Frequently Asked Questions

How does the velocity-based linearisation approach differ from the work by Kaminer and Rugh ?

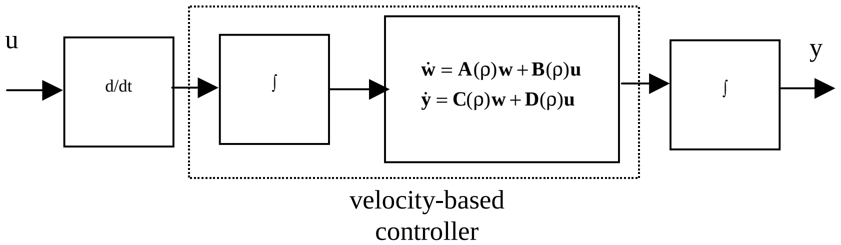
The approach of Kaminer and Rugh simply ensures that the equilibrium linearisations of the implemented nonlinear controller match the linear controllers generated by the classical gain-scheduling design procedure. This approach is firmly rooted in classical gain-scheduling ideas and, in addition to its own particular difficulties (LEITH, D.J., LEITHEAD, W.E., 1998, Comments on "Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant". Automatica, 34, 1041-1043; 2000, Further Comments on "Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant". Automatica, 36, 173-174), suffers from the usual limitations of classical gain-scheduling approaches including: in particular, an inherent restriction to near equilibrium operation (arising from the use of equilibrium linearisations). Velocity-based linearisation techniques provide a new framework which generalises linearisation theory itself. There is a velocity-based linearisation associated with every operating point (explicitly including off-equilibrium points) not only equilibrium points. The applications of this new framework are very wide ranging. In the context of gain-scheduling design, for example, the restriction to near equilibrium operation can be completely relaxed

*Doesn't the velocity-based approach just amount to scheduling on filtered quantities as studied in Shamma (SHAMMA, J.S., ATHANS, M., 1992, Gain Scheduling: Potential Hazards and Possible Remedies. IEEE Control Systems Magazine, **12**, 101-107) and others i.e. use filtered state/input for scheduling rather than instantaneous values ?*

No. On the face of it, there is little in common between these approaches.

Is it really admissible to include differentiators in the velocity-based formulation ?

Differentiation of noisy signals is most certainly not advocated. The differentiation in the velocity-based formulation is purely formal in nature. The velocity-based formulation is used for analysis and design purposes. When it comes to implementation, it is necessary to adopt a realisation which avoids explicit differentiation. This can be achieved in many ways but is particularly straightforward in a control systems context when the controller contains integral action (as is usually the case). By partitioning the controller dynamics such that the pure integrator acts on the input to the controller, the integrator may be formally combined with the differentiator in the velocity-based formulation (this is a formal step, no unstable pole-zero cancellation is required in the implemented controller) leading to an implementation without explicit differentiation action as required.



Doesn't the presence of an integrator-differentiator pair in the velocity-based linearisation mean that it is uncontrollable ?

For analysis and design purposes, the input to the velocity based linearisation is dr/dt, not the original input, r. With regard to implementation of a velocity-based controller, the combination of the integrator and differentiator noted above is purely formal (carried out analytically at the design stage). No unstable pole-zero cancellation is required and there is no associated uncontrollable/unobservable mode.

What issues are associated with the apparent increase in order of the velocity-based linearisation relative to the original dynamics ?

The velocity-based formulation, obtained by differentiating, associated with the nonlinear dynamics

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{r}) \qquad (1)$$

is

$$\dot{\mathbf{x}} = \mathbf{w}$$
$$\dot{\mathbf{w}} = \nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x},\mathbf{r})\mathbf{w} + \nabla_{\mathbf{r}}\mathbf{F}(\mathbf{x},\mathbf{r})\dot{\mathbf{r}} \tag{2}$$

Clearly, if the original dynamics involve n states then the velocity-based formulation, (2), involve 2n states. The following points may be noted.

1. The initial conditions for the **w** and **x** states in (2) are not independent but, from (1), are related by

   $$\mathbf{w}=\mathbf{F}(\mathbf{x},\mathbf{r}) \tag{3}$$

   When the initial conditions satisfy this constraint, it follows from the integrability (2) that the solutions are strictly confined to an n-dimensional manifold within the 2n dimensional state space. Here, integrability refers to the relationship between the coefficients $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x},\mathbf{r}), \nabla_{\mathbf{r}}\mathbf{F}(\mathbf{x},\mathbf{r})$ in (2) and $\mathbf{F}(\mathbf{x},\mathbf{r})$ in (1) rather than integrability of the dynamics with respect to time. It is emphasised that the n-dimensional nature of the velocity-based dynamics is an *exact* property which follows directly from the integrability of the velocity-based equations. The n-dimensional manifold is not associated with pseudo-order reduction nor with a singular perturbation type of fast/slow time-scale separation. When (3) is explicitly invertible in the sense that **x** may be expressed directly in terms of **w** and **r**, then the velocity-based dynamics, (2), may be explicitly reformulated in n-dimensional form as

   $$\dot{\mathbf{w}} = \nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x}(\mathbf{w},\mathbf{r}),\mathbf{r})\mathbf{w} + \nabla_{\mathbf{r}}\mathbf{F}(\mathbf{x}(\mathbf{w},\mathbf{r}),\mathbf{r})\dot{\mathbf{r}} \tag{4}$$

   In a control design context, the controller velocity-based linearisation family will typically be parameterised by the same scheduling variable, ρ, as the plant family. When a measurement of the scheduling variable is available, the controller may be directly scheduled on this quantity. This automatically ensures that the constraint, (3), is satisfied.

2. The velocity-based framework is an open one in the sense that it place no restrictions on the design methodology employed. It is obviously possible to conceive of design approaches which, for whatever reason, neglect the constraint (3). This implies the use of an over-bounding approach whereby the exact system is subsumed within some larger class of systems to which the design procedure is then applied. Of course, such an approach may, in general, introduce a degree of conservativeness. The additional degree of freedom introduced by neglecting (3) can be made explicit. Let $\mathbf{x_o}$, $\mathbf{w_o}$, $\mathbf{r_o}$ denote, respectively, the initial conditions for **x**, **w** and **r** in the velocity-based formulation, (2). It follows immediately from the integrability of (2) that

   $$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x},\mathbf{r}) + \mathbf{k} \tag{5}$$

   where $\mathbf{k}=\mathbf{w_o}-\mathbf{F}(\mathbf{x_o},\mathbf{r_o})$. When the constraint, (3), is satisfied, **k** is identically zero. Otherwise, the original dynamics are altered by the constant offset, **k**. That is, the class of over-bounding systems consists precisely of the class of systems (5) associated with an appropriate set of offsets, **k**. It should be noted that, once particular initial conditions have been chosen and the value of **k** is thereby fixed, the velocity-based dynamics, (2), evolve on an n-dimensional manifold within the 2n dimensional state space.

3. It is often useful to reformulate the dynamics, (1), as

   $$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{r} + \mathbf{f}(\rho) \tag{6}$$

   where **A**, **B** are appropriately dimensioned constant matrices, $\mathbf{f}(\bullet)$ is a nonlinear function and $\rho(\mathbf{x},\mathbf{r})$ embodies the nonlinear dependence of the dynamics on the state and input with $\nabla_{\mathbf{x}}\rho$, $\nabla_{\mathbf{r}}\rho$ constant. Trivially, this reformulation can always be achieved by letting $\rho = [\mathbf{x}^T \;\; \mathbf{r}^T]^T$. However, the nonlinearity of the system is frequently dependent on only a subset of the states and inputs, in which case the dimension of ρ is reduced. The formulation, (6), defines a scheduling variable ρ which explicitly embodies the nonlinear dependence of the dynamics. Differentiating, the associated velocity-based formulation is

   $$\dot{\mathbf{x}} = \mathbf{w} \tag{7}$$
   $$\dot{\mathbf{w}} = (\mathbf{A}+\nabla\mathbf{f}(\rho))\mathbf{w} + (\mathbf{B}+\nabla\mathbf{f}(\rho))\,\dot{\mathbf{r}} \tag{8}$$

   Since the **w** dynamics depend on **x** only through the scheduling variable, ρ, (7)-(8) may be simplified to

   $$\dot{\rho} = \mathbf{M}\mathbf{w} + \mathbf{N}\dot{\mathbf{r}} \tag{9}$$
   $$\dot{\mathbf{w}} = (\mathbf{A}+\nabla\mathbf{f}(\rho))\mathbf{w} + (\mathbf{B}+\nabla\mathbf{f}(\rho))\,\dot{\mathbf{r}} \tag{10}$$

   where $\mathbf{M}=\nabla_{\mathbf{x}}\rho$, $\mathbf{N}=\nabla_{\mathbf{r}}\rho$. Letting q denote the dimension of ρ, it can be seen that the dynamics, (7)-(8), involve n+q states rather than 2n states. This formulation therefore often reduces the additional degrees of freedom introduced by over-bounding approaches. For example, when the original dynamics, (1), are such that every value of $[\rho^T\ \mathbf{w}^T]^T$ is reachable (a trivial example is when ρ depends only on the input, $\mathbf{r}$), then over-bounding approaches are tight and involve no conservativeness.