# Memristors and Nonlinear Programming

Dr. Bharathwaj "Bharath" Muthuswamy

Visiting Assistant Professor in Computer Science
The College of New Jersey
muthuswb@tcnj.edu

March 28th 2018

## About me...

- BS (2002), MS (2005), PhD (2009) in EECS from the University of California, Berkeley
  (advisors: Dr. Leon O. Chua, Dr. Pravin P. Varaiya)

# About me...

- BS (2002), MS (2005), PhD (2009) in EECS from the University of California, Berkeley
  (advisors: Dr. Leon O. Chua, Dr. Pravin P. Varaiya)
    - For my MS, I worked on biomimetic bipedal robotics using Central Pattern Generators (I did not work on this after 2006)
    - For my PhD, my primary contribution was designing, implementing and rigorously proving the existence of chaos in the Muthuswamy-Chua system (circuit): an inductor-capacitor-memristor circuit in series (parallel)

# About me...

- BS (2002), MS (2005), PhD (2009) in EECS from the University of California, Berkeley
(advisors: Dr. Leon O. Chua, Dr. Pravin P. Varaiya)
    - For my MS, I worked on biomimetic bipedal robotics using Central Pattern Generators (I did not work on this after 2006)
    - For my PhD, my primary contribution was designing, implementing and rigorously proving the existence of chaos in the Muthuswamy-Chua system (circuit): an inductor-capacitor-memristor circuit in series (parallel)
- Areas of interest:
    - Nonlinear Dynamics (Circuits). Specifically: chaotic circuits and memristors
    - Embedded (FPGA) Systems and Education

# Presentation Goal and Organization

- Goal of this talk: Discuss ongoing work from an idea developed at TCNJ - memristor applications to nonlinear programming (specifically, convex optimization)

# Presentation Goal and Organization

- Goal of this talk: Discuss ongoing work from an idea developed at TCNJ - memristor applications to nonlinear programming (specifically, convex optimization)
- Organization:
    - Introduction to the Memristor
        - Fundamental Circuit Variables and Elements
        - Memristor Mathematical Formulation (Gedanken-Experiment)
        - Memristive Models of RRAM (Resistive Random Access Memory)

# Presentation Goal and Organization

- Goal of this talk: Discuss ongoing work from an idea developed at TCNJ - memristor applications to nonlinear programming (specifically, convex optimization)
- Organization:
  - Introduction to the Memristor
    - Fundamental Circuit Variables and Elements
    - Memristor Mathematical Formulation (Gedanken-Experiment)
    - Memristive Models of RRAM (Resistive Random Access Memory)
  - The Memristor Crossbar Array

# Presentation Goal and Organization

- Goal of this talk: Discuss ongoing work from an idea developed at TCNJ - memristor applications to nonlinear programming (specifically, convex optimization)
- Organization:
  - Introduction to the Memristor
    - Fundamental Circuit Variables and Elements
    - Memristor Mathematical Formulation (Gedanken-Experiment)
    - Memristive Models of RRAM (Resistive Random Access Memory)
  - The Memristor Crossbar Array
  - Memristors and Nonlinear Programming
    - Introduction to Nonlinear Programming
    - Introduction to ADMM (Alternative Direction Method of Multipliers)
    - ADMM Using the Memristor Crossbar Array

