Full-envelope Flight Control Using an Adaptive Critic Neural Network

Silvia Ferrari

Advisor: Prof. Robert F. Stengel

Princeton University

FAA/NASA Joint University Program on Air Transportation, MIT, Cambridge, MA

October 10, 2001

Introduction

- Classical/neural synthesis of control systems
  Linear control theory
  Artificial neural networks

- Adaptive critics
  Learn in real time
  Cope with noise
  Cope with many variables
  Plan over time in a complex way
  ...

- Adaptation takes place during every time interval:

Action network takes immediate control action

Critic network estimates projected cost

Motivation

- Provide full envelope control

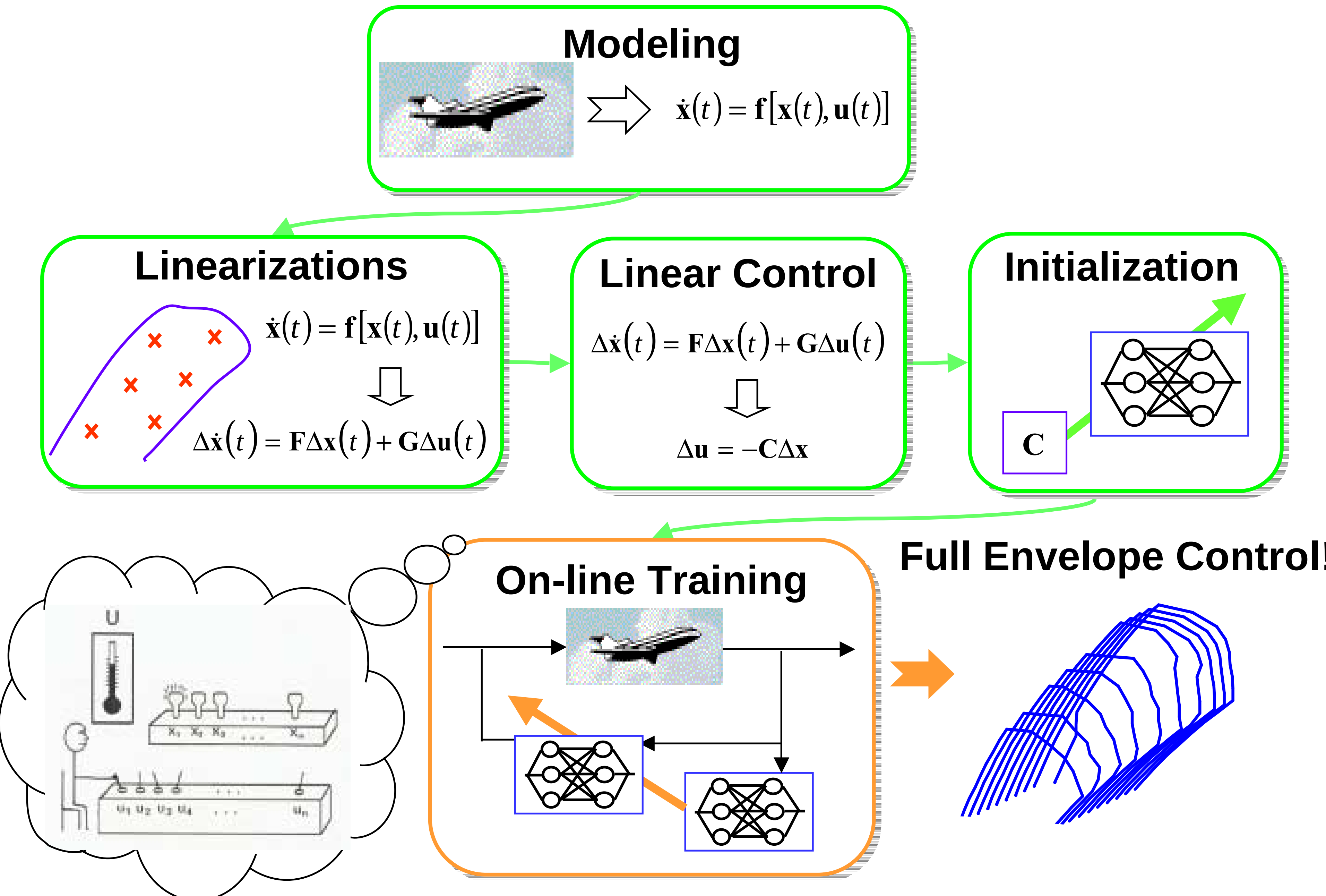
- Multiphase learning
  Initialization (off-line), motivated by linear controllers
  On-line training, during full-scale simulations or aircraft testing

- On-line training improves performance w.r.t. linear controllers:
  Differences between actual and assumed dynamic models
  Nonlinear effects not captured in linearizations

- Potential applications:
  Incorporate pilot's knowledge into controller *a-priori*
  Uninhabited air vehicles control
  Aerobatic flight control

