $\hat{\theta}$. If θ were known, then including a term $-\theta x$ in the control law would cancel the effects of the uncertainty. If $\hat{\theta} \to \theta$ over time, then including the term $-\hat{\theta} x$ in the control law would also cancel the effects of the uncertainty over time. This approach is referred to as *indirect adaptive control* [SMO02].

An indirect approach to adaptive control is made up of an *approximator* (often referred to as an *identifier* in the adaptive control literature) that is used to estimate unknown plant parameters and a certainty equivalence control scheme in which the plant controller is designed, assuming that the parameter estimates are their true values. Here the adjustable approximator is used to model some component of the system. Since the approximation is used in the control law, it is possible to determine if we have a good estimate of the plant dynamics. If the approximation is good (i.e., we know how the plant should behave), then it is easy to meet our control objectives. If, on the other hand, the plant output moves in the wrong direction, then we may assume that our estimate is incorrect and should be adjusted accordingly.

As an example of an indirect adaptive controller, consider the cruise control problem where we have an approximator that is used to estimate the vehicle mass and aerodynamic drag. Assume that the vehicle dynamics may be approximated by

$$m\dot{x} = -\rho x^2 + u,$$

where m is the vehicle mass, ρ is the coefficient of aerodynamic drag, x is the vehicle velocity, and u is the plant input. Assume that an approximator has been defined so that estimates of the mass and drag are found such that $\hat{m} \to m$ and $\hat{\rho} \to \rho$. Then the control law

$$u = \hat{\rho} x^2 + \hat{m} v(t)$$

may be used so that $\dot{x} = v(t)$ when $\hat{m} = m$ and $\hat{\rho} = \rho$. Here $v(t)$ may be considered a new control input that is defined to drive x to any desired value [SMO02].

5.1.2 Linear Stationary Systems and Operators

Basics of Kalman State–Space Theory

It is well–known that linear multiple input–multiple output (MIMO) control systems can always be put into Kalman canonical state–space form of order n, with m inputs and k outputs. In the case of *continual time systems* we have the state and output equations of the form[1]

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(t)\,\mathbf{x}(t) + \mathbf{D}(t)\,\mathbf{u}(t), \end{aligned} \tag{5.3}$$

while in case of *discrete time systems* we have the state and output equations of the form

$$\begin{aligned} \mathbf{x}(n+1) &= \mathbf{A}(n)\,\mathbf{x}(n) + \mathbf{B}(n)\,\mathbf{u}(n), \\ \mathbf{y}(n) &= \mathbf{C}(n)\,\mathbf{x}(n) + \mathbf{D}(n)\,\mathbf{u}(n). \end{aligned} \tag{5.4}$$

Both in (5.3) and in (5.4) the variables have the following meaning:

$\mathbf{x}(t) \in \mathbb{X}$ is an $n-$vector of *state variables* belonging to the *state space* $\mathbb{X} \subset \mathbb{R}^n$;

$\mathbf{u}(t) \in \mathbb{U}$ is an $m-$vector of *inputs* belonging to the *input space* $\mathbb{U} \subset \mathbb{R}^m$;

$\mathbf{y}(t) \in \mathbb{Y}$ is a $k-$vector of *outputs* belonging to the *output space* $\mathbb{Y} \subset \mathbb{R}^k$;

$\mathbf{A}(t) : \mathbb{X} \to \mathbb{X}$ is an $n \times n$ matrix of *state dynamics*;

$\mathbf{B}(t) : \mathbb{U} \to \mathbb{X}$ is an $n \times m$ matrix of *input map*;

$\mathbf{C}(t) : \mathbb{X} \to \mathbb{Y}$ is an $k \times n$ matrix of *output map*;

$\mathbf{D}(t) : \mathbb{U} \to \mathbb{Y}$ is an $k \times m$ matrix of *input–output transform*.

Input $\mathbf{u}(t) \in \mathbb{U}$ can be empirically determined by trial and error; it is properly defined by optimization process called *Kalman regulator*, or more generally (in the presence of noise), by *Kalman filter* (even better, *extended Kalman filter* to deal with stochastic nonlinearities) [Kal60].

Now, the most common special case of the general Kalman model (5.3), with constant state, input and output matrices (and relaxed boldface vector–matrix notation), is the so–called *stationary linear model* Such systems frequently serve as a baseline, against which other control systems are measured.

[1] In our covariant form, (5.4) reads

$$\dot{x}^i = a^i_j x^j + b^i_k u^k, \qquad y^i = c^i_j x^j + d^i_k u^k, \qquad (i,j = 1,...,n;\ k = 1,...,m).$$

We follow a common notational convention and let u denote the vector of *inputs*, y the vector of *outputs* and assume that they can be related through an intermediary *state* variable x according to the equations

$$\dot{x} = Ax + Bu, \qquad y = Cx. \tag{5.5}$$

We refer to this as the deterministic *stationary linear model.* The stationary linear system (5.5) defines a variety of operators, in particular those related to: (i) regulators, (ii) end point controls, (iii) servomechanisms, and (iv) repetitive modes (see [Bro01]).

Regulator Problem and the Steady State Operator

Consider a variable, or set of variables, associated with a dynamical system. They are to be maintained at some desired values in the face of changing circumstances. There exist a second set of parameters that can be adjusted so as to achieve the desired regulation. The effecting variables are usually called *inputs* and the affected variables called *outputs.* Specific examples include the regulation of the thrust of a jet engine by controlling the flow of fuel, as well as the regulation of the oxygen content of the blood using the respiratory rate.

Now, there is the steady state operator of particular relevance for the regulator problem. It is

$$y_\infty = -CA^{-1}Bu_\infty,$$

which describes the map from constant values of u to the equilibrium value of y. It is defined whenever A is invertible but the steady state value will only be achieved by a real system if, in addition, the eigenvalues of A have negative real parts. Only when the rank of $CA^{-1}B$ equals the dimension of y can we steer y to an arbitrary steady state value and hold it there with a constant u. A nonlinear version of this problem plays a central role in robotics where it is called the *inverse kinematics problem* (see, e.g., [MLS94]).

End Point Control Problem and the Adjustment Operator

Here we have inputs, outputs and trajectories. In this case the shape of the trajectory is not of great concern but rather it is the end point that is of primary importance. Standard examples include rendezvous problems such as one has in space exploration.

Now, the operator of relevance for the end point control problem, is the operator

$$x(T) = \int_0^T \exp[A(T-\sigma)]\, Bu(\sigma)\, d\sigma.$$

If we consider this to define a map from the mD L_2 space $L_2^m[0,T]$ (where u takes on its values) into $\mathbb{R}^m$ then, if it is an onto map, it has a Moore–Penrose (least squares) inverse

$$u(\sigma) = B^T \exp[A^T(T-\sigma)] \ (W[0,T])^{-1} (x(T) - \exp(AT)\, x(0)),$$

with the symmetric positive definite matrix W, the *controllability Gramian*, being given by

$$W[0,T] = \int_0^T \exp[A(T-\sigma)]\, BB^T \exp[A^T(T-\sigma)]\, d\sigma.$$

Servomechanism Problem and the Corresponding Operator

Here we have inputs, outputs and trajectories, as above, and an associated dynamical system. In this case, however, it is desired to cause the outputs to follow a trajectory specified by the input. For example, the control of an airplane so that it will travel along the flight path specified by the flight controller.

Now, because we have assumed that A, B and C are constant

$$y(t) = C \exp(At)\, x(0) + \int_0^t C \exp[A(T-\tau)]\, Bu(\tau)\, d\tau,$$

and, as usual, the *Laplace transform* $\mathcal{L}$, defined as a pair of inverse maps $\mathcal{L} = \{F, f\} : \mathbb{R} \leftrightarrows \mathbb{C}$,

$$F(s) = \{\mathcal{L}f(t)\}(s) = \int_0^\infty e^{-st} f(t)\, dt, \qquad (t \in \mathbb{R},\, s \in \mathbb{C})$$
$$f(t) = \{\mathcal{L}^{-1}F(s)\}(t) = \frac{1}{2\pi i}\int_{\gamma - i\infty}^{\gamma + i\infty} e^{st} F(s)\, ds,$$

– can be used to convert convolution to multiplication. This brings out the significance of the Laplace transform pair

$$C \exp(At) B \quad \overset{\mathcal{L}}{\Longleftrightarrow} \quad C(Is - A)^{-1} B \tag{5.6}$$

as a means of characterizing the input–output map of a linear model with constant coefficients.

Repetitive Mode Problem and the Corresponding Operator

Here again one has some variable, or set of variables, associated with a dynamical system and some inputs which influence its evolution. The task has elements which are repetitive and are to be done efficiently. Examples from biology include the control of respiratory processes, control of the pumping action of the heart, control of successive trials in practicing a athletic event.

The relevant operator is similar to the servomechanism operator, however the constraint that u and x are periodic means that the relevant diagonalization is provided by Fourier series, rather than the Laplace transform. Thus, in the Fourier domain, we are interested in a set of complex matrices

$$G(iw_i) = C(iw_i - A)^{-1}B, \qquad (w_i = 0, w_0, 2w_0, ...)$$

More general, but still deterministic, models of the input–state–output relation are afforded by the *nonlinear affine control system* (see, e.g., [Isi89])

$$\dot{x}(t) = f(x(t)) + g(x(t))\,u(t), \qquad y(t) = h(x(t));$$

and the still more general *fully nonlinear control system*

$$\dot{x}(t) = f(x(t), u(t)), \qquad y(t) = h(x(t)).$$

Feedback Changes the Operator

No idea is more central to automatic control than the idea of feedback. When an input is altered on the basis of the difference between the actual output of the system and the desired output, the system is said to involve *feedback*. Man made systems are often constructed by starting with a basic element such as a motor, a burner, a grinder, etc. and then adding sensors and the hardware necessary to use the measurement generated by the sensors to regulate the performance of the basic element. This is the *essence of feedback control*. Feedback is often contrasted with open loop systems in which the inputs to the basic element is determined without reference to any measurement of the trajectories. When the word feedback is used to describe naturally occurring systems, it is usually implicit that the behavior of the system can best be explained by pretending that it was designed as one sees man made systems being designed [Bro01].

In the context of linear systems, the effect of feedback is easily described. If we start with the stationary linear system (5.5) with u being the controls and y being the measured quantities, then the effect of feedback is to replace u by $u - Ky$ with K being a matrix of feedback gains. The closed–loop equations are then

$$\dot{x} = (A - BKC)\,x + Bu, \qquad y = Cx.$$

Expressed in terms of the Laplace transform pairs (5.6), feedback effects the transformation

$$\big(C\exp(At)B; C(Is - A)^{-1}B\big) \longmapsto C\exp(A{-}BKC)^t B; C(Is{-}A{+}BKC)^{-1}B.$$

Using such a transformation, it is possible to alter the dynamics of a system in a significant way. The modifications one can effect by feedback include influencing the location of the eigenvalues and consequently the stability of the system. In fact, if K is m by p and if we wish to select a gain matrix K so that $A - BKC$ has eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$, it is necessary to insure that

$$\det\begin{pmatrix} C(I\lambda_1 - A)^{-1}B & -I \\ I & K \end{pmatrix} = 0, \qquad (i = 1, 2, ..., n).$$

Now, if CB is invertible then we can use the relationship $C\dot{x} = CAx + CBu$ together with $y = Cx$ to write $\dot{y} = CAx + CBu$. This lets us solve for u and recast the system as

$$\dot{x} = (A - B(CB)^{-1}CA)\,x + B(CB)^{-1}\dot{y},$$
$$u = (CB)^{-1}\dot{y} - (CB)^{-1}CA\,x.$$

Here we have a set of equations in which the roles of u and y are reversed. They show how a choice of y determines x and how x determines u [Bro01].

5.1.3 Stability and Boundedness

Let a time–varying dynamical system may be expressed as

$$\dot{x}(t) = f(t, x(t)), \tag{5.7}$$

where $x \in \mathbb{R}^n$ is an nD vector and $f : \mathbb{R}^+ \times D \to \mathbb{R}^n$ with $D = \mathbb{R}^n$ or $D = B_h$ for some $h > 0$, where $B_h = \{x \in \mathbb{R}^n : |x| < h\}$ is a ball centered at the origin with a radius of h. If $D = \mathbb{R}^n$ then we say that the dynamics of the system are defined *globally*, whereas if $D = B_h$ they are only defined *locally*. We do not consider systems whose dynamics are defined over disjoint subspaces of R. It is assumed that $f(t, x)$ is piecemeal continuous in t and Lipschitz in x for existence and uniqueness of state solutions. As an example, the linear system $\dot{x}(t) = Ax(t)$ fits the form of (5.7) with $D = \mathbb{R}^n$ [SMO02].

Assume that for every x_0 the initial value problem

$$\dot{x}(t) = f(t, x(t)), \qquad x(t_0) = x_0,$$

possesses a unique solution $x(t, t_0, x_0)$; it is called a solution to (5.7) if $x(t, t_0, x_0) = x_0$ and $\frac{d}{dt}x(t, t_0, x_0) = f(t, x(t, t_0, x_0))$ [SMO02].

A point $x_e \in \mathbb{R}^n$ is called an *equilibrium point* of (5.7) if $f(t, x_e) = 0$ for all $t \geq 0$. An equilibrium point x_e is called an *isolated equilibrium point* if there exists an $\rho > 0$ such that the ball around x_e, $B_\rho(x_e) = \{x \in \mathbb{R}^n : |x - x_e| < \rho\}$, contains no other equilibrium points besides x_e [SMO02].

The equilibrium $x_e = 0$ of (5.7) is said to be *stable in the sense of Lyapunov* if for every $\epsilon > 0$ and any $t_0 \geq 0$ there exists a $\delta(\epsilon, t_0) > 0$ such that $|x(t, t_0, x_0)| < \epsilon$ for all $t \geq t_0$ whenever $|x_0| < \delta(\epsilon, t_0)$ and $x(t, t_0, x_0) \in B_h(x_e)$ for some $h > 0$. That is, the equilibrium is stable if when the system (5.7) starts close to x_e, then it will stay close to it. Note that stability is a property of an equilibrium, not a system. A system is stable if all its equilibrium points are stable. Stability in the sense of Lyapunov is a local property. Also, notice that the definition of stability is for a single equilibrium $x_e \in \mathbb{R}^n$ but actually such an equilibrium is a trajectory of points that satisfy the differential equation in (5.7). That is, the equilibrium x_e is a solution to the differential equation (5.7), $x(t, t_0, x_0) = x_e$ for $t \geq 0$. We call any set such that when the initial condition of (5.7) starts in the set and stays in the set for all $t \geq 0$, an *invariant set*. As an example, if $x_e = 0$ is an equilibrium, then the set containing only the point x_e is an invariant set, for (5.7) [SMO02].

If δ is independent of t_0, that is, if $\delta = \delta(\epsilon)$, then the equilibrium x_e is said to be *uniformly stable*. If in (5.7) f does not depend on time (i.e., $f(x)$), then x_e being stable is equivalent to it being uniformly stable. Uniform stability is also a local property.

The equilibrium $x_e = 0$ of (5.7) is said to be *asymptotically stable* if it is stable and for every $t_0 \geq 0$ there exists $\eta(t_0) > 0$ such that $\lim_{t\to\infty} |x(t, t_0, x_0)| = 0$ whenever $|x_0| < \eta(t_0)$. That is, it is asymptotically stable if when it starts close to the equilibrium it will converge to it. Asymptotic stability is also a local property. It is a stronger stability property since it requires that the solutions to the ordinary differential equation converge to zero in addition to what is required for stability in the sense of Lyapunov.

The equilibrium $x_e = 0$ of (5.7) is said to be *uniformly asymptotically stable* if it is uniformly stable and for every $\epsilon > 0$ and and $t_0 \geq 0$, there exist a $\delta_0 > 0$ independent of t_0 and ϵ, and a $T(\epsilon) > 0$ independent of t_0, such that $|x(t, t_0, x_0) - x_e| \leq \epsilon$ for all $t \geq t_0 + T(\epsilon)$ whenever $|x_0 - x_e| < \delta(\epsilon)$. Again, if in (5.7) f does not depend on time (i.e., $f(x)$), then x_e being asymptotically stable is equivalent to it being uniformly asymptotically stable. Uniform asymptotic stability is also a local property.

The set $X_d \subset \mathbb{R}^n$ of all $x_0 \in \mathbb{R}^n$ such that $|x(t, t_0, x_0)| \to 0$ as $t \to \infty$ is called the *domain of attraction* of the equilibrium $x_e = 0$ of (5.7). The equilibrium $x_e = 0$ is said to be *asymptotically stable* in the large if $X_d \subset \mathbb{R}^n$. That is, an equilibrium is asymptotically stable in the large if no matter where the system starts, its state converges to the equilibrium asymptotically. This is a global property as opposed to the earlier stability definitions that characterized local properties. This means that for asymptotic stability in the large, the local property of asymptotic stability holds for $B_h(x_e)$ with $h = \infty$ (i.e., on the whole state–space).

The equilibrium $x_e = 0$ is said to be *exponentially stable* if there exists an $\alpha > 0$ and for every $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that $|x(t, t_0, x_0)| \leq \epsilon e^{-\alpha(t-t_0)}$, whenever $|x_0| < \delta(\epsilon)$ and $t \geq t_0 \geq 0$. The constant α is sometimes called the *rate of convergence*. Exponential stability is sometimes said to be a 'stronger' form of stability since in its presence we know that system trajectories decrease exponentially to zero. It is a local property; here is its global version. The equilibrium point $x_e = 0$ is *exponentially stable in the large* if there exists $\alpha > 0$ and for any $\beta > 0$ there exists $\epsilon(\beta) > 0$ such that $|x(t, t_0, x_0)| \leq \epsilon(\beta)e^{-\alpha(t-t_0)}$, whenever $|x_0| < \beta$ and $t \geq t_0 \geq 0$.

An equilibrium that is not stable is called *unstable*.

Closely related to stability is the concept of *boundedness*, which is, however, a global property of a system in the sense that it applies to trajectories (solutions) of the system that can be defined over all of the state–space [SMO02].

A solution $x(t, t_0, x_0)$ of (5.7) is *bounded* if there exists a $\beta > 0$, that may depend on each solution, such that $|x(t, t_0, x_0)| < \beta$ for all $t \geq t_0 \geq 0$. A system is said to possess *Lagrange stability* if for each $t_0 \geq 0$ and $x_0 \in \mathbb{R}^n$, the solution $x(t, t_0, x_0)$ is bounded. If an equilibrium is asymptotically stable in the large or exponentially stable in the large then the system for which the equilibrium is defined is also Lagrange stable (but not necessarily vice versa). Also, if an equilibrium is stable, it does not imply that the system for which the equilibrium is defined is Lagrange stable since there may be a way to pick x_0 such that it is near an unstable equilibrium and $x(t, t_0, x_0) \to \infty$ as $t \to \infty$.