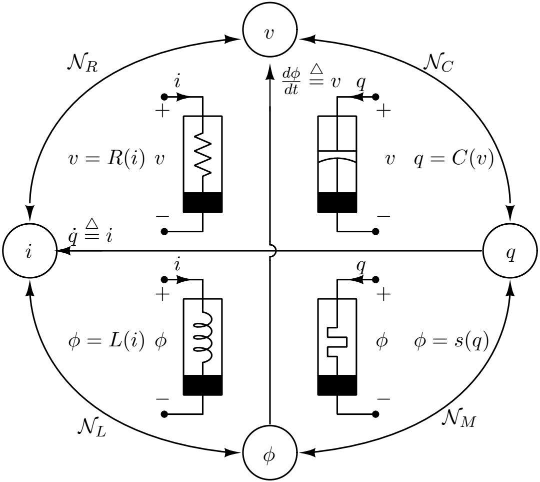
# Presentation Goal and Organization

- Goal of this talk: Discuss ongoing work from an idea developed at TCNJ - memristor applications to nonlinear programming (specifically, convex optimization)
- Organization:
  - Introduction to the Memristor
    - Fundamental Circuit Variables and Elements
    - Memristor Mathematical Formulation (Gedanken-Experiment)
    - Memristive Models of RRAM (Resistive Random Access Memory)
  - The Memristor Crossbar Array
  - Memristors and Nonlinear Programming
    - Introduction to Nonlinear Programming
    - Introduction to ADMM (Alternative Direction Method of Multipliers)
    - ADMM Using the Memristor Crossbar Array
  - Analog emulation of Memristor RRAM

# Presentation Goal and Organization

- Goal of this talk: Discuss ongoing work from an idea developed at TCNJ - memristor applications to nonlinear programming (specifically, convex optimization)
- Organization:
  - Introduction to the Memristor
    - Fundamental Circuit Variables and Elements
    - Memristor Mathematical Formulation (Gedanken-Experiment)
    - Memristive Models of RRAM (Resistive Random Access Memory)
  - The Memristor Crossbar Array
  - Memristors and Nonlinear Programming
    - Introduction to Nonlinear Programming
    - Introduction to ADMM (Alternative Direction Method of Multipliers)
    - ADMM Using the Memristor Crossbar Array
  - Analog emulation of Memristor RRAM
  - Conclusion and Q/A

# Fundamental Circuit Variables and Elements



# Memristor: Mathematical Formulation (Gedanken-Experiment)

# Memristor: Mathematical Formulation (Gedanken-Experiment)

- A memristor (menductor) defines a relationship between $\phi$ and $q$ ($q$ and $\phi$):

$$\phi \triangleq s(q) \tag{1}$$

# Memristor: Mathematical Formulation (Gedanken-Experiment)

- A memristor (menductor) defines a relationship between $\phi$ and $q$ ($q$ and $\phi$):

$$\phi \triangleq s(q) \qquad (1)$$

- In terms of current and voltage:

$$v(t) = M(q(t))i(t) \qquad (2)$$

Here, $M(q(t)) = M(\int_{-\infty}^{t} i(\tau)d\tau)$. $M$ is the memristance function

# Memristor: Mathematical Formulation (Gedanken-Experiment)

- A memristor (menductor) defines a relationship between $\phi$ and $q$ ($q$ and $\phi$):

$$\phi \triangleq s(q) \tag{1}$$

- In terms of current and voltage:

$$v(t) = M(q(t))i(t) \tag{2}$$

  Here, $M(q(t)) = M(\int_{-\infty}^{t} i(\tau)d\tau)$. $M$ is the memristance function
- Eq. (2) defines an ideal memristor (Josephson junctions)
- Generalization to a nonideal memristor:

$$\begin{aligned} v(t) &= M(\mathbf{x}(t))i(t) \\ \dot{\mathbf{x}}(t) &= f(\mathbf{x}, i) \end{aligned} \tag{3}$$

# Memristor: Mathematical Formulation (Gedanken-Experiment)

- A memristor (menductor) defines a relationship between $\phi$ and $q$ ($q$ and $\phi$):

$$\phi \triangleq s(q) \tag{1}$$

- In terms of current and voltage:

$$v(t) = M(q(t))i(t) \tag{2}$$

  Here, $M(q(t)) = M(\int_{-\infty}^{t} i(\tau)d\tau)$. $M$ is the memristance function
- Eq. (2) defines an ideal memristor (Josephson junctions)
- Generalization to a nonideal memristor:

$$\begin{aligned} v(t) &= M(\mathbf{x}(t))i(t) \\ \dot{\mathbf{x}}(t) &= f(\mathbf{x}, i) \end{aligned} \tag{3}$$

