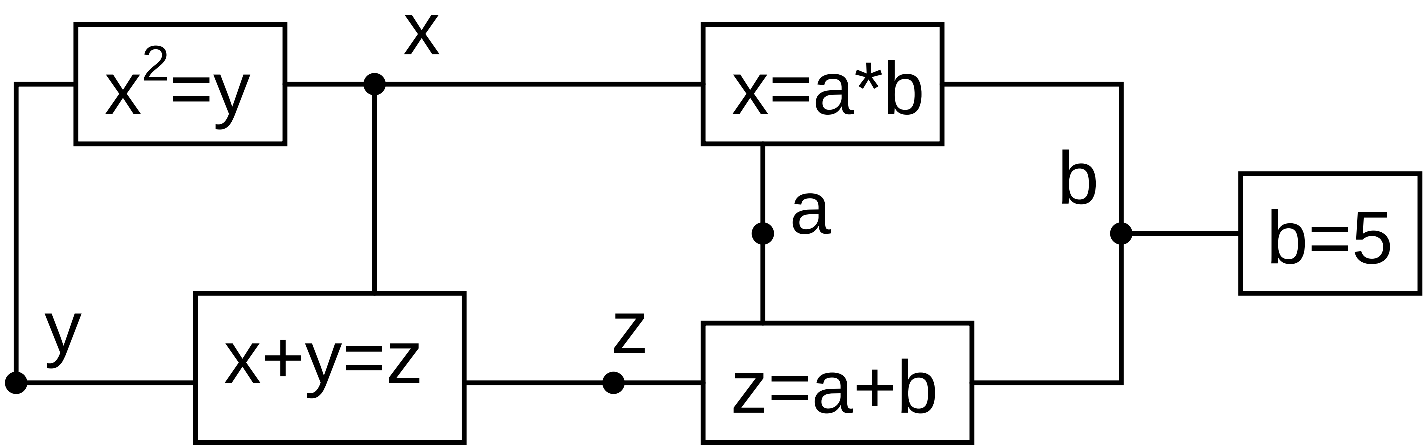
Constraint Satisfaction Modules

A Methodology for Analog Circuit Design

Piotr Mitros

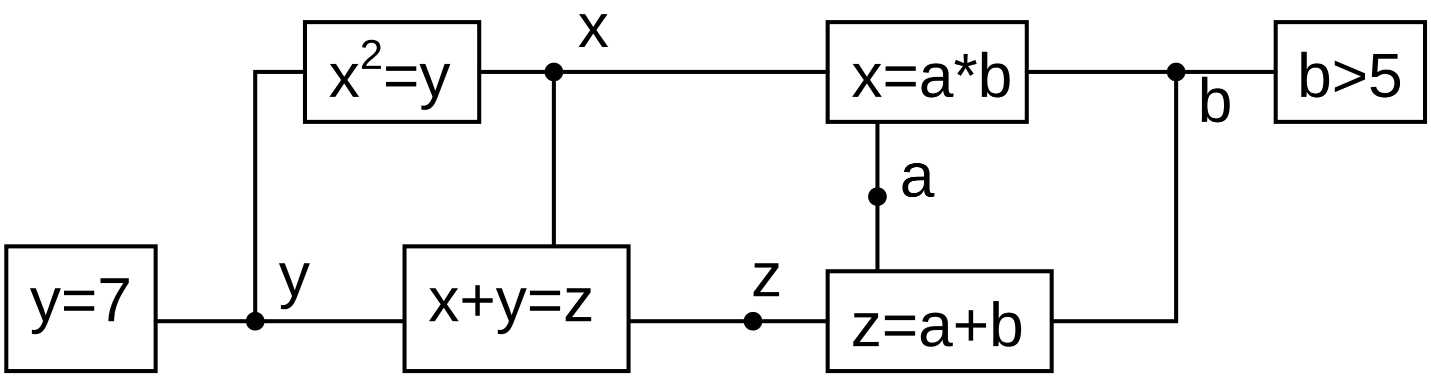
pmitros@mit.edu

Massachusetts Institute of Technology


x
x^2=y
x=a*b
b
a
b=5
y
x+y=z
z
z=a+b

Overview

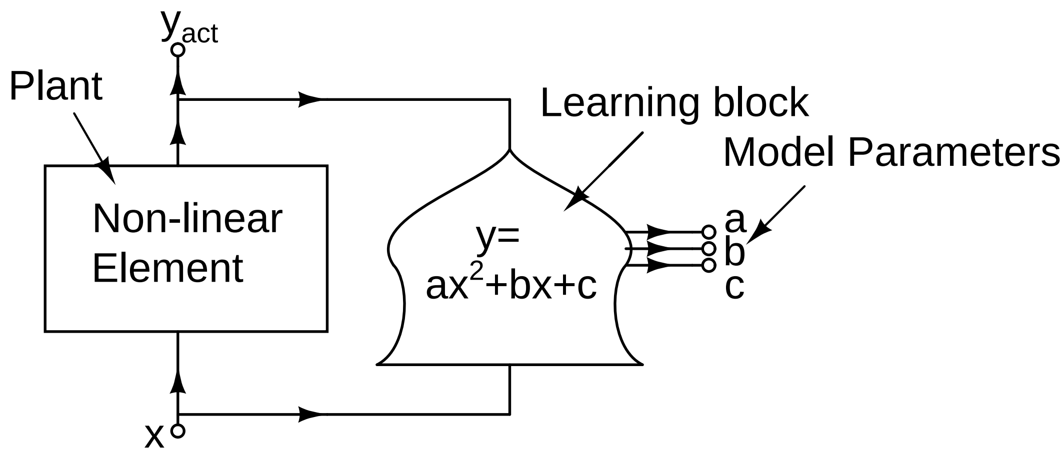
- Solving constrained equations [CHUA] [DENNIS]
- Time varying equations – modeling
- Linearization



$x^2 = y$	$x = a \cdot b$	$b > 5$
$y = 7$	$x + y = z$	$z = a + b$

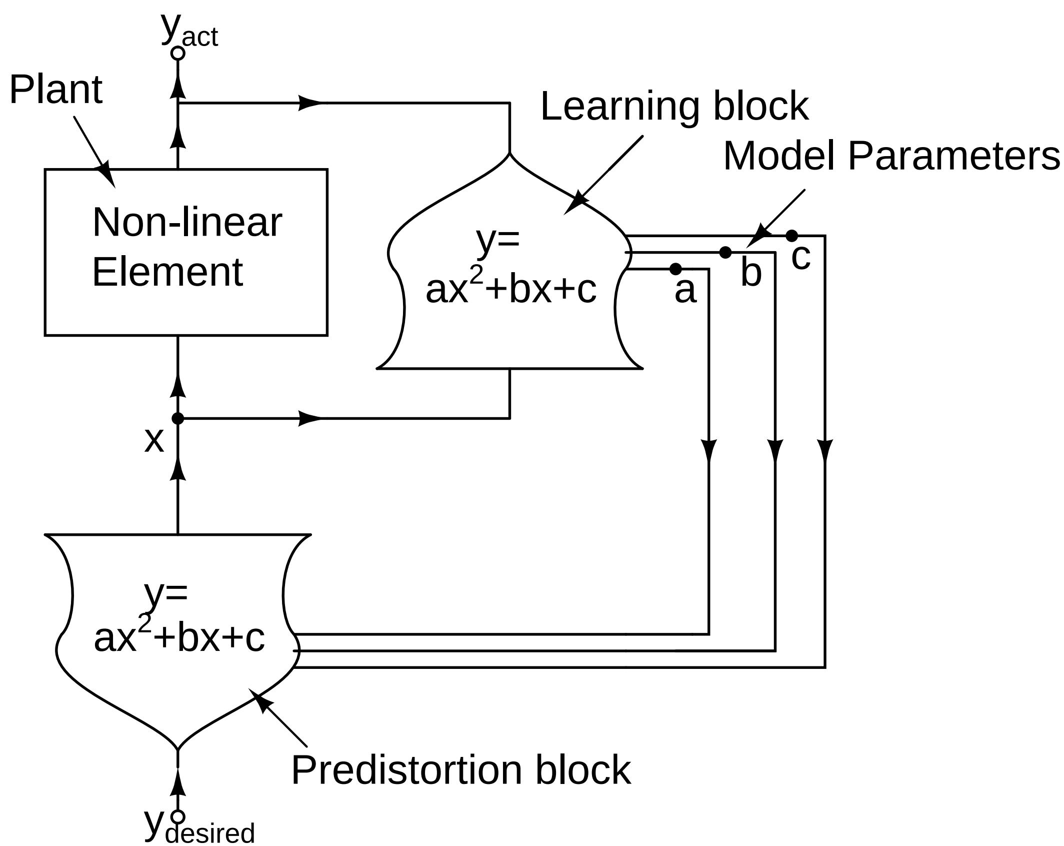
Overview

- Solving constrained equations [CHUA] [DENNIS]
- Time varying equations – modeling
- Linearization

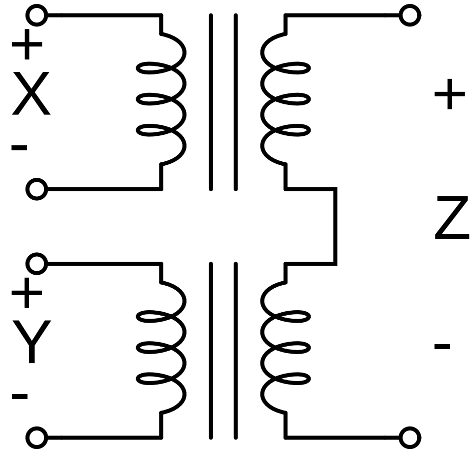


Overview

- Solving constrained equations [CHUA] [DENNIS]
- Time varying equations – modeling
- Linearization



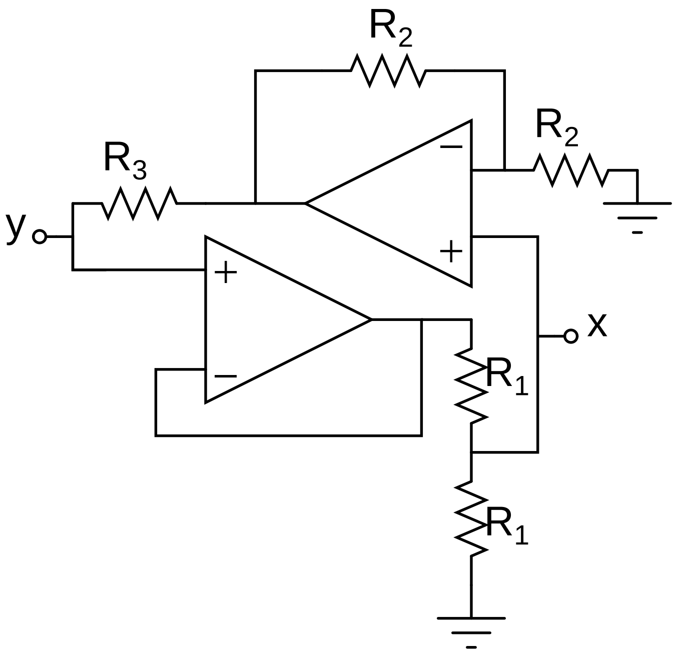
Transformers



$$x + y = z$$

- [Mallock, 1933]
- [Seidel, Knight, 1995]

Active Transformer



Generalized Active Transformer

Take transformer relation:

$$y = 2x$$

Generalized Active Transformer

Take transformer relation:

$$y = 2x$$

Objective function:

$$L(x, y) = (y - 2x)^2$$

Generalized Active Transformer

Take transformer relation:

$$y = 2x$$

Objective function:

$$L(x, y) = (y - 2x)^2$$

Minimize objective function:

$$\min_{x,y} L(x, y)$$

$$\min_{x,y} (y - 2x)^2$$

Generalized Active Transformer, cont.

Minimize objective function:

$$\min_{x,y} (y - 2x)^2$$

Generalized Active Transformer, cont.