The solutions $x(t, t_0, x_0)$ are *uniformly bounded* if for any $\alpha > 0$ and $t_0 \geq 0$, there exists a $\beta(\alpha) > 0$ (independent of t_0) such that if $|x_0| < \alpha$, then $|x(t, t_0, x_0)| < \beta(\alpha)$ for all $t \geq t_0 \geq 0$. If the solutions are uniformly bounded then they are bounded and the system is Lagrange stable.

The solutions $x(t, t_0, x_0)$ are said to be *uniformly ultimately bounded* if there exists some $B > 0$, and if corresponding to any $\alpha > 0$ and $t_0 > 0$ there exists a $T(\alpha) > 0$ (independent of t_0) such that $|x_0| < \alpha$ implies that $|x(t, t_0, x_0)| < B$ for all $t \geq t_0 + T(\alpha)$. Hence, a system is said to be uniformly ultimately bounded if eventually all trajectories end up in a $B-$neighborhood of the origin.

5.1.4 Lyapunov's Stability Method

A. M. Lyapunov invented two methods to analyze stability [SMO02]. In his *indirect method* he showed that if we linearize a system about an equilibrium point, certain conclusions about local stability properties can be made (e.g., if the eigenvalues of the linearized system are in the left half plane then the equilibrium is stable but if one is in the right half plane it is unstable).

In his *direct method* the stability results for an equilibrium $x_e = 0$ of (5.7) depend on the existence of an appropriate *Lyapunov function* $V : D \to \mathbb{R}$ where $D = \mathbb{R}^n$ for global results (e.g., asymptotic stability in the large) and $D = B_h$ for some $h > 0$, for local results (e.g., stability in the sense of Lyapunov or asymptotic stability). If V is continuously differentiable with respect to its arguments then the derivative of V with respect to t along the solutions of (5.7) is

$$\dot{V}(t, x) = \partial_t V + \partial_x V f(t, x).$$

As an example, suppose that (5.7) is autonomous, and let $V(x)$ is a quadratic form $V(x) = x^T P x$ where $x \in \mathbb{R}^n$ and $P = P^T$. Then, $\dot{V}(x) = \frac{\partial V}{\partial x} f(t, x) = \dot{x}^T P x + x^T P \dot{x} = 2x^T P \dot{x}$ [SMO02].

Lyapunov's direct method provides for the following ways to test for stability. The first two are strictly for local properties while the last two have local and global versions.

- *Stable*: If $V(t, x)$ is continuously differentiable, positive definite, and $\dot{V}(t, x) \leq 0$, then $x_e = 0$ is stable.

- *Uniformly stable*: If $V(t, x)$ is continuously differentiable, positive definite, decrescent[2], and $V(t, x) \leq 0$, then $x_e = 0$ is uniformly stable.

[2] A C^0-function $V(t, x) : \mathbb{R}^+ \times B_h \to \mathbb{R}(V(t, x) : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R})$ is said to be *decrescent* if there exists a strictly increasing function γ defined on $[0, r)$ for some

- *Uniformly asymptotically stable*: If $V(t,x)$ is continuously differentiable, positive definite, and decrescent, with negative definite $\dot{V}(t,x)$, then $x_e = 0$ is uniformly asymptotically stable (uniformly asymptotically stable in the large if all these properties hold globally).

- *Exponentially stable*: If there exists a continuously differentiable $V(t,x)$ and $c, c_1, c_2, c_3 > 0$ such that

$$c_1 \left|x\right|^c \leq V(t,x) \leq c_2 \left|x\right|^c, \qquad \dot{V}(t,x) \leq -c_{31} \left|x\right|^c, \tag{5.8}$$

for all $x \in B_h$ and $t \geq 0$, then $x_e = 0$ is exponentially stable. If there exists a continuously differentiable function $V(t,x)$ and equations (5.8) hold for some $c, c_1, c_2, c_3 > 0$ for all $x \in \mathbb{R}^n$ and $t \geq 0$, then $x_e = 0$ is exponentially stable in the large [SMO02].

5.2 The Basis of Modern Geometric Control

In this section we present the basics of modern geometric control, as currently used in modern biomechanics.

5.2.1 Feedback Linearization

Exact Feedback Linearization

The idea of feedback linearization is to algebraically transform the nonlinear system dynamics into a fully or partly linear one so that the linear control techniques can be applied. Note that this is not the same as a conventional linearization using Jacobians. In this subsection we will present the modern, geometric, Lie–derivative based techniques for exact feedback linearization of nonlinear control systems.

The Lie Derivative and Lie Bracket in Control Theory

Recall (see (2.4.1) above) that given a scalar function $h(x)$ and a vector–field $f(x)$, we define a new scalar function, $\mathcal{L}_f h = \nabla h f$, which is the Lie derivative of h w.r.t. f, i.e., the directional derivative of h along the direction of the vector f. Repeated Lie derivatives can be defined recursively:

$$\mathcal{L}_f^0 h = h, \qquad \mathcal{L}_f^i h = \mathcal{L}_f \left(\mathcal{L}_f^{i-1} h\right) = \nabla \left(\mathcal{L}_f^{i-1} h\right) f, \qquad (\text{for} \;\; i = 1, 2, ...)$$

Or given another vector–field, g, then $\mathcal{L}_g \mathcal{L}_f h(x)$ is defined as

$$\mathcal{L}_g \mathcal{L}_f h = \nabla \left(\mathcal{L}_f h\right) g.$$

$r > 0$ (defined on $[0, \infty)$) such that $V(t,x) \leq \gamma(|x|)$ for all $t \geq 0$ and $x \in B_h$ for some $h > 0$.

For example, if we have a control system

$$\dot{x} = f(x), \qquad y = h(x),$$

with the state $x = x(t)$ and the the output y, then the derivatives of the output are:

$$\dot{y} = \frac{\partial h}{\partial x}\dot{x} = \mathcal{L}_f h, \qquad \text{and} \qquad \ddot{y} = \frac{\partial L_f h}{\partial x}\dot{x} = \mathcal{L}_f^2 h.$$

Also, recall that the curvature of two vector–fields, g_1, g_2, gives a non–zero Lie bracket (2.4.1), $[g_1, g_2]$ (see Figure 5.3). Lie bracket motions can generate new directions in which the system can move.

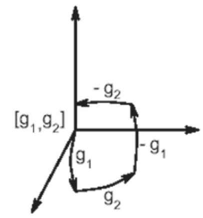

Fig. 5.3. 'Lie bracket motion' is possible by appropriately modulating the control inputs (see text for explanation).

In general, the Lie bracket of two vector–fields, $f(x)$ and $g(x)$, is defined by

$$[f, g] = Ad_f g = \nabla g f - \nabla f g = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g,$$

where $\nabla f = \partial f / \partial x$ is the Jacobian matrix. We can define Lie brackets recursively,

$$Ad_f^0 g = g, \qquad Ad_f^i g = [f, Ad_f^{i-1} g], \qquad (\text{for} \quad i = 1, 2, ...)$$

Lie brackets have the properties of bilinearity, skew–commutativity and Jacobi identity.

For example, if

$$f = \begin{pmatrix} \cos x_2 \\ x_1 \end{pmatrix}, \qquad g = \begin{pmatrix} x_1 \\ 1 \end{pmatrix},$$

then we have

$$[f, g] = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \cos x_2 \\ x_1 \end{pmatrix} - \begin{pmatrix} 0 & -\sin x_2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos x_2 + \sin x_2 \\ -x_1 \end{pmatrix}.$$

Input/Output Linearization

Given a single–input single–output (SISO) system

$$\dot{x} = f(x) + g(x)\,u, \qquad y = h(x), \tag{5.9}$$

we want to formulate a linear–ODE relation between output y and a new input v. We will investigate (see [Isi89, SI89, Wil00]):

- How to generate a linear input/output relation.
- What are the internal dynamics and zero–dynamics associated with the input/output linearization?
- How to design stable controllers based on the I/O linearization.

This linearization method will be exact in a finite domain, rather than tangent as in the local linearization methods, which use Taylor series approximation. Nonlinear controller design using the technique is called exact feedback linearization.

Algorithm for Exact Feedback Linearization

We want to find a nonlinear compensator such that the closed–loop system is linear (see Figure 5.4). We will consider only affine SISO systems of the type (5.9), i.e, $\dot{x} = f(x) + g(x)\,u$, $y = h(x)$, and we will try to construct a *control law* of the form

$$u = p(x) + q(x)\,v, \tag{5.10}$$

where v is the setpoint, such that the *closed–loop nonlinear system*

$$\dot{x} = f(x) + g(x)\,p(x) + g(x)\,q(x)\,v, \qquad y = h(x),$$

is linear from command v to y.

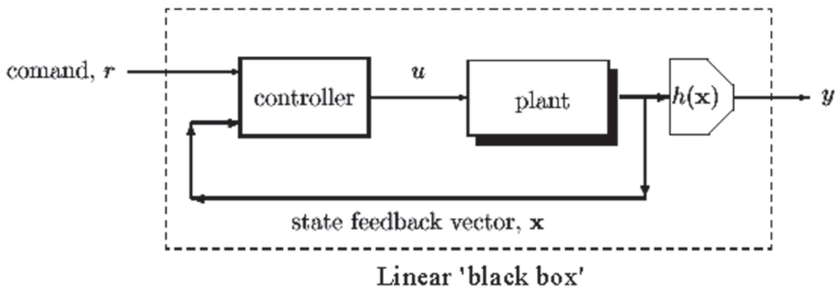

Fig. 5.4. Feedback linearization (see text for explanation).

The main idea behind the feedback linearization construction is to find a nonlinear change of coordinates which transforms the original system into one which is linear and controllable, in particular, a chain of integrators. The difficulty is finding the output function $h(x)$ which makes this construction possible.

We want to design an exact nonlinear feedback controller. Given the nonlinear affine system, $\dot{x} = f(x) + g(x)$, $y = h(x)$,.we want to find the controller functions $p(x)$ and $q(x)$. The unknown functions inside our controller (5.10) are given by:

$$p(x) = \frac{-\left(\mathcal{L}_f^r h(x) + \beta_1 \mathcal{L}_f^{r-1} h(x) + ... + \beta_{r-1} \mathcal{L}_f h(x) + \beta_r h(x)\right)}{\mathcal{L}_g \mathcal{L}_f^{r-1} h(x)},$$

$$q(x) = \frac{1}{\mathcal{L}_g \mathcal{L}_f^{r-1} h(x)}, \tag{5.11}$$

which are comprised of Lie derivatives, $\mathcal{L}_f h(x)$. Here, the *relative order*, r, is the smallest integer r such that $\mathcal{L}_g \mathcal{L}_f^{r-1} h(x) \neq 0$. For linear systems r is the difference between the number of poles and zeros.

To obtain the *desired response*, we choose the r parameters in the β polynomial to describe how the output will respond to the setpoint, v (pole–placement).

$$\frac{d^r y}{dt^r} + \beta_1 \frac{d^{r-1} y}{dt^{r-1}} + ... + \beta_{r-1} \frac{dy}{dt} + \beta_r y = v.$$

Here is the proposed algorithm [Isi89, SI89, Wil00]):

1. Given nonlinear SISO process, $\dot{x} = f(x, u)$, and output equation $y = h(x)$, then:
2. Calculate the relative order, r.
3. Choose an rth order desired linear response using pole–placement technique (i.e., select β). For this could be used a simple rth order low–pass filter such as a Butterworth filter.
4. Construct the exact linearized nonlinear controller (5.11), using Lie derivatives and perhaps a symbolic manipulator (Mathematica or Maple).
5. Close the loop and obtain a linear input–output black–box (see Figure 5.4).
6. Verify that the result is actually linear by comparing with the desired response.

Relative Degree

A nonlinear SISO system

$$\dot{x} = f(x) + g(x)\, u, \qquad y = h(x),$$

is said to have *relative degree* r at a point x_o if (see [Isi89, NS90])

1. $L_g L_f^k h(x) = 0$ for all x in a neighborhood of x_o and all $k < r - 1$; and

2. $L_g L_f^{r-1} h(x_o) \neq 0$.

For example, *controlled Van der Pol oscillator* has the state space form

$$\dot{x} = f(x) + g(x)\, u = \begin{bmatrix} x_2 \\ 2\omega\zeta\,(1-\mu x_1^2)\, x_2 - \omega^2 x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

Suppose the output function is chosen as $y = h(x) = x_1$. In this case we have

$$L_g h(x) = \frac{\partial h}{\partial x} g(x) = \begin{bmatrix} 1 \; 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0, \qquad \text{and}$$
$$L_f h(x) = \frac{\partial h}{\partial x} f(x) = \begin{bmatrix} 1 \; 0 \end{bmatrix} \begin{bmatrix} x_2 \\ 2\omega\zeta\,(1-\mu x_1^2)\, x_2 - \omega^2 x_1 \end{bmatrix} = x_2.$$

Moreover

$$L_g L_f h(x) = \frac{\partial (L_f h)}{\partial x} g(x) = \begin{bmatrix} 0 \; 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1,$$

and thus we see that the Vand der Pol oscillator system has relative degree 2 at any point x_o.

However, if the output function is, for instance $y = h(x) = \sin x_2$, then $L_g h(x) = \cos x_2$. The system has relative degree 1 at any point x_o, provided that $(x_o)_2 \neq (2k+1)\,\pi/2$. If the point x_o is such that this condition is violated, no relative degree can be defined.

As another example, consider a *linear system* in the state space form

$$\dot{x} = A\,x + B\,u, \qquad y = C\,x.$$

In this case, since $f(x) = A\,x$, $g(x) = B$, $h(x) = C\,x$, it is easily seen that

$$L_f^k h(x) = C\,A^k\,x, \qquad \text{and therefore,}$$
$$L_g L_f^k h(x) = C\,A^k\,B.$$

Thus, the integer r is characterized by the conditions

$$C\,A^k\,B = 0, \qquad \text{for all } k < r-1$$
$$C\,A^{r-1}\,B \neq 0, \qquad \text{otherwise.}$$

It is well–known that the integer satisfying these conditions is exactly equal to the *difference* between the degree of the denominator polynomial and the degree of the numerator polynomial of the transfer function

$$H(s) = C\,(sI - A)^{-1} B$$

of the system.

Approximative Feedback Linearization

Consider a SISO system

$$\dot{x} = f(x) + g(x)\,u, \tag{5.12}$$

where f and g are smooth vector–fields defined on a compact contractible region M of $\mathbb{R}^n$ containing the origin. (Typically, M is a closed ball in $\mathbb{R}^n$.) We assume that $f(0) = 0$, i.e., that the origin is an equilibrium for $\dot{x} = f(x)$. The classical problem of feedback linearization can be stated as follows: find in a neighborhood of the origin a smooth change of coordinates $z = \Phi(x)$ (a local diffeomorphism) and a smooth feedback law $u = k(x) + l(x)\,u_{new}$ such that the closed–loop system in the new coordinates with new control is linear,

$$\dot{z} = Az + B\,u_{new},$$

and controllable (see [BH96]). We usually require that $\Phi(0) = 0$. We assume that the system (5.12) has the *linear controllability* property

$$\dim(\mathrm{span}\{g, Ad_f g, ..., Ad_f^{n-1} g\}) = n, \qquad \text{for all } \; x \in M \tag{5.13}$$

(where Ad_f^i are iterated Lie brackets of f and g). We define the *characteristic distribution* for (5.12)

$$\mathcal{D} = \mathrm{span}\{g, Ad_f g, ..., Ad_f^{n-2} g\},$$

which is an $(n-1)$D smooth distribution by assumption of linear controllability (5.13). We call any nowhere vanishing $1-$form ω annihilating $\mathcal{D}$ a characteristic $1-$form for (5.12). All the characteristic $1-$forms for (5.12) can be represented as multiples of some fixed characteristic $1-$form ω_0 by a smooth nowhere vanishing function (zero–form) β. Suppose that there is a nonvanishing β so that $\beta\omega_0$ is exact, i.e., $\beta\omega_0 = d\alpha$ for some smooth function α, where d denotes the exterior derivative. Then ω_0 is called *integrable* and is called an integrating factor for ω_0. The following result is standard in nonlinear control: Suppose that the system (5.12) has the linear controllability property (5.13) on M. Let $\mathcal{D}$ be the characteristic distribution and ω_0 be a characteristic $1-$form for (5.12). The following statements are equivalent:

1. Equation (5.12) is feedback linearizable in a neighborhood of the origin in M;
2. D is involutive in a neighborhood of the origin in M; and
3. ω_0 is integrable in a neighborhood of the origin in M.

As is well known, a generic nonlinear system is not feedback linearizable for $n > 2$. However, in some cases, it may make sense to consider *approximate feedback linearization*.

Namely, if one can find a feedback linearizable system close to (5.12), there is hope that a control designed for the feedback linearizable system and applied to (5.12) will give satisfactory performance if the feedback linearizable system is close enough to (5.12). The first attempt in this direction goes back to [Kre84], where it was proposed to apply to (5.12) a change of variables and feedback that yield a system of the form

$$\dot{z} = Az + B\,u_{new} + O(z, u_{new}),$$

where the term $O(z, u_{new})$ contains higher–order terms. The aim was to make $O(z, u_{new})$ of as high order as possible. Then we can say that the system (5.12) is approximately feedback linearized in a small neighborhood of the origin. Later [HT93] introduced a new algorithm to achieve the same goal with fewer steps.

Another idea has been investigated in [HSK92]. Roughly speaking, the idea was to neglect nonlinearities in (5.12) responsible for the failure of the involutivity condition in above theorem. This approach happened to be successful in the ball–and–beam system, when neglect of centrifugal force acting on ball yielded a feedback linearizable system. Application of a control scheme designed for the system with centrifugal force neglected to the original system gave much better results than applying a control scheme based on classical Jacobian linearization. This approach has been further investigated in [XH94, XH95] for the purpose of approximate feedback linearization about the manifold of constant operating points. However, a general approach to deciding which nonlinearities should be neglected to get the best approximation has not been set forth.

