Controle ótimo um passo à frente via otimização e aplicações

Optimization-based one step ahead optimal control

Amit Bhaya$^1$

amit@nacad.ufrj.br, COPPE/Programa de Engenharia Elétrica$^1$

DME/IM, UFRJ 2023.
Outline

- What is control & optimal control?
- What is one step ahead optimal control (OSAOC)?
- Optimal control via optimization
- Some examples
- Concluding remarks
What is control?

- Dynamical system $\dot{x} = f(x)$: object of study in mathematics
- Vector field on RHS is “God-given”
- Mathematician’s job is to describe evolution of state $x$: equilibria, limit cycles, etc.
- Control theory studies $\dot{x} = f(x, u)$ where $u$ is a control input
- Control engineer’s job is to choose control input $u$ so that evolution described by RHS vector field behaves in prespecified fashion.
- For example, can $u$ be chosen so that all trajectories go to (a globally stable) equilibrium, or limit cycle?
What is control?

- Dynamical system $\dot{x} = f(x)$: object of study in mathematics
- Vector field on RHS is “God-given”
- Mathematician’s job is to describe evolution of state $x$: equilibria, limit cycles, etc.
- Control theory studies $\dot{x} = f(x, u)$ where $u$ is a control input
- Control engineer’s job is to choose control input $u$ so that evolution described by RHS vector field behaves in prespecified fashion.
- For example, can $u$ be chosen so that all trajectories go to (a globally stable) equilibrium, or limit cycle?
What is control?

- Dynamical system $\dot{x} = f(x)$: object of study in mathematics
- Vector field on RHS is “God-given”
- Mathematician’s job is to describe evolution of state $x$: equilibria, limit cycles, etc.

- Control theory studies $\dot{x} = f(x, u)$ where $u$ is a control input
- Control engineer’s job is to choose control input $u$ so that evolution described by RHS vector field behaves in prespecified fashion.
- For example, can $u$ be chosen so that all trajectories go to (a globally stable) equilibrium, or limit cycle?
What is control?

- Dynamical system $\dot{x} = f(x)$: object of study in mathematics
- Vector field on RHS is “God-given”
- Mathematician’s job is to describe evolution of state $x$: equilibria, limit cycles, etc.
- Control theory studies $\dot{x} = f(x, u)$ where $u$ is a control input
- Control engineer’s job is to choose control input $u$ so that evolution described by RHS vector field behaves in prespecified fashion.
- For example, can $u$ be chosen so that all trajectories go to (a globally stable) equilibrium, or limit cycle?
What is control?

- Dynamical system \( \dot{x} = f(x) \): object of study in mathematics
- Vector field on RHS is “God-given”
- Mathematician’s job is to describe evolution of state \( x \): equilibria, limit cycles, etc.
- Control theory studies \( \dot{x} = f(x, u) \) where \( u \) is a control input
- Control engineer’s job is to choose control input \( u \) so that evolution described by RHS vector field behaves in prespecified fashion.
- For example, can \( u \) be chosen so that all trajectories go to (a globally stable) equilibrium, or limit cycle?
What is control?

- Dynamical system \( \dot{x} = f(x) \): object of study in mathematics
- Vector field on RHS is “God-given”
- Mathematician’s job is to describe evolution of state \( x \): equilibria, limit cycles, etc.
- Control theory studies \( \dot{x} = f(x, u) \) where \( u \) is a control input
- Control engineer’s job is to choose control input \( u \) so that evolution described by RHS vector field behaves in prespecified fashion.
- For example, can \( u \) be chosen so that all trajectories go to (a globally stable) equilibrium, or limit cycle?
Linear control: discrete-time case

- Dynamical system \( x(k + 1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m} \): most well-understood case, complete theory.

- In this case, without control \((u(k) = 0, \forall k)\), origin is stable if spectral radius \( \rho(A) < 1 \).

- What if spectral radius \( \rho(A) > 1 \)?

- Assume that state \( x(k) \) is measured and available to engineer.