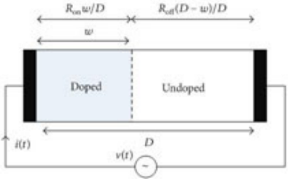
- Discharge tubes (internal state $x$ = number of conduction electrons $n$):

$$v = M(n)i \tag{4}$$

$$\frac{dn}{dt} = -\beta n + \alpha M(n)i^2 \tag{5}$$

# Memristive Models of RRAM

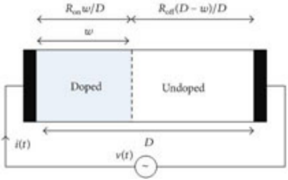
- RRAM (ReRAM) memristive models were popularized by HP (2008)



$$M(w(t)) = R_{on}\frac{w(t)}{D} + R_{off}\left(\frac{D-w(t)}{D}\right) \tag{6}$$

# Memristive Models of RRAM

- RRAM (ReRAM) memristive models were popularized by HP (2008)

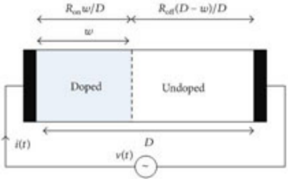


$$M(w(t)) = R_{on}\frac{w(t)}{D} + R_{off}\left(\frac{D-w(t)}{D}\right) \tag{6}$$

- Concept of forming and destroying conductive filaments through an insulating material

# Memristive Models of RRAM

- RRAM (ReRAM) memristive models were popularized by HP (2008)

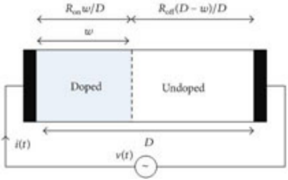


$$M(w(t)) = R_{on}\frac{w(t)}{D} + R_{off}\left(\frac{D-w(t)}{D}\right) \tag{6}$$

- Concept of forming and destroying conductive filaments through an insulating material
- For HP RRAM, oxygen vacancies act as conductive filaments

# Memristive Models of RRAM

- RRAM (ReRAM) memristive models were popularized by HP (2008)

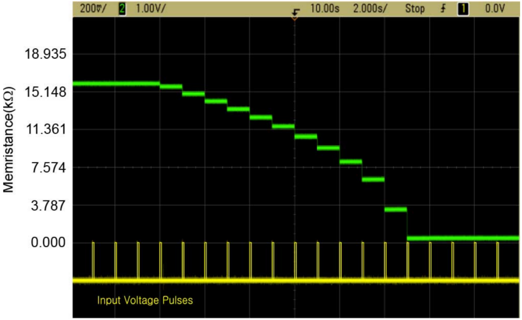


$$M(w(t)) = R_{on}\frac{w(t)}{D} + R_{off}\left(\frac{D-w(t)}{D}\right) \quad (6)$$

- Concept of forming and destroying conductive filaments through an insulating material
- For HP RRAM, oxygen vacancies act as conductive filaments
- Usually, $R_{off} >> R_{on}$

# Memristive Models of RRAM

- Changing memristance for input current pulses



# Memristive Models of RRAM

- Potential advantages (as compared to traditional three-terminal transistor based memory):

# Memristive Models of RRAM

- Potential advantages (as compared to traditional three-terminal transistor based memory):
  - Since a memristor is a two-terminal device, smaller footprint ⇒ higher memory densities

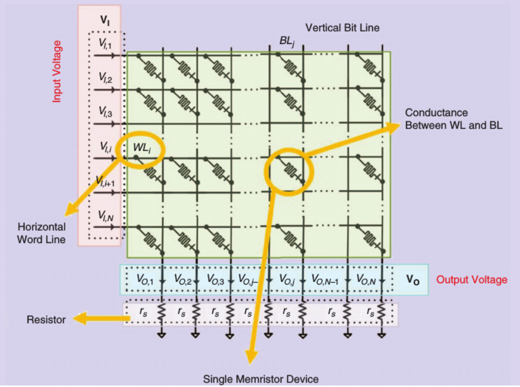
# Memristive Models of RRAM

- Potential advantages (as compared to traditional three-terminal transistor based memory):
  - Since a memristor is a two-terminal device, smaller footprint ⇒ higher memory densities
  - Non-volatile (resistance is retained when $v(t) = 0$) analog (continuous change in resistance values) memory