Table of Contents

Aircraft Control Design Approach



Linear Control Design

Linearizations:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)]$$

$$\Downarrow$$

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\Delta\mathbf{u}(t)$$

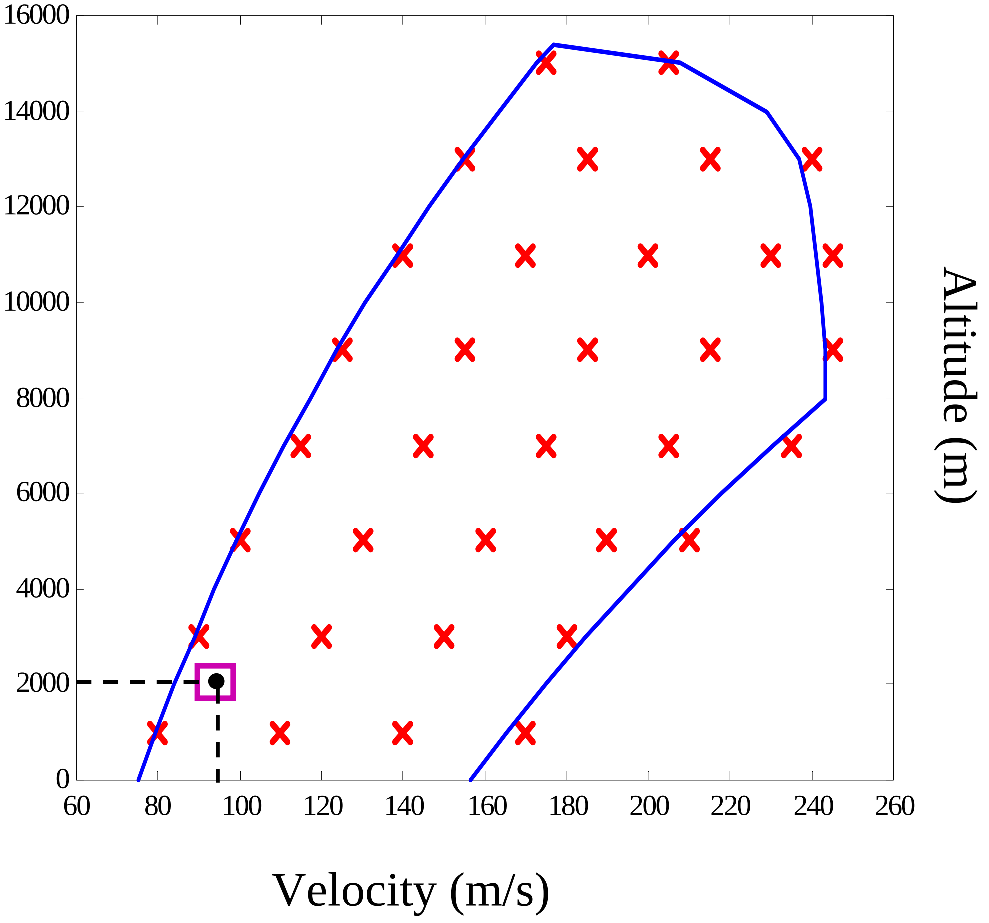
$$\Downarrow$$

$$\begin{cases} \Delta\dot{\mathbf{x}}_L(t) = \mathbf{F}_L\Delta\mathbf{x}_L(t) + \mathbf{G}_L\Delta\mathbf{u}_L(t) \\ \Delta\dot{\mathbf{x}}_{LD}(t) = \mathbf{F}_{LD}\Delta\mathbf{x}_{LD}(t) + \mathbf{G}_{LD}\Delta\mathbf{u}_{LD}(t) \end{cases}$$

Linear control design:

- Longitudinal
- Lateral-directional

Aircraft Flight Envelope $\{V, H\}$: ($\gamma = \mu = \beta = 0$)



Proportional Integral Linear Control Law

Quadratic cost function:

$$J = \lim_{t_f \to \infty} \frac{1}{2} \int_0^{t_f} L\left[\mathbf{x}_a(\tau), \tilde{\mathbf{u}}(\tau)\right] d\tau$$

$$= \lim_{t_f \to \infty} \frac{1}{2} \int_0^{t_f} \left[\mathbf{x}_a^T(\tau)\mathbf{Q}\mathbf{x}_a(\tau) + 2\mathbf{x}_a^\tau(\tau)\mathbf{M}\tilde{\mathbf{u}}(\tau) + \tilde{\mathbf{u}}^T(\tau)\mathbf{R}\tilde{\mathbf{u}}(\tau)\right] d\tau$$

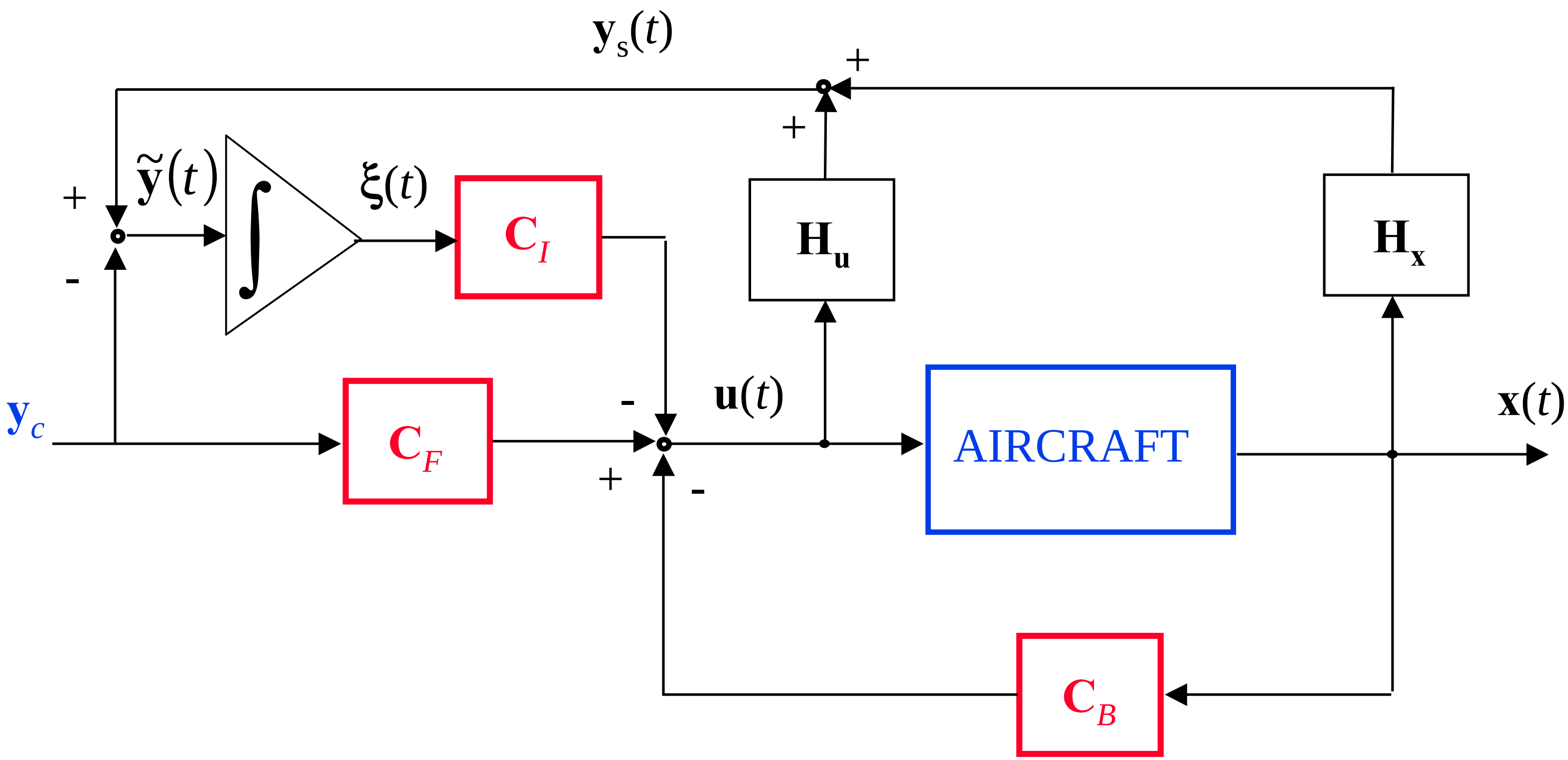
where $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_c$, $\boldsymbol{\xi}(t) = \int_0^t \tilde{\mathbf{y}}(\tau) d\tau$, and $\mathbf{x}_a \equiv \left[\tilde{\mathbf{x}}^T \quad \boldsymbol{\xi}^T\right]^T$

***Minimizing Linear Control Law*:**

$$\tilde{\mathbf{u}}(t) = -\mathbf{C}\mathbf{x}_a(t) = -\mathbf{C}_B\tilde{\mathbf{x}}(t) - \mathbf{C}_I\boldsymbol{\xi}(t)$$

Linear Proportional-Integral Controller

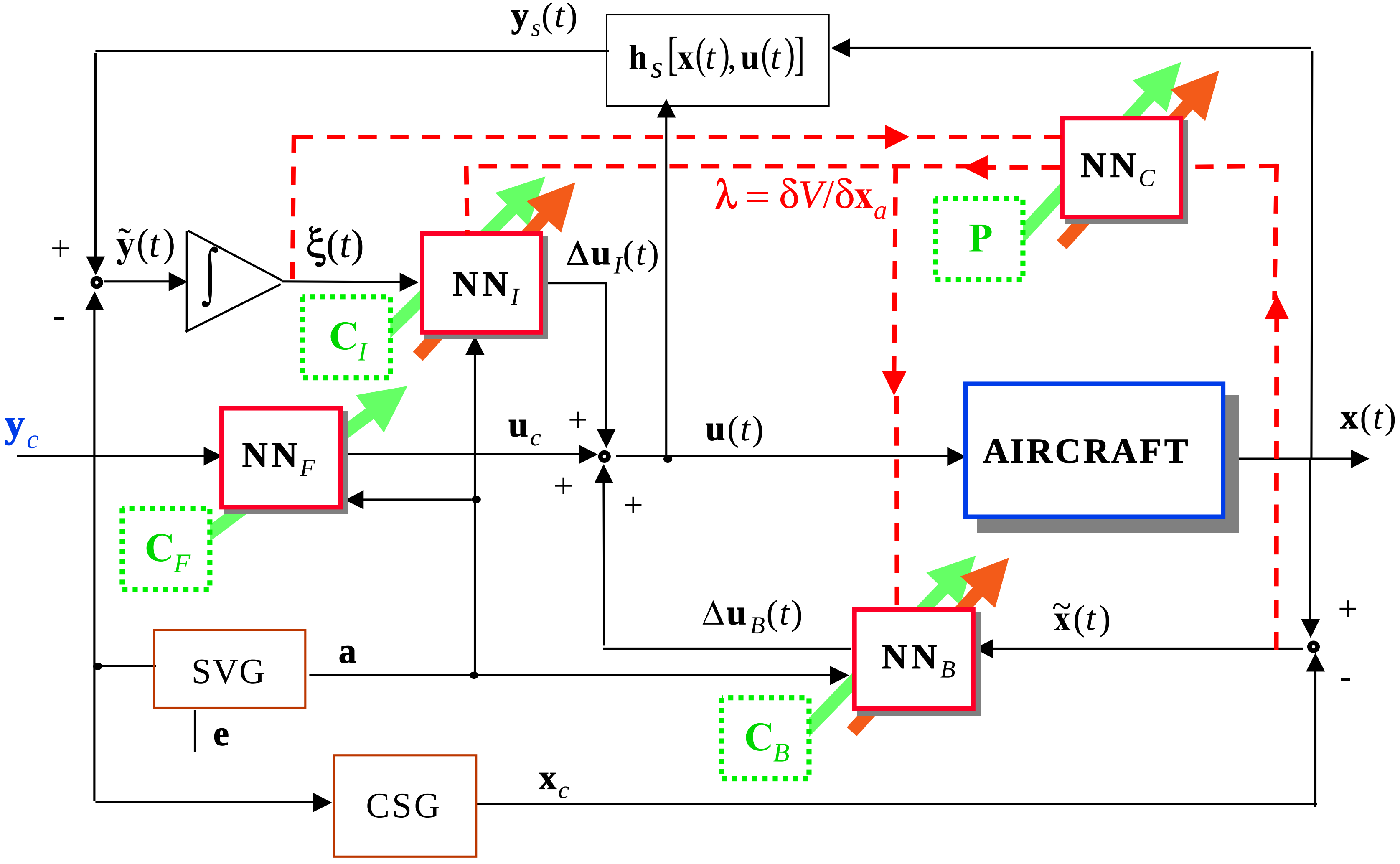
Closed-loop stability: $\mathbf{x}(t) \rightarrow \mathbf{x}_c, \quad \mathbf{u}(t) \rightarrow \mathbf{u}_c, \quad \tilde{\mathbf{y}}(t) \rightarrow 0$



Omitting Δ's, for simplicity:

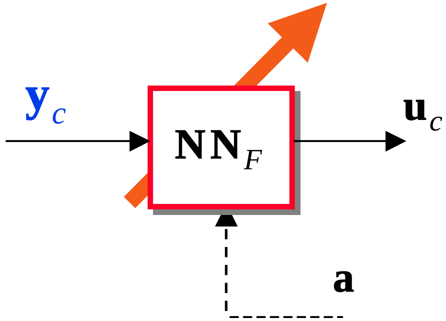
$\tilde{\mathbf{y}}(t) = \mathbf{y}_S(t) - \mathbf{y}_C, \quad \tilde{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_C, \ldots, \qquad \mathbf{y}_c$ = desired output, $(\mathbf{x}_c, \mathbf{u}_c)$ = set point.

Proportional-Integral Neural Network Controller



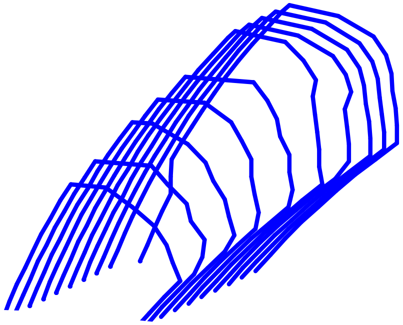
Where: $\tilde{\mathbf{y}}(t) \to 0$, $\mathbf{y}_S(t) \to \mathbf{y}_C$, : Algebraic Training, : On-line Training.

Role of Forward Neural Network in the Control System



- $\mathbf{NN}_F$ represents commanded trim control settings, $\mathbf{u}_c = \mathbf{NN}_F(\mathbf{y}_c, \mathrm{a})$,.

  scheduled by **a**, throughout <u>full flight envelope</u>:
  $OR = \{V, H, \gamma, \mu, \beta\}$



- Trim settings commanded by $\mathbf{y}_c$ are defined as

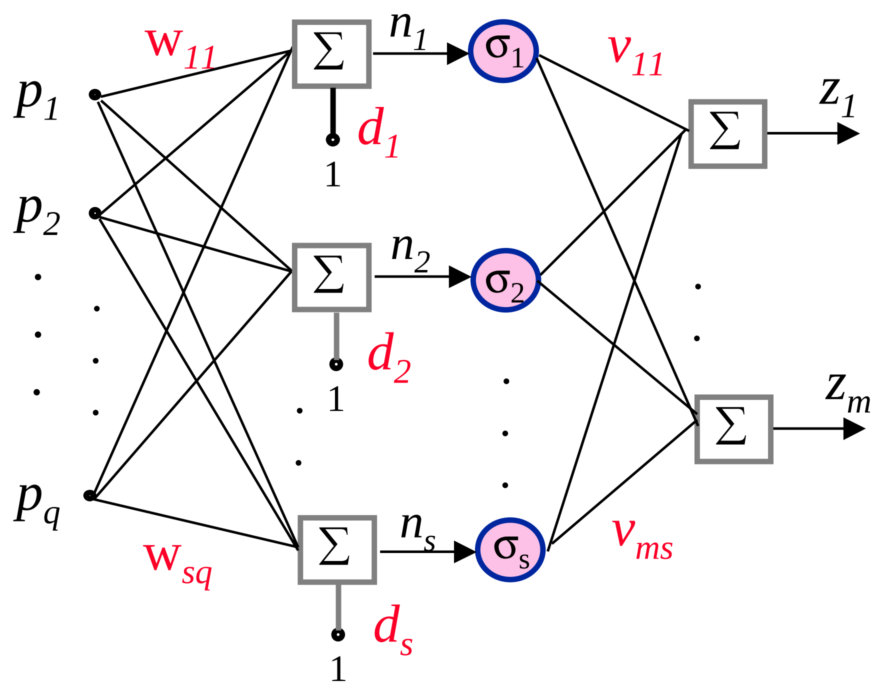
  $$\mathbf{f}[\mathbf{x}_c(t), \mathbf{u}_c(t), \mathbf{p}(t)] = \mathbf{0},$$

Trim Map Modelling:

$$U_c(\mathbf{x}_c, \mathbf{p}) \equiv \left\{ \mathbf{u}_c^k : \quad \left(\mathbf{x}_c^k, \mathbf{p}^k\right) \in OR, \quad \mathbf{f}\left(\mathbf{x}_c^k, \mathbf{p}^k, \mathbf{u}_c^k\right) = 0, \quad k = 1, \quad \ldots, \quad p \right\}$$

One Layer Sigmoidal Forward Neural Network

Training set:
$\{\mathbf{y}_c^{\,k}, \mathbf{a}^k, \mathbf{u}_c^{\,k}\}_{k\,=\,1,\,\ldots,\,p}$



Output: $\mathbf{z} = \mathbf{NN}(\mathbf{p})$

Input: $\mathbf{p}$

Input weight matrix:

$\mathbf{W} \equiv \{w_{ij}\}, (s \times q)$

Output weight matrix:

$\mathbf{V} \equiv \{v_{ij}\}, (m \times s)$

Input bias: $\mathbf{d}$

q - inputs, s - nodes, m - outputs

$$\sigma(n) \equiv \frac{\left(e^n - 1\right)}{\left(e^n + 1\right)}, \quad \begin{cases} -\infty < n < \infty \\ -1 < \sigma(n) < 1 \end{cases}$$

Forward Neural Network Initialization Equations

Requirements:

$\mathbf{z}(\mathbf{y}_c{}^k, \mathbf{a}_k) = \mathbf{u}_c{}^k, \forall\, k$

Training set:

$\{\mathbf{y}_c{}^k, \mathbf{a}^k, \mathbf{u}_c{}^k\}_{k = 1, \ldots, p}$

Initialization equations:

$$u_{l_c}^k = \sum_{i=1}^{s} v_{li}\,\sigma\left(n_i^k\right), \qquad l = 1, \ldots, m, \text{ and } k = 1, \ldots, p$$

$$n_i^k = \sum_{j=1}^{q} w_{ij}\, p_j^k + d_i, \quad \text{with} \quad \mathbf{p}^k = [\mathbf{y}_c^{k^T} \quad \mathbf{a}^{k^T}]^T$$

In matrix form, assuming n_i^k is known:

$$\boxed{\mathbf{u}_{l_c} = \mathbf{S}\mathbf{v}_l}$$

$$\mathbf{S} \equiv \begin{bmatrix} \sigma\left(n_1^1\right) & \sigma\left(n_2^1\right) & \cdots & \sigma\left(n_s^1\right) \\ \sigma\left(n_1^2\right) & \sigma\left(n_2^2\right) & \cdots & \sigma\left(n_s^2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma\left(n_1^p\right) & \sigma\left(n_2^p\right) & \cdots & \sigma\left(n_s^p\right) \end{bmatrix}$$

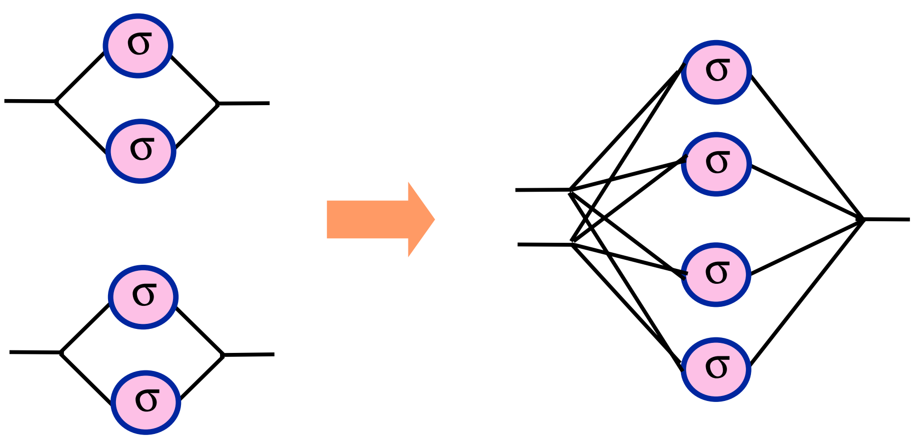
Forward Neural Network Initialization Equations Solution

Existence of solution:

- No solution iff rank ($\mathbf{S}$ | $\mathbf{u}_{lc}$) $\neq$ rank ($\mathbf{S}$)
- Unique solution iff rank ($\mathbf{S}$ | $\mathbf{u}_{lc}$) = rank ($\mathbf{S}$) = s
- An ($s - r$)-family of solutions iff rank ($\mathbf{S}$ | $\mathbf{u}_{lc}$) = rank ($\mathbf{S}$) = $r < s$

Suggested methods of solution:

- Reduce number of nodes until $s = r$, i.e., eliminate columns in $\mathbf{S}$
- Exact algebraic solution where s is chosen equal to p (square $\mathbf{S}$)
- Approximate solution using pseudoinverse, $\mathbf{v}_l = \mathbf{S}^{PI}\mathbf{u}_{lc}$, with $s < p$:



Network Superpositi on

Initialized Full-envelope Forward Neural Network

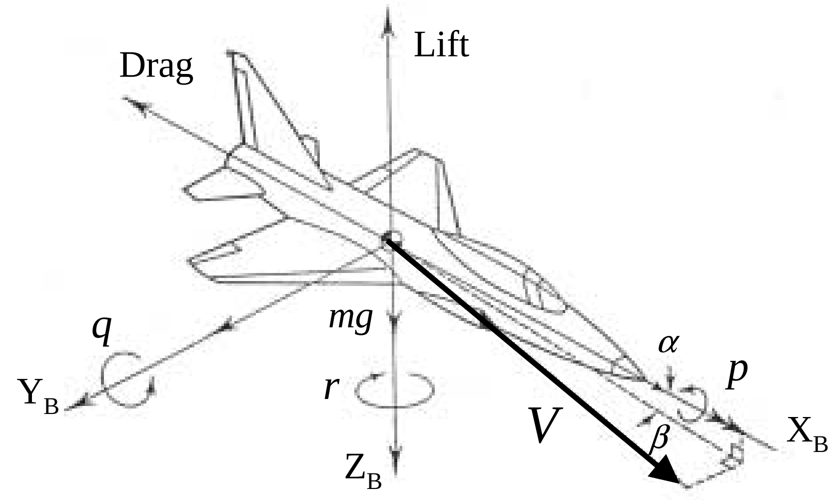
Full-envelope command input:

$$\mathbf{y}_c = [V_c \gamma_c \mu_c \beta_c]^T$$

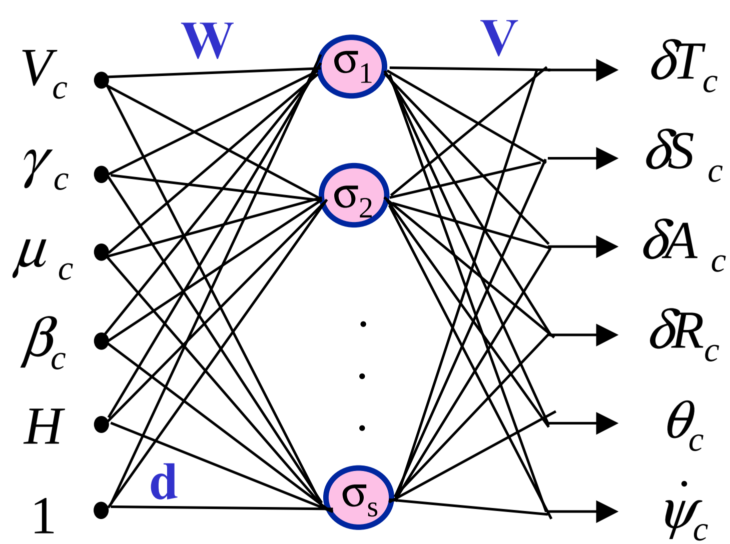
$$\Downarrow$$

$$\mathbf{x}_c = [V_c \gamma_c q_c \theta_c r_c \beta_c p_c \mu_c]^T$$

$$\mathbf{u}_c = [\delta T_c \ \delta S_c \ \delta A_c \ \delta R_c]^T$$



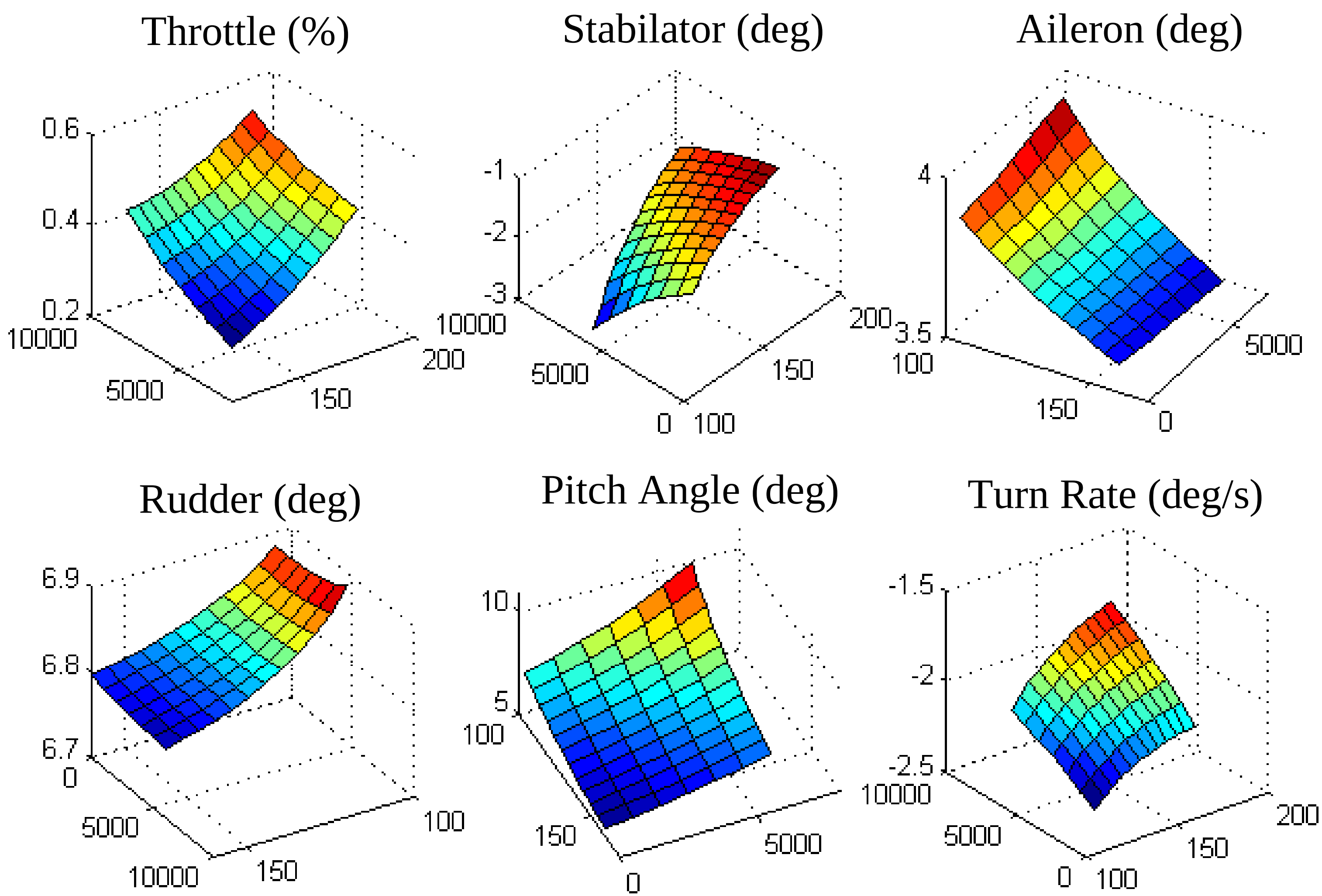
Full-envelope forward neural network (NN_F):



s = 200 (nodes)

Actual Full-envelope Aircraft Trim Map

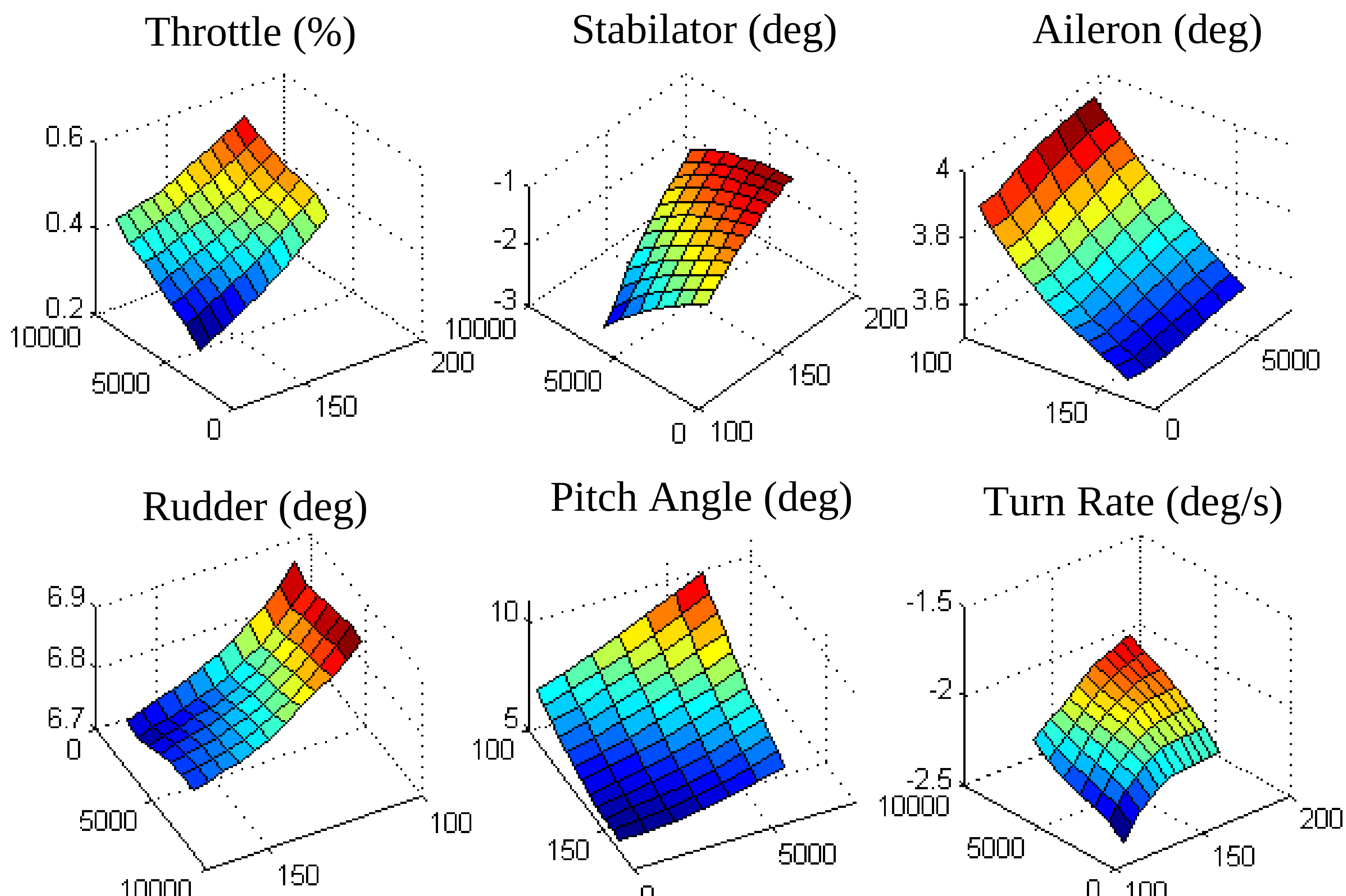
Trim settings (from $\mathbf{f}[\mathbf{x}_c(t), \mathbf{u}_c(t), \mathbf{p}(t)] = \mathbf{0}$) vs. V (m/s) and H (m):



constant γ, μ, and β

Full-envelope Neural Network Modeling of Aircraft Trim Map

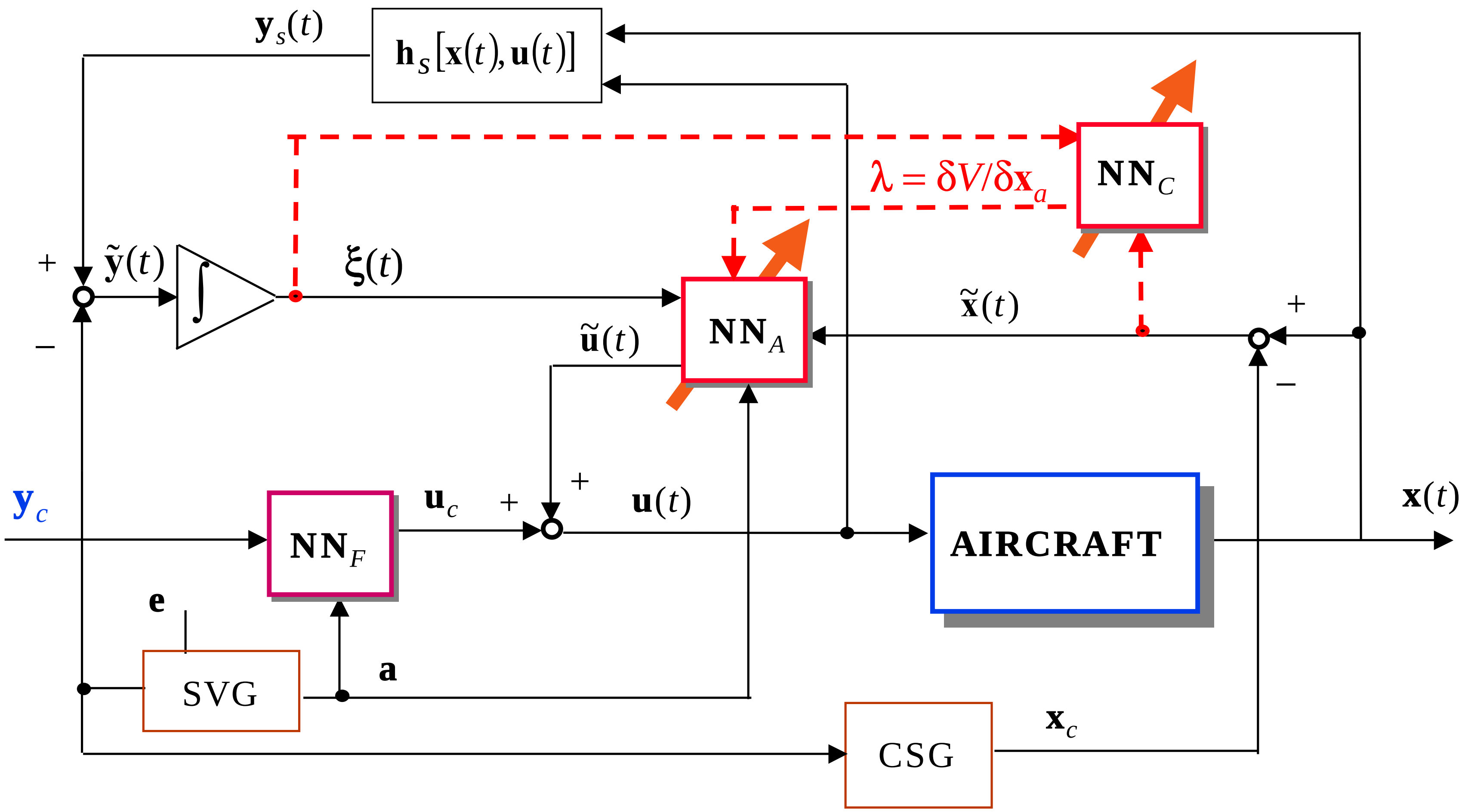
Forward neural network output vs. *V* (m/s) and *H* (m):



Error ~ $O(10^{-5})$

constant γ, μ, and β

Proportional-Integral Neural Network Controller: Action and Critic Networks Implementation

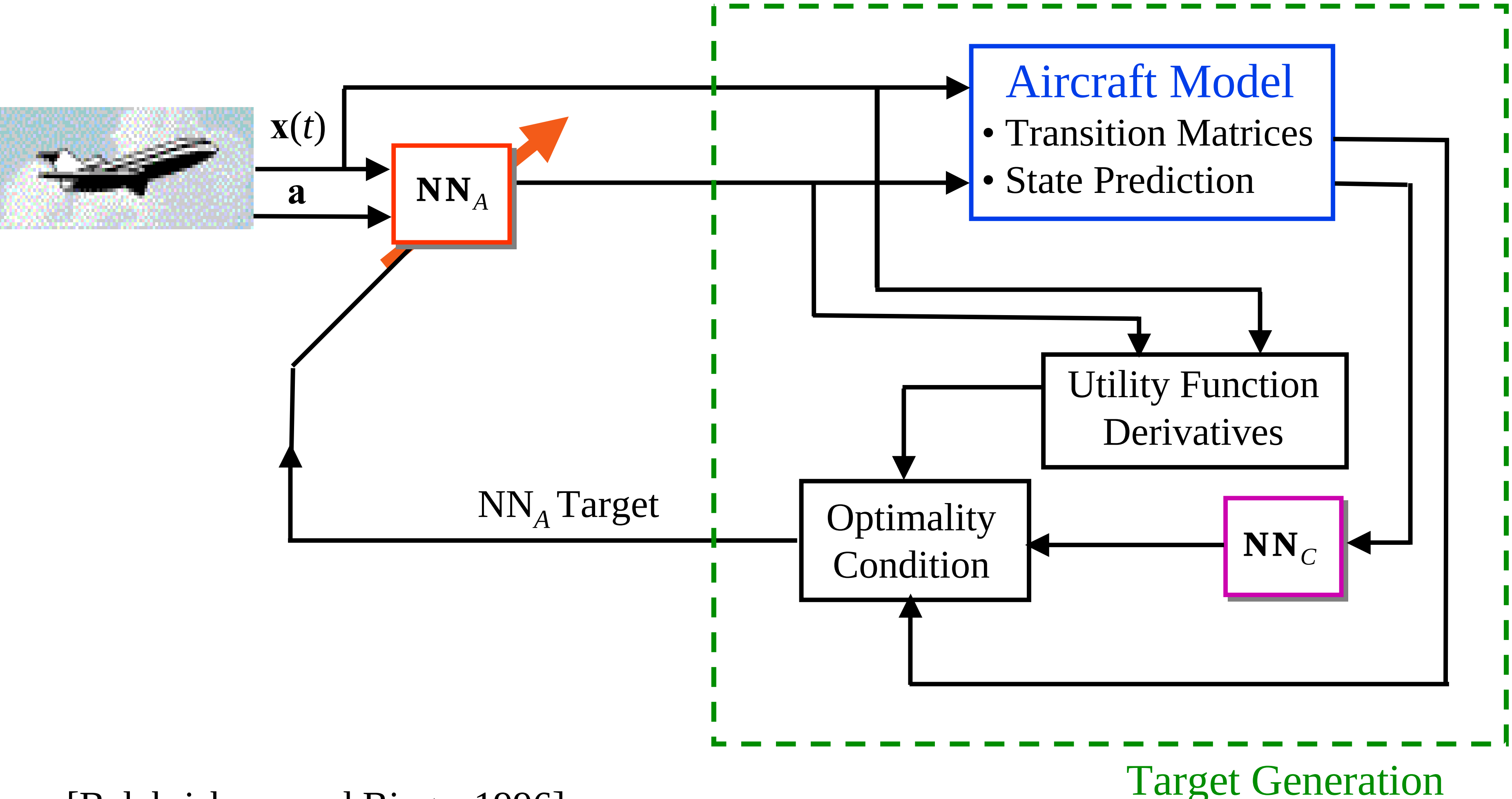


$$V(t) = -\lim_{t_f \to \infty} \frac{1}{2} \int_{t_f}^{t} \left[\mathbf{x}_a^T(\tau)\mathbf{Q}\mathbf{x}_a(\tau) + 2\mathbf{x}_a^{\tau}(\tau)\mathbf{M}\tilde{\mathbf{u}}(\tau) + \tilde{\mathbf{u}}^T(\tau)\mathbf{R}\tilde{\mathbf{u}}(\tau) \right] d\tau,$$

: On-line Training

Adaptive Critic Implementation:
Action Network On-line Training

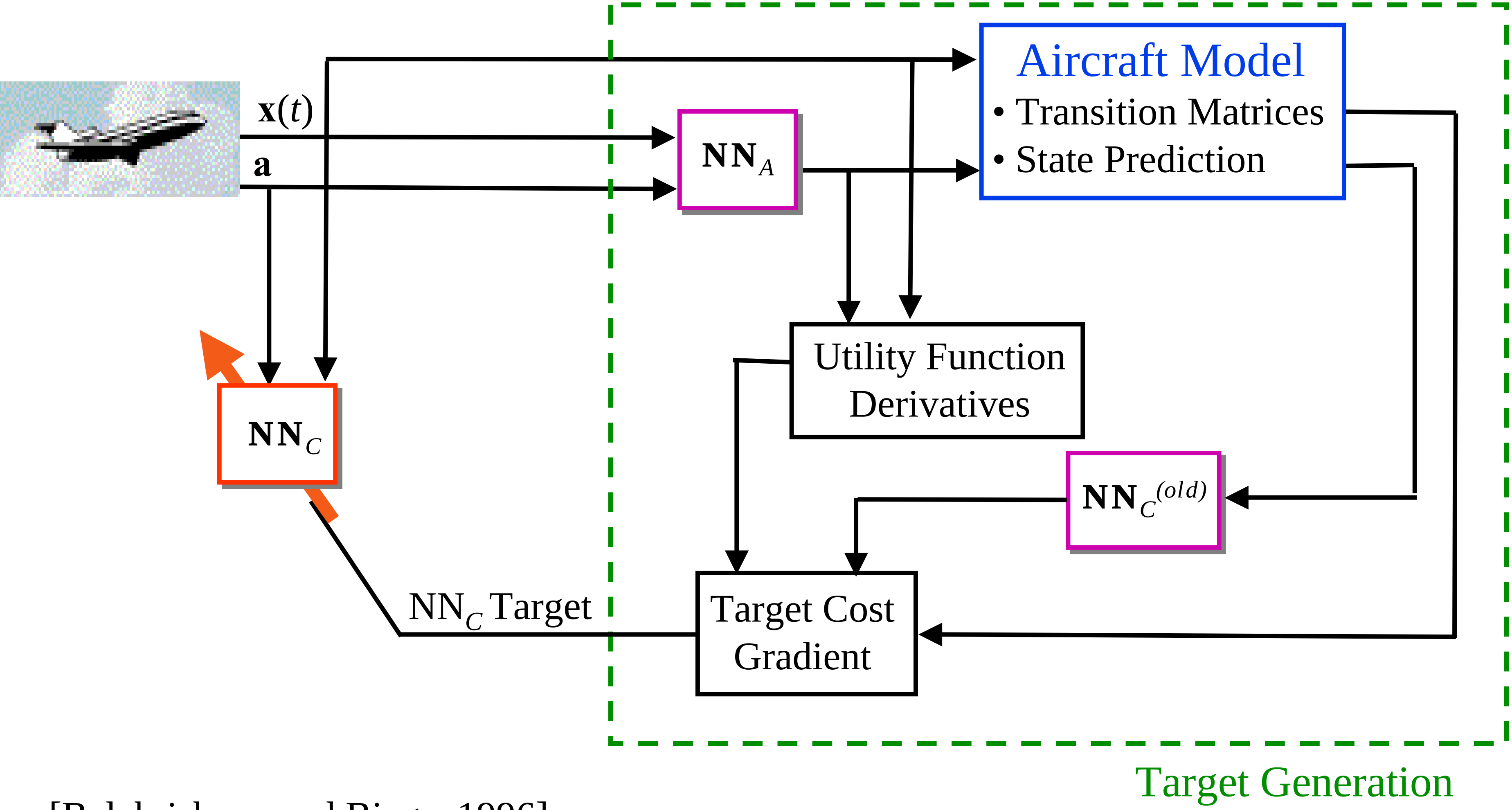
Train action network, at time *t*, holding the critic parameters fixed