All of the above–mentioned work dealt with applying a change of coordinates and a preliminary feedback so that the resulting system looks like linearizable part plus nonlinear terms of highest possible order around an equilibrium point or an equilibrium manifold. However, in many applications one requires a large region of operation for the nonlinearizable system. In such a case, demanding the nonlinear terms to be neglected to be of highest possible order may, in fact, be quite undesirable. One might prefer that the nonlinear terms to be neglected be small in a uniform sense over the region of operation. In tis section we propose an approach to approximate feedback linearization that uses a change of coordinates and a preliminary feedback to put a system (5.12) in a perturbed Brunovsky form,

$$\dot{z} = Az + B\,u_{new} + P(z) + Q(z)\,u_{new}), \tag{5.14}$$

where $P(z)$ and $Q(z)$ vanish at $z = 0$ and are 'small' on M. We obtain upper bounds on uniform norms of P and Q (depending on some measures of noninvolutivity of $\mathcal{D}$) on any compact, contractible M.

A different, indirect approach was presented in [BH96]. In this section, the authors present an approach for finding feedback linearizable systems that approximate a given SISO nonlinear system on a given compact region of the state–space. First, they it is shown that if the system is close to being involutive, then it is also close to being linearizable. Rather than working directly with the characteristic distribution of the system, the authors work with characteristic $1-$forms, i.e., with the $1-$forms annihilating the characteristic distribution. It is shown that homotopy operators can be used to decompose a given characteristic $1-$form into an exact and an antiexact part. The exact part is used to define a change of coordinates to a normal form that looks like a linearizable part plus nonlinear perturbation terms. The nonlinear terms in this normal form depend continuously on the antiexact part, and they vanish whenever the antiexact part does. Thus, the antiexact part of a given characteristic $1-$form is a measure of nonlinearizability of the system. If the nonlinear terms are small, by neglecting them we get a linearizable system approximating the original system. One can design control for the original system by designing it for the approximating linearizable system and applying it to the original one. We apply this approach for design of locally stabilizing feedback laws for nonlinear systems that are close to being linearizable.

Let us start with approximating characteristic $1-$forms by exact forms using *homotopy operators* (compare with (2.15) above). Namely, on any contractible region M one can define a linear operator H that satisfies

$$\omega = d(H\omega) + Hd\omega \tag{5.15}$$

for any form ω. The homotopy identity (5.15) allows to decompose any given $1-$form into the *exact part* $d(H\omega)$ and an 'error part' $\epsilon = Hd\omega$, which we call the *antiexact part of* ω. For given ω_0 annihilating $\mathcal{D}$ and a scaling factor β we define $\alpha_\beta = H\beta w_0$ and $\epsilon_\beta = Hd\beta w_0$. The $1-$form ϵ_β measures how exact $\omega_\beta = \beta w_0$ is. If it is zero, then ω_β is exact and the system (5.12) is linearizable, and the zero–form α_β and its first $n-1$ Lie derivatives along f are the new coordinates. In the case that ω_0 is not exactly integrable, i.e., when no exact integrating factor β exists, we choose β so that $d\beta w_0$ is *smallest* in some sense (because this also makes ϵ_β small). We call this β an *approximate integrating factor for* ω_0. We use the zero–form α_β and its first $n-1$ Lie derivatives along f as the new coordinates as in the linearizable case. In those new coordinates the system (5.12) is in the form

$$\dot{z} = Az + Bru + Bp + Eu,$$

where r and p are smooth functions, $r \neq 0$ around the origin, and the term E (the obstruction to linearizablity) depends linearly on ϵ_β and some of its derivatives. We choose $u = r^{-1}(u_{new} - p)$, where u_{new} is a new control variable. After this change of coordinates and control variable the system is of the form (5.14) with $Q = r^{-1}E$, $P = -r^{-1}pE$. We obtain estimates on the uniform norm of Q and P (via estimates on r, p, and E) in terms of the error $1-$form ϵ_β, for any fixed β, on any compact, contractible manifold M. Most important is that Q and P depend in a continuous way on ϵ_β and some of its derivatives, and they vanish whenever ϵ does (see [BH96]).

5.2.2 Controllability

Linear Controllability

A system is *controllable* if the set of all states it can reach from initial state $x_0 = x(0)$ at the fixed time $t = T$ contains a ball $\mathcal{B}$ around x_0. Again, a system is *small time locally controllable* (STLC) iff the ball $\mathcal{B}$ for $t \leq T$ contains a neighborhood of x_0.[3]

In the case of a linear system in the standard state–space form (see subsection (3.5.2) above)

$$\dot{x} = Ax + Bu, \tag{5.16}$$

where A is the $n \times n$ *state matrix* and B is the $m \times n$ *input matrix*, all controllability definitions coincide, i.e.,

$$0 \to x(T), \qquad x(0) \to 0, \qquad x(0) \to x(T),$$

where T is either fixed or free.

Rank condition states: System (5.16) is controllable iff the matrix

$$W_n = \left(B\, AB \ldots A^{n-1}B\right) \qquad \text{has full rank.}$$

In the case of nonlinear systems the corresponding result is obtained using the formalism of Lie brackets, as Lie algebra is to nonlinear systems as matrix algebra is to linear systems.

Nonlinear Controllability

Nonlinear MIMO–systems are generally described by differential equations of the form (see [Isi89, NS90, Goo98]):

$$\dot{x} = f(x) + g_i(x)\, u^i, \qquad (i = 1, ..., n), \tag{5.17}$$

defined on a smooth $n-$manifold M, where $x \in M$ represents the state of the control system, $f(x)$ and $g_i(x)$ are vector–fields on M and the u^i are control inputs, which belong to a set of *admissible controls*, $u^i \in U$. The system (5.17) is called *driftless*, or *kinematic*, or *control linear* if $f(x)$ is identically zero; otherwise, it is called a *system with drift*, and the vector–field $f(x)$ is called the *drift term*. The flow $\phi_t^g(x_0)$ represents the solution of the differential equation $\dot{x} = g(x)$ at time t starting from x_0. Geometric way to understand the *controllability* of the system (5.17) is to understand the geometry of the vector–fields $f(x)$ and $g_i(x)$.

[3] The above definition of controllability tells us only whether or not something can reach an open neighborhood of its starting point, but does not tell us how to do it. That is the point of the *trajectory generation*.

Example: Car–Parking Using Lie Brackets

In this popular example, the driver has two different transformations at his disposal. He can turn the steering wheel, or he can drive the car forward or back. Here, we specify the state of a car by four coordinates: the (x, y) coordinates of the center of the rear axle, the direction θ of the car, and the angle ϕ between the front wheels and the direction of the car. L is the constant length of the car. Therefore, the configuration manifold of the car is 4D, $M = (x, y, \theta, \phi)$.

Using (5.17), the driftless car kinematics can be defined as:

$$\dot{x} = g_1(x)\, u_1 + g_2(x)\, u_2, \tag{5.18}$$

with two vector–fields $g_1, g_2 \in \mathcal{X}^k(M)$.

The infinitesimal transformations will be the vector–fields

$$g_1(x) \equiv \text{DRIVE} = \cos\theta \frac{\partial}{\partial x} + \sin\theta \frac{\partial}{\partial y} + \frac{\tan\phi}{L} \frac{\partial}{\partial \theta} \equiv \begin{pmatrix} \cos\theta \\ \sin\theta \\ \frac{1}{L}\tan\phi \\ 0 \end{pmatrix},$$

$$\text{and} \qquad g_2(x) \equiv \text{STEER} = \frac{\partial}{\partial \phi} \equiv \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Now, STEER and DRIVE do not commute; otherwise we could do all your steering at home before driving of on a trip. Therefore, we have a Lie bracket

$$[g_2, g_1] \equiv [\text{STEER}, \text{DRIVE}] = \frac{1}{L\cos^2\phi} \frac{\partial}{\partial \theta} \equiv \text{ROTATE}.$$

The operation $[g_2, g_1] \equiv$ ROTATE $\equiv$ [STEER,DRIVE] is the infinitesimal version of the sequence of transformations: steer, drive, steer back, and drive back, i.e.,

$$\{\text{STEER}, \text{DRIVE}, \text{STEER}^{-1}, \text{DRIVE}^{-1}\}.$$

Now, ROTATE can get us out of some parking spaces, but not tight ones: we may not have enough room to ROTATE out. The usual tight parking space restricts the DRIVE transformation, but not STEER. A truly tight parking space restricts STEER as well by putting your front wheels against the curb.

Fortunately, there is still another commutator available:

$$[g_1, [g_2, g_1]] \equiv [\text{DRIVE}, [\text{STEER}, \text{DRIVE}]] = [[g_1, g_2], g_1] \equiv$$
$$[\text{DRIVE}, \text{ROTATE}] = \frac{1}{L\cos^2\phi}\left(\sin\theta \frac{\partial}{\partial x} - \cos\theta \frac{\partial}{\partial y}\right) \equiv \text{SLIDE}.$$

The operation $[[g_1, g_2], g_1] \equiv$ SLIDE $\equiv$ [DRIVE,ROTATE] is a displacement at right angles to the car, and can get us out of any parking place. We just need to remember to steer, drive, steer back, drive some more, steer, drive back, steer back, and drive back:

$$\{\text{STEER}, \text{DRIVE}, \text{STEER}^{-1}, \text{DRIVE}, \text{STEER}, \text{DRIVE}^{-1}, \text{STEER}^{-1}, \text{DRIVE}^{-1}\}.$$

We have to reverse steer in the middle of the parking place. This is not intuitive, and no doubt is part of the problem with parallel parking.

Thus from only two controls u_1 and u_2 we can form the vector–fields DRIVE $\equiv g_1$, STEER $\equiv g_2$, ROTATE $\equiv [g_2, g_1]$, and SLIDE $\equiv [[g_1, g_2], g_1]$, allowing us to move anywhere in the configuration manifold M. The car kinematics $\dot{x} = g_1 u_1 + g_2 u_2$ is thus expanded as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \text{DRIVE} \cdot u_1 \ + \text{STEER} \cdot u_2 \ \equiv \begin{pmatrix} \cos\theta \\ \sin\theta \\ \frac{1}{L}\tan\phi \\ 0 \end{pmatrix} \cdot u_1 \ + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \cdot u_2 \ .$$

The *parking theorem* says: One can get out of any parking lot that is larger than the car.

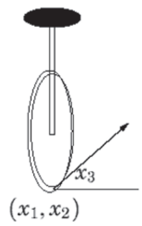

Fig. 5.5. The unicycle problem (see text for explanation).

The Unicycle Example

Now, consider the unicycle example (see Figure 5.5). Here we have

$$g_1 = \begin{pmatrix} \cos x_3 \\ \sin x_3 \\ 0 \end{pmatrix}, \qquad g_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad [g_1, g_2] = \begin{pmatrix} \sin x_3 \\ -\cos x_3 \\ 0 \end{pmatrix}.$$

The unicycle system is full rank and therefore controllable.

Controllability Condition

Nonlinear controllability is an extension of linear controllability. The nonlinear MIMO system

$$\dot{x} = f(x) + g(x)\,u \qquad \text{is controllable}$$

if the set of vector–fields $\{g, [f, g], ..., [f^{n-1}, g]\}$ is independent.

For example, for the kinematic car system of the form (5.18), the *nonlinear controllability criterion* reads: If the Lie bracket tree:

$g_1,\ g_2,\ [g_1, g_2],\ [[g_1, g_2], g_1],\ [[g_1, g_2], g_2],\ [[[g_1, g_2], g_1], g_1],\ [[[g_1, g_2], g_1], g_2],$ $[[[g_1, g_2], g_2], g_1],\ [[[g_1, g_2], g_2], g_2], ...$

– has *full rank* then the system is *controllable* [Isi89, NS90, Goo98]. In this case the combined input

$$(u_1, u_2) = \begin{cases} (1, 0), & t \in [0, \varepsilon] \\ (0, 1), & t \in [\varepsilon, 2\varepsilon] \\ (-1, 0), & t \in [2\varepsilon, 3\varepsilon] \\ (0, -1), & t \in [3\varepsilon, 4\varepsilon] \end{cases}$$

gives the motion $x(4\varepsilon) = x(0) + \varepsilon^2\,[g_1, g_2] + O(\varepsilon^3)$, with the flow given by (see (2.20) below)

$$F_t^{[g_1,g_2]} = \lim_{n\to\infty} \left(F^{-g_2}_{\sqrt{t/n}} F^{-g_1}_{\sqrt{t/n}} F^{g_2}_{\sqrt{t/n}} F^{g_1}_{\sqrt{t/n}} \right)^n .$$

Distributions

In control theory, the set of all possible directions in which the system can move, or the set of all points the system can reach, is of obvious fundamental importance. Geometrically, this is related to *distributions*.

A *distribution* $\Delta \subset \mathcal{X}^k(M)$ on the manifold M is a subbundle of its tangent bundle TM, which assigns a subspace of the tangent space T_xM to each point $x \in M$ in a smooth way. The dimension of $\Delta(x)$ over $\mathbb{R}$ at a point $x \in M$ is called the rank of Δ at x.

A distribution Δ is involutive if, for any two vector–fields $X, Y \in \Delta$, their Lie bracket $[X, Y] \in \Delta$.

A function $f \in C^k(M)$ is called an *integral* of Δ if $df(x) \in \Delta^0(x)$ for each $x \in M$. An *integral manifold* of Δ is a submanifold N of M such that $T_xN \subset \Delta(x)$ for each $x \in N$. A distribution Δ is *integrable* if, for any $x \in M$, there is a submanifold $N \subset M$, whose dimension is the same as the rank of Δ at x,.containing x such that the tangent bundle, TN, is exactly Δ restricted to N, i.e., $TN = \Delta|_N$. Such a submanifold is called the *maximal integral manifold* through x.

It is natural to consider distributions generated by the vector–fields appearing in the sequence of flows (2.19). In this case, consider the distribution defined by

$$\Delta = \mathrm{span}\{f; g_1 ... g_m\},$$

where the span is taken over the set of smooth real–valued functions. Denote by $\bar{\Delta}$ the *involutive closure* of the distribution Δ, which is the closure of Δ

under bracketing. Then, $\bar{\Delta}$ is the smallest subalgebra of $\mathcal{X}^k(M)$ which contains $\{f; g_1...g_m\}$. We will often need to 'add' distributions. Since distributions are, pointwise, vector spaces, define the sum of two distributions,

$$(\Delta_1 + \Delta_2)(x) = \Delta_1(x) + \Delta_2(x).$$

Similarly, define the intersection

$$(\Delta_1 \cap \Delta_2)(x) = \Delta_1(x) \cap \Delta_2(x).$$

More generally, we can arrive at a distribution via a *family of vector–fields*, which is simply a subset $\mathcal{V} \subset \mathcal{X}^k(M)$. Given a family of vector–fields $\mathcal{V}$, we may define a distribution on M by

$$\Delta_{\mathcal{V}}(x) = \langle X(x) | X \in \mathcal{V} \rangle_{\mathbb{R}}.$$

Since $\mathcal{X}^k(M)$ is a Lie algebra, we may ask for the smallest Lie subalgebra of $\mathcal{X}^k(M)$ which contains a family of vector–fields $\mathcal{V}$. It will be denoted as $\overline{Lie}(\mathcal{V})$, and will be represented by the set of vector–fields on M generated by repeated Lie brackets of elements in $\mathcal{V}$. Let $\mathcal{V}^{(0)} = \mathcal{V}$ and then iteratively define a sequence of families of vector–fields by

$$\mathcal{V}^{(i+1)} = \mathcal{V}^{(i)} \cup \{[X, Y] | X \in \mathcal{V}^{(0)} = \mathcal{V} \text{ and } Y \in \mathcal{V}^{(i)}\}.$$

Now, every element of $\overline{Lie}(\mathcal{V})$ is a linear combination of repeated Lie brackets of the form

$$[Z_k, [Z_{k-1}, [\cdots, [Z_2, Z_1] \cdots]]]$$

where $Z_i \in \mathcal{V}$ for $i = 1, ..., k$.

Foliations

Related to integrable distributions are foliations.

Frobenius' theorem asserts that integrability and involutivity are equivalent, at least locally. Thus, associated with an involutive distribution is a partition Φ of M into disjoint connected immersed submanifolds called *leaves*. This partition Φ is called a *foliation*. More precisely, a foliation $\mathcal{F}$ of a smooth manifold M is a collection of disjoint immersed submanifolds of M whose disjoint union equals M. Each connected submanifold of $\mathcal{F}$ is called a *leaf* of the foliation. Given an integrable distribution Δ, the collection of maximal integral manifolds for Δ defines a foliation on M, denoted by $\mathcal{F}_D$.

A foliation $\mathcal{F}$ of M defines an equivalence relation on M whereby two points in M are equivalent if they lie in the same leaf of $\mathcal{F}$. The set of equivalence classes is denoted $M/\mathcal{F}$ and is called the *leaf space* of $\mathcal{F}$. A foliation $\mathcal{F}$ is said to be simple if $M/\mathcal{F}$ inherits a manifold structure so that the projection from M to $M/\mathcal{F}$ is a surjective submersion.

In control theory, foliation leaves are related to the set of points that a control system can reach starting from a given initial condition. A foliation Φ of M defines an equivalence relation on M whereby two points in M are equivalent if they lie in the same leaf of Φ. The set of equivalence classes is denoted M/Φ and is called the *leaf space* of Φ.

Philip Hall Basis

Given a set of vector–fields $\{g_1 ... g_m\}$, define the *length* of a *Lie product* as

$$l(g_i) = 1, \qquad l([A, B]) = l(A) + l(B), \qquad (\text{for} \;\; i = 1, ..., m),$$

where A and B may be Lie products. A *Philip Hall basis* is an ordered set of Lie products $H = \{B_i\}$ satisfying:

1. $g_i \in H, \qquad (i = 1, ..., m)$;
2. If $l(B_i) < l(B_j)$, then $B_i < B_j$; and
3. $[B_i, B_j] \in H$ iff
   (a) $B_i, B_j \in H$ and $B_i < B_j$, and
   (b) either $B_j = g_k$ for some k or $B_j = [B_l, B_r]$ with $B_l, B_r \in H$ and $B_l \leq B_i$.

Essentially, the ordering aspect of the Philip Hall basis vectors accounts for skew symmetry and Jacobi identity to determine a basis.