# Memristive Models of RRAM

- Potential advantages (as compared to traditional three-terminal transistor based memory):
  - Since a memristor is a two-terminal device, smaller footprint ⇒ higher memory densities
  - Non-volatile (resistance is retained when $v(t) = 0$) analog (continuous change in resistance values) memory
- However, a variety of fabrication challenges associated with RRAM (discussed in analog emulation of memristors)

# Memristor Crossbar Array Schematic



# Memristor Crossbar Array Schematic

- Output voltage(s) $\mathbf{V_O}$ as a function of input voltage(s) $\mathbf{V_I}$:

# Memristor Crossbar Array Schematic

- Output voltage(s) $\mathbf{V_O}$ as a function of input voltage(s) $\mathbf{V_I}$:

$$\begin{bmatrix} V_{O,1} \\ \vdots \\ V_{O,M} \end{bmatrix} = \begin{bmatrix} \frac{g_{1,1}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \frac{g_{1,2}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \cdots & \frac{g_{1,N}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{g_{M,1}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \frac{g_{M,2}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \cdots & \frac{g_{M,N}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} \end{bmatrix} \begin{bmatrix} V_{I,1} \\ \vdots \\ V_{I,N} \end{bmatrix} \tag{7}$$

$$\mathbf{V_O} = \mathbf{C}\mathbf{V_I} \tag{8}$$

# Memristor Crossbar Array Schematic

- Output voltage(s) $\mathbf{V_O}$ as a function of input voltage(s) $\mathbf{V_I}$:

$$\begin{bmatrix} V_{O,1} \\ \vdots \\ V_{O,M} \end{bmatrix} = \begin{bmatrix} \frac{g_{1,1}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \frac{g_{1,2}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \cdots & \frac{g_{1,N}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{g_{M,1}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \frac{g_{M,2}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} & \cdots & \frac{g_{M,N}}{g_s + \sum\limits_{k=1}^{N} g_{1,K}} \end{bmatrix} \begin{bmatrix} V_{I,1} \\ \vdots \\ V_{I,N} \end{bmatrix} \tag{7}$$

$$\mathbf{V_O} = \mathbf{C}\mathbf{V_I} \tag{8}$$

- Notice that a memristor crossbar array has $O(1)$ time complexity with respect to analog matrix multiplicaton

# Memristor Crossbar Array Schematic

- Output voltage(s) $\mathbf{V_O}$ as a function of input voltage(s) $\mathbf{V_I}$:

$$\begin{bmatrix} V_{O,1} \\ \vdots \\ V_{O,M} \end{bmatrix} = \begin{bmatrix} \frac{g_{1,1}}{g_s + \sum_{k=1}^{N} g_{1,K}} & \frac{g_{1,2}}{g_s + \sum_{k=1}^{N} g_{1,K}} & \cdots & \frac{g_{1,N}}{g_s + \sum_{k=1}^{N} g_{1,K}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{g_{M,1}}{g_s + \sum_{k=1}^{N} g_{1,K}} & \frac{g_{M,2}}{g_s + \sum_{k=1}^{N} g_{1,K}} & \cdots & \frac{g_{M,N}}{g_s + \sum_{k=1}^{N} g_{1,K}} \end{bmatrix} \begin{bmatrix} V_{I,1} \\ \vdots \\ V_{I,N} \end{bmatrix} \tag{7}$$

$$\mathbf{V_O} = \mathbf{C}\mathbf{V_I} \tag{8}$$

- Notice that a memristor crossbar array has $O(1)$ time complexity with respect to analog matrix multiplicaton
- Primary challenges in memristor crossbar array technology: scalability (addressed by using arbiters), process-variablity (hot topic of research), negative coefficients...

# Memristor Crossbar Array Schematic

- Negative coefficients...