Minimize objective function:

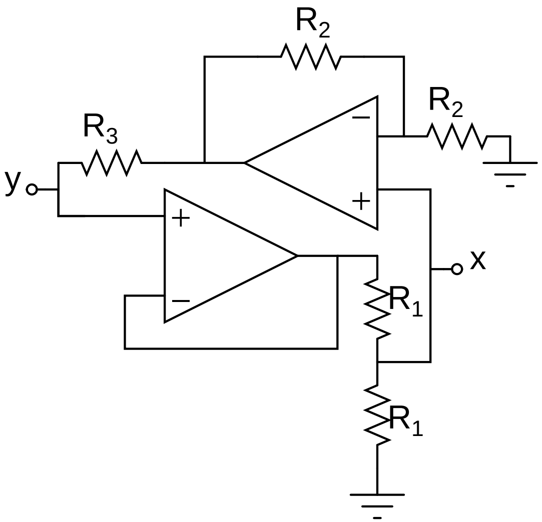
$$\min_{x,y} (y - 2x)^2$$

Output currents:

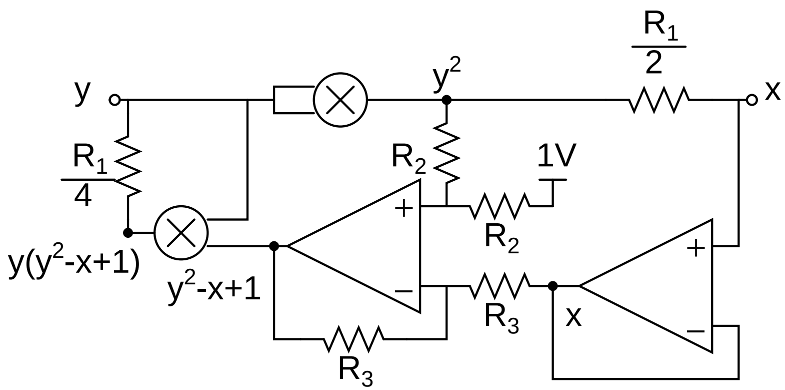
$$I_x = -\frac{dL}{dx} = -8x + 4y$$

$$I_y = -\frac{dL}{dy} = 4x - 2y$$

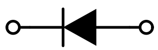
Constraint blocks



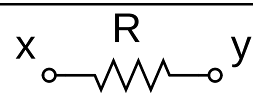
$x = 2y$



$x = y^2$

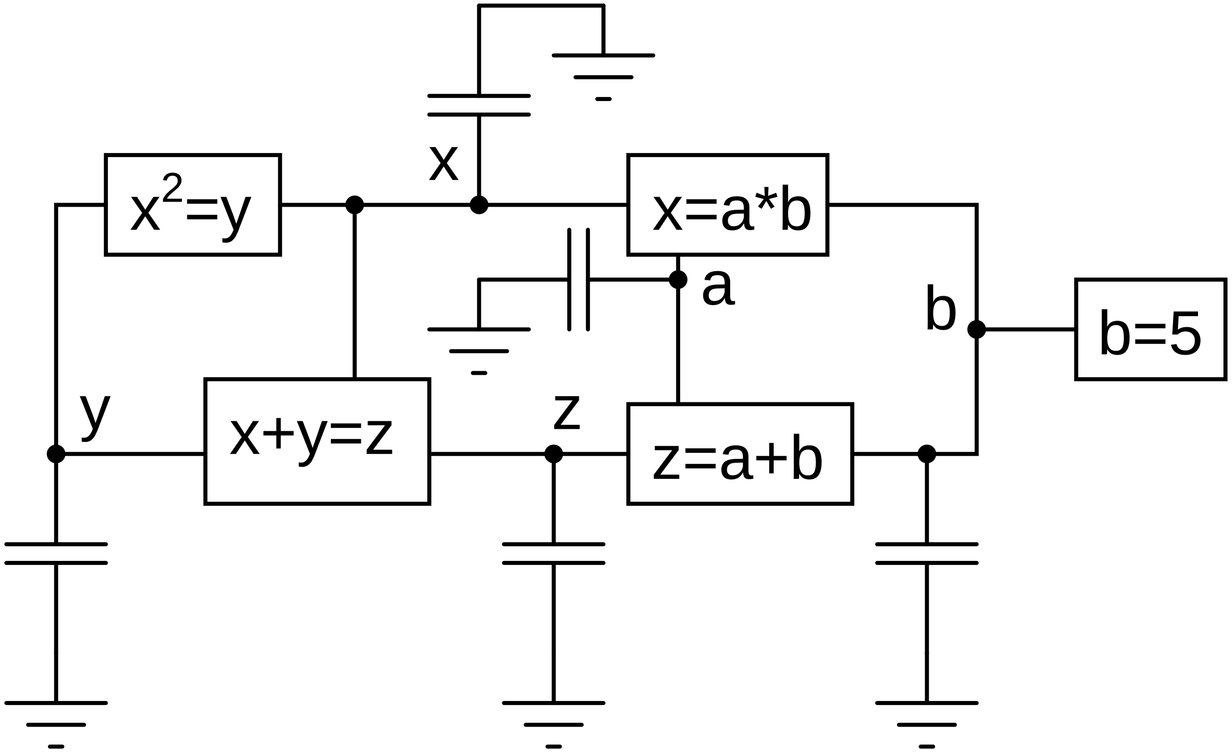


$x > y$

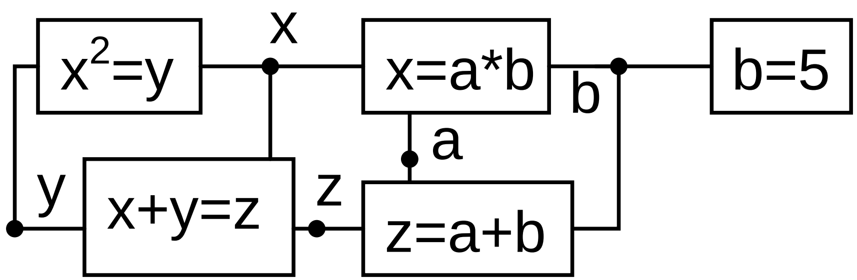


$x \approx y$

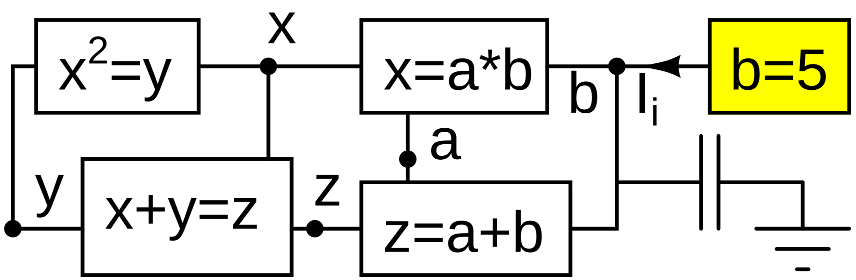
Constraint Networks



Stability (definitions)

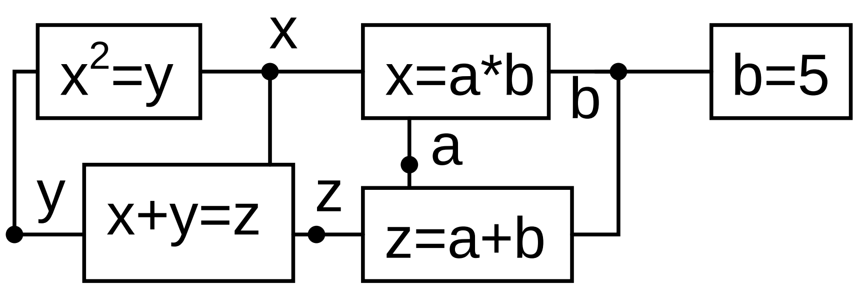


Stability (definitions)



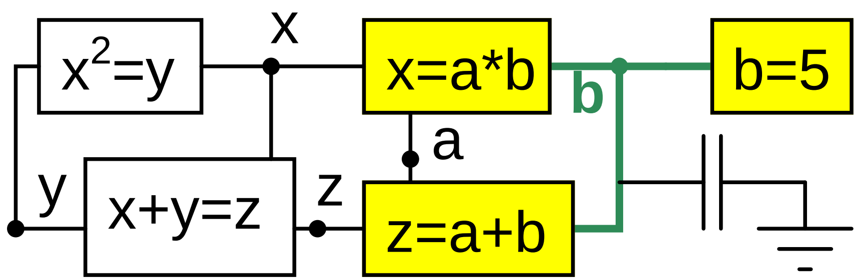
$$L_i \quad = \quad (b-5)^2$$

Stability (definitions)



$$
\begin{aligned}
L_i &= (b-5)^2 \\
L_{global} &= (x^2-y)^2+(x-ab)^2+(b-5)^2+ \\
&\quad (x+y-z)^2+(z-a-b)^2
\end{aligned}
$$

Stability (definitions)

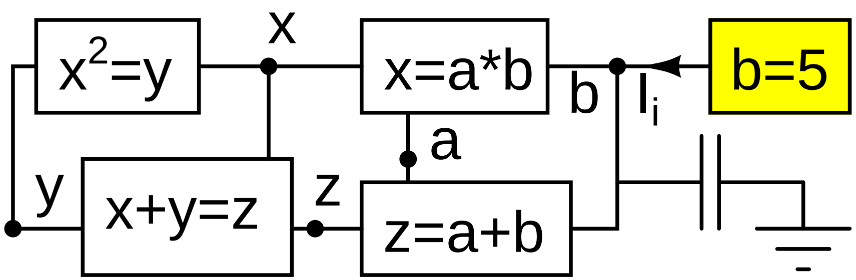


$$L_i = (b-5)^2$$

$$L_{global} = (x^2-y)^2 + (x-ab)^2 + (b-5)^2 + \\ (x+y-z)^2 + (z-a-b)^2$$

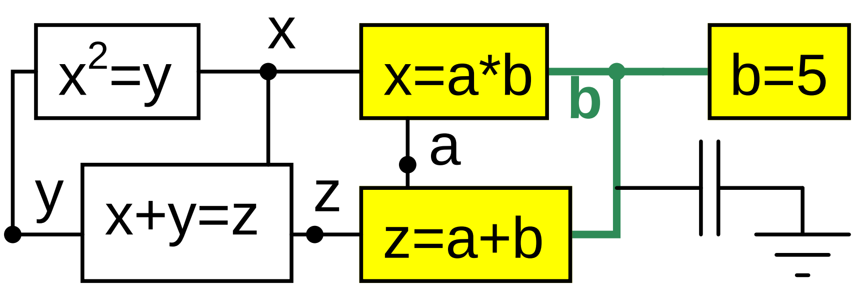
$$L_b = (x-ab)^2 + (b-5)^2 + (z-a-b)^2$$

Stability (dynamics)



$$I_i = -\frac{dL_i}{dV_b} \qquad (L_i = (b-5)^2)$$

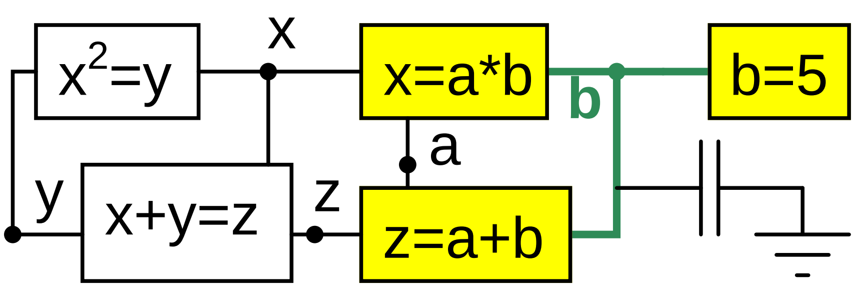
Stability (dynamics)



$$I_i = -\frac{dL_i}{dV_b} \qquad (L_i = (b-5)^2)$$

$$I_b = \sum_{i \in \mathcal{N}_b} -\frac{dL_i}{dV_b}$$

Stability (dynamics)

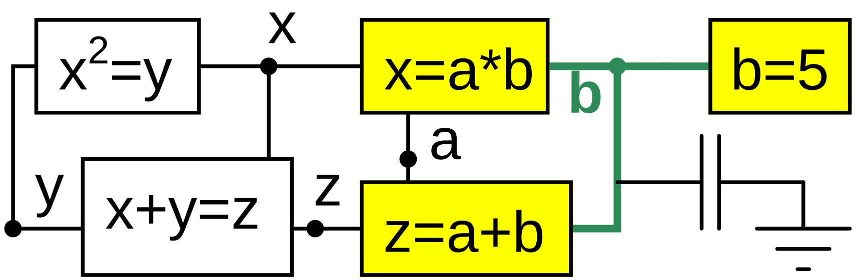


$$I_i = -\frac{dL_i}{dV_b} \qquad (L_i = (b-5)^2)$$

$$I_b = \sum_{i \in \mathcal{N}_b} -\frac{dL_i}{dV_b}$$

$$\frac{dV_b}{dt} = \frac{I_b}{C} = \frac{1}{C}\left(\sum_{i \in \mathcal{N}_b} -\frac{dL_i}{dV_b}\right)$$

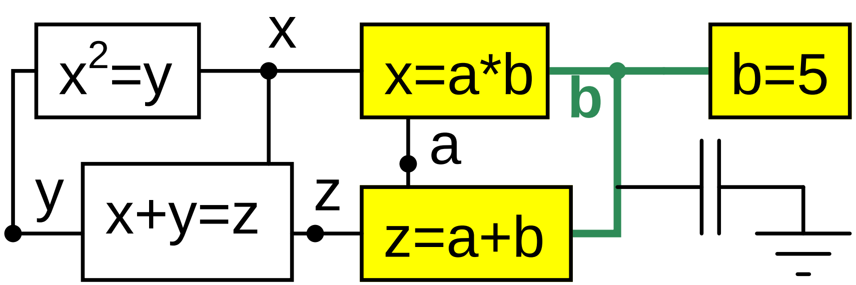
Stability (grand finale)



$$\frac{dV_b}{dt} = \frac{1}{C}\left(\sum_{i\in\mathcal{N}_b} -\frac{dL_i}{dV_b}\right)$$

$$\frac{dL_b}{dt} =$$

Stability (grand finale)



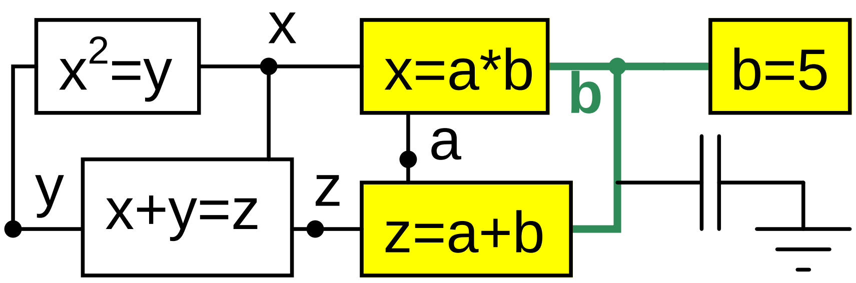
$$\frac{dV_b}{dt} = \frac{1}{C}\left(\sum_{i \in \mathcal{N}_b} -\frac{dL_i}{dV_b}\right)$$

Rate of change in voltage

Effect of Δ voltage on objective function

$$\frac{dL_b}{dt} = \frac{dV_b}{dt}\frac{dL_b}{dV_b}$$

Stability (grand finale)



$$\frac{dV_b}{dt} = \frac{1}{C}\left(\sum_{i \in \mathcal{N}_b} -\frac{dL_i}{dV_b}\right)$$