5.3 Modern Control Techniques for Mechanical Systems

In this section we present modern control techniques for mechanical systems, as used in modern biomechanics research. Much of the existing work on control of mechanical systems has relied on the presence of specific structure. The most common examples of the types of structure assumed are symmetry (conservation laws) and constraints. While it may seem counter–intuitive that constraints may help in control theory, this is sometimes in fact the case. The reason is that the constraints provide extra forces (forces of constraint) which can be used to advantage. probably, the most interesting work is done from the Lagrangian (respectively Hamiltonian) perspective where we study systems whose Lagrangians are 'kinetic energy minus potential energy' (resp. 'kinetic energy plus potential energy'). For these *simple mechanical control systems*, the controllability questions are different than those typically asked in nonlinear control theory. In particular, one is often more interested in what happens to configurations rather than states, which are configurations and velocities (resp. momenta) for these systems (see [Lew95, LM97]).

5.3.1 Abstract Control System

In general, a nonlinear control system Σ can be represented as a triple (Σ, M, f), where M is the system's *state–space* manifold with the tangent bundle TM and the general fibre bundle E, and f is a smooth map, such that the following bundle diagram commutes [Man98]

$$\begin{tikzcd} E \arrow[rr, "\psi"] \arrow[dr, "\pi"'] & & TM \arrow[dl, "\pi_M"] \\ & M & \end{tikzcd}$$

where $\psi : (x, u) \mapsto (x, f(x, u))$, π_M is the natural projection of TM on M, the projection $\pi : E \to M$ is a smooth fibre bundle, and the fibers of E represent the *input spaces*. If one chooses fibre–respecting coordinates (x, u) for E, then locally this definition reduces to $\psi : (x, u) \mapsto (x, \psi(x, u))$, i.e.,

$$\dot{x} = \psi(x, u).$$

The specific form of the map ψ, usually used in nonlinear control, is $\psi : (x, u) \mapsto (x, f(x) + g(x, u))$, with $g(x, 0) = 0$, producing standard nonlinear system equation

$$\dot{x} = f(x) + g(x, u).$$

5.3.2 Controllability of a Linear Control System

Consider a linear biomechanical control system:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{5.19}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $A \in L(\mathbb{R}^n, \mathbb{R}^n)$, and $B \in L(\mathbb{R}^m, \mathbb{R}^n)$. One should think of $t \mapsto u(t)$ as being a specified input signal, i.e., a function on the certain time interval, $[0, T]$. Now, control theory wants to design the signal to make the *state* $t \mapsto x(t)$ do what we want. What this is may vary, depending on the situation at hand. For example, one may want to steer from an initial state x^i to a final state x_f, perhaps in an optimal way. Or, one may wish to design $u : \mathbb{R}^n \to \mathbb{R}^m$ so that some state, perhaps $x = 0$, is stable for the dynamical system $\dot{x}(t) = Ax + Bu(x)$, which is called *state feedback* (often one asks that u be linear). One could also design u to be a function of both x and t, etc.

One of the basic control questions is *controllability*, which comes in many guises. Basically we are asking for 'reachable' points. In particular,

$$\mathcal{R}(0) = \mathrm{span}_{\mathbb{R}}\{[B|AB|...|A^{n-1}B]\},$$

which is the smallest $A-$invariant subspace containing $\mathrm{Im}(B)$, denotes the set of points reachable from $0 \in \mathbb{R}^n$. For the linear system (5.19), the basic controllability questions have definite answers. We want to do something similar for a class of simple mechanical systems [Lew95, LM97].

5.3.3 Affine Control System and Local Controllability

The nonlinear control system that we most often consider in human–like biomechanics has state–space M, a smooth $n-$manifold, and is *affine* in the controls. Thus it has the form (see [Lew95, LM97])

$$\dot{x} = f(x) + u^a g_a(x), \qquad (x \in M), \tag{5.20}$$

where $f, g_1, ..., g_m$ are vector–fields on M. The *drift* vector–field $f = f(x)$ describes how the system would evolve in the absence of any inputs. Each of the *control* vector–fields $g_1, ..., g_m$ specifies a direction in which one can supply actuation. To fully specify the control system properly, one should also specify the type of control action to be considered. Here we consider our controls to be taken from the set: $U = \{u : \mathbb{R} \to \mathbb{R}^m \,|\, u \text{ is piecewise constant}\}$. This class of controls is sufficient to deal with all analytic control systems. More generally, one may wish to consider measurable functions which take their values in a subset of $\mathbb{R}^m$.

Given an *affine control system* (5.20), it is possible to define a family of vector–fields on M by: $V_\Sigma = \{f + u^a g_a \,|\, u \in \mathbb{R}^m\}$.

A solution of the system (5.20) is a pair (γ, u), where $\gamma : [0, T] \to M$ is a piecewise smooth curve on M and $u \in U$ such that

$$\dot{\gamma}(t) = f(\gamma(t)) + u^a(t)\, g_a(\gamma(t)), \qquad \text{for each} \quad t \in [0, T].$$

The *reachable set* from x_0 in time T is

$$\begin{aligned} \mathcal{R}(x_0, T) = \{x | \exists \gamma : [0, T] \to M \quad &\text{and} \\ u \; : \; [0, T] \to \mathbb{R}^m \quad &\text{satisfying} \;\; (5.20) \\ \text{with} \;\; \gamma(0) = x_0 \quad \text{and} \quad &\gamma(T) = x\}. \end{aligned}$$

Note that since the system has drift f, when we reach the point $\gamma(T)$ we will not remain there if this is not an equilibrium point for f. Also, we have, $\mathcal{R}(x_0, \leq T) = \cup_{0<t\leq T} \mathcal{R}(x_0, T)$.

Let $x_0 \in M$, let V be a neighborhood of x_0, and let $T > 0$. We say that equation (5.20) represents a *locally accessible system* at x_0 if $\mathcal{R}(x_0, \leq T)$ contains an open subset of M for each V and for each T sufficiently small. Furthermore, we say that the system (5.20) is *small–time local controllability* (STLC, see [Sus83, Sus87]), if it is locally accessible and if x_0 is in the interior of $\mathcal{R}(x_0, \leq T)$ for each V and for each T sufficiently small.

5.3.4 Lagrangian Control Systems

Simple Mechanical Control Systems

As a motivation/prototype of a simple mechanical control system, consider a simple *robotic leg* (see Figure 5.6), in which inputs are: (1) an internal torque

F^1 moving the leg relative to the body and (2) a force F^2 extending the leg. This system is 'controllable' in the sense that, starting from rest, one can reach any configuration from a given initial configuration. However, as a traditional control system, it is not controllable because of conservation of angular momentum. If one asks for the *states* (i.e., configurations and velocities) reachable from configurations with zero initial velocity, one finds that not all states are reachable. This is a consequence of the fact that angular momentum is conserved, even with inputs. Thus if one starts with zero momentum, the momentum will remain zero (this is what enables one to treat the system as *nonholonomic*). Nevertheless, all configurations are accessible. This suggests that the question of controllability is different depending on whether one is interested in configurations or states. We will be mainly interested in reachable configurations. Considering the system with just one of the two possible input forces is also interesting. In the case where we are just allowed to use F^2, the possible motions are quite simple; one can only move the ball on the leg back and forth. With just the force F^1 available, things are a bit more complicated. But, for example, one can still say that no matter how you apply the force, the ball with never move 'inwards' [Lew95, LM97].

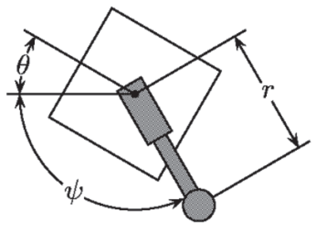

Fig. 5.6. A simple robotic leg (see text for explanation).

In general, simple mechanical control systems are characterized by:

- An nD configuration manifold M;
- A Riemannian metric g on M;
- A potential energy function V on M; and
- m linearly independent $1-$forms, $F^1, ..., F^m$ on M (input forces; e.g., in the case of the simple robotic leg, $F^1 = d\theta - d\psi$ and $F^2 = dr$).

When we say these systems are not amenable to liberalization–based methods, we mean that their liberalizations at zero velocity are not controllable, and that they are not feedback linearizable. This makes simple mechanical control systems a non–trivial class of nonlinear control systems, especially from the point of view of control design.

As a basic example to start with, consider a *planar rigid body* (see Figure 5.7), with coordinates (x, y, θ). Inputs are (1) force pointing towards center of mass, $F^1 = \cos\theta dx + \sin\theta dy$, (2) force orthogonal to line to center of mass,

$F^2 = -\sin\theta dx + \cos\theta dy - hd\theta$, and (3) torque at center of mass $F^3 = d\theta$. The planar rigid body, although seemingly quite simple, can be actually interesting. Clearly, if one uses all three inputs, the system is *fully actuated*, and so boring for investigating reachable configurations. But if one takes various combinations of one or two inputs, one gets a pretty nice sampling of what can happen for these systems. For example, all possible combinations of two inputs allow one to reach all configurations. Using F^1 or F^3 alone give simple, 1D reachable sets, similar to using F^2 for the robotic leg (as we are always starting with zero initial velocity). However, if one is allowed to only use F^2, then it is not quite clear what to expect, at least just on the basis of intuition.

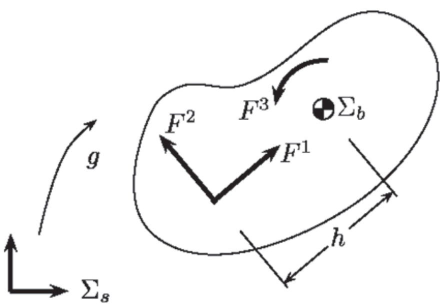

Fig. 5.7. Coordinate systems of a planar rigid body.

It turns out that our simplifying assumptions, i.e., zero initial velocity and restriction of our interest to configurations (i.e., as all problem data is on M, we expect answers to be describable using data on M), makes our task much simpler. In fact, the computations without these assumptions have been attempted, but have yet to yield coherent answers.

Now, we are interested in how do the input 1−forms $F^1, ..., F^m$ interact with the unforced mechanics of the system as described by the kinetic energy Riemannian metric. That is, what is the analogue of linear system's 'the smallest $A-$invariant subspace containing $\mathrm{Im}(B)$' – for simple mechanical control systems?

Motion and Controllability in Affine Connections

If we start with the local Riemannian metric form $g \longmapsto g_{ij}(q)\, dq^i dq^j$, then we have a kinetic energy Lagrangian $L(q,v) = g_{ij}(q)\, \dot{q}^i \dot{q}^j$, and consequently the Euler–Lagrange equations (3.4) are given as [Lew98]

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}^i}\right) - \frac{\partial L}{\partial q^i} \equiv g_{ij}\ddot{q}^j + \left(\frac{\partial g_{ij}}{\partial q^k} - \frac{1}{2}\frac{\partial g_{jk}}{\partial q^i}\right)\dot{q}^j \dot{q}^k = u_a F_i^a, \qquad (i = 1, ..., n).$$

Now multiply this by g^{li} and take the symmetric part of the coefficient of $\dot{q}^j \dot{q}^k$ to get $\ddot{q}^l + \Gamma^l_{jk}\, \dot{q}^j \dot{q}^k = u^a Y^l_a$, $l = 1, ..., n$, where $\Gamma^i_{jk}$are the Christoffel symbols

(3.9) for the Levi–Civita connection ∇ (see (2.5.1) above). So, the equations of motion are

$$\nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) = u^a(t)\, Y_a\left(\gamma(t)\right),$$

where $Y_a = (F^a)^\sharp$, $a = 1, ..., m$. Here $^\sharp : T^*M \to TM$ is the 'musical' isomorphism associated with the Riemannian metric g.

Now, there is nothing to be gained by using a Levi–Civita connection, or by assuming that the vector–fields come from $1-$forms. At this point, perhaps the generalization to an arbitrary affine connection seems like a senseless abstraction. However, as we shall see, this abstraction allows us to include another large class of mechanical control systems. So we will study the control system

$$\nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) = u^a(t)\, Y_a\left(\gamma(t)\right) [+Y_0\left(\gamma(t)\right)], \tag{5.21}$$

with ∇ a general affine connection on M, and $Y_1 ..., Y_m$ linearly independent vector–fields on M. The 'optional' term $Y_0 = Y_0\left(\gamma(t)\right)$ in (5.21) indicates how potential energy may be added. In this case $Y_0 = -\operatorname{grad} V$ (however, one looses nothing by considering a general vector–field instead of a gradient) [Lew98].

A *solution* to (5.21) is a pair (γ, u) satisfying (5.21) where $\gamma : [0, T] \to M$ is a curve and $u : [0; T] \to \mathbb{R}^m$ is bounded and measurable.

Let U be a neighborhood of $q_0 \in M$ and denote by $\mathcal{R}^U_M(q_0, T)$ those points in M for which there exists a solution (γ, u) with the following properties:

1. $\gamma(t) \in U$ for $t \in [0, T]$;
2. $\dot{\gamma}(0) = 0_q$; and
3. $\gamma(T) \in T_q M$.

Also $\mathcal{R}^U_M(q_0, \leq T) = \cup_{0 \leq t \leq T} \mathcal{R}^U_M(q_0, t)$. Now, regarding the local controllability, we are only interested in points which can be reached without taking 'large excursions'. Control problems which are local in this way have the advantage that they can be characterized by Lie brackets. So, we want to describe our *reachable set* $\mathcal{R}^U_M(q, \leq T)$ for the simple mechanical control system (5.21). The system (5.21) is *locally configuration accessible* (LCA) at q if there exists $T > 0$ so that $\mathcal{R}^U_M(q, \leq t)$ contains a non–empty open subset of M for each neighborhood U of q and each $t \in]0, T]$. Also, (5.21) is *locally configuration controllable* (LCC) at q if there exists $T > 0$ so that $\mathcal{R}^U_M(q, \leq t)$ contains a neighborhood of q for each neighborhood U of q and each $t \in]0, T]$. Although sound very similar, the notions of local configuration accessibility and local configuration controllability are genuinely different (see Figure 5.8). Indeed, one need only look at the example of the robotic leg with the F^1 input. In this example one may show that the system is LCA, but is not LCC [Lew98].

Local Configuration Accessibility

The accessibility problem is solved by looking at Lie brackets. For this we need to recall the definition of the *vertical lift* [Lew98]:

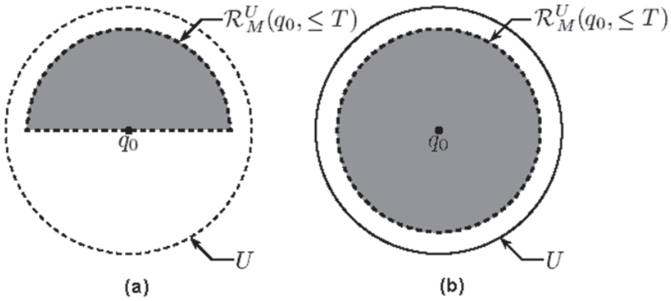

Fig. 5.8. Difference between the notions of local configuration accessibility (a), and local configuration controllability (b).

$$\mathrm{verlift}(Y(v_q)) = \left.\frac{d}{dt}\right|_{t=0}(v_q + tY(q)),$$

in local coordinates, if $Y = Y^i\partial_{q^i}$, then $\mathrm{verlift}(Y) = Y^i\partial_{v^i}$. Now we can rewrite (5.21) in the first–order form:

$$\dot{v} = Z(v) + u^a\,\mathrm{verlift}(Y_a(v)),$$

where Z is the geodesic spray for ∇.

We evaluate all brackets at 0_q (recall that $T_{0_q}TM \simeq T_qM \oplus T_qM$). Here, the first component we think of as being the 'horizontal' bit which is tangent to the zero section in TM, and we think of the second component as being the 'vertical' bit which is the tangent space to the fibre of $\tau_M : TM \to M$.

To get an answer to the local configuration accessibility problem, we employ standard nonlinear control techniques involving Lie brackets. Doing so gives us our first look at the *symmetric product*, $\langle X : Y\rangle = \nabla_X Y + \nabla_Y X$. Our sample brackets suggest that perhaps the only things which appear in the bracket computations are symmetric products and Lie brackets of the input vector–fields $Y_1, ..., Y_m$.

Here are some sample brackets:

(i) $[Z, \mathrm{verlift}(Y_a)](0_q) = (-Y_a(q), 0)$;
(ii) $[\mathrm{verlift}(Y_a), [Z, \mathrm{verlift}(Y_b)]](0_q) = (0, \langle Y_a : Y_b\rangle(q))$;
(iii) $[[Z, \mathrm{verlift}(Y_a)], [Z, \mathrm{verlift}(Y_b)]](0_q) = ([Y_a, Y_b](q), 0)$.

Now, let C_{ver} be the closure of $\mathrm{span}\{Y_1, ..., Y_m\}$ under symmetric product. Also, let C_{hor} be the closure of C_{ver} under Lie bracket. So, we assume C_{ver} and C_{hor} to be distributions (i.e., of constant rank) on M. The closure of $\mathrm{span}\{Z, \mathrm{verlift}(Y_1), ..., \mathrm{verlift}(Y_m)\}$ under Lie bracket, when evaluated at 0_q, is then the distribution

$$q \mapsto C_{hor}(q) \oplus C_{ver}(q) \subset T_qM \oplus T_qM.$$

Proving that the involutive closure of span$\{Z, \mathrm{verlift}(Y_1), ..., \mathrm{verlift}(Y_m)\}$ is equal at 0_q to $C_{hor}(q) \oplus C_{ver}(q)$ is a matter of computing brackets, samples of which are given above, and seeing the patterns to suggest an inductive proof. The brackets for these systems are very structured. For example, the brackets of input vector–fields are identically zero. Many other brackets vanish identically, and many more vanish when evaluated at 0_q.

C_{hor} is integrable: let Λ_q be the maximal integral manifold through $q \in M$. Then, $\mathcal{R}_M^U(q, \leq T)$ is contained in Λ_q, and $\mathcal{R}_M^U(q, \leq T)$ contains a non–empty open subset of Λ_q. In particular, if $rank(C_{hor}) = n$ then (5.21) is LCA [Lew95, LM97]. This theorem gives a 'computable' description of the reachable sets (in the sense that we can compute Λ_q by solving some over–determined nonlinear PDE's). But it does not give the kind of insight that we had with the 'smallest $A-$invariant subspace containing Im(B)'.

Recall that a submanifold N of M is *totally geodesic* if every geodesic with initial velocity tangent to N remains on N. This can be weakened to distributions: a distribution D on M is *geodesically invariant* if for every geodesic $\gamma : [0, T] \to M$, $\dot{\gamma}(0) \in D_{\gamma(0)}$ implies $\dot{\gamma}(t) \in D_{\gamma(t)}$ for $t \in]0, T]$.