# Memristor Crossbar Array Schematic

- Negative coefficients...

$$\begin{aligned}\mathbf{V_0} &= \mathbf{C}\mathbf{V_I} \\ &= \left(\mathbf{C}_+ - (-\mathbf{C}_-)\right)\mathbf{V_I} \\ &= \left(\mathbf{C}_+ - (\mathbf{B}\mathbf{D})\right)\mathbf{V_I} \qquad (9)\end{aligned}$$

# Memristor Crossbar Array Schematic

- Negative coefficients...

$$\begin{aligned}\mathbf{V_0} &= \mathbf{C}\mathbf{V_I} \\ &= (\mathbf{C}_+ - (-\mathbf{C}_-))\,\mathbf{V_I} \\ &= (\mathbf{C}_+ - (\mathbf{BD}))\,\mathbf{V_I} \qquad (9)\end{aligned}$$

- Mathematical "trick" of auxiliary variables in defining:
    - $\tilde{N}$: number of nonzero columns of $-\mathbf{C}_-$
    - $\mathbf{B} \in \mathbb{R}^{N \times \tilde{N}}$: nonzero columns of $-\mathbf{C}_-$
    - $\mathbf{D} \in \mathbb{R}^{\tilde{N} \times N}$: indices of the nonzero columns of $-\mathbf{C}_-$

# Memristor Crossbar Array Schematic

- Negative coefficients...

$$\begin{aligned}\mathbf{V_0} &= \mathbf{C}\mathbf{V_I} \\ &= \left(\mathbf{C}_+ - (-\mathbf{C}_-)\right)\mathbf{V_I} \\ &= \left(\mathbf{C}_+ - (\mathbf{B}\mathbf{D})\right)\mathbf{V_I} \qquad (9)\end{aligned}$$

- Mathematical "trick" of auxiliary variables in defining:
  - $\tilde{N}$: number of nonzero columns of $-\mathbf{C}_-$
  - $\mathbf{B} \in \mathbb{R}^{N \times \tilde{N}}$: nonzero columns of $-\mathbf{C}_-$
  - $\mathbf{D} \in \mathbb{R}^{\tilde{N} \times N}$: indices of the nonzero columns of $-\mathbf{C}_-$
- Usually $\tilde{N} << N$, size of resulting memristor crossbar array is $(N + \tilde{N}) \times (N + \tilde{N})$

# Memristor Crossbar Array Schematic

- Example...

# Memristor Crossbar Array Schematic

- Example...

$$
\begin{aligned}
\mathbf{C} &= \begin{bmatrix} 1 & -3 & 0 \\ 0 & -0.75 & 4 \\ -2 & -5 & 6 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 6 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 0 \\ 0 & 0.75 & 0 \\ 2 & 5 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 6 \end{bmatrix} - \begin{bmatrix} 0 & 3 \\ 0 & 0.75 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
&= \left(\mathbf{C}_{+} - (\mathbf{B}\mathbf{D})\right)
\end{aligned}
\tag{10}
$$

# Memristor Crossbar Array Schematic

- Example...

$$\begin{aligned}
\mathbf{C} &= \begin{bmatrix} 1 & -3 & 0 \\ 0 & -0.75 & 4 \\ -2 & -5 & 6 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 6 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 0 \\ 0 & 0.75 & 0 \\ 2 & 5 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 6 \end{bmatrix} - \begin{bmatrix} 0 & 3 \\ 0 & 0.75 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
&= (\mathbf{C}_+ - (\mathbf{BD}))
\end{aligned} \tag{10}$$

- Now, a variety of nonlinear programming problems can be mapped onto the memristor crossbar array...