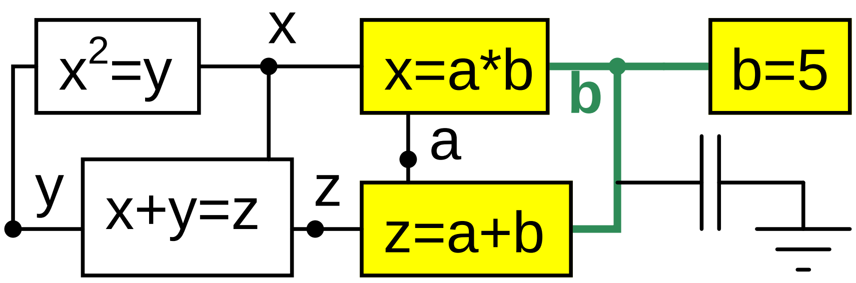
Rate of change in voltage

Effect of Δ voltage on objective function

$$\frac{dL_b}{dt} = \frac{dV_b}{dt}\frac{dL_b}{dV_b}$$

$$\frac{dL_b}{dV_b} = \left(\sum_{i \in \mathcal{N}_b} \frac{dL_i}{dV_b}\right)$$

Stability (grand finale)



$$\frac{dV_b}{dt} = \frac{1}{C}\left(\sum_{i\in\mathcal{N}_b} -\frac{dL_i}{dV_b}\right)$$

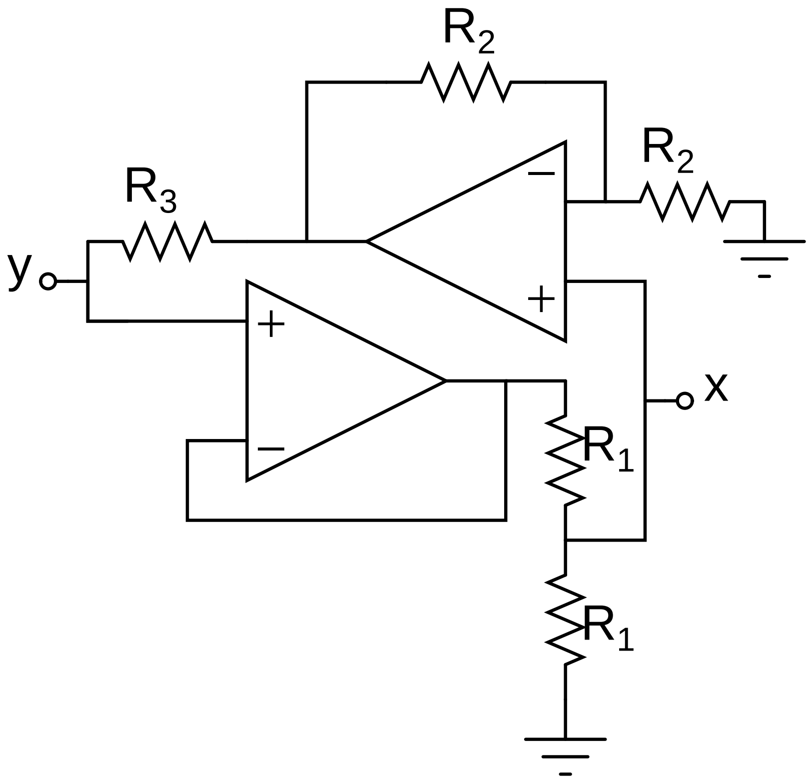
Rate of change in voltage

Effect of Δ voltage on objective function

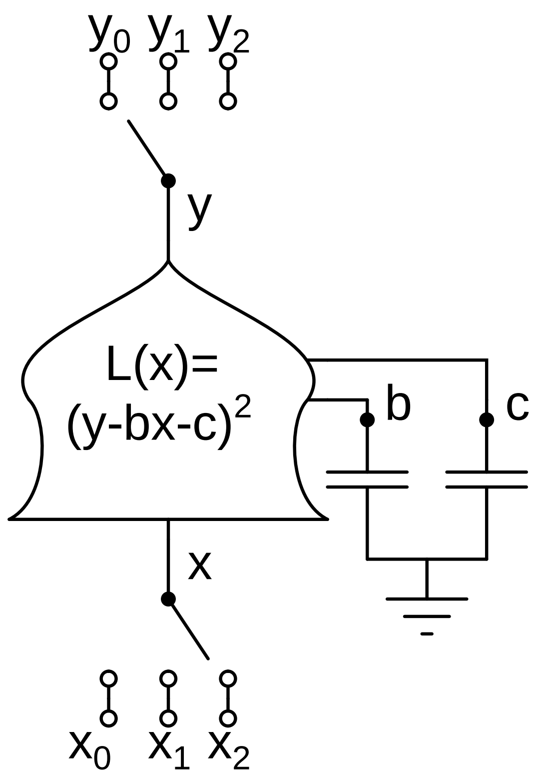
$$\frac{dL_b}{dt} = \frac{dV_b}{dt}\,\frac{dL_b}{dV_b} =$$

$$\frac{1}{C}\left(\sum_{i\in\mathcal{N}_b} -\frac{dL_i}{dV_b}\right)\left(\sum_{i\in\mathcal{N}_b} \frac{dL_i}{dV_b}\right) \leq 0$$

Underconstrained case

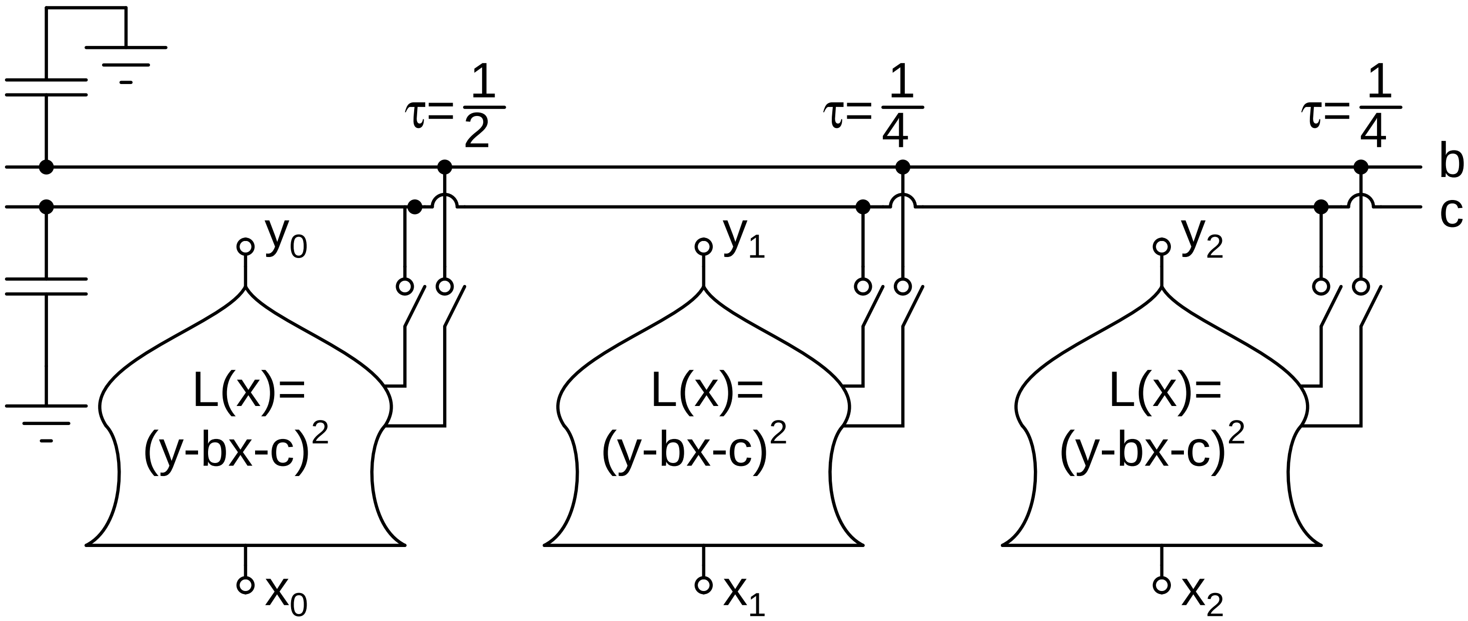


Underconstrained case



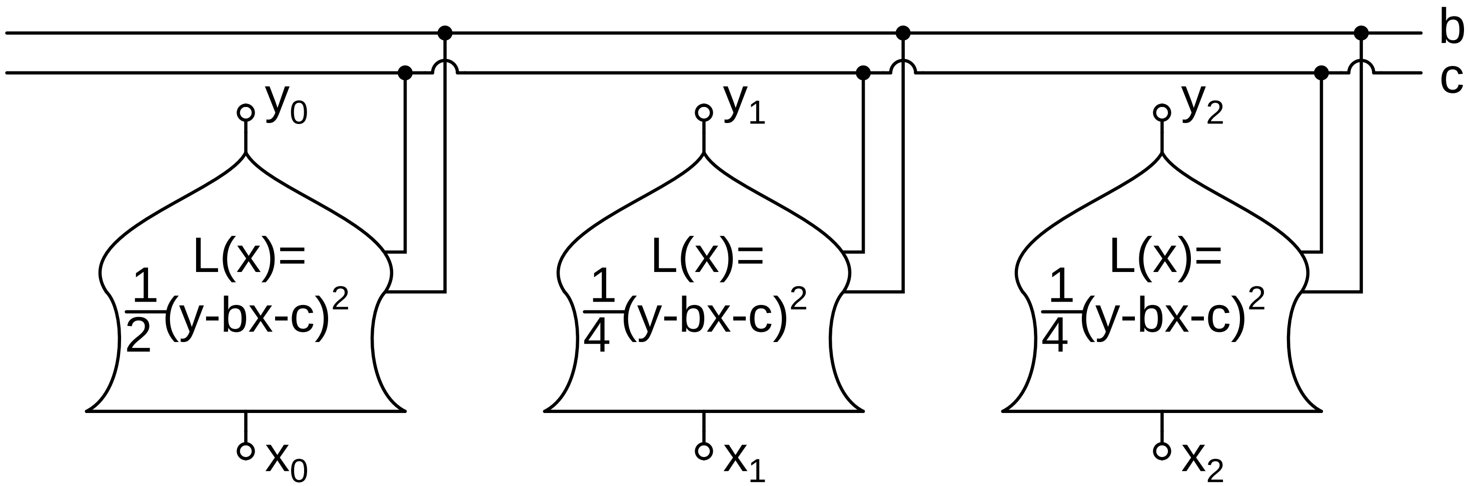
$$y = bx + c$$

Underconstrained case



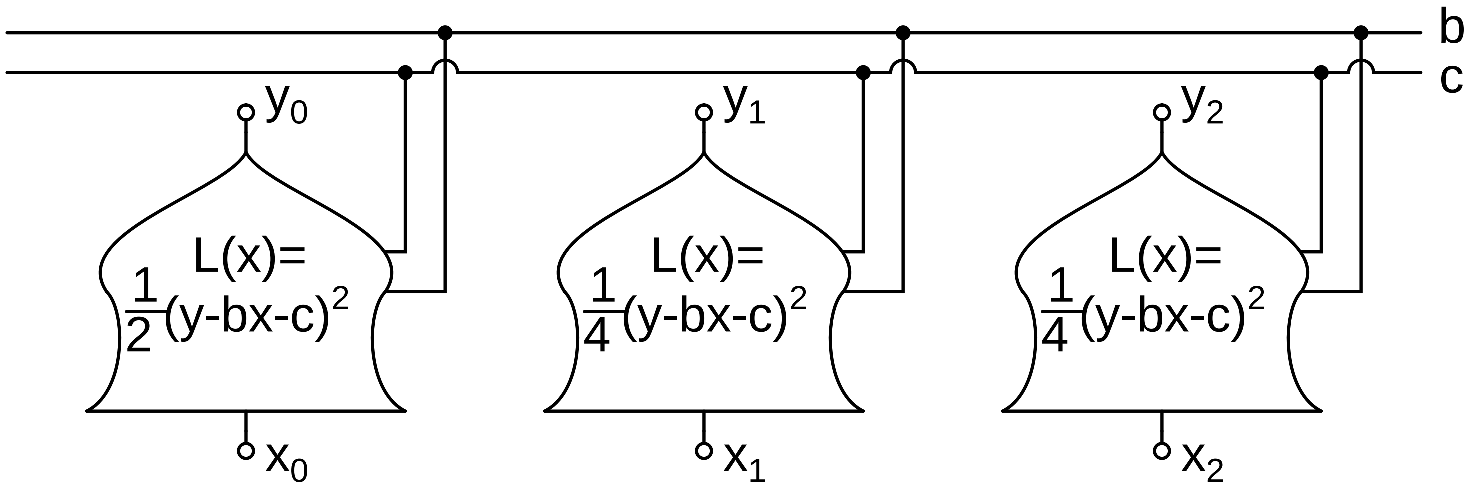
$$y = bx + c$$

Underconstrained case



$$y = bx + c$$

Underconstrained case



$$y = bx + c$$

$$\min_{b,c} \quad \begin{array}{l} \frac{1}{2}\left(y_0 - bx_0 - c\right)^2 + \\ \frac{1}{4}\left(y_1 - bx_1 - c\right)^2 + \\ \frac{1}{4}\left(y_2 - bx_2 - c\right)^2 \end{array}$$

General linear regression

$$\sum_i \underbrace{c_i}_{\text{Model parameters}} \underbrace{f_i(x_1, \ldots, x_n)}_{\text{Functions}} = g(x_1, \ldots, x_n)$$