D is geodesically invariant i it is closed under symmetric product [Lew98]. This theorem says that the symmetric product plays for geodesically invariant distributions the same role the Lie bracket plays for integrable distributions. This result was key in providing the geometric description of the reachable configurations.

An integrable distribution is *geodesically generated distribution* if it is the involutive closure of a geodesically invariant distribution. This basically means that one may reach all points on a leaf with geodesics lying in some subdistribution. The picture one should have in mind with the geometry of the reachable sets is a foliation of M by geodesically generated (immersed) submanifolds onto which the control system restricts if the initial velocity is zero. The idea is that when we start with zero velocity we remain on leaves of the foliation defined by C_{hor} [LM97]. Note that for cases when the affine connection possesses no geodesically invariant distributions, the system (5.21) is automatically LCA. This is true, for example, of S^2 with the affine connection associated with its round metric.

Clearly C_{ver} is the smallest geodesically invariant distribution containing span$\{Y_1, ..., Y_m\}$. Also, C_{hor} is geodesically generated by span span$\{Y_1, ..., Y_m\}$. Thus $\mathcal{R}_M^U$ is contained in, and contains a non–empty open subset of, the distribution geodesically generated by span$\{Y_1, ..., Y_m\}$. Note that the pretty decomposition we have for systems with no potential energy does not exist at this point for systems with potential energy.

Local Configuration Controllability

The problem of configuration controllability is harder than the one of configuration accessibility. Following [LM99], we will call a symmetric product in $\{Y_1, ..., Y_m\}$ *bad* if it contains an even number of each of the input vector–fields.

Otherwise we will call it *good*. The *degree* is the total number of vector–fields. For example, $\langle\langle Y_a : Y_b\rangle : \langle Y_a : Y_b\rangle\rangle$ is bad and of degree 4, and $\langle Y_a : \langle Y_b : Y_b\rangle\rangle$ is good and of degree 3. If each bad symmetric product at q is a linear combination of good symmetric products of lower degree, then (5.21) is LCC at q.

Now, the single–input case can be solved completely: The system (5.21) with $m = 1$ is LCC iff $\dim(M) = 1$ [LM99].

Systems With Nonholonomic Constraints

Let us now add to the data a distribution D defining nonholonomic constraints. One of the interesting things about this affine connection approach is that we can easily integrate into our framework systems with nonholonomic constraints. As a simple example, consider a *rolling disk* (see Figure 5.9), with two inputs: (1) a 'rolling' torque, $F^1 = d\theta$ and (2) a 'spinning' torque, $F^2 = d\phi$. It can be analyzed as a nonholonomic system (see [Lew99]).

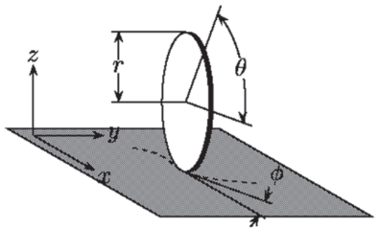

Fig. 5.9. Rolling disk problem (see text for explanation).

The control equations for a simple mechanical control system with constraints are:

$$\nabla_{\dot\gamma(t)}\dot\gamma(t) = \lambda(t) + u^a(t)\, Y_a\left(\gamma(t)\right)\left[-\operatorname{grad} V\left(\gamma(t)\right)\right], \qquad \dot\gamma(t) \in D_{\gamma(t)},$$

where $\lambda(t) \in D^{\perp}_{\gamma(t)}$ are *Lagrange multipliers*.

Examples

1. Recall that for the simple robotic leg (Figure 5.6) above, Y_1 was internal torque and Y_2 was extension force. Now, in the following three cases:
(i) both inputs active – this system is LCA and LCC (satisfies sufficient condition);
(ii) Y_1 only, it is LCA but not LCC; and
(iii) Y_2 only, it is not LCA.
In theses three cases, C_{hor} is generated by the following linearly independent vector–fields:

(i) both inputs: $\{Y_1, Y_2, [Y_1, Y_2]\}$;
(ii) Y_1 only: $\{Y_1, \langle Y_1 : Y_1\rangle, \langle Y_1 : \langle Y_1 : Y_1\rangle\rangle\}$; and
(iii) Y_2 only: $\langle Y_2\rangle$.
Recall that with both inputs the system was not accessible in TM as a consequence of conservation of angular momentum. With the input Y_2 only, the control system behaves very simply when given zero initial velocity. The ball on the end of the leg just gets moved back and forth. This reflects the foliation of M by the maximal integral manifolds of C_{hor}, which are evidently 1D in this case. With the Y_1 input, recall that the ball will always go 'outwards' no matter what one does with the input. Thus the system is not LCC. But apparently (since $rank(C_{hor}) = \dim(M)$) one can reach a non–empty open subset of M. The behavior exhibited in this case is typical of what one can expect for single–input systems with no potential energy.

2. For the planar rigid body (Figure 5.7) above, we have the following five cases:
(i) Y_1 and Y_2 active, this system is LCA and LCC (satisfies sufficient condition);
(ii) Y_1 and Y_3, it is LCA and LCC (satisfies sufficient condition);
(iii) Y_1 only or Y_3 only, not LCA;
(iv) Y_2 only, LCA but not LCC; and
(v) Y_2 and Y_3: LCA and LCC (fails sufficient condition).

Now, with the inputs Y_1 or Y_3 alone, the motion of the system is simple. In the first case the body moves along the line connecting the point of application of the force and the center of mass, and in the other case the body simply rotates. The equations in (x, y, θ) coordinates are

$$\ddot{x} = \frac{\cos\theta}{m}u^1 - \frac{\sin\theta}{m}u^2, \qquad \ddot{y} = \frac{\sin\theta}{m}u^1 + \frac{\cos\theta}{m}u^2, \qquad \ddot{\theta} = \frac{1}{J}\left(u^3 - hu^2\right),$$

which illustrates that the $\theta-$equation decouples when only Y_3 is applied. We make a change of coordinates for the case where we have only Y_1: $(\xi, \eta, \psi) = (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta, \theta)$. In these coordinates we have

$$\ddot{\xi} - 2\dot{\eta}\dot{\psi} - \xi\dot{\psi}^2 = \frac{1}{m}u^1, \qquad \ddot{\eta} + 2\dot{\xi}\dot{\psi} - \eta\dot{\psi}^2 = 0, \qquad \dot{\psi} = 0,$$

which illustrates the decoupling of the $\xi-$equation in this case.

C_{hor} has the following generators:
(i) Y_1 and Y_2: $\{Y_1, Y_2, [Y_1, Y_2]\}$;
(ii) Y_1 and Y_3: $\{Y_1, Y_3, [Y_1, Y_3]\}$;
(iii) Y_1 only or Y_3 only: $\{Y_1\}$ or $\{Y_3\}$;
(iv) Y_2 only: $\{Y_2, \langle Y_2 : Y_2\rangle, \langle Y_2 : \langle Y_2 : Y_2\rangle\rangle\}$;
(v) Y_2 and Y_3 $\{Y_2, Y_3, [Y_2, Y_3]\}$.

3. Recall that for the rolling disk (Figure 5.9) above, Y_1 was 'rolling' input and Y_2 was 'spinning' input. Now, in the following three cases:
(i) Y_1 and Y_2 active, this system is LCA and LCC (satisfies sufficient condition);

(ii) Y_1 only: not LCA; and
(iii) Y_2 only: not LCA.
In theses three cases, C_{hor} has generators:
(i) Y_1 and Y_2: $\{Y_1, Y_2, [Y_1, Y_2], [Y_2, [Y_1, Y_2]]\}$;
(ii) Y_1 only: $\{Y_1\}$; and
(iii) Y_2 only: $\{Y_2\}$.
The rolling disk passes the good/bad symmetric product test. Another way to show that it is LCC is to show that the inputs allow one to follow any curve which is admitted by the constraints. Local configuration controllability then follows as the constraint distribution for the rolling disk has an involutive closure of maximal rank [Lew99].

Categorical Structure of Control Affine Systems

Control affine systems make a category $\mathcal{CAS}$ (see [Elk99]). The category $\mathcal{CAS}$ has the following data:

- An *object* in $\mathcal{CAS}$ is a pair $\sum = (M, \mathfrak{F} = \{f_0, f_1, ..., f_m\})$ where $\mathfrak{F}$ is a family of vector–fields
$$\dot{x}(t) = f_0(x(t)) + u^a(t) f_a(x(t))$$
on the manifold M.
- A *morphism* sending $\sum = (M, \mathfrak{F} = \{f_0, f_1, ..., f_m\})$ to $\sum' = (M', \mathfrak{F}' = \{f'_0, f'_1, ..., f'_{m'}\})$ is a triple $(\psi, \lambda_0, \Lambda)$ where $\psi : M \to M'$, $\lambda_0 : M \to \mathbb{R}^{m'}$, and $\Lambda : M \to L(\mathbb{R}^m, \mathbb{R}^{m'})$ are smooth maps satisfying:
  1. $T_x\psi(f_a(x)) = \Lambda_a^\alpha(x) f'_\alpha(\psi(x)),\ \ a \in \{1, ..., m\}$, and
  2. $T_x\psi(f_0(x)) = f'_0(\psi(x)) + \lambda_0^\alpha f'_\alpha(\psi(x))$.

  This corresponds to a change of state–input by
$$(x, u) \longmapsto (\psi(x), \lambda_0(x) + \Lambda(x)u).$$

Elkin [Elk99] discusses equivalence, inclusion, and factorization in the category $\mathcal{CAS}$. Using categorical language, he considers local equivalence for various classes of nonlinear control systems, including single–input systems, systems with involutive input distributions, and systems with three states and two inputs.

5.3.5 Lie–Adaptive Control in Human–Like Biomechanics

In this subsection we develop the concept of *machine learning* in the framework of Lie derivative control formalism (see (5.2.1) above). Consider an nD, SISO system in the standard affine form (5.9), rewritten here for convenience:
$$\dot{x}(t) = f(x) + g(x)\, u(t), \qquad y(t) = h(x), \tag{5.22}$$

As already stated, the feedback control law for the system (5.22) can be defined using Lie derivatives $\mathcal{L}_f h$ and $\mathcal{L}_g h$ of the system's output h along the vector–fields f and g.

If the SISO system (5.22) is a relatively simple (quasilinear) system with *relative degree* $r = 1$ it can be rewritten in a quasilinear form

$$\dot{x}(t) = \gamma_i(t)\, f_i(x) + d_j(t)\, g_j(x)\, u(t), \tag{5.23}$$

where γ_i $(i = 1, ..., n)$ and d_j $(j = 1, ..., m)$ are system's parameters, while f_i and g_j are smooth vector–fields.

In this case the feedback control law for *tracking* the *reference signal* $y_R = y_R(t)$ is defined as (see [Isi89, NS90])

$$u = \frac{-\mathcal{L}_f h + \dot{y}_R + \alpha\,(y_R - y)}{\mathcal{L}_g h}, \tag{5.24}$$

where α denotes the feedback gain.

Obviously, the problem of reference signal tracking is relatively simple and straightforward if we know all the system's parameters $\gamma_i(t)$ and $d_j(t)$ of (5.23). The question is can we apply a similar control law if the system parameters are unknown?

Now we have much harder problem of *adaptive signal tracking*. However, it appears that the feedback control law can be actually cast in a similar form (see [SI89],[Gom94]):

$$\widehat{u} = \frac{-\widehat{\mathcal{L}_f h} + \dot{y}_R + \alpha\,(y_R - y)}{\widehat{\mathcal{L}_g h}}, \tag{5.25}$$

where Lie derivatives $\mathcal{L}_f h$ and $\mathcal{L}_g h$ of (5.24) have been replaced by their estimates $\widehat{\mathcal{L}_f h}$ and $\widehat{\mathcal{L}_g h}$, defined respectively as

$$\widehat{\mathcal{L}_f h} = \widehat{\gamma}_i(t)\, \mathcal{L}_{f_i} h, \qquad \widehat{\mathcal{L}_g h} = \widehat{d}_j(t)\, \mathcal{L}_{g_i} h,$$

in which $\widehat{\gamma}_i(t)$ and $\widehat{d}_j(t)$ are the estimates for $\gamma_i(t)$ and $d_j(t)$.

Therefore, we have the straightforward control law even in the uncertain case, provided that we are able to estimate the unknown system parameters. Probably the best known *parameter update law* is based on the so–called *Lyapunov criterion* (see [SI89]) and given by

$$\dot{\psi} = -\gamma\, \epsilon\, W, \tag{5.26}$$

where $\psi = \{\gamma_i - \widehat{\gamma}_i, d_j - \widehat{d}_j\}$ is the parameter estimation error, $\epsilon = y - y_R$ is the output error, and γ is a positive constant, while the matrix W is defined as:

$$W = \left[W_1^T\, W_2^T\right]^T, \qquad \text{with}$$

$$W_1 = \begin{bmatrix} \mathcal{L}_{f_1} h \\ \vdots \\ \mathcal{L}_{f_n} h \end{bmatrix}, \qquad W_2 = \begin{bmatrix} \mathcal{L}_{g_1} h \\ \vdots \\ \mathcal{L}_{g_m} h \end{bmatrix} \cdot \frac{-\widehat{\mathcal{L}_f h} + \dot{y}_R + \alpha\,(y_R - y)}{\widehat{\mathcal{L}_g h}}.$$

The proposed adaptive control formalism (5.25–5.26) can be efficiently applied wherever we have a problem of tracking a given signal with an output of a SISO–system (5.22–5.23) with unknown parameters.

5.3.6 Intelligent Robot Control: Interaction with Environment

Here we show a dynamic model of the robot interacting with the environment [KV98, KV03a, KV03b]. The robot dynamics is described by a vector differential equation

$$H(q)\,\ddot{q} + h(q,\dot{q}) + J^T(q)\,F = \tau,$$

where, $q = q(t)$ is an nD vector of robot generalized coordinates; $H(q)$ is an $n \times n$ positive definite matrix of inertia moments of the manipulation mechanics; $h(q,\dot{q})$ is an nD nonlinear function of centrifugal, Coriolis, and gravitational moments; $\tau = \tau(t)$ is an nD vector of input control; $J^T(q)$ is an $n \times n$ Jacobian matrix connecting the velocities of robot end–effector and the velocities of robot generalized coordinates; and $F = F(t)$ is an mD vector of generalized forces, or, of generalized forces and moments from the environment acting on the end–effector.

In the frame of robot joint coordinates, the model of environment dynamics can be presented in the form

$$M(q)\,\ddot{q} + L(q,\dot{q}) = S^T(q)\,F,$$

where $M(q) \in \mathbb{R}^{n\times n}$ is a nonsingular matrix; $L(q,\dot{q}) \in \mathbb{R}^n$ is a nonlinear vector function; and $S^T(q) \in \mathbb{R}^{n\times n}$ is the matrix with $rank(S) = n$.

The end–effector of the manipulator is constrained on static geometric surfaces, $\Phi(q) = 0$, where $\Phi(q) \in \mathbb{R}^m$ is the holonomic constraint function.

In practice, it is convenient to adopt a simplified model of the environment, taking into account the dominant effects, such as stiffness, $F = K'(x - x_0)$, or an environment damping during the tool motion, $F = B'x$, where $K' \in \mathbb{R}^{n\times n}$, $B' \in \mathbb{R}^{n\times n}$ are semidefinite matrices describing the environment stiffness and damping, respectively, and $x_0 \in \mathbb{R}^n$ denotes the coordinate vector in Cartesian coordinates of the point of contact between the end–effector (tool) and a constraint surface. However, it is more appropriate to adopt the relationship defined by specification of the target impedance

$$F = M'\Delta\ddot{x} + B'\Delta\dot{x} + K'\Delta x, \qquad \text{where} \qquad \Delta x = x - x_0,$$

and M' is a positive definite inertia matrix. The matrices M', B', K' define the target impedance which can be selected to correspond to various objectives of the given manipulation task.

In the case of contact with the environment, the robot control task can be described as robot motion along a programmed trajectory $q_p(t)$ representing a twice continuously differentiable function, when a desired force of interaction

$F_p(t)$ acts between the robot and the environment. Thus, the programmed motion $q_p(t)$ and the desired interaction force $F_p(t)$ must satisfy the following relation

$$F_p(t) \equiv f\left(q_p(t), \dot{q}_p(t), \ddot{q}_p(t)\right).$$

The control problem for robot interacting with dynamic environment is to define the control $\tau(t)$ for $t \geq t_0$, that satisfies the target conditions

$$\lim_{t\to\infty} q(t) \to q_p(t), \qquad \lim_{t\to\infty} F(t) \to F_p(t).$$

As a first example, the control algorithm based on stabilization of the robot motion with a preset quality of transient responses is considered, which has the form

$$\tau = H(q)[\ddot{q}_p - KP\eta - KD\dot{\eta}] + h(q,\dot{q}) + J^T(q)F.$$

The familly of desired transient responses is specified by the vector differential equation

$$\ddot{\eta} = -KP\eta - KD\dot{\eta}, \qquad \eta(t) = q(t) - q_p(t), \tag{5.27}$$

where $KP \in \mathbb{R}^{n\times n}$ is the diagonal matrix of position feedback gains, and $KD \in R^{n\times n}$ is the diagonal matrix of velocity feedback gains. The right side of (5.27), i.e., PD–regulator is chosen such that the system defined by (5.27) is asymptotically stable in the whole. The values of matrices KP and KD can be chosen according to algebraic stability conditions.

The proposed control law represents a version of the well–known computed torque method including force term which uses dynamic robot model and the available on–line information from the position, velocity and force sensors. Here the model of robot environment does not have any influence on the performance of the control algorithm.

As the second example, control algorithm based on stabilization of the interaction force with a preset quality of transient responses is considered, which has the form

$$\begin{aligned}\tau &= H(q)\,M^{-1}(q)\,[-L(q,\dot{q}) + S^T(q)\,F] + h(q,\dot{q})\\ &\quad + J^T(q)\left\{F_p - \int_{t_0}^{t}\left[KFP\,\mu(\omega) + KFI\int_{t_0}^{t}\mu(\omega)\,dt\right]d\omega\right\},\end{aligned}$$

where $\mu(t) = F(t) - F_p(t)$; $KFP \in \mathbb{R}^{n\times n}$ is the matrix of proportional force feedback gains; and $KFI \in \mathbb{R}^{n\times n}$ is the matrix of integral force feedback gains. Here, it has been assumed that the interaction force in transient process should behave according to the following differential equation

$$\dot{\mu}(t) = Q(\mu), \qquad Q(\mu) = -KFP\,\mu - KFI\int_{t_0}^{t}\mu\,dt. \tag{5.28}$$

PI force regulator (continuous vector function of Q) is chosen such that the system defined by (5.28) is asymptotically stable in the whole. IN this case, environment dynamics model has explicit influence on the performance of contact control algorithm, also having influence on PI force local gains. It is clear that without knowing a sufficiently accurate environment model (parameters of matrices $M(q), L(q, \dot{q}), S(q)$) it is not possible to determine the nominal contact force $F_p(t)$.