# Introduction to Nonlinear Programming

- Nonlinear programming is essentially the concept of finding an optimal solution to a problem, given a set of constraints

# Introduction to Nonlinear Programming

- Nonlinear programming is essentially the concept of finding an optimal solution to a problem, given a set of constraints
- Example: Consider the problem of sizing devices on an IC (Integrated Circuit) - task of choosing the width and length of each device in the IC:

# Introduction to Nonlinear Programming

- Nonlinear programming is essentially the concept of finding an optimal solution to a problem, given a set of constraints
- Example: Consider the problem of sizing devices on an IC (Integrated Circuit) - task of choosing the width and length of each device in the IC:
    - Variables: width, length of each device

# Introduction to Nonlinear Programming

- Nonlinear programming is essentially the concept of finding an optimal solution to a problem, given a set of constraints
- Example: Consider the problem of sizing devices on an IC (Integrated Circuit) - task of choosing the width and length of each device in the IC:
  - Variables: width, length of each device
  - Function to be minimized (Objective function): Total power dissipated

# Introduction to Nonlinear Programming

- Nonlinear programming is essentially the concept of finding an optimal solution to a problem, given a set of constraints
- Example: Consider the problem of sizing devices on an IC (Integrated Circuit) - task of choosing the width and length of each device in the IC:
    - Variables: width, length of each device
    - Function to be minimized (Objective function): Total power dissipated
    - Constraint functions: timing threshold, device area, total area etc.

# Convex Optimization

- Mathematically ($f_0$: objective function, $\mathbf{f}_i$: constraint function(s) $f_0, \mathbf{f}_i : \mathbb{R}^n \rightarrow \mathbb{R}$):

$$\text{minimize} : f_0(\mathbf{x}) \tag{11}$$

$$\text{subject to} : \mathbf{f}_i(\mathbf{x}) \leq \mathbf{b}_i \quad i = 1, \cdots, m \tag{12}$$

# Convex Optimization

- Mathematically ($f_0$: objective function, $\mathbf{f}_i$: constraint function(s) $f_0, \mathbf{f}_i : \mathbb{R}^n \to \mathbb{R}$):

$$\text{minimize} : f_0(\mathbf{x}) \tag{11}$$

$$\text{subject to} : \mathbf{f}_i(\mathbf{x}) \leq \mathbf{b}_i \quad i = 1, \cdots, m \tag{12}$$

- We have a convex optimization problem iff $f_0, \mathbf{f}_i$ are convex:

$$f_k(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_k(\mathbf{x}) + \beta f_k(\mathbf{y}) \tag{13}$$

with $\alpha + \beta \geq 1, \alpha \geq 0, \beta \geq 0$

# Introduction to ADMM

- ADMM : Alternative Direction Method of Multipliers - decompose optimization variable into two simpler groups

# Introduction to ADMM

- ADMM : Alternative Direction Method of Multipliers - decompose optimization variable into two simpler groups
- One of the main steps in ADMM involves solving the optimization problem:

$$\text{minimize}: \frac{\rho}{2}\|\mathbf{x}-\alpha\|_2^2 \tag{14}$$

$$\text{subject to}: \mathbf{G}\mathbf{x}=\mathbf{h} \tag{15}$$

where: $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable, $\alpha$ is a "coupling" parameter, $\mathbf{G} \in \mathbb{R}^{l\times n}$ and $\mathbf{h} \in \mathbb{R}^l$ are parameters.

# ADMM Using the Memristor Crossbar Array

- Using the Karush-Kuhn-Tucker (KKT) necessary and sufficient conditions, we get the following equivalent equation to solve:

$$\rho(\mathbf{x} - \alpha) + \mathbf{G}^T\lambda = \mathbf{0} \tag{16}$$

$$\mathbf{G}\mathbf{x} = \mathbf{h} \tag{17}$$

where $\lambda \in \mathbb{R}^l$ is the Lagrange multiplier. Eqs. (16) and (17) can be rewritten as:

# ADMM Using the Memristor Crossbar Array

- Using the Karush-Kuhn-Tucker (KKT) necessary and sufficient conditions, we get the following equivalent equation to solve:

$$\rho(\mathbf{x} - \alpha) + \mathbf{G}^T \lambda = \mathbf{0} \tag{16}$$

$$\mathbf{G}\mathbf{x} = \mathbf{h} \tag{17}$$

where $\lambda \in \mathbb{R}^l$ is the Lagrange multiplier. Eqs. (16) and (17) can be rewritten as:

$$\begin{bmatrix} \rho\mathbf{I} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \rho\alpha \\ \mathbf{h} \end{bmatrix} \tag{18}$$