Model parameters Functions

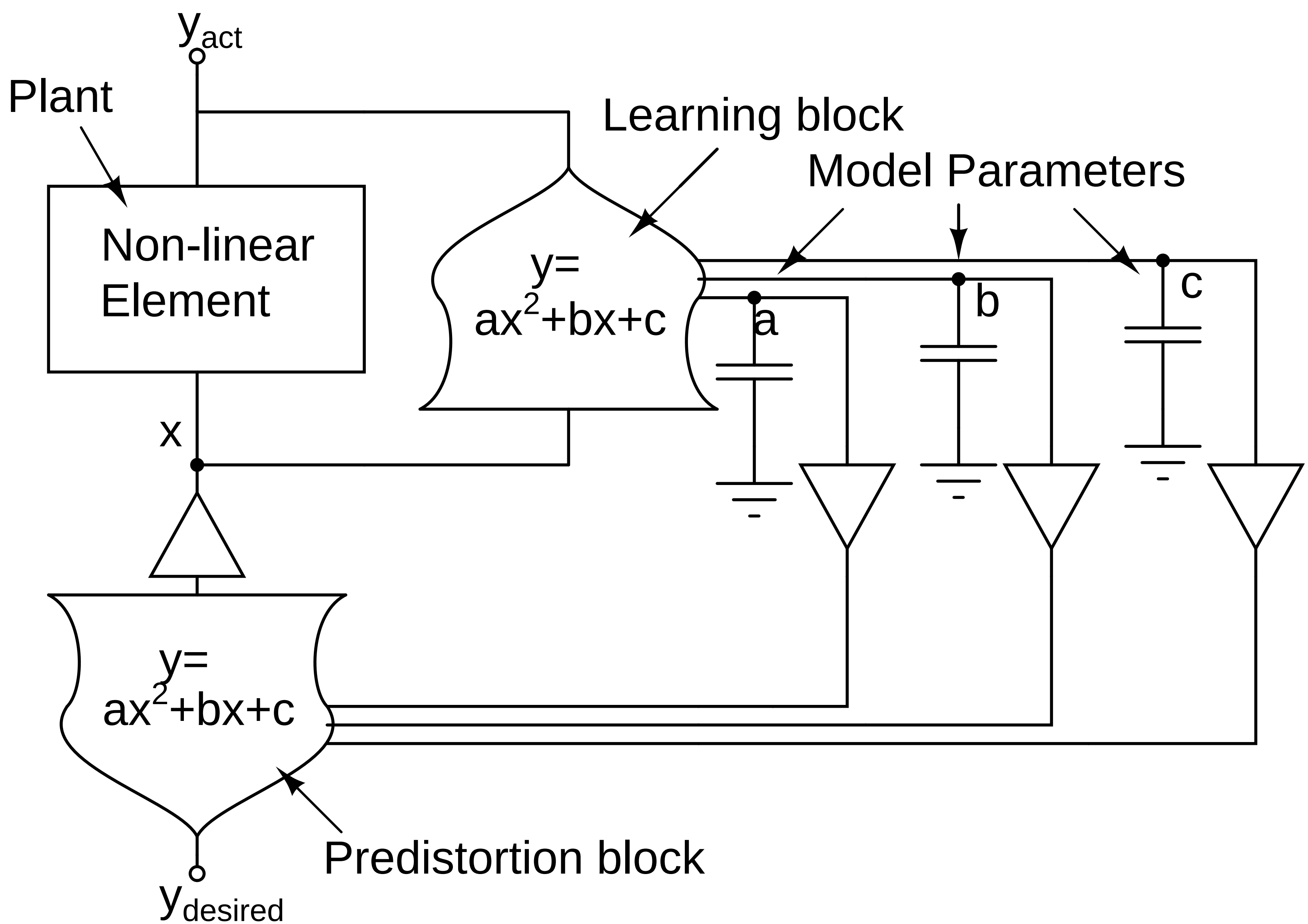
General linear regression

$$\sum_i \underbrace{c_i}_{\text{Model parameters}} \underbrace{f_i(x_1, \ldots, x_n)}_{\text{Functions}} = g(x_1, \ldots, x_n)$$

Model parameters Functions

- Taylor approximation, Fourier series, discretization, wavelets, conic sections, Chebyshev polynomials, etc.
- (Robust) stability proofs in non-limit case

Linearization



Compiling

Given a circuit, often possible to compile to a more efficient implementation:

- Common subexpression elimination
- Bidirectional $\rightarrow$ unidirectional
- Common buffers
- Common current output
- Approximations
- Circuit-specific simplifications
- Etc.

Example – Simplifed linearizer

- Example: Model a nonlinearity by $x = c_1 y^2 + c_2 y + c_3$

Example – Simplifed linearizer

- Example: Model a nonlinearity by $x = c_1 y^2 + c_2 y + c_3$
- Notation:

$$x = \sum_i c_i f_i(y)$$

$$f_1(y) = y^2 \qquad f_2(y) = y \qquad f_3(y) = 1$$

Example – Simplifed linearizer

Given: $x = \sum_i c_i f_i(y_{act})$

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\,(y_{act})(x - \sum_i c_i f_i(y_{act})) \\ \vdots \\ f_n\,(y_{act})(x - \sum_i c_i f_i(y_{act})) \end{bmatrix} =$$

Example – Simplifed linearizer

Given: $x = \sum_i c_i f_i(y_{act})$

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1(y_{act})(x - \sum_i c_i f_i(y_{act})) \\ \vdots \\ f_n(y_{act})(x - \sum_i c_i f_i(y_{act})) \end{bmatrix} =$$

$$\begin{bmatrix} f_1(y_{act}) \\ \vdots \\ f_n(y_{act}) \end{bmatrix} \cdot \left(x - \sum_i c_i f_i(y_{act}) \right)$$

- Direction term
- Scaling term

Example – Simplifed linearizer

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right) \\ \vdots \\ f_n\left(y_{act}\right) \end{bmatrix} \cdot \left(x - \sum_i c_i f_i(y_{act}) \right)$$

Example – Simplifed linearizer

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right) \\ \vdots \\ f_n\left(y_{act}\right) \end{bmatrix} \cdot \left(x - \sum_i c_i f_i(y_{act})\right)$$

If the **model** is monotonic:

$$\text{sign}\left(x - \sum_i c_i f_i\left(y_{act}\right)\right) = \text{sign}(y_{des} - y_{act})$$

Example – Simplifed linearizer

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right) \\ \vdots \\ f_n\left(y_{act}\right) \end{bmatrix} \cdot \left(x - \sum_i c_i f_i(y_{act})\right)$$

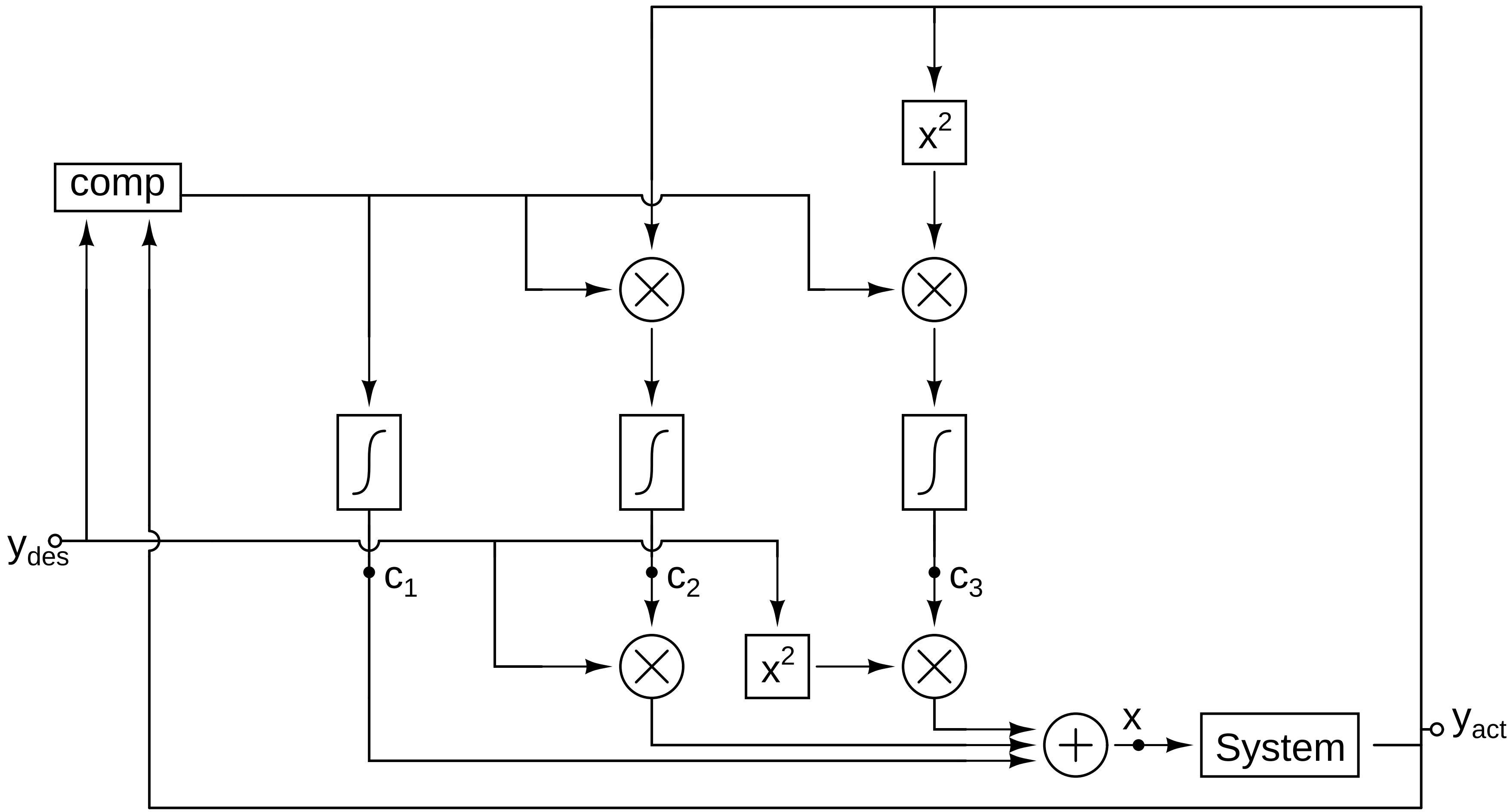
If the **model** is monotonic:

$$\text{sign}\left(x - \sum_i c_i f_i\left(y_{act}\right)\right) = \text{sign}(y_{des} - y_{act})$$

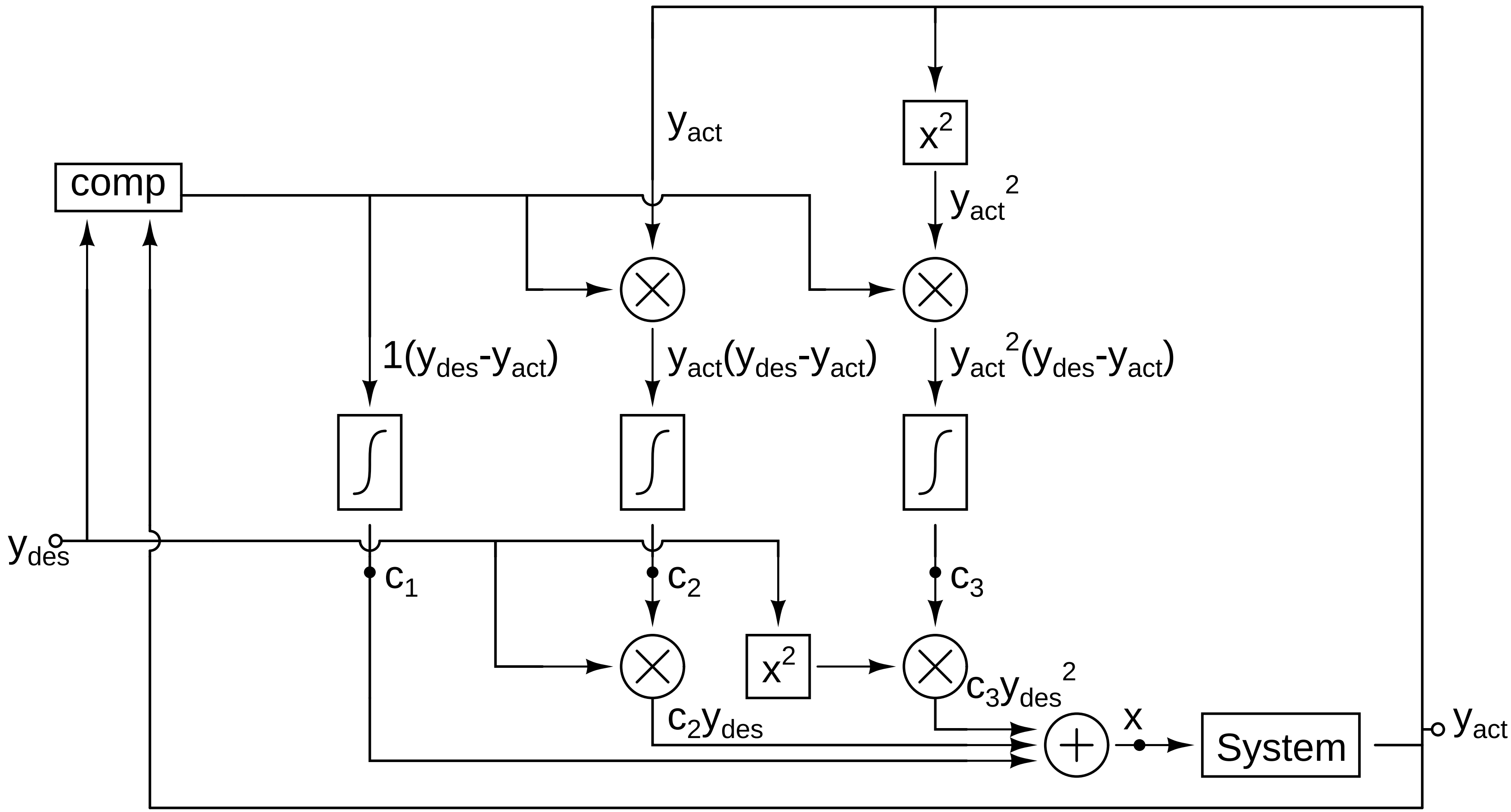
Approximate as:

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = \begin{bmatrix} f_1\left(y_{act}\right) \\ \vdots \\ f_n\left(y_{act}\right) \end{bmatrix} \cdot (y_{des} - y_{act})$$

Linearizer Block Diagram

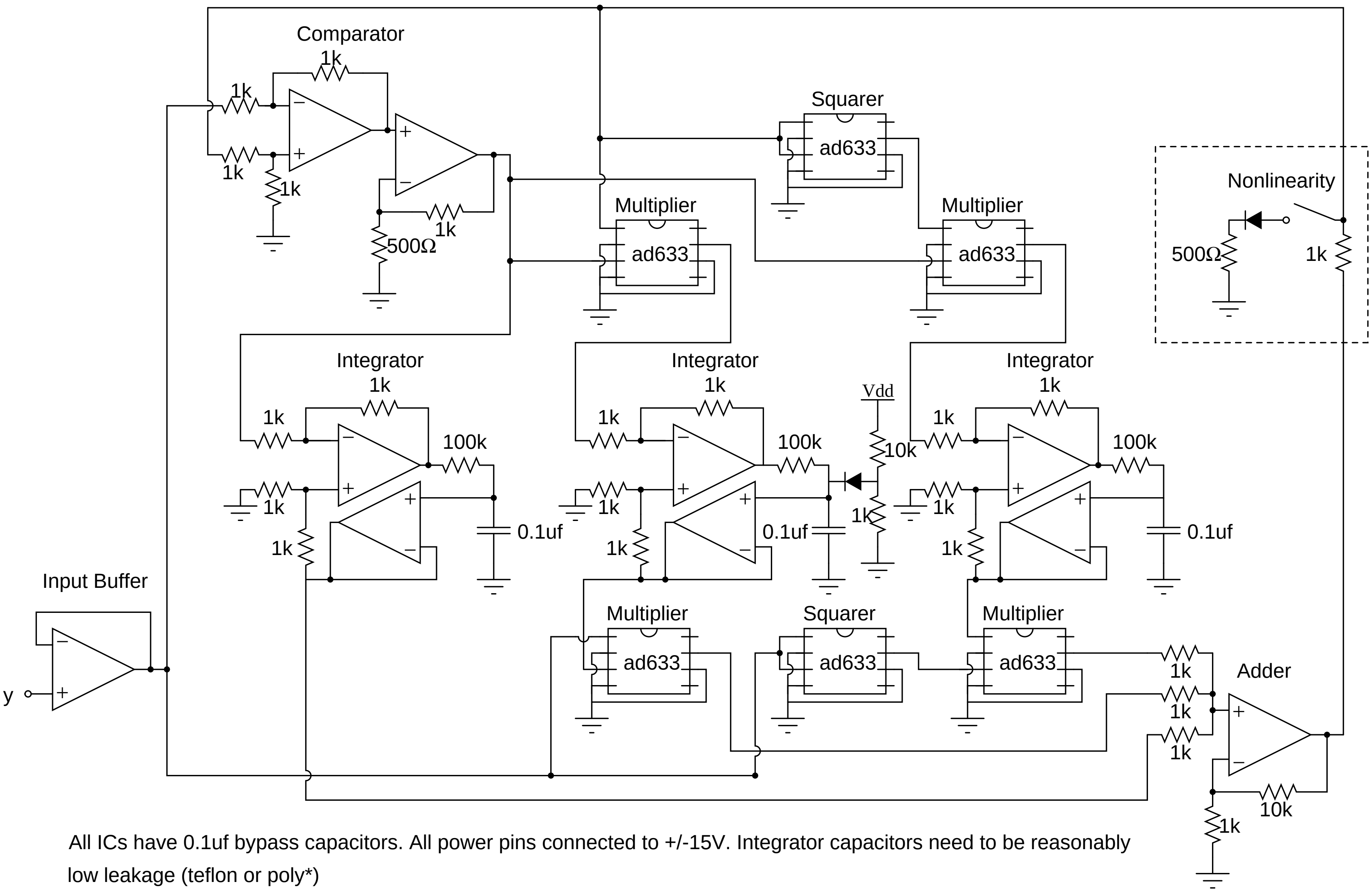


Linearizer Block Diagram



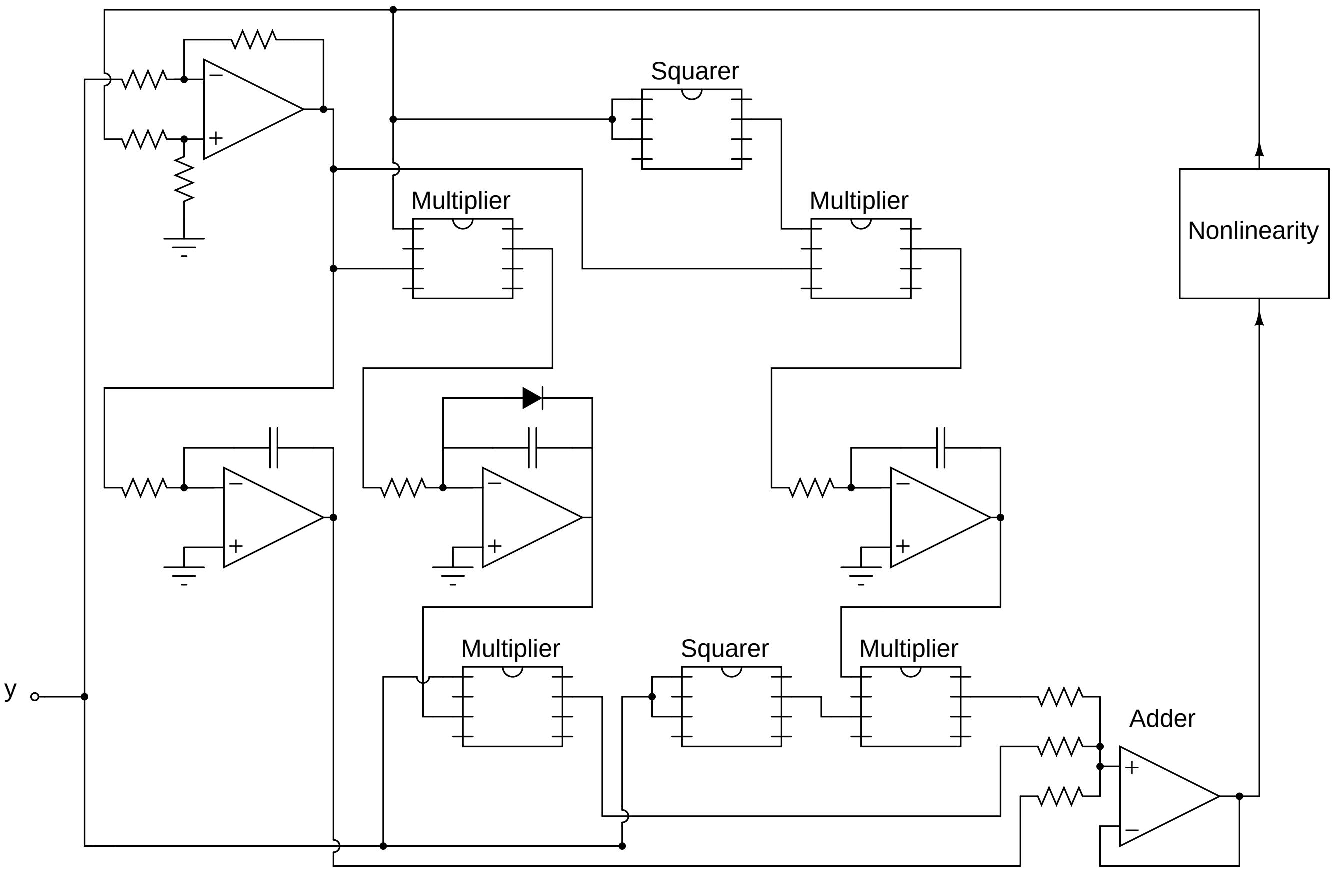
$$\dot{c}_i = f_i\,(y_{act}) \cdot (y_{des} - y_{act})$$

Linearizer Circuit Detail

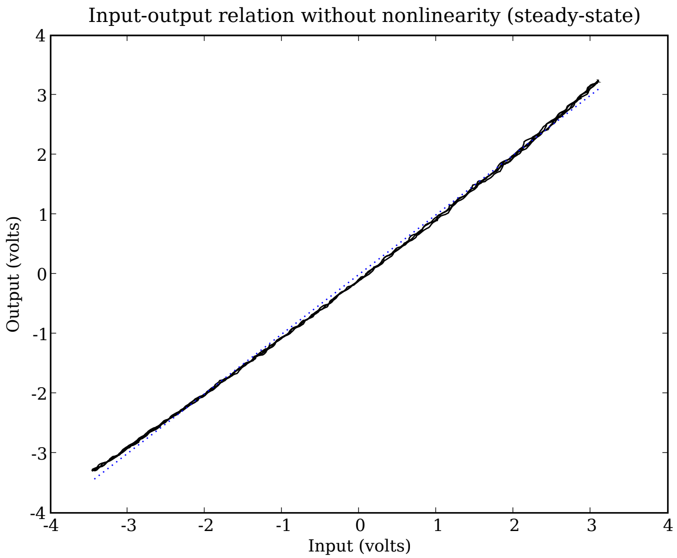


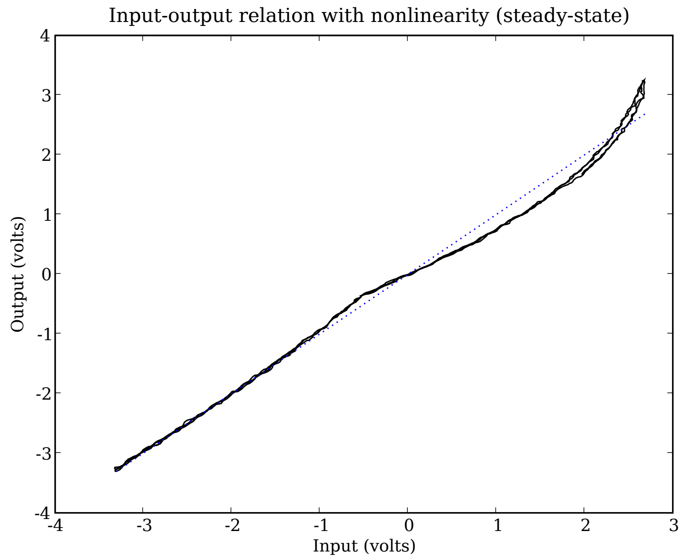
All ICs have 0.1uf bypass capacitors. All power pins connected to +/-15V. Integrator capacitors need to be reasonably low leakage (teflon or poly*)