- Key idea of full state feedback: choose \( u(k) =Fx(k) \)

- Dynamical system under feedback becomes
  \( x(k + 1) = (A + BF)x(k) \)

- New question: given \( A, B \), can \( F \) be chosen such that \( \rho(A + BF) < 1 \)?

- Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system \( x(k + 1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m} \): most well-understood case, complete theory.

- In this case, without control \((u(k) = 0, \forall k)\), origin is stable if spectral radius \(\rho(A) < 1\).

- What if spectral radius \(\rho(A) > 1\)?
- Assume that state \(x(k)\) is measured and available to engineer.
- Key idea of full state feedback: choose \(u(k) = Fx(k)\)
- Dynamical system under feedback becomes \(x(k + 1) = (A + BF)x(k)\)
- New question: given \(A, B\), can \(F\) be chosen such that \(\rho(A + BF) < 1\)?
- Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system $x(k+1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}$: most well-understood case, complete theory.

- In this case, without control ($u(k) = 0, \forall k$), origin is stable if spectral radius $\rho(A) < 1$.

- What if spectral radius $\rho(A) > 1$?

  - Assume that state $x(k)$ is measured and available to engineer.
  - Key idea of full state feedback: choose $u(k) = Fx(k)$
  - Dynamical system under feedback becomes $x(k+1) = (A + BF)x(k)$
  - New question: given $A, B$, can $F$ be chosen such that $\rho(A + BF) < 1$?
  - Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system $x(k+1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}$: most well-understood case, complete theory.

- In this case, without control ($u(k) = 0, \forall k$), origin is stable if spectral radius $\rho(A) < 1$.

- What if spectral radius $\rho(A) > 1$?

- Assume that state $x(k)$ is measured and available to engineer.

  - Key idea of full state feedback: choose $u(k) = Fx(k)$

  - Dynamical system under feedback becomes
    $x(k+1) = (A + BF)x(k)$

  - New question: given $A, B$, can $F$ be chosen such that $\rho(A + BF) < 1$?

  - Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system $x(k + 1) = Ax(k) + Bu(k), x \in R^n, u \in R^m, A \in R^{n \times n}, B \in R^{n \times m}$: most well-understood case, complete theory.
- In this case, without control ($u(k) = 0, \forall k$), origin is stable if spectral radius $\rho(A) < 1$.
- What if spectral radius $\rho(A) > 1$?
- Assume that state $x(k)$ is measured and available to engineer.
- Key idea of full state feedback: choose $u(k) = Fx(k)$
  - Dynamical system under feedback becomes $x(k + 1) = (A + BF)x(k)$
  - New question: given $A, B$, can $F$ be chosen such that $\rho(A + BF) < 1$?
  - Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system \( x(k + 1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m} \): most well-understood case, complete theory.
- In this case, without control \((u(k) = 0, \forall k)\), origin is stable if spectral radius \(\rho(A) < 1\).
- What if spectral radius \(\rho(A) > 1\)?
- Assume that state \(x(k)\) is measured and available to engineer.
- Key idea of full state feedback: choose \(u(k) = Fx(k)\)
- Dynamical system under feedback becomes \(x(k + 1) = (A + BF)x(k)\)
- New question: given \(A, B\), can \(F\) be chosen such that \(\rho(A + BF) < 1\)?
- Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system $x(k+1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}$: most well-understood case, complete theory.

- In this case, without control ($u(k) = 0, \forall k$), origin is stable if spectral radius $\rho(A) < 1$.

- What if spectral radius $\rho(A) > 1$?

- Assume that state $x(k)$ is measured and available to engineer.

- Key idea of full state feedback: choose $u(k) = Fx(k)$

- Dynamical system under feedback becomes $x(k+1) = (A + BF)x(k)$

- New question: given $A, B$, can $F$ be chosen such that $\rho(A + BF) < 1$?

- Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
Linear control: discrete-time case

- Dynamical system $x(k+1) = Ax(k) + Bu(k), x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}$: most well-understood case, complete theory.

- In this case, without control ($u(k) = 0, \forall k$), origin is stable if spectral radius $\rho(A) < 1$.