# ADMM Using the Memristor Crossbar Array

- Using the Karush-Kuhn-Tucker (KKT) necessary and sufficient conditions, we get the following equivalent equation to solve:

$$\rho(\mathbf{x} - \alpha) + \mathbf{G}^T\lambda = \mathbf{0} \tag{16}$$

$$\mathbf{G}\mathbf{x} = \mathbf{h} \tag{17}$$

where $\lambda \in \mathbb{R}^l$ is the Lagrange multiplier. Eqs. (16) and (17) can be rewritten as:

$$\begin{bmatrix} \rho\mathbf{I} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \rho\alpha \\ \mathbf{h} \end{bmatrix} \tag{18}$$

- The system in Eq. (18) can be efficiently mapped to memristor crossbars by configuring their memristance values according to $\mathbf{C} = \begin{bmatrix} \rho\mathbf{I} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix}$

# ADMM Using the Memristor Crossbar Array

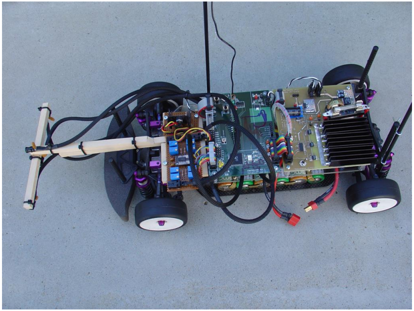
- Sample application: Model Predictive Control (MPC) reformulated as a Mixed Linear Complementarity Problem (MLCP) For Autonomous Vehicles!

# ADMM Using the Memristor Crossbar Array

- Sample application: Model Predictive Control (MPC) reformulated as a Mixed Linear Complementarity Problem (MLCP) For Autonomous Vehicles!
- Proposal for a TCNJ ECE Senior Design Project in AY 2018 - 2019...

# ADMM Using the Memristor Crossbar Array

- Sample application: Model Predictive Control (MPC) reformulated as a Mixed Linear Complementarity Problem (MLCP) For Autonomous Vehicles!
- Proposal for a TCNJ ECE Senior Design Project in AY 2018 - 2019...



- NATCAR, Berkeley EECS 192

# Analog Emulation of Memristor RRAM

- Some major issues with current RRAM technology:
  - Lack of proper models that predict physical device behavior

# Analog Emulation of Memristor RRAM

- Some major issues with current RRAM technology:
  - Lack of proper models that predict physical device behavior
  - Memristive (RRAM) devices fall well short of CMOS in terms of endurance

# Analog Emulation of Memristor RRAM

- Some major issues with current RRAM technology:
    - Lack of proper models that predict physical device behavior
    - Memristive (RRAM) devices fall well short of CMOS in terms of endurance
- Since memristive RRAM models and devices are still a topic of research...

# Analog Emulation of Memristor RRAM

- Some major issues with current RRAM technology:
  - Lack of proper models that predict physical device behavior
  - Memristive (RRAM) devices fall well short of CMOS in terms of endurance
- Since memristive RRAM models and devices are still a topic of research...
- ...there is a need for robust (analog) emulators

# CMOS-based Analog Emulation of Memristor RRAM: Block Diagram

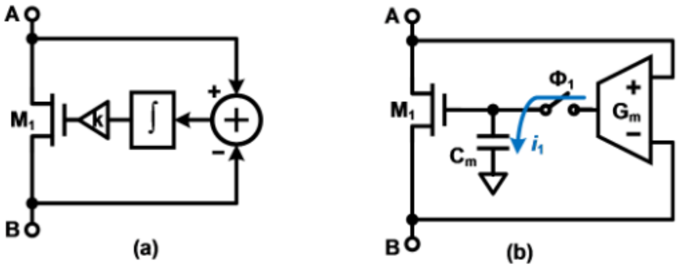
- Analog emulation of memristors has been done using opamps etc.