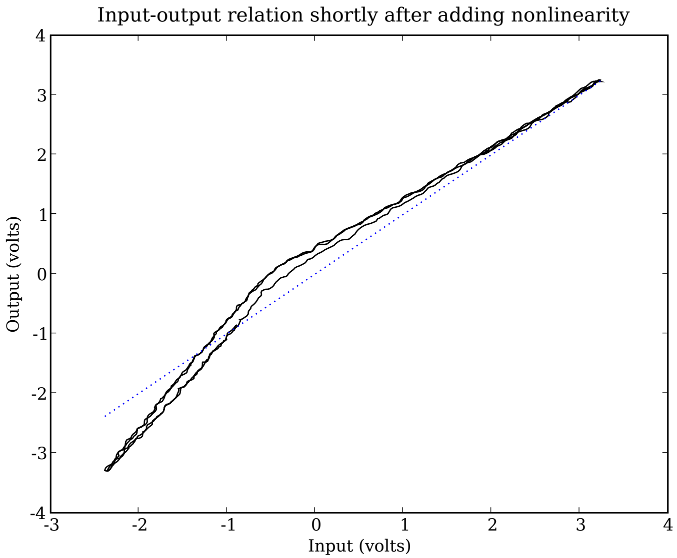
Linearizer Circuit Detail

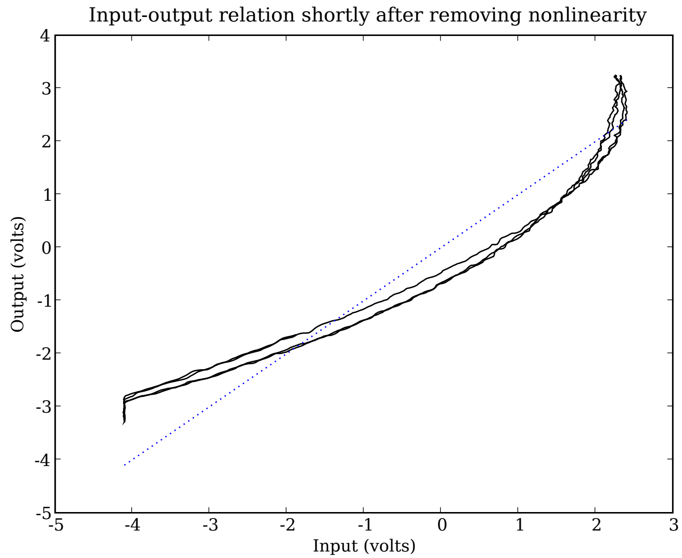


Scope traces

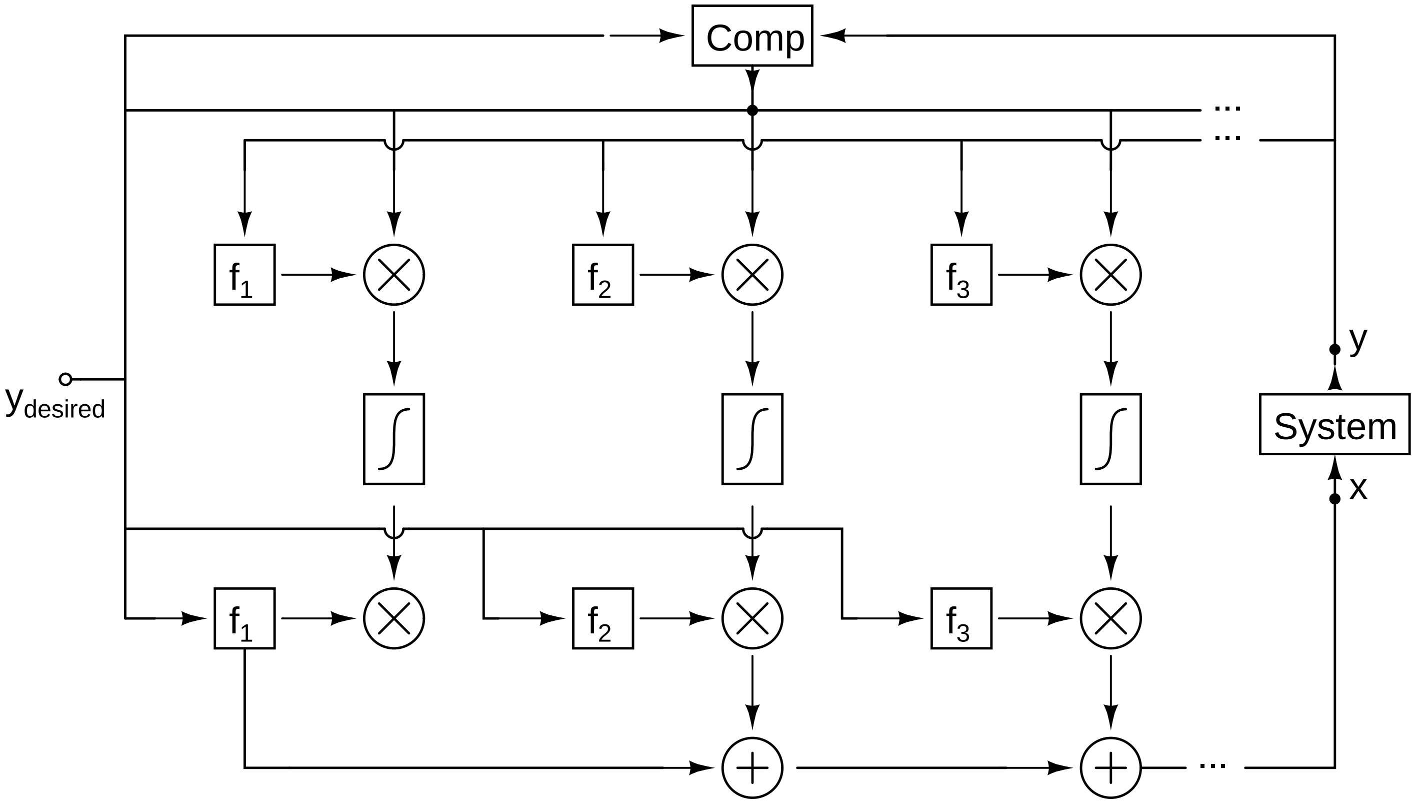








Robustness



$$x = \sum_i c_i f_i(y)$$

Future work

- Develop applications
  - New uses for methodology
  - Applications for existing uses
    - Super operational amplifier
    - Cartesian feedback
    - . . .
- Control systems with memory
- Parallelism
- Dynamic integration rate
- . . .

Contributions

- Demonstrated a methodology for analog circuit design
  - Local stability criterion
  - Bidirectional information flow
- Uses:
  - Equation solvers
  - Modelers
  - Linearizers

Contributions

- Demonstrated a methodology for analog circuit design
  - Local stability criterion
  - Bidirectional information flow
- Uses:
  - Equation solvers
  - Modelers
  - Linearizers

Questions?

Robust stability – low frequency

Let $\phi_j = \sum_{i \in \mathcal{N}_j} \frac{dL_i}{dV_i}$

Given error current E

$$\frac{dL}{dt} = \sum_j (E - \phi_j)\phi_j = \sum_j E\phi_j - \phi_j^2 \leq 0$$

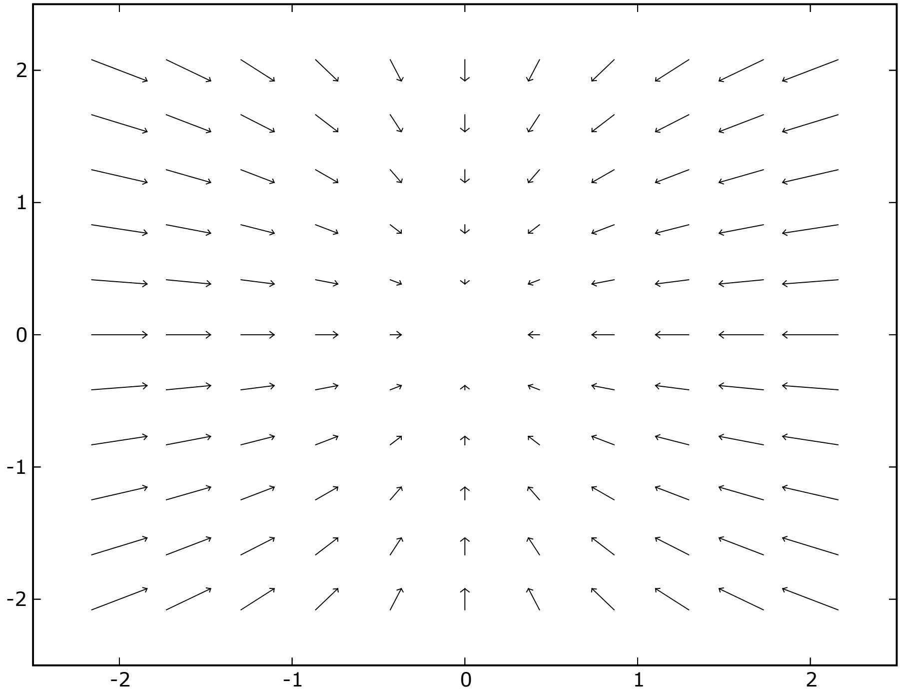
Gives robust stability (stricter bounds exist). Weakens to:

$$\sum_j |E| \leq \sum_j |\phi_j|$$

Won't be true globally!

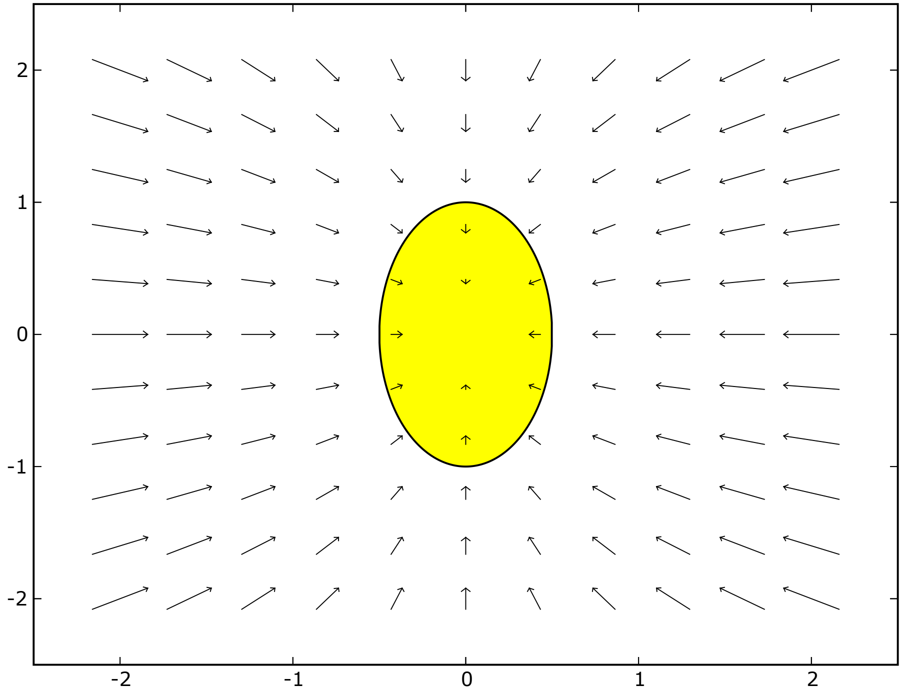
Robust stability – Low frequency

Given a Vector Field:



Robust stability – Low frequency

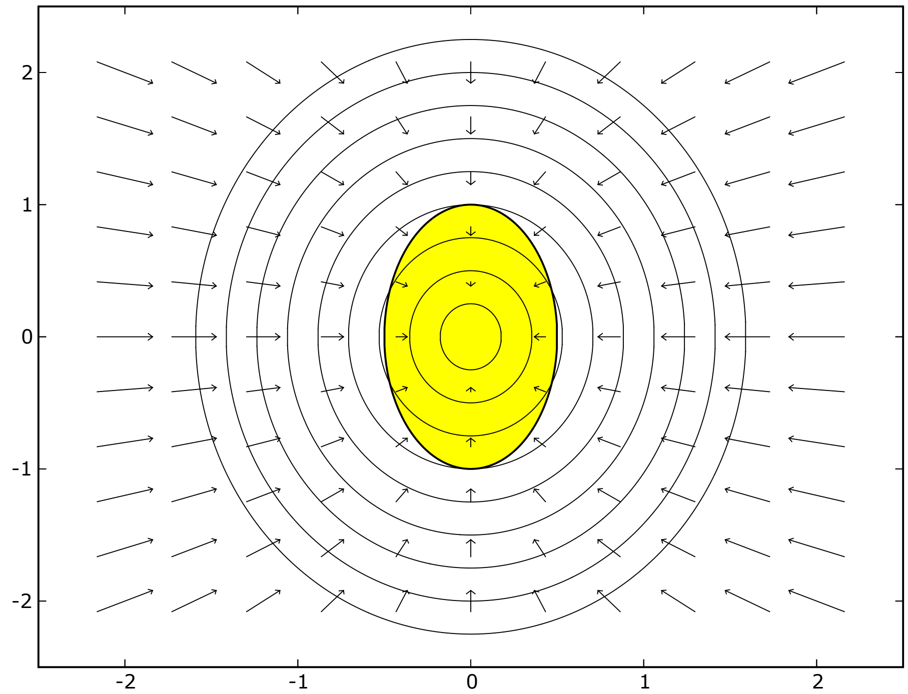
Find region holds everywhere where: $\sum_j |E| \leq \sum_j |\phi_j|$



For nice L, excludes only small region around minimum

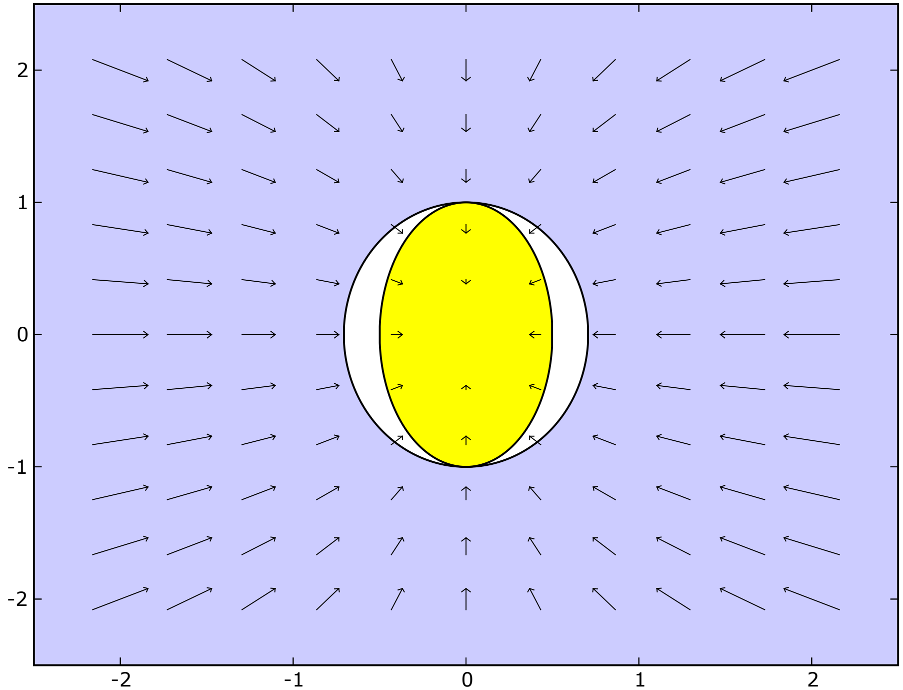
Robust stability – Low frequency

Look at level curves:



Robust stability – Low frequency

Pick smallest level curve containing region where stability critereon does not hold:



Robust stability – LaSalle

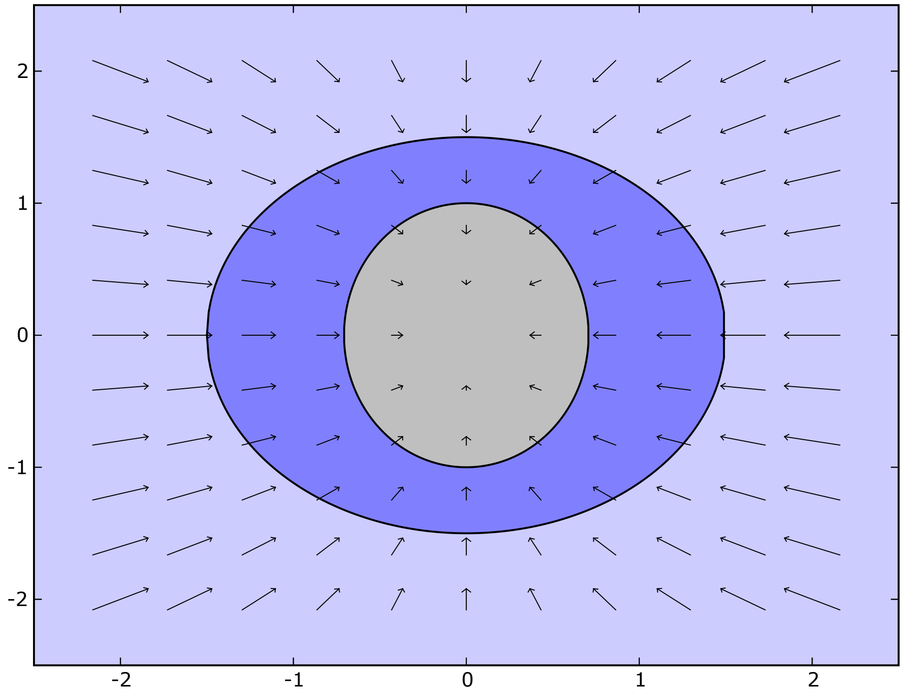
Define new Lyapunov-like function:

$$f'(x) = \begin{cases} f(x) & : \quad x \notin B_1 \\ 0 & : \quad x \in B_1 \end{cases}$$