- What if spectral radius $\rho(A) > 1$?

- Assume that state $x(k)$ is measured and available to engineer.

- Key idea of full state feedback: choose $u(k) = Fx(k)$

- Dynamical system under feedback becomes $x(k+1) = (A + BF)x(k)$

- New question: given $A, B$, can $F$ be chosen such that $\rho(A + BF) < 1$?

- Yes! Under simple algebraic condition (unstable “modes” are “controllable”).
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost) $J(x, u) = \int_0^T (x(t)^T Qx(t) + u(t)^T Ru(t)) dt$ (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
- Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (=fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost)
  \[ J(x, u) = \int_0^T (x(t)^T Qx(t) + u(t)^T Ru(t)) \, dt \] (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
- Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (=fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost) $J(x, u) = \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt$ (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
- Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (=fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
  - In most cases, we wish to do this optimally.
  - Given an objective function (performance index, cost)
    \[ J(x, u) = \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) \, dt \] (for example)
  - Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
  - Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (=fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost)
  $$J(x, u) = \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$
  (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
- Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost)
  $$J(x, u) = \int_0^T \left( x(t)^T Q x(t) + u(t)^T R u(t) \right) dt$$
  (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
- Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (=fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost)
  $J(x, u) = \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt$ (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
  - Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
What is optimal control?

- Dynamical system $\dot{x} = f(x, u)$: continuous-time case.
- Suppose that there is some mission objective: e.g., landing a spacecraft on the moon.
- Also suppose that there are physical constraints: energy (=fuel) has weight, thus limited
- New problem: design control $u$ such that trajectory $x$ fulfills mission & does not violate constraints.
- In most cases, we wish to do this optimally.
- Given an objective function (performance index, cost)
  $$J(x, u) = \int_{0}^{T} (x(t)^T Qx(t) + u(t)^T Ru(t)) \, dt$$
  (for example)
- Can $u(t)$ be chosen such that cost $J(x, u)$ is minimized?
- Yes! Complete theory: Bolza, Bliss, Pontryagin, Bellman.
Optimal control: mathematical formulation

- Given a dynamical system \( \dot{x} = f(x, u) \)
- and a cost function \( J(x, u) \)
- minimize \( J(x, u) \), by choice of \( u(t) \), \( t \in [0, T] \), subject to the dynamics \( \dot{x} = f(x, u) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy an ODE with RHS dependent on the choice \( u \).
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state \( x \) and the control \( u \)!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system $\dot{x} = f(x, u)$
- and a cost function $J(x, u)$
  - minimize $J(x, u)$, by choice of $u(t), t \in [0, T]$, subject to the dynamics $\dot{x} = f(x, u)$
- Control $u$ is a function; trajectory $x$ must satisfy an ODE with RHS dependent on the choice $u$.
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state $x$ and the control $u$!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system $\dot{x} = f(x, u)$
- and a cost function $J(x, u)$
- minimize $J(x, u)$, by choice of $u(t)$, $t \in [0, T]$, subject to the dynamics $\dot{x} = f(x, u)$
- Control $u$ is a function; trajectory $x$ must satisfy an ODE with RHS dependent on the choice $u$.
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state $x$ and the control $u$!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system $\dot{x} = f(x, u)$
- and a cost function $J(x, u)$
- minimize $J(x, u)$, by choice of $u(t), t \in [0, T]$, subject to the dynamics $\dot{x} = f(x, u)$
- Control $u$ is a function; trajectory $x$ must satisfy an ODE with RHS dependent on the choice $u$.
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state $x$ and the control $u$!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system $\dot{x} = f(x, u)$
- and a cost function $J(x, u)$
- minimize $J(x, u)$, by choice of $u(t), t \in [0, T]$, subject to the dynamics $\dot{x} = f(x, u)$
- Control $u$ is a function; trajectory $x$ must satisfy an ODE with RHS dependent on the choice $u$.
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state $x$ and the control $u$!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system \( \dot{x} = f(x, u) \)
- and a cost function \( J(x, u) \)
- minimize \( J(x, u) \), by choice of \( u(t), \ t \in [0, T] \), subject to the dynamics \( \dot{x} = f(x, u) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy an ODE with RHS dependent on the choice \( u \).
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state \( x \) and the control \( u \)!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system \( \dot{x} = f(x, u) \)
- and a cost function \( J(x, u) \)
- minimize \( J(x, u) \), by choice of \( u(t), t \in [0, T] \), subject to the dynamics \( \dot{x} = f(x, u) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy an ODE with RHS dependent on the choice \( u \).
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state \( x \) and the control \( u \)!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system $\dot{x} = f(x, u)$
- and a cost function $J(x, u)$
- minimize $J(x, u)$, by choice of $u(t), t \in [0, T]$, subject to the dynamics $\dot{x} = f(x, u)$
- Control $u$ is a function; trajectory $x$ must satisfy an ODE with RHS dependent on the choice $u$.
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state $x$ and the control $u$!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).