5.4 Neural Path Integral Motion Controller

Recall that human motion is naturally driven by synergistic action of more than 600 skeletal muscles. While the muscles generate driving torques in the moving joints, subcortical neural system performs both local and global (loco)motion control: first reflexly controlling contractions of individual muscles, and then orchestrating all the muscles into synergetic actions in order to produce efficient movements. While the local reflex control of individual muscles is performed on the *spinal control level*, the global integration of all the muscles into coordinated movements is performed within the *cerebellum*.

All hierarchical subcortical neuro–muscular physiology, from the bottom level of a single muscle fiber, to the top level of cerebellar muscular synergy, acts as a *temporal* $< out|in >$ *reaction*, in such a way that the higher level acts as a command/control space for the lower level, itself representing an abstract image of the lower one:

1. At the *muscular level*, we have *excitation–contraction dynamics* [Hat78, Hat77b], in which $< out|in >$ is given by the following sequence of nonlinear diffusion processes (see Appendix for details): *neural-action-potential*$\rightsquigarrow$*synaptic-potential*$\rightsquigarrow$*muscular-action-potential*$\rightsquigarrow$*excitation-contraction-coupling*$\rightsquigarrow$*muscle-tension-generating* [Iva91]. Its purpose is the generation of muscular forces, to be transferred into driving torques within the joint anatomical geometry.
2. At the *spinal level*, $< out|in >$ is given by *autogenetic–reflex stimulus–response control* [Hou79]. Here we have a neural image of all individual muscles. The main purpose of the spinal control level is to provide both positive and negative feedbacks to stabilize generated muscular forces within the 'homeostatic' (or, more appropriately, 'homeokinetic') limits. The individual muscular actions are combined into flexor–extensor (or agonist–antagonist) pairs, mutually controlling each other. This is the mechanism of *reciprocal innervation of agonists and inhibition of antagonists*. It has a purely mechanical purpose to form the so–called *equivalent muscular actuators* (EMAs), which would generate driving torques $T_i(t)$ for all movable joints.
3. At the *cerebellar level*, $< out|in >$ is given by *sensory–motor integration* [HBB96]. Here we have an abstracted image of all autogenetic reflexes.

> The main purpose of the cerebellar control level is integration and fine tuning of the action of all active EMAs into a synchronized movement, by *supervising* the individual autogenetic reflex circuits. At the same time, to be able to perform in new and unknown conditions, the cerebellum is continuously adapting its own neural circuitry by unsupervised (self–organizing) learning. Its action is subconscious and automatic, both in humans and in animals.

Naturally, we can ask the question: Can we assign a single $< out|in >$ measure to all these neuro–muscular stimulus–response reactions? We think that we can do it; so in this Letter, we propose the concept of *adaptive sensory–motor transition amplitude* as a unique measure for this temporal $< out|in >$ relation. Conceptually, this $< out|in > -amplitude$ can be formulated as the '*neural path integral*' (see Appendix for details):

$$< out|in > \equiv \underset{amplitude}{\langle motor|sensory \rangle} = \int \mathcal{D}[w,x]\, \mathrm{e}^{\mathrm{i}\, S[x]}. \tag{5.29}$$

Here, the integral is taken over all *activated* (or, 'fired') *neural pathways* $x^i = x^i(t)$ of the cerebellum, connecting its input *sensory*−state with its output *motor*−state, symbolically described by *adaptive neural measure* $\mathcal{D}[w,x]$, defined by the weighted product (of discrete time steps)

$$\mathcal{D}[w,x] = \lim_{n\longrightarrow\infty} \prod_{t=1}^{n} w^i(t)\, dx^i(t), \tag{5.30}$$

in which the *synaptic weights* $w^i = w^i(t)$, included in all active neural pathways $x^i = x^i(t)$, are updated by the unsupervised *Hebbian–like learning* rule [Heb49]:

$$w^i(t+1) = w^i(t) + \frac{\sigma}{\eta}(w^i_d(t) - w^i_a(t)), \tag{5.31}$$

where $\sigma = \sigma(t)$, $\eta = \eta(t)$ represent local neural *signal* and *noise* amplitudes, respectively, while superscripts d and a denote *desired* and *achieved* neural states, respectively. Theoretically, equations (5.29–6.7) define an $\infty-$*dimensional neural network*. Practically, in a computer simulation we can use $10^7 \leq n \leq 10^8$, roughly corresponding to the number of neurons in the cerebellum.

The exponent term $S[x]$ in equation (5.29) represents the *autogenetic–reflex action*, describing reflexly–induced motion of all active EMAs, from their initial *stimulus*−state to their final *response*−state, along the family of extremal (i.e., Euler–Lagrangian) paths $x^i_{\min}(t)$. ($S[x]$ is properly derived in (5.34–5.35) below.)

5.4.1 Spinal Autogenetic Reflex Control

Recall (from Introduction) that at the spinal control level we have the autogenetic reflex *motor servo* [Hou79], providing the local, reflex feedback loops for individual muscular contractions. A voluntary contraction force F of human skeletal muscle is reflexly excited (positive feedback $+F^{-1}$) by the responses of its *spindle receptors* to stretch and is reflexly inhibited (negative feedback $-F^{-1}$) by the responses of its *Golgi tendon organs* to contraction. Stretch and unloading reflexes are mediated by combined actions of several autogenetic neural pathways, forming the *motor servo*.

In other words, branches of the afferent fibers also synapse with with interneurons that inhibit motor neurons controlling the antagonistic muscles – *reciprocal inhibition*. Consequently, the stretch stimulus causes the antagonists to relax so that they cannot resists the shortening of the stretched muscle caused by the main reflex arc. Similarly, firing of the Golgi tendon receptors causes inhibition of the muscle contracting too strong and simultaneous *reciprocal activation* of its antagonist. Both mechanisms of reciprocal inhibition and activation performed by the autogenetic circuits $+F^{-1}$ and $-F^{-1}$, serve to generate the well–tuned EMA–driving torques T_i.

Now, once we have properly defined the symplectic musculo–skeletal dynamics [Iva04] on the biomechanical (momentum) phase–space manifold T^*M^N, we can proceed in formalizing its hierarchical subcortical neural control. By introducing the *coupling Hamiltonians* $H^m = H^m(q,p)$, selectively corresponding *only* to the $M \leq N$ *active joints*, we define the *affine Hamiltonian control function* $H_{aff} : T^*M^N \to \mathbb{R}$, in local canonical coordinates on T^*M^N given by (adapted from [NS90] for the biomechanical purpose)

$$H_{aff}(q,p) = H_0(q,p) - H^m(q,p)\, T_m, \qquad (m = 1, \dots, M \leq N), \tag{5.32}$$

where $T_m = T_m(t,q,p)$ are now *feedback torque one–forms* (different from the initial driving torques T_i acting in all the joints). Using the affine Hamiltonian function (5.32), we get the *affine Hamiltonian servo–system* [Iva04],

$$\begin{aligned} \dot{q}^i &= \frac{\partial H_0(q,p)}{\partial p_i} - \frac{\partial H^m(q,p)}{\partial p_i} T_m, \\ \dot{p}_i &= -\frac{\partial H_0(q,p)}{\partial q^i} + \frac{\partial H^m(q,p)}{\partial q^i} T_m, \\ q^i(0) &= q^i_0, \quad p_i(0) = p^0_i, \qquad (i = 1, \dots, N; \quad m = 1, \dots, M \leq N). \end{aligned} \tag{5.33}$$

The affine Hamiltonian control system (5.33) gives our formal description for the autogenetic spinal motor–servo for all $M \leq N$ activated (i.e., working) EMAs.

5.4.2 Cerebellum – the Comparator

Having, thus, defined the spinal reflex control level, we proceed to model the top subcortical commander/controller, the *cerebellum*. It is a brain region anatomically located at the bottom rear of the head (the hindbrain), directly above the brainstem, which is important for a number of subconscious and automatic motor functions, including motor learning. It processes information received from the motor cortex, as well as from proprioceptors and visual and equilibrium pathways, and provides 'instructions' to the motor cortex and other subcortical motor centers (like the basal nuclei), which result in proper balance and posture, as well as smooth, coordinated skeletal movements, like walking, running, jumping, driving, typing, playing the piano, etc. Patients with cerebellar dysfunction have problems with precise movements, such as walking and balance, and hand and arm movements. The cerebellum looks *similar in all animals*, from fish to mice to humans. This has been taken as evidence that it performs a common function, such as regulating motor learning and the timing of movements, in all animals. Studies of simple forms of motor learning in the vestibulo–ocular reflex and eye–blink conditioning are demonstrating that timing and amplitude of learned movements are encoded by the cerebellum.

The cerebellum is responsible for coordinating precisely timed $< out|in >$ activity by integrating motor output with ongoing sensory feedback (see Figure 5.10). It receives extensive projections from sensory–motor areas of the cortex and the periphery and directs it back to premotor and motor cortex [Ghe90]. This suggests a role in sensory–motor integration and the timing and execution of human movements. The cerebellum stores patterns of motor control for frequently performed movements, and therefore, its circuits are changed by experience and training. It was termed the *adjustable pattern generator* in the work of J. Houk and collaborators [HBB96]. Also, it has become the inspiring 'brain–model' in the recent robotic research [SA98, Sch98].

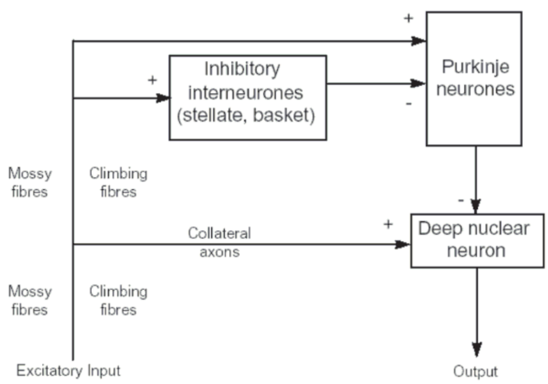

Fig. 5.10. Schematic $< out|in >$ organization of the primary cerebellar circuit. In essence, excitatory inputs, conveyed by collateral axons of Mossy and Climbing fibers activate directly neurones in the Deep cerebellar nuclei. The activity of these latter is also modulated by the inhibitory action of the cerebellar cortex, mediated by the Purkinje cells.

Comparing the number of its neurons ($10^7 - 10^8$), to the size of conventional neural networks, suggests that artificial neural nets *cannot* satisfactorily model the function of this sophisticated 'super–bio–computer', as its dimensionality is virtually infinite. Despite a lot of research dedicated to its structure and function (see [HBB96] and references there cited), the real nature of the cerebellum still remains a 'mystery'.

5.4.3 Hamiltonian Action and Neural Path Integral

Here, we propose a *quantum–like adaptive control* approach to modelling the 'cerebellar mystery'. Corresponding to the affine Hamiltonian control function (5.32) we define the *affine Hamiltonian control action*,

$$S_{aff}[q,p] = \int_{t_{in}}^{t_{out}} d\tau \left[p_i \dot{q}^i - H_{aff}(q,p)\right]. \tag{5.34}$$

From the affine Hamiltonian action (5.34) we further derive the associated expression for the *neural phase–space path integral* (in normal units), representing the *cerebellar sensory–motor amplitude* $< out|in >$,

$$\begin{aligned} \left\langle q^i_{out}, p^{out}_i | q^i_{in}, p^{in}_i \right\rangle &= \int \mathcal{D}[w,q,p]\, \mathrm{e}^{\mathrm{i}\, S_{aff}[q,p]} \qquad (5.35)\\ &= \int \mathcal{D}[w,q,p] \exp\left\{ \mathrm{i} \int_{t_{in}}^{t_{out}} d\tau \left[p_i \dot{q}^i - H_{aff}(q,p)\right]\right\}, \\ \text{with} \qquad \int \mathcal{D}[w,q,p] &= \int \prod_{\tau=1}^{n} \frac{w^i(\tau) dp_i(\tau) dq^i(\tau)}{2\pi}, \end{aligned}$$

where $w_i = w_i(t)$ denote the cerebellar synaptic weights positioned along its neural pathways, being continuously updated using the Hebbian–like self–organizing learning rule (6.7). Given the transition amplitude $< out|in >$ (5.35), the *cerebellar sensory–motor transition probability* is defined as its absolute square, $| < out|in > |^2$.

In (5.35), $q^i_{in} = q^i_{in}(t)$, $q^i_{out} = q^i_{out}(t)$; $p^{in}_i = p^{in}_i(t)$, $p^{out}_i = p^{out}_i(t)$; $t_{in} \leq t \leq t_{out}$, for all discrete time steps, $t = 1, ..., n \longrightarrow \infty$, and we are allowing for the affine Hamiltonian $H_{aff}(q,p)$ to depend upon all the ($M \leq N$) EMA–angles and angular momenta collectively. Here, we actually systematically took a discretized differential time limit of the form $t_\sigma - t_{\sigma-1} \equiv d\tau$ (both σ and τ denote discrete time steps) and wrote $\frac{(q^i_\sigma - q^i_{\sigma-1})}{(t_\sigma - t_{\sigma-1})} \equiv \dot{q}^i$. For technical details regarding the path integral calculations on Riemannian and symplectic manifolds (including the standard regularization procedures), see [Kla97, Kla00].

Now, motor learning occurring in the cerebellum can be observed using functional MR imaging, showing changes in the cerebellar action potential, related to the motor tasks (see, e.g., [Mas02]). To account for these electro–physiological currents, we need to add the *source* term $J_i(t) q^i(t)$ to the affine

Hamiltonian action (5.34), (the current $J_i = J_i(t)$ acts as a source $J_i A^i$ of the *cerebellar electrical potential* $A^i = A^i(t)$),

$$S_{aff}[q,p,J] = \int_{t_{in}}^{t_{out}} d\tau \left[p_i \dot{q}^i - H_{aff}(q,p) + J_i q^i \right],$$

which, subsequently gives the cerebellar path integral with the action potential source, coming either from the motor cortex or from other subcortical areas.

Note that the standard *Wick rotation*: $t \mapsto \mathrm{i}t$ (see [Kla97, Kla00]), makes all our path integrals real, i.e.,

$$\int \mathcal{D}[w,q,p]\, \mathrm{e}^{\mathrm{i}\, S_{aff}[q,p]} \quad \underrightarrow{Wick} \quad \int \mathcal{D}[w,q,p]\, \mathrm{e}^{-\, S_{aff}[q,p]},$$

while their subsequent discretization gives the standard thermodynamic *partition functions*,

$$Z = \sum_j \mathrm{e}^{-w_j E^j / T}, \tag{5.36}$$

where E^j is the energy eigenvalue corresponding to the affine Hamiltonian $H_{aff}(q,p)$, T is the temperature–like environmental control parameter, and the sum runs over all energy eigenstates (labelled by the index j). From (6.16), we can further calculate all statistical and thermodynamic system properties (see [Fey72]), as for example, *transition entropy* $S = k_B \ln Z$, etc.

5.5 Brain–Like Control Functor in Human–Like Biomechanics

In this final section we propose our most recent model [IB05] of the complete biomechanical *brain–like control functor*. This is a neuro–dynamical reflection on our *covariant force law*, $F_i = m g_{ij} a^j$, and its associated *covariant force functor* $\mathcal{F}_* : TT^*M \to TTM$ (see section 2.7 above).

Traditional hierarchical robot control (see, e.g., [VS82]) consists of three levels: the *executive* control–level (at the bottom) performs tracking of nominal trajectories in internal–joint coordinates, the *strategic* control–level (at the top) performs 'planning' of trajectories of an end–effector in external–Cartesian coordinates, and the *tactical* control–level (in the middle) connects other two levels by means of inverse kinematics.

The modern version of the hierarchical robot control includes decision–making done by the neural (or, neuro–fuzzy) classifier to adapt the (manipulator) control to dynamically changing environment.

On the other hand, the so–called 'intelligent' approach to robot control typically represents a form of function approximation, which is itself based on some combination of neuro–fuzzy–genetic computations. Many special issues and workshops focusing on physiological models for robot control reflect the increased attention for the development of *cerebellar models* [Sma99, SA98, Sch99, Sch98, Arb98] for learning robot control with functional decomposition, where the main result could be formulated as: *the cerebellum is more then just the function approximator*.

In this section we try to fit between these three approaches for humanoid control, emphasizing the role of muscle–like robot actuators. We propose a new, physiologically based, tensor–invariant, hierarchical force control (FC, for short) for the physiologically realistic biomechanics. We consider the muscular torque one–forms F_i as the most important component of human–like motion; therefore we propose the sophisticated hierarchical system for the subtle F_i–control: corresponding to the spinal, the cerebellar and cortical levels of human motor control. F_i are first set–up as testing input–signals to biomechanics, and then covariantly updated as feedback $1-$forms u_i on each FC–level. On the spinal FC–level the nominal joint–trajectory tracking is proposed in the form of affine Hamiltonian control; here the driving torques are given corrections by spinal–reflex controls. On the cerebellar FC–level, the relation is established between canonical joint coordinates q^i, p_i and gradient *neural–image coordinates* x^i, y_i, representing bidirectional, self–organized, associative memory machine; here the driving torques are given the cerebellar corrections. On the cortical FC–level the topological 'hyper–joystick' is proposed as the central FC command–space, selector, with the fuzzy–logic feedback–control map defined on it, giving the cortical corrections to the driving torques.

The model of the spinal FC–level formulated here resembles *autogenetic motor servo*, acting on the spinal–reflex level of the human locomotor control. The model of the cerebellar FC–level formulated here mimics the self–organizing, associative function of the excitatory granule cells and the inhibitory Purkinje cells of the cerebellum [HBB96]. The model of the cortical FC–level presented in this section mimics the synergistic *regulation of locomotor conditioned reflexes* by the cerebellum [HBB96].

We believe that (already mentioned) extremely high order of the driving force redundancy in biomechanics justifies the formulation of the three–level force control system. Also, both brain–like control systems can be easily extended to provide $SE(3)-$based force control for moving inverse kinematics (IK) chains of legs and arms.