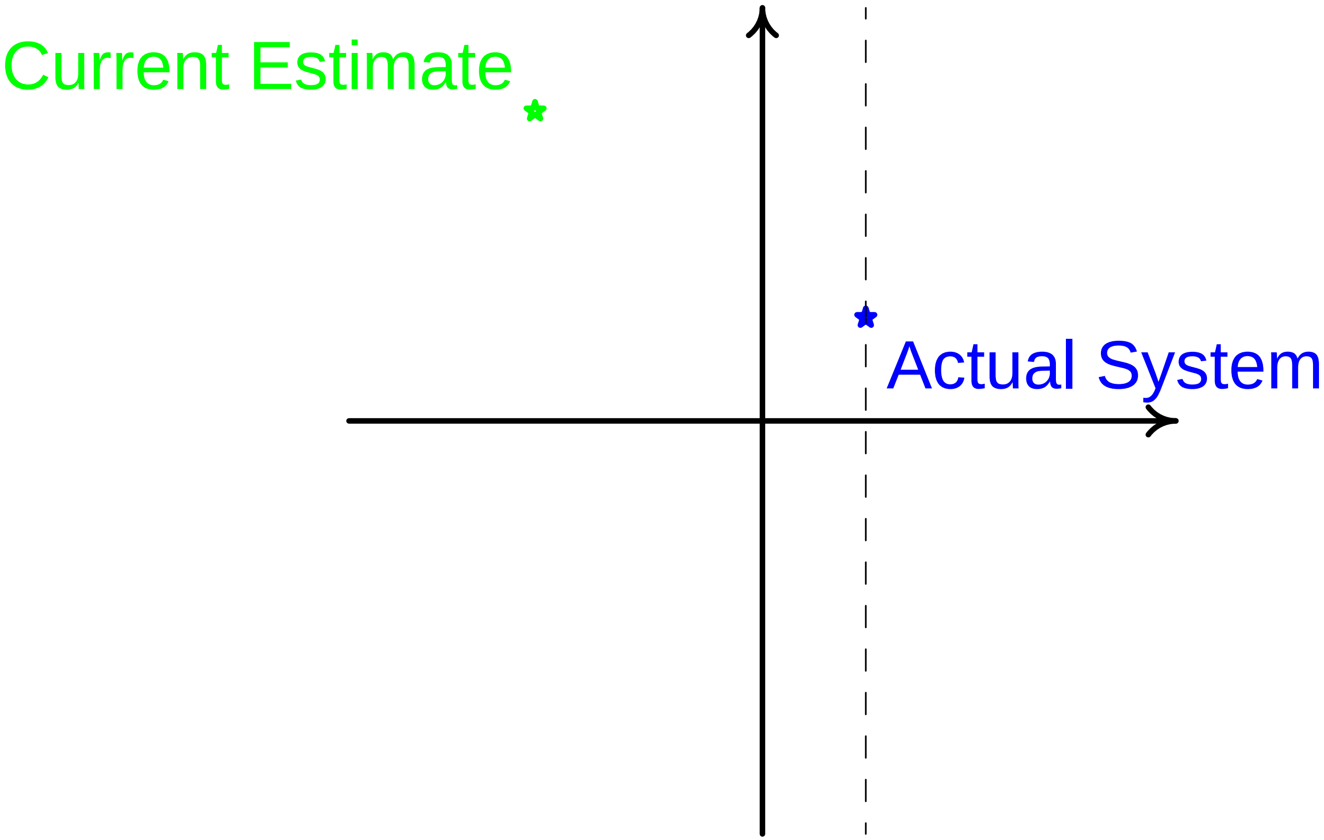
LaSalle shows the system will converge to the region. At worst, small (low frequency) oscillations.

Robust stability – Linearity

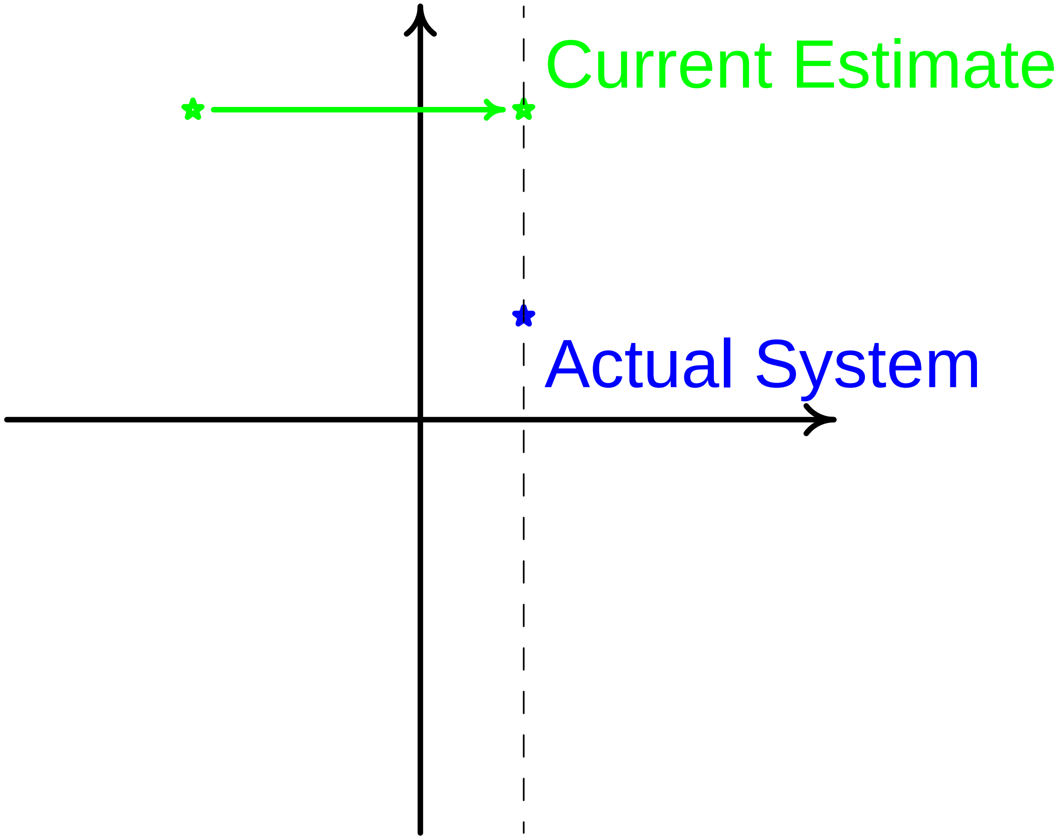
For some specific systems we can do better! If B_1 is small, and the system has a larger region where it is essencially modelled as linear, global stability follows from scale-invariance of linear systems.