[Balakrishnan and Biega, 1996]

Adaptive Critic Implementation:
Critic Network On-line Training

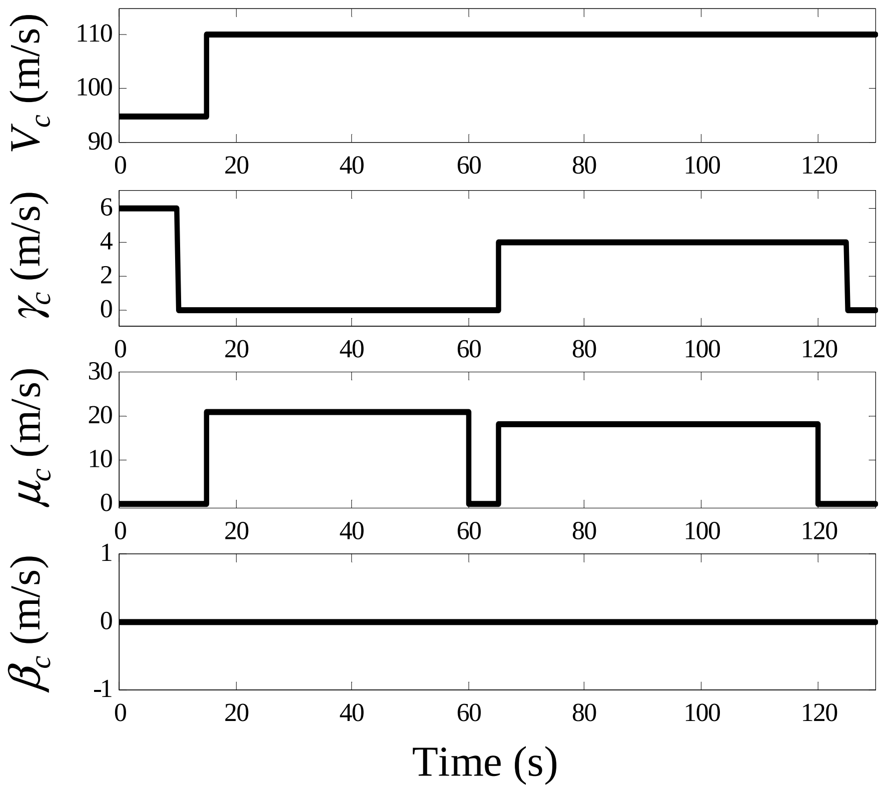
Train critic network, at time t, holding the action parameters fixed



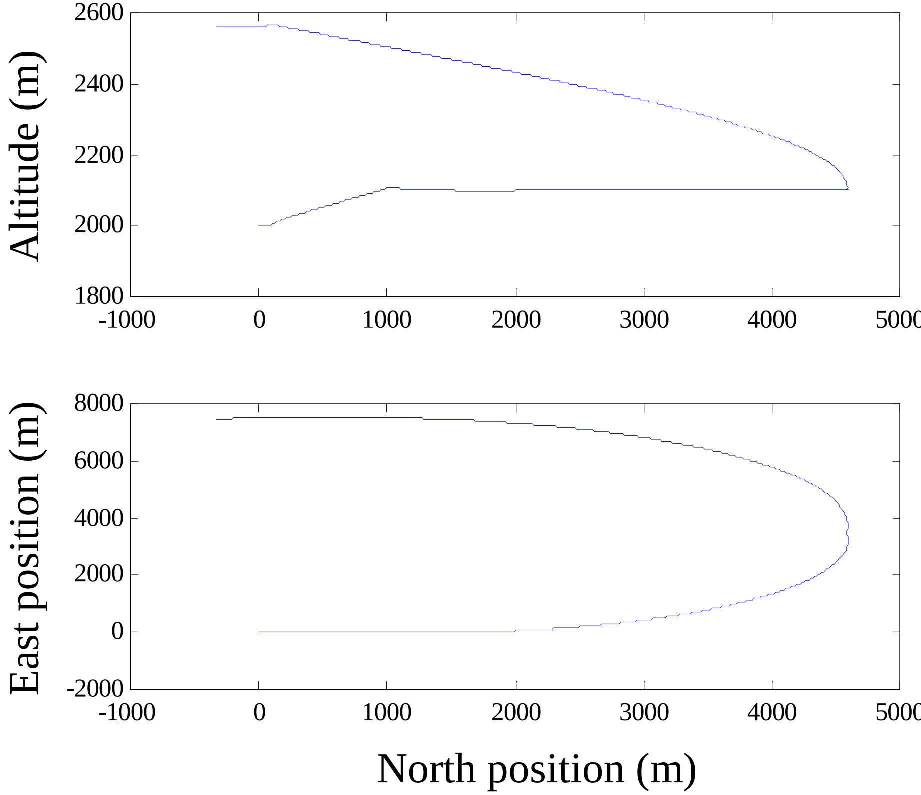
[Balakrishnan and Biega, 1996]

Example: Full-envelope Large-angle Aircraft Maneuver

Command-input time history:



Resulting trajectory:



Summary and Conclusions

- Adaptive critic flight control design:

  ❖ Algebraic pre-training based on available linear control knowledge

  ❖ On-line training during simulations (full envelope, severe conditions)

  Objective: improve global aircraft control performance

- Achievements:

  Systematic approach for designing nonlinear control systems

  Innovative neural network (off-line and on-line) training techniques

- Results: improved performance during full-envelope large-angle maneuvers

Future Work:

Continue testing and implement constrained adaptive critic designs!