# CMOS-based Analog Emulation of Memristor RRAM: Block Diagram

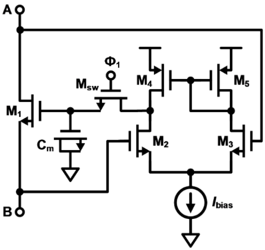
- Analog emulation of memristors has been done using opamps etc.
- But we would like solutions that are easy to fabricate in CMOS

# CMOS-based Analog Emulation of Memristor RRAM: Block Diagram

- Analog emulation of memristors has been done using opamps etc.
- But we would like solutions that are easy to fabricate in CMOS

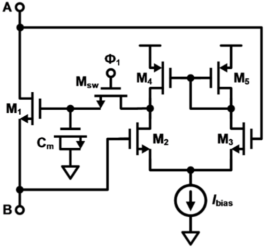


# CMOS-based Analog Emulation of Memristor RRAM: Implementation



$$
\begin{aligned}
I &\approx K_n \frac{W}{L}\left(V_G - V_{C_M} - V_{T_N}\right)\cdot V_{AB} \\
&= K_n \frac{W}{L}\left(x - V_{C_M} - V_{T_N}\right)\cdot V_{AB} \qquad (19)
\end{aligned}
$$

# CMOS-based Analog Emulation of Memristor RRAM: Implementation



$$
\begin{aligned}
I &\approx K_n \frac{W}{L}\left(V_G - V_{C_M} - V_{T_N}\right)\cdot V_{AB} \\
&= K_n \frac{W}{L}\left(x - V_{C_M} - V_{T_N}\right)\cdot V_{AB} \qquad (19)
\end{aligned}
$$

$$
\begin{aligned}
\dot{x} &= \frac{G_M(V_{AB})}{C_M} \\
&= f(V_{AB}) \qquad (20)
\end{aligned}
$$

# Conclusion, Q/A and References

- Ongoing research:
  - Implement convex optimization problems, say MLCP, via ADMM on memristor crossbar array (work jointly done with TCNJ faculty (Dr. Adegbege) and students (Paul B., Jake B., Matt K.)):
    - Simulation framework for memristors (crossbar array): QUCS (Quite Universal Circuit Simulator), MAPP (Berkeley Model and Algorithm Prototyping Platform)
    - Synthesize memristor crossbar array using Analog RRAM memristor emulator
    - Senior design project for AY 2018-2019
  - Memristive models for atrial fibrillation (with Dr. S. T. Mathew)
  - Noise induced chaos in the Muthuswamy-Chua system (with Dr. K. Ganesan and Dr. S. Banerjee)
- Questions?

# References

- "Memristor - The Missing Circuit Element". Chua, L. O. IEEE Transactions on Circuit Theory. 18(5), pp. 507 - 519, 1971.
- "Memristive Devices and Systems". Chua, L. O. and Kang, S. M. Proceedings of the IEEE. 64(2), pp. 209 - 223, 1976.
- "Introduction to Nonlinear Circuits and Networks". Muthuswamy, B. and Banerjee, S. Springer, 2018.
- "A Memristor-Based Optimization Framework for Artificial Intelligence Applications". Liu, S. et. al. IEEE Circuits and Systems Magazine, pp. 29 - 44, First Quarter 2018.
- "Convex Optimization". Boyd, S. and Vandengerghe, L. Cambridge University Press, 1st edition, 2004.
- "Algorithm-Hardware Co-Optimization of the Memristor-Based Framework for Solving SOCP and Homogeneous QCQP Problems". Ren, A. et. al. 22nd IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 788 - 793, 2017.
- "Memristor Emulator for Memristor Circuit Applications". Kim, H. et. al. IEEE Transactions on Circuits and Systems - I : Regular Papers, 59(10), pp. 2422 - 2431, October 2012.
- "A Compact CMOS Memristor Emulator Circuit and its Applications". Saxena, V. arXiv:1711.06819v1, November 18th 2017.
- "Well-Posed Models of Memristive Devices". Wang, T. and Roychowdhury, J. arXiv:1605.04897v1, May 15th 2016.
- "Analog Circuit for Real-time Optimization of Constrained Control". Levenson, R. M. and Adegbege, A. 2016 American Control Conference (ACC), pp. 6947 - 6952.