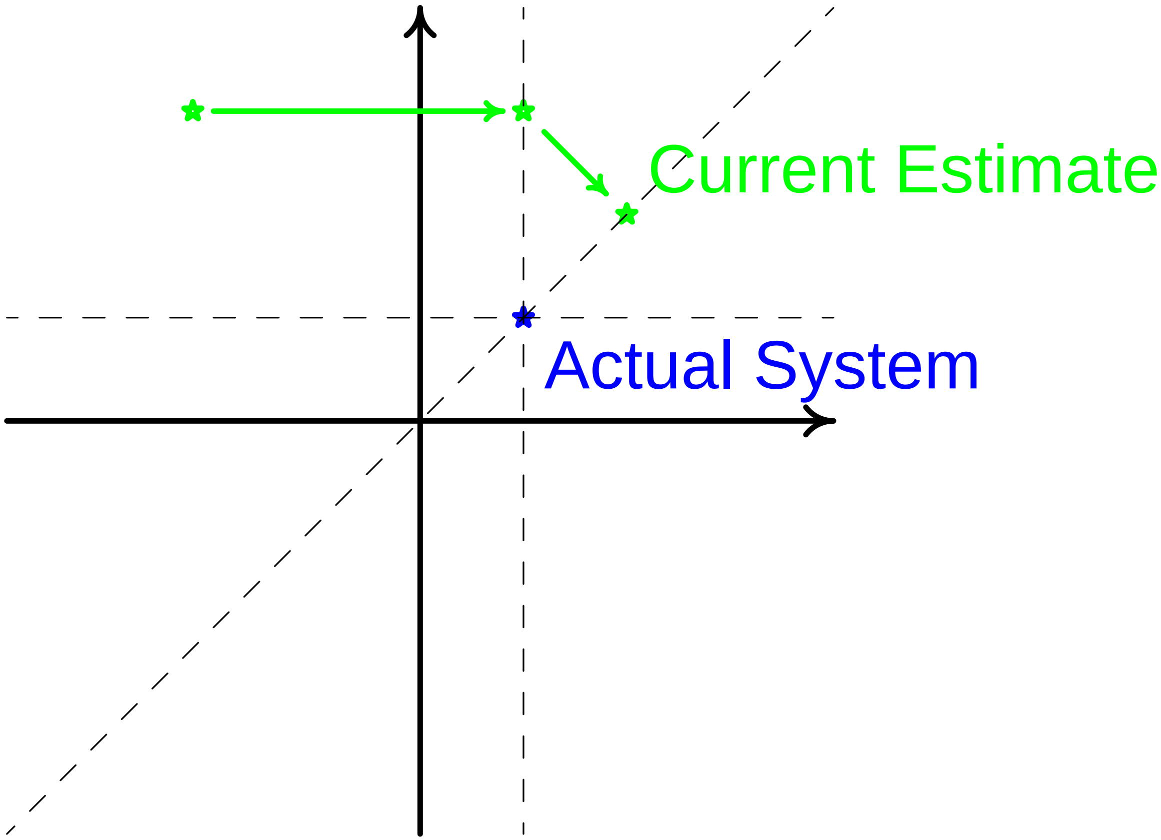
GLR Stability



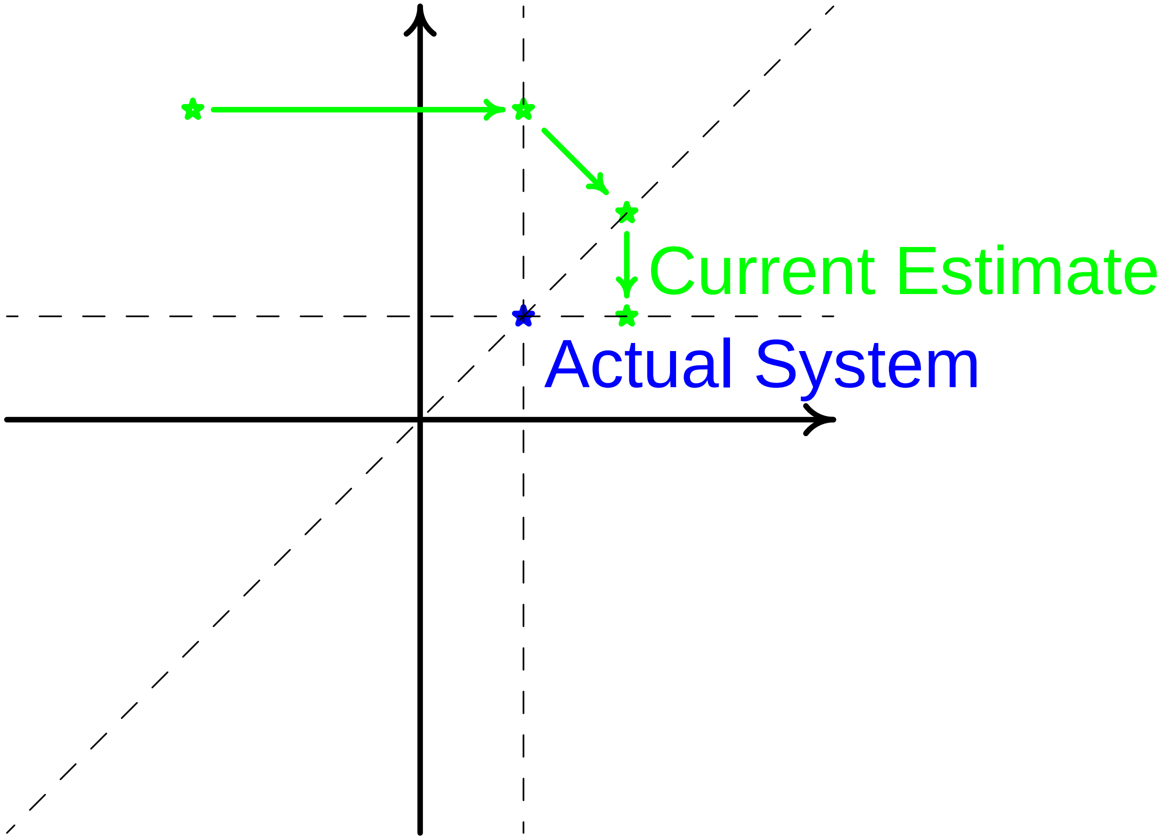
GLR Stability



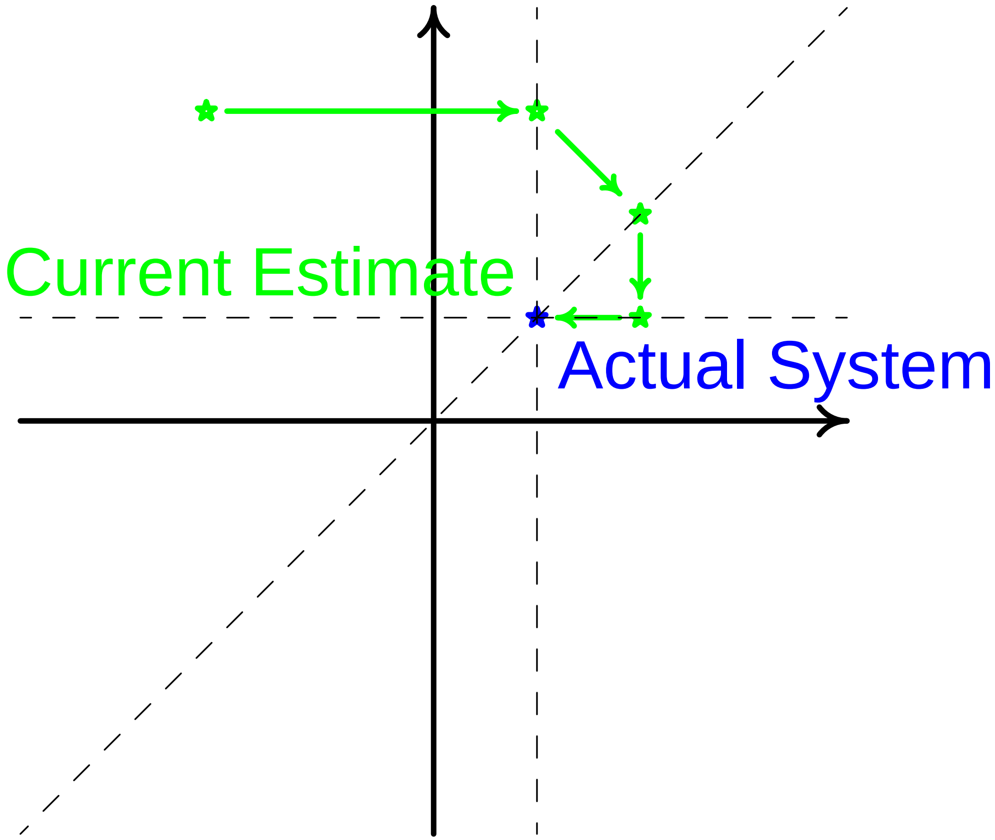
GLR Stability



GLR Stability



GLR Stability



GLR stability - robust

Model of the form:

$$\sum_i c_i f_i(x) = 0$$

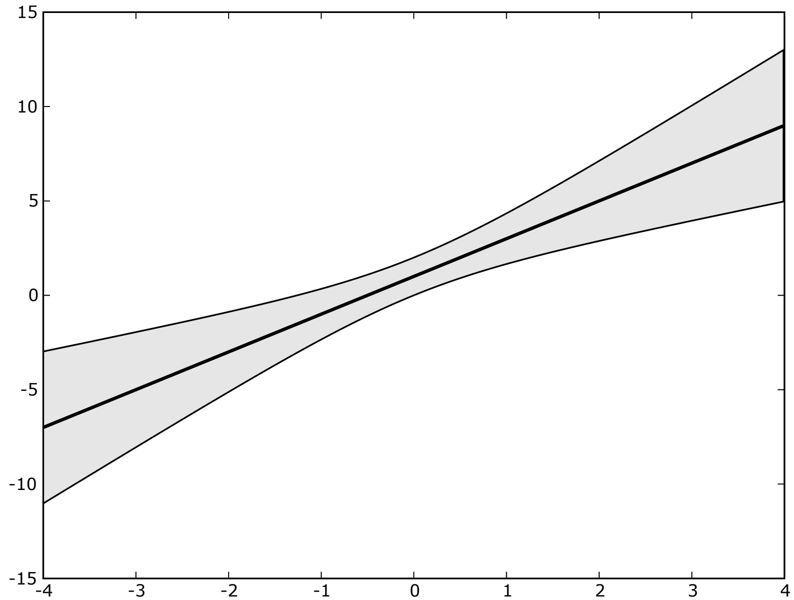
The model can approximate S to within a ball of radius r if:

$$\forall x \text{s.t.} S(x) = 0 \exists \epsilon_1, ..., \epsilon_n \text{s.t. :}$$

$$(c_1 + \epsilon_1) f_1(x) + ... + (c_n + \epsilon_n) f_n(x) = 0, \sum_i \epsilon_i^2 < r^2$$

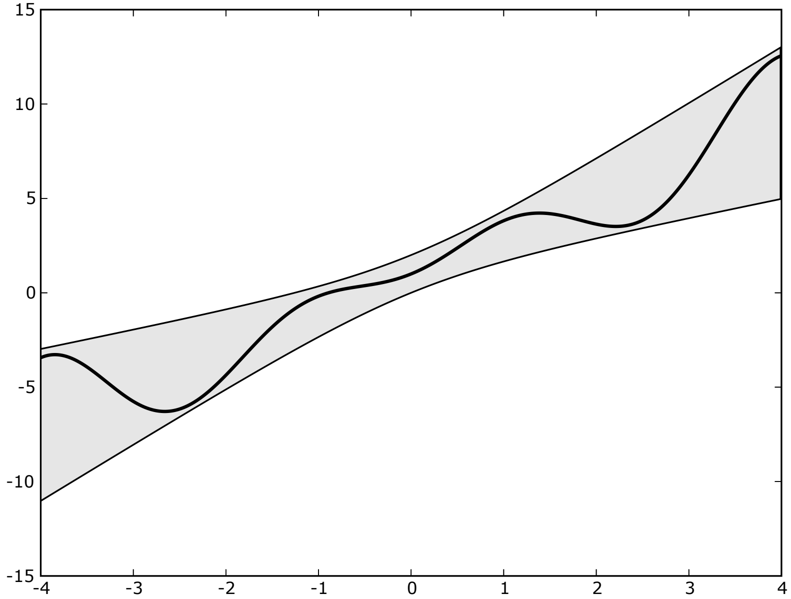
GLR stability - ball

For the model $y = ax + b$ the ball of radius one around $(a, b) = (2, 1)$:



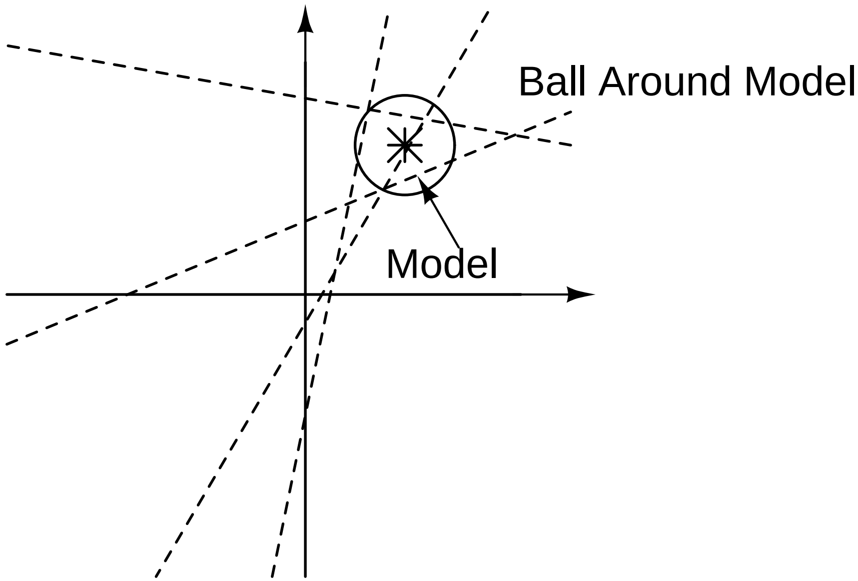
GLR stability - ball

Sample function in the ball:



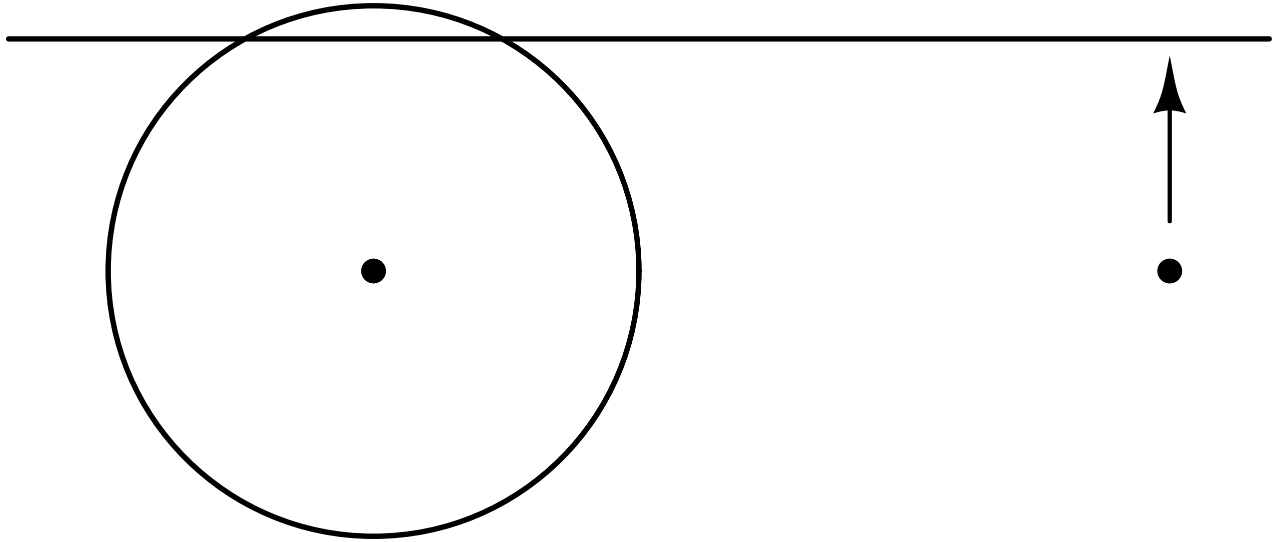
GLR stability - lines and ball

Every hyperplane defined by a point **x** passes through a ball around the model parameters:

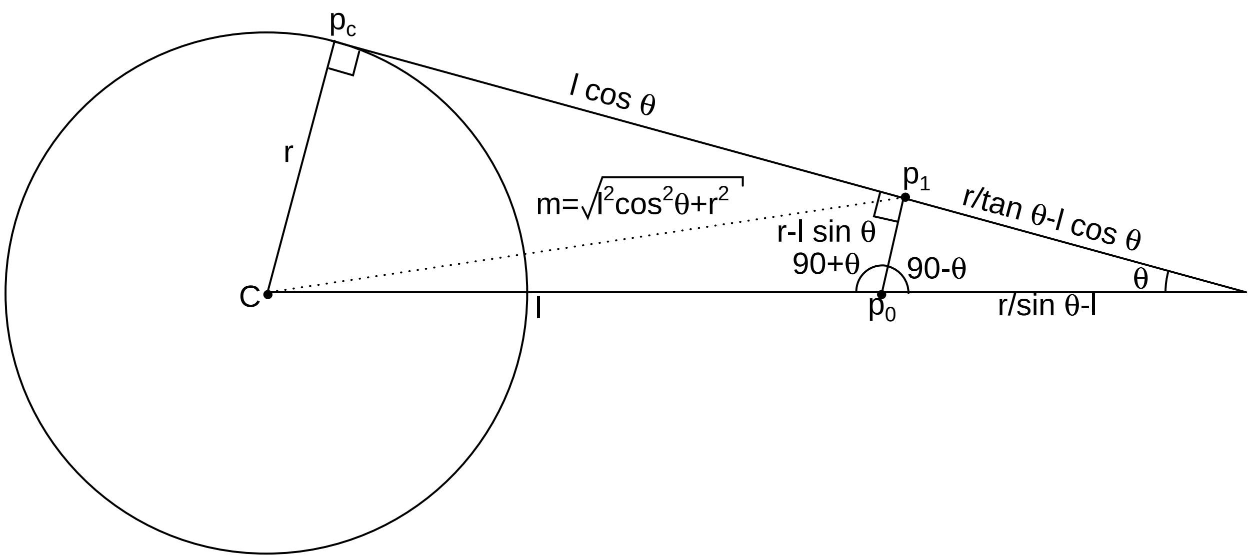


GLR stability - Divergence

Model can diverge!



GLR stability - Model



- Converges if $|\sin\theta| > \frac{r}{l}$
- Need increasingly small angles to diverge as l increases
- Worst-case divergence goes as $r\sqrt{t}$
- Divergence translates to local models

GLR stability - continuous

$$\frac{dl}{dt} = \pm r \sin\theta - l \sin^2\theta$$

Then, for convergence:

$$E\left(\frac{dl}{dt}\right) < 0$$

Taking the worst-case,

$$E(r\,|\sin\theta| - l\sin^2\theta) < 0$$

Which gives the event horizon:

$$\sin\theta = \frac{r}{l}$$

GLR stability - discrete

$$m = \sqrt{l^2 \cos^2 \theta + r^2}$$

$$m^2 = l^2(1 - \sin^2 \theta) + r^2$$

$$m^2 - l^2 = l^2 \sin^2 \theta + r^2$$

Expected magnitude of the drift:

$$E(m^2 - l^2) = E(r^2 - l^2 \sin^2 \theta)$$

For new point closer than old point: $E(m^2 - l^2) < 0$

$$E(r^2 - l^2 \sin^2 \theta) < 0$$

Same event horizon as continuous time case:

$$|\sin \theta| = \frac{r}{l}$$

Cartesian feedback