Functor Control Machine

In this subsection we define the functor control–machine (compare with section (3.5) above), for the learning control with functional decomposition, by a two–step generalization of the Kalman's theory of linear MIMO–feedback systems. The first generalization puts the Kalman's theory into the pair of mutually dual linear categories $\mathtt{Vect}$ and $\mathtt{Vect}^*$ of vector spaces and linear operators, with a 'loop–functor' representing the closed–loop control, thus formulating the unique, categorical formalism valid both for the discrete and continual MIMO–systems.

We start with the unique, *feedforward continual–sequential state equation*

$$\dot{x}(t+1) = Ax(t) + Bu(t), \qquad y(t) = Cx(t), \tag{5.37}$$

where the finite–dimensional vector spaces of *state* $X \ni x$, *input* $U \ni u$, and *output* $Y \ni y$ have the corresponding linear operators, respectively $A : X \to X$, $B : U \to X$, and $C : X \to Y$. The modular system theory comprises the *system dynamics*, given by a pair (X, A), together with a *reachability map* $e : U \to X$ of the pair (B, A), and an *observability map* $m : X \to Y$ of the pair (A, C). If the reachability map e is surjection the system dynamics (X, A) is called *reachable*; if the observability map m is injection the system dynamics (X, A) is called *observable*. If the system dynamics (X, A) is both reachable and observable, a *composition* $r = m \circ e : U \to Y$ defines the *total system's response*, which is given by solution of equation (5.37). If the unique solution to the continual–sequential state equation exists, it gives the answer to the (minimal) *realization problem*: find the system S that realizes the given response $r = m \circ e : U \to Y$ (in the smallest number of discrete states and in the shortest time).

The inverse map $r^{-1} = e^{-1} \circ m^{-1} : Y \to U$ of the total system's response $r : U \to Y$ defines the linear *feedback operator* $K : Y \to U$, given by standard feedback equation

$$u(t) = Ky(t). \tag{5.38}$$

In categorical language, the feedforward system dynamics in the category $\mathtt{Vect}$ is a pair (X, A), where $X \in \mathtt{Ob}(\mathtt{Vect})$ is an object in $\mathtt{Vect}$ and $A : X \to X \in \mathtt{Mor}(\mathtt{Vect})$ is a $\mathtt{Vect}$–morphism. A *feedforward decomposable system* in $\mathtt{Vect}$ is such a sixtuple $S \equiv (X, A, U, B, Y, C)$ that (X, A) is the system dynamics in $\mathtt{Vect}$, a $\mathtt{Vect}$–morphism $B : U \to X$ is an *input map*, and a $\mathtt{Vect}$–morphism $C : X \to Y$ is an *output map*. Any object in $\mathtt{Vect}$ is characterized by mutually dual notions of its *degree* (a number of its input morphisms) and its *codegree* (a number of its output morphisms). Similarly, any decomposable system S in $\mathtt{Vect}$ has a *reachability map* given by an epimorphism $e = A \circ B : U \to X$ and its dual *observability map* given by a monomorphism $m = C \circ A : X \to Y$; their composition $r = m \circ e : U \to Y$ in $\mathtt{Mor}(\mathtt{Vect})$ defines the total system's response in $\mathtt{Vect}$ given by the unique solution of the continual–sequential state equation (5.37) [IS01].

The dual of the total system's response, defined by the feedback equation (5.38), is the *feedback morphism* $K = e^{-1} \circ m^{-1} : Y \to U$ belonging to the dual category $\mathtt{Vect}^*$.

In this way, the linear, closed–loop, continual–sequential MIMO–system (5.37–5.38) represents the *linear iterative loop functor* $\mathcal{L} : \mathtt{Vect} \Rightarrow \mathtt{Vect}^*$.

Our second generalization represents a *natural system process* $\Xi[\mathcal{L}]$, that transforms the linear loop functor $\mathcal{L} : \mathtt{Vect} \Rightarrow \mathtt{Vect}^*$ – into the *nonlinear loop functor* $\mathcal{NL} : \mathcal{CAT} \Rightarrow \mathcal{CAT}^*$ between two mutually dual nonlinear categories $\mathcal{CAT}$ and $\mathcal{CAT}^*$. We apply the natural process Ξ, separately

1. To the feedforward decomposable system $S \equiv (X, A, U, B, Y, C)$ in $\mathtt{Vect}$, and
2. To the feedback morphism $K = e^{-1} \circ m^{-1} : Y \to U$ in $\mathtt{Vect}^*$.

Under the action of the natural process Ξ, the linear feedforward system dynamics (X, A) in $\mathtt{Vect}$ transforms into a nonlinear feedforward Ξ–dynamics $(\Xi[X], \Xi[A])$ in $\mathcal{CAT}$, represented by a *nonlinear feedforward decomposable system*, $\Xi[S] \equiv (\Xi[X], \Xi[A], \Xi[U], \Xi[B], \Xi[Y], \Xi[C])$.

The reachability map transforms into the *input process* $\Xi[e] = \Xi[A] \circ \Xi[B] : \Xi[U] \longrightarrow \Xi[X]$, while its dual, observability map transforms into the *output process* $\Xi[m] = \Xi[C] \circ \Xi[A] : \Xi[X] \longrightarrow \Xi[Y]$. In this way the total response of the linear system $r = m \circ e : U \to Y$ in $\mathtt{Mor}(\mathtt{Vect})$ transforms into the *nonlinear system behavior*, $\Xi[r] = \Xi[m] \circ \Xi[e] : \Xi[U] \longrightarrow \Xi[Y]$ in $\mathtt{Mor}(\mathcal{CAT})$. Obviously, $\Xi[r]$, if exists, is given by a nonlinear Ξ–transform of the linear state equations (5.37–5.38).

Analogously, the linear feedback morphism $K = e^{-1} \circ m^{-1} : Y \to U$ in $\mathtt{Mor}(\mathtt{Vect}^*)$ transforms into the nonlinear feedback morphism $\Xi[K] = \Xi[e^{-1}] \circ \Xi[m^{-1}] : \Xi[Y] \to \Xi[U]$ in $\mathtt{Mor}(\mathcal{CAT}^*)$.

In this way, the natural system process $\Xi : \mathcal{L} \Rrightarrow \mathcal{NL}$ is established. That means that the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{CAT} \Rightarrow \mathcal{CAT}^*$ is defined out of the linear, closed–loop, continual–sequential MIMO–system (5.37).

In this section we formulate the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{CAT} \Rightarrow \mathcal{CAT}^*$ for various hierarchical levels of muscular–like FC.

Spinal Control Level

Our first task is to establish the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{EX} \Rightarrow \mathcal{EX}^*$ on the category $\mathcal{EX}$ of spinal FC–level.

Recall that our dissipative, driven δ–Hamiltonian biomechanical system on the configuration manifold M is, in local canonical–symplectic coordinates $q^i, p_i \in U_p$ on the momentum phase–space manifold T^*M, given by autonomous equations

$$\dot{q}^i = \frac{\partial H_0}{\partial p_i} + \frac{\partial R}{\partial p_i}, \qquad (i = 1, \dots, N) \tag{5.39}$$

$$\dot{p}_i = F_i - \frac{\partial H_0}{\partial q^i} + \frac{\partial R}{\partial q^i}, \tag{5.40}$$

$$q^i(0) = q^i_0, \qquad p_i(0) = p^0_i, \tag{5.41}$$

including contravariant equation (5.39) – the velocity vector–field, and covariant equation (5.40) – the force 1–form, together with initial joint angles q^i_0 and momenta p^0_i. Here the physical Hamiltonian function $H_0 : T^*M \to \mathbb{R}$ represents the total biomechanical energy function, in local canonical coordinates $q^i, p_i \in U_p$ on T^*M given by

$$H_0(q, p) = \frac{1}{2} g^{ij}\, p_i\, p_j + V(q),$$

where $g^{ij} = g^{ij}(q, m)$ denotes the contravariant material metric tensor.

Now, the *control Hamiltonian function* $H_\gamma : T^*M \to \mathbb{R}$ of FC is in local canonical coordinates on T^*M defined by [NS90]

$$H_\gamma(q, p, u) = H_0(q, p) - q^i\, u_i, \qquad (i = 1, \ldots, N) \tag{5.42}$$

where $u_i = u_i(t, q, p)$ are feedback–control 1−forms, representing the spinal FC–level $u-$corrections to the covariant torques $F_i = F_i(t, q, p)$.

Using $\delta-$Hamiltonian biomechanical system (5.39–5.41) and the control Hamiltonian function (5.42), control $\gamma_\delta-$Hamiltonian FC–system can be defined as

$$\begin{aligned}
\dot{q}^i &= \frac{\partial H_\gamma(q, p, u)}{\partial p_i} + \frac{\partial R(q, p)}{\partial p_i}, \\
\dot{p}_i &= F_i - \frac{\partial H_\gamma(q, p, u)}{\partial q^i} + \frac{\partial R(q, p)}{\partial q^i}, \\
o^i &= -\frac{\partial H_\gamma(q, p, u)}{\partial u_i}, \qquad (i = 1, \ldots, N) \\
q^i(0) &= q^i_0, \qquad p_i(0) = p^0_i,
\end{aligned}$$

where $o^i = o^i(t)$ represent FC natural outputs which can be different from commonly used joint angles.

If nominal reference outputs $o^i_R = o^i_R(t)$ are known, the simple PD *stiffness–servo* [Whi87] could be formulated, via *error function* $e(t) = o^j - o^j_R$, in covariant form

$$u_i = K_o \delta_{ij}(o^j - o^j_R) + K_{\dot{o}} \delta_{ij}(\dot{o}^j - \dot{o}^j_R), \tag{5.43}$$

where Ks are the control–gains and δ_{ij} is the Kronecker tensor.

If natural outputs o^i *actually are* the joint angles and nominal canonical trajectories $\left(q^i_R = q^i_R(t),\, p^R_i = p^R_i(t)\right)$ are known, then the stiffness–servo (5.43) could be formulated in canonical form as

$$u_i = K_q \delta_{ij}(q^i - q^i_R) + K_p(p_i - p^R_i).$$

Now, using the fuzzified $\mu-$Hamiltonian biomechanical system with fuzzy system numbers (i.e, imprecise segment lengths, masses and moments of inertia, joint dampings and muscular actuator parameters)

$$\dot{q}^i = \frac{\partial H_0(q, p, \sigma_\mu)}{\partial p_i} + \frac{\partial R}{\partial p_i}, \tag{5.44}$$

$$\dot{p}_i = \bar{F}_i - \frac{\partial H_0(q, p, \sigma_\mu)}{\partial q^i} + \frac{\partial R}{\partial q^i}, \tag{5.45}$$

$$q^i(0) = \bar{q}^i_0, \qquad p_i(0) = \bar{p}^0_i, \qquad (i = 1, \ldots, N), \tag{5.46}$$

(see 3.5.4 above) and the control Hamiltonian function (5.42), $\gamma_\mu-$Hamiltonian FC–system can be defined as

$$\dot{q}^i = \frac{\partial H_\gamma(q,p,u,\sigma_\mu)}{\partial p_i} + \frac{\partial R(q,p)}{\partial p_i},$$
$$\dot{p}_i = \bar{F}_i - \frac{\partial H_\gamma(q,p,u,\sigma_\mu)}{\partial q^i} + \frac{\partial R(q,p)}{\partial q^i},$$
$$\bar{o}^i = -\frac{\partial H_\gamma(q,p,u,\sigma_\mu)}{\partial u_i}, \qquad q^i(0) = \bar{q}^i_0, \qquad p_i(0) = \bar{p}^0_i,$$

where $\bar{o}^i = \bar{o}^i(t)$ represent the fuzzified natural outputs.

Finally, applying stochastic forces (diffusion fluctuations $B_{ij}[q^i(t),t]$ and discontinuous jumps in the form of ND Wiener process $W^j(t)$), i.e., using the fuzzy–stochastic $[\mu\sigma]-$Hamiltonian biomechanical system

$$dq^i = \left(\frac{\partial H_0(q,p,\sigma_\mu)}{\partial p_i} + \frac{\partial R}{\partial p_i}\right) dt, \tag{5.47}$$
$$dp_i = B_{ij}[q^i(t),t]\, dW^j(t) + \left(\bar{F}_i - \frac{\partial H_0(q,p,\sigma_\mu)}{\partial q^i} + \frac{\partial R}{\partial q^i}\right) dt, \tag{5.48}$$
$$q^i(0) = \bar{q}^i_0, \qquad p_i(0) = \bar{p}^0_i. \tag{5.49}$$

(see 3.5.4 above), and the control Hamiltonian function (5.42),$\gamma_{\mu\sigma}-$Hamiltonian FC–system can be defined as

$$dq^i = \left(\frac{\partial H_\gamma(q,p,u,\sigma_\mu)}{\partial p_i} + \frac{\partial R(q,p)}{\partial p_i}\right) dt,$$
$$dp_i = B_{ij}[q^i(t),t]\, dW^j(t) + \left(\bar{F}_i - \frac{\partial H_\gamma(q,p,u,\sigma_\mu)}{\partial q^i} + \frac{\partial R(q,p)}{\partial q^i}\right) dt,$$
$$d\bar{o}^i = -\frac{\partial H_\gamma(q,p,u,\sigma_\mu)}{\partial u_i} dt, \qquad (i = 1,\dots,N)$$
$$q^i(0) = \bar{q}^i_0, \qquad p_i(0) = \bar{p}^0_i.$$

If we have the case that not all of the configuration joints on the configuration manifold M are active in the specified robot task, we can introduce the coupling Hamiltonians $H^j = H^j(q,p)$, $j = 1,\dots, M \leq N$, corresponding to the system's active joints, and we come to *affine Hamiltonian function* $H_a : T^*M \to \mathbb{R}$, in local canonical coordinates on T^*M given as [NS90]

$$H_a(q,p,u) = H_0(q,p) - H^j(q,p)\, u_j. \tag{5.50}$$

Using $\delta-$Hamiltonian biomechanical system (5.39–5.41) and the affine Hamiltonian function (5.50), affine $a_\delta-$Hamiltonian FC–system can be defined as

$$\dot{q}^i = \frac{\partial H_0(q,p)}{\partial p_i} - \frac{\partial H^j(q,p)}{\partial p_i} u_j + \frac{\partial R}{\partial p_i}, \tag{5.51}$$

$$\dot{p}_i = F_i - \frac{\partial H_0(q,p)}{\partial q^i} + \frac{\partial H^j(q,p)}{\partial q^i} u_j + \frac{\partial R}{\partial q^i}, \tag{5.52}$$

$$o^i = -\frac{\partial H_a(q,p,u)}{\partial u_i} = H^j(q,p), \tag{5.53}$$

$$q^i(0) = q_0^i, \qquad p_i(0) = p_i^0, \tag{5.54}$$

$$(i = 1, \dots, N; \qquad j = 1, \dots, M \leq N).$$

Using the Lie–derivative exact feedback linearization (see (5.2.1) above), and applying the *constant relative degree* r (see [Isi89, SI89]) to all N joints of the affine a_δ–Hamiltonian FC–system (5.51–5.54), the control law for asymptotic tracking the reference outputs o_R^j could be formulated as

$$u_j = \frac{\dot{o}_R^{(r)j} - \mathcal{L}_f^{(r)} H^j + \sum_{s=1}^r \gamma_{s-1}(o_R^{(s-1)j} - \mathcal{L}_f^{(s-1)} H^j)}{\mathcal{L}_g \mathcal{L}_f^{(r-1)} H^j},$$

where standard MIMO–vector–fields f and g are given by

$$f = \left(\frac{\partial H_0}{\partial p_i}, -\frac{\partial H_0}{\partial q^i} \right), \qquad g = \left(-\frac{\partial H^j}{\partial p_i}, \frac{\partial H^j}{\partial q^i} \right)$$

and γ_{s-1} are the coefficients of linear differential equation of order r for the error function $e(t) = o^j - o_R^j$

$$e^{(r)} + \gamma_{r-1} e^{(r-1)} + \dots + \gamma_1 e^{(1)} + \gamma_0 e = 0.$$

Using the fuzzified μ–Hamiltonian biomechanical system (5.44–5.46) and the affine Hamiltonian function (5.50), affine a_μ–Hamiltonian FC–system can be defined as

$$\begin{aligned}
\dot{q}^i &= \frac{\partial H_0(q,p,\sigma_\mu)}{\partial p_i} - \frac{\partial H^j(q,p,\sigma_\mu)}{\partial p_i} u_j + \frac{\partial R(q,p)}{\partial p_i}, \\
\dot{p}_i &= \bar{F}_i - \frac{\partial H_0(q,p,\sigma_\mu)}{\partial q^i} + \frac{\partial H^j(q,p,\sigma_\mu)}{\partial q^i} u_j + \frac{\partial R(q,p)}{\partial q^i}, \\
\bar{o}^i &= -\frac{\partial H_a(q,p,u,\sigma_\mu)}{\partial u_i} = H^j(q,p,\sigma_\mu), \\
q^i(0) &= \bar{q}_0^i, \qquad p_i(0) = \bar{p}_i^0, \qquad (i = 1, \dots, N;\ j = 1, \dots, M \leq N).
\end{aligned}$$

Using the fuzzy–stochastic $[\mu\sigma]$–Hamiltonian biomechanical system (5.47–5.49) and the affine Hamiltonian function (5.50), affine $a_{\mu\sigma}$–Hamiltonian FC–system can be defined as

$$
\begin{aligned}
dq^i &= \left(\frac{\partial H_0(q,p,\sigma_\mu)}{\partial p_i} - \frac{\partial H^j(q,p,\sigma_\mu)}{\partial p_i} u_j + \frac{\partial R(q,p)}{\partial p_i} \right) dt, \\
dp_i &= B_{ij}[q^i(t),t]\, dW^j(t) \qquad + \\
&\quad \left(\bar{F}_i - \frac{\partial H_0(q,p,\sigma_\mu)}{\partial q^i} + \frac{\partial H^j(q,p,\sigma_\mu)}{\partial q^i} u_j + \frac{\partial R(q,p)}{\partial q^i} \right) dt, \\
d\bar{\sigma}^i &= -\frac{\partial H_a(q,p,u,\sigma_\mu)}{\partial u_i} dt = H^j(q,p,\sigma_\mu)\, dt, \\
q^i(0) &= \bar{q}^i_0, \qquad p_i(0) = \bar{p}^0_i, \qquad (i = 1,\dots,N;\ j = 1,\dots,\ M \le N).
\end{aligned}
$$

Being high–degree and highly nonlinear, all of these affine control systems are extremely sensitive upon the variation of parameters, inputs, and initial conditions. The sensitivity function S of the affine Hamiltonian $H_a(q,p,u)$ upon the parameters β_i (representing segment lengths L_i, masses m_i, moments of inertia J_i and joint dampings b_i, see [IS01, Iva91]), is in the case of $a_\delta-$Hamiltonian FC–system defined as

$$
S(H,\beta) = \frac{\beta_i}{H_a(q,p,u)} \frac{\partial H_a(q,p,u)}{\partial \beta_i},
$$

and similarly in other two $a_\mu-$ and $a_{\mu\sigma}-$ cases.