Kalman, Bellman: early 1960s.
Optimal control: mathematical formulation

- Given a dynamical system $\dot{x} = f(x, u)$
- and a cost function $J(x, u)$
- minimize $J(x, u)$, by choice of $u(t), t \in [0, T]$, subject to the dynamics $\dot{x} = f(x, u)$
- Control $u$ is a function; trajectory $x$ must satisfy an ODE with RHS dependent on the choice $u$.
- This is a difficult problem of variational calculus.
- Even when there are no additional constraints on the state $x$ and the control $u$!
- Without additional constraints: complete theory in LQ case (linear system, quadratic cost)
- LQ case: optimal control is found in feedback form (solving Riccati equation or by dynamic programming).
- Kalman, Bellman: early 1960s.
Given a dynamical system
\[ x(k+1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
and a cost function \( J(x(k), u(k)) \)
minimize \( J(x(k), u(k)) \), by choice of \( u(k), k \in [1, K] \), subject to the dynamics \( x(k+1) = f(x(k), u(k)) \)
Control \( u \) is a function; trajectory \( x \) must satisfy a recurrence relation dependent on the choice \( u \).
The recurrence relations
\[ x(k+1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
can be regarded as \( K \) equality constraints
We can now reinterpret the optimal control problem as an optimization problem.
Optimal control: discrete-time case

- Given a dynamical system
  \[ x(k + 1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
- and a cost function \( J(x(k), u(k)) \)
  - minimize \( J(x(k), u(k)) \), by choice of \( u(k), k \in [1, K] \), subject to the dynamics \( x(k + 1) = f(x(k), u(k)) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy a recurrence relation dependent on the choice \( u \).
- The recurrence relations
  \[ x(k + 1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
  can be regarded as \( K \) equality constraints
- We can now reinterpret the optimal control problem as an optimization problem.
Optimal control: discrete-time case

- Given a dynamical system
  \[ x(k+1) = f(x(k), u(k)), \ k = 1, 2, \ldots, K \]
- and a cost function \( J(x(k), u(k)) \)
- minimize \( J(x(k), u(k)) \), by choice of \( u(k), k \in [1, K] \), subject to the dynamics \( x(k+1) = f(x(k), u(k)) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy a recurrence relation dependent on the choice \( u \).
- The recurrence relations
  \[ x(k+1) = f(x(k), u(k)), \ k = 1, 2, \ldots, K \] can be regarded as \( K \) equality constraints
- We can now reinterpret the optimal control problem as an optimization problem.
Optimal control: discrete-time case

- Given a dynamical system
  \[ x(k + 1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
- and a cost function \( J(x(k), u(k)) \)
- minimize \( J(x(k), u(k)) \), by choice of \( u(k), \quad k \in [1, K] \), subject to the dynamics \( x(k + 1) = f(x(k), u(k)) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy a recurrence relation dependent on the choice \( u \).
- The recurrence relations
  \[ x(k + 1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
  can be regarded as \( K \) equality constraints
- We can now reinterpret the optimal control problem as an optimization problem.