The three affine FC–level systems a_δ, a_μ and $a_{\mu\sigma}$, resemble (in a fuzzy–stochastic–Hamiltonian form), Houk's autogenetic motor servo of muscle spindle and Golgi tendon proprioceptors [Hou79], correcting the covariant driving torques $F_i = F_i(t,q,p)$ by local 'reflex controls' $u_i(t,q,p)$. They form the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{EX} \Rightarrow \mathcal{EX}^*$.

Cerebellar Control Level

Our second task is to establish the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{TA} \Rightarrow \mathcal{TA}^*$ on the category $\mathcal{TA}$ of the cerebellar FC–level. Here we propose an oscillatory neurodynamical $(x,y,\boldsymbol{\omega})$–system (adapted from [IJB99a]), a bidirectional, self–organized, associative–memory machine, resembling the function of a set of excitatory granule cells and inhibitory Purkinje cells in the middle layer of the cerebellum (see [EIS67, HBB96]). The neurodynamical $(x,y,\boldsymbol{\omega})$–system acts on *neural–image manifold* M^N_{im} of the configuration manifold M^N as a pair of smooth, '$1-1$' and 'onto' maps (Ψ,Ψ^{-1}), where $\Psi : M^N \to M^N_{im}$ represents the feedforward map, and $\Psi^{-1} : M^N_{im} \to M^N$ represents the feedback map. Locally, it is defined in Riemannian neural coordinates $x^i, y_i \in V_y$ on M^N_{im}, which are in bijective correspondence with symplectic joint coordinates $q^i, p_i \in U_p$ on T^*M.

The $(x,y,\boldsymbol{\omega})$–system is formed out of two distinct, yet nonlinearly–coupled neural subsystems, with $A^i(q)$ (A.37) and $B_i(p)$ (5.58) as system inputs, and the feedback–control $1-$forms u_i (5.63) as system outputs:

1. Granule cells excitatory (contravariant) and Purkinje cells inhibitory (covariant) activation (x, y)–dynamics (A.35–5.58), defined respectively by a vector–field $x^i = x^i(t) : M \to TM$, representing a cross–section of the tangent bundle TM, and a 1–form $y_i = y_i(t) : M \to T^*M$, representing a cross–section of the cotangent bundle T^*M; and
2. Excitatory and inhibitory unsupervised learning $(\boldsymbol{\omega})$–dynamics (5.58–5.60) generated by random differential Hebbian learning process (5.61–5.63), defined respectively by contravariant synaptic tensor–field $\omega^{ij} = \omega^{ij}(t) : M \to TTM^N_{im}$ and covariant synaptic tensor–field $\omega_{ij} = \omega_{ij}(t) : M \to T^*T^*M$, representing cross–sections of contravariant and covariant tensor bundles, respectively.

The system equations are defined as

$$\dot{x}^i = A^i(q) + \omega^{ij}\, f_j(y) - x^i, \tag{5.55}$$

$$\dot{y}_i = B_i(p) + \omega_{ij}\, f^j(x) - y_i, \tag{5.56}$$

$$A^i(q) = K_q(q^i - q^i_R), \tag{5.57}$$

$$B_i(p) = K_p(p^R_i - p_i), \tag{5.58}$$

$$\dot{\omega}^{ij} = -\omega^{ij} + I^{ij}(x, y), \tag{5.59}$$

$$\dot{\omega}_{ij} = -\omega_{ij} + I_{ij}(x, y), \tag{5.60}$$

$$I^{ij} = f^i(x)\, f^j(y) + \dot{f}^i(x)\, \dot{f}^j(y) + \sigma^{ij}, \tag{5.61}$$

$$I_{ij} = f_i(x)\, f_j(y) + \dot{f}_i(x)\, \dot{f}_j(y) + \sigma_{ij}, \tag{5.62}$$

$$u_i = \frac{1}{2}(\delta_{ij}\, x^i + y_i), \qquad (i, j = 1, \dots, N). \tag{5.63}$$

Here $\boldsymbol{\omega}$ is a symmetric 2nd order synaptic tensor–field; $I^{ij} = I^{ij}(x, y, \sigma)$ and $I_{ij} = I_{ij}(x, y, \sigma)$ respectively denote contravariant–excitatory and covariant–inhibitory random differential Hebbian innovation–functions with tensorial Gaussian noise $\boldsymbol{\sigma}$ (in both variances); fs and $\dot{f}$s denote sigmoid activation functions ($f = \tanh(.)$) and corresponding signal velocities ($\dot{f} = 1 - f^2$), respectively in both variances;

$A^i(q)$ and $B_i(p)$ are contravariant–excitatory and covariant–inhibitory neural inputs to granule and Purkinje cells, respectively; u_i are the corrections to the feedback–control 1–forms on the cerebellar FC–level.

Nonlinear activation (x, y)–dynamics (A.35–5.58), describes a two–phase biological neural oscillator field, in which excitatory neural field excites inhibitory neural field, which itself reciprocally inhibits the excitatory one. (x, y)–dynamics represents a nonlinear extension of a linear, Lyapunov–stable, conservative, gradient system, defined in local neural coordinates $x^i, y_i \in V_y$ on T^*M as

$$\dot{x}^i = -\frac{\partial \Phi}{\partial y_i} = \omega^{ij} y_j - x^i, \qquad \dot{y}_i = -\frac{\partial \Phi}{\partial x^i} = \omega_{ij} x^j - y_i. \tag{5.64}$$

The gradient system (5.64) is derived from scalar, neuro-synaptic action potential $\Phi : T^*M \to \mathbb{R}$, given by a negative, smooth bilinear form in $x^i, y_i \in V_y$ on T^*M as

$$-2\Phi = \omega_{ij}x^i x^j + \omega^{ij}y_i y_j - 2x^i y_i, \qquad (i,j = 1,\dots,N), \tag{5.65}$$

which itself represents a Ψ–image of the Riemannian metrics $g : TM \to \mathbb{R}$ on the configuration manifold M.

The nonlinear oscillatory activation (x, y)–dynamics (A.35–5.58) is obtained from the linear conservative dynamics (5.64) by adding configura-tion–dependent inputs A^i and B_i, as well as sigmoid activation functions f_j and f^j, respectively. It represents an interconnected pair of excitatory and inhibitory neural fields.

Both variant–forms of learning $(\boldsymbol{\omega})$–dynamics (5.59–5.60) are given by generalized unsupervised (self–organizing) Hebbian learning scheme (see [Kos92]) in which $\dot{\omega}_{ij}$ (resp. $\dot{\omega}^{ij}$) denotes the new–update value, $-\omega_{ij}$ (resp. $-\omega^{ij}$) corresponds to the old value and $I_{ij}(x^i, y_j)$ (resp. $I^{ij}(x^i, y_j)$) is the innovation function of the symmetric 2nd order synaptic tensor-field $\boldsymbol{\omega}$. The nonlinear innovation functions I_{ij} and I^{ij} are defined by random differential Hebbian learning process (5.61–5.62). As $\boldsymbol{\omega}$ is symmetric and zero-trace coupling synaptic tensor, the conservative linear activation dynamics (5.64) is equivalent to the rule that the state of each neuron (in both neural fields) is changed in time iff the scalar action potential Φ (5.65), is lowered. Therefore, the scalar action potential Φ represents the monotonically decreasing Lyapunov function (such that $\dot{\Phi} \leq 0$) for the conservative linear dynamics (5.64), which converges to a local minimum or ground state of Φ. That is to say, the system (5.64) moves in the direction of decreasing the scalar action potential Φ, and when both $\dot{x}^i = 0$ and $\dot{y}_i = 0$ for all $i = 1,\dots,N$, the steady state is reached.

In this way, the neurodynamical $(x, y, \boldsymbol{\omega})-$system acts as tensor–invariant self–organizing (excitatory / inhibitory) associative memory machine, resembling the set of granule and Purkinje cells of cerebellum [HBB96].

The feedforward map $\Psi : M \to M$ is realized by the inputs $A^i(q)$ and $B_i(p)$ to the $(x, y, \boldsymbol{\omega})$–system, while the feedback map $\Psi^{-1} : M \to M$ is realized by the system output, i.e., the feedback–control 1−forms $u_i(x, y)$. These represent the cerebellar FC–level corrections to the covariant torques $F_i = F_i(t, q, p)$.

The tensor–invariant form of the oscillatory neurodynamical $(x, y, \boldsymbol{\omega})$–system (A.35–5.63) resembles the associative action of the granule and Purkinje cells in the tunning of the limb cortico–rubro–cerebellar recurrent network [HBB96], giving the cerebellar correction $u_i(x, y)$ to the covariant driving torques $F_i = F_i(t, q, p)$. In this way $(x, y, \boldsymbol{\omega})$–system forms the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{TA} \Rightarrow \mathcal{TA}^*$.

Cortical Control Level

Our third task is to establish the nonlinear loop functor $L = \Xi[\mathcal{L}] : \mathcal{ST} \Rightarrow \mathcal{ST}^*$ on the category $\mathcal{ST}$ of the cortical FC–level.

Recall that for the purpose of cortical control, the purely rotational biomechanical manifold M could be firstly reduced to *N–torus* and subsequently transformed to *N–cube* ('hyper–joystick'), using the following geometric techniques (see (2.4.4) above).

Denote by S^1 the constrained unit circle in the complex plane. This is an Abelian Lie group. We have two reduction homeomorphisms

$$SO(3) \gtrsim SO(2) \rhd SO(2) \rhd SO(2), \qquad \text{and} \qquad SO(2) \approx S^1,$$

where '$\rhd$' denotes the noncommutative semidirect product.

Next, let I^N be the unit cube $[0,1]^N$ in $\mathbb{R}^N$ and '$\sim$' an equivalence relation on $\mathbb{R}^N$ obtained by 'gluing' together the opposite sides of I^N, preserving their orientation. Therefore, M can be represented as the quotient space of $\mathbb{R}^N$ by the space of the integral lattice points in $\mathbb{R}^N$, that is a constrained torus T^N:

$$\mathbb{R}^N/Z^N = I^N/\sim \cong \prod_{i=1}^{N} S_i^1 \equiv \{(q^i,\, i=1,\dots,N) : \mathrm{mod}\ 2\pi\} = T^N.$$

In the same way, the momentum phase–space manifold T^*M can be represented by T^*T^N.

Conversely by 'ungluing' the configuration space we get the primary unit cube. Let '$\sim^*$' denote an equivalent decomposition or 'ungluing' relation. By the *Tychonoff product–topology theorem*, for every such quotient space there exists a 'selector' such that their quotient models are homeomorphic, that is, $T^N/\sim^* \approx A^N/\sim^*$. Therefore I_q^N represents a 'selector' for the configuration torus T^N and can be used as an N–directional '$\hat{q}$–command–space' for FC. Any subset of DOF on the configuration torus T^N representing the joints included in the general biomechanics has its simple, rectangular image in the rectified $\hat{q}$–command space – selector I_q^N, and any joint angle q^i has its rectified image $\hat{q}^i$.

In the case of an end–effector, $\hat{q}^i$ reduces to the position vector in external–Cartesian coordinates z^r $(r = 1,\dots,3)$. If orientation of the end–effector can be neglected, this gives a topological solution to the standard inverse kinematics problem.

Analogously, all momenta $\hat{p}_i$ have their images as rectified momenta $\hat{p}_i$ in the $\hat{p}$–command space – selector I_p^N. Therefore, the total momentum phase–space manifold T^*T^N obtains its 'cortical image' as the $\widehat{(q,p)}$–command space, a trivial $2N$D bundle $I_q^N \times I_p^N$.

Now, the simplest way to perform the feedback FC on the cortical $\widehat{(q,p)}$–command space $I_q^N \times I_p^N$, and also to mimic the cortical–like behavior [1,2], is to use the $2N$D fuzzy–logic controller, in pretty much the same way as in popular 'inverted pendulum' examples [Kos92, Kos96].

We propose the fuzzy feedback–control map Ξ that maps all the rectified joint angles and momenta into the feedback–control 1–forms

$$\Xi : (\hat{q}^i(t),\, \hat{p}_i(t)) \mapsto u_i(t, q, p), \tag{5.66}$$

so that their corresponding universes of discourse, $\hat{M}^i = (\hat{q}^i_{max} - \hat{q}^i_{min})$, $\hat{P}_i = (\hat{p}_i^{max} - \hat{p}_i^{min})$ and $U_i = (u_i^{max} - u_i^{min})$, respectively, are mapped as

$$\Xi : \prod_{i=1}^{N} \hat{M}M^i \times \prod_{i=1}^{N} \hat{P}_i \to \prod_{i=1}^{N} U_i. \tag{5.67}$$

The $2N$–D map Ξ (5.66–5.67) represents a *fuzzy inference system*, defined by (adapted from [IJB99b]):

1. *Fuzzification* of the crisp *rectified* and *discretized* angles, momenta and controls using Gaussian–bell membership functions

$$\mu_k(\chi) = \exp[-\frac{(\chi - m_k)^2}{2\sigma_k}], \qquad (k = 1, 2, \dots, 9),$$

where $\chi \in D$ is the common symbol for $\hat{q}^i$, $\hat{p}_i$ and $u_i(q,p)$ and D is the common symbol for M^i, $\hat{P}_i$ and $_i$; the mean values m_k of the seven partitions of each universe of discourse D are defined as $m_k = \lambda_k D + \chi_{min}$, with partition coefficients λ_k uniformly spanning the range of D, corresponding to the set of nine linguistic variables $L = \{NL, NB, NM, NS, ZE, PS, PM, PB, PL\}$; standard deviations are kept constant $\sigma_k = D/9$. Using the linguistic vector L, the 9×9 FAM (fuzzy associative memory) matrix (a 'linguistic phase–plane'), is heuristically defined for each human joint, in a symmetrical weighted form

$$\mu_{kl} = \varpi_{kl} \exp\{-50[\lambda_k + u(q,p)]^2\}, \qquad (k, l = 1, 2, \dots, 9)$$

with weights $\varpi_{kl} \in \{0.6, 0.6, 0.7, 0.7, 0.8, 0.8, 0.9, 0.9, 1.0\}$.
2. *Mamdani inference* is used on each FAM–matrix μ_{kl} for all human joints: (i) $\mu(\hat{q}^i)$ and $\mu(\hat{p}_i)$ are combined inside the fuzzy IF–THEN rules using AND (Intersection, or Minimum) operator,

$$\mu_k[\bar{u}_i(q,p)] = \min_l\{\mu_{kl}(\hat{q}^i),\, \mu_{kl}(\hat{p}_i)\}.$$

(ii) the output sets from different IF–THEN rules are then combined using OR (Union, or Maximum) operator, to get the final output, fuzzy–covariant torques,

$$\mu[u_i(q,p)] = \max_k\{\mu_k[\bar{u}_i(q,p)]\}.$$

3. *Defuzzification* of the fuzzy controls $\mu[u_i(q,p)]$ with the 'center of gravity' method
$$u_i(q,p) = \frac{\int \mu[u_i(q,p)]\, du_i}{\int du_i},$$
to update the crisp feedback–control 1−forms $u_i = u_i(t,q,p)$. These represent the cortical FC–level corrections to the covariant torques $F_i = F_i(t,q,p)$.

Operationally, the construction of the cortical $\widehat{(q,p)}$–command space $I_q^N \times I_p^N$ and the $2N$D feedback map Ξ (5.66–5.67), mimic the regulation of locomotor conditioned reflexes by the motor cortex [HBB96], giving the cortical correction to the covariant driving torques F_i. Together they form the nonlinear loop functor $\mathcal{NL} = \Xi[\mathcal{L}] : \mathcal{ST} \Rightarrow \mathcal{ST}^*$.

A sample output from the leading human–motion simulator, *Human Biodynamics Engine* (developed by the authors in Defence Science & Technology Organisation, Australia), is given in Figure 5.5, giving the sophisticated 264 DOF analysis of adult male running with the speed of 5 m/s.

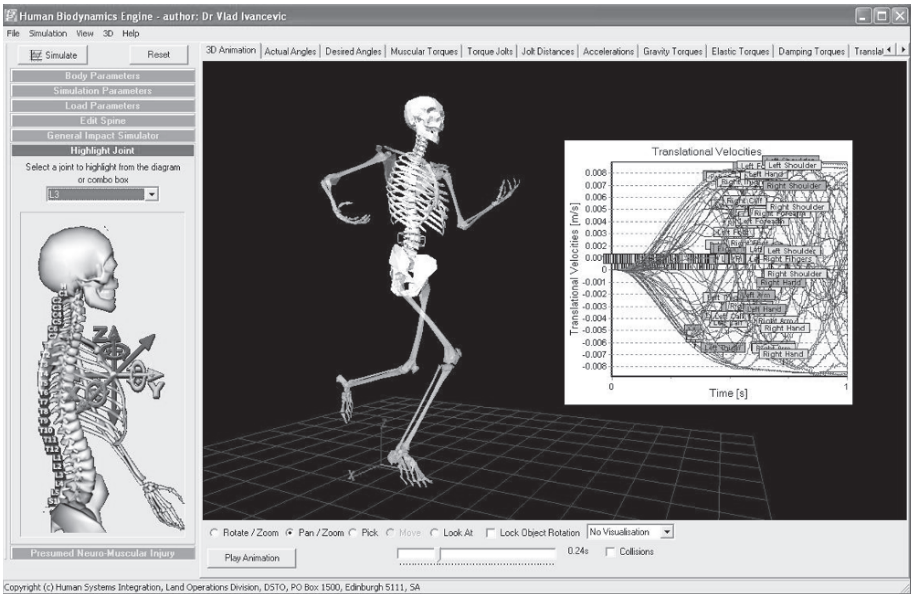

Fig. 5.11. Sample output from the Human Biodynamics Engine: running with the speed of 5 m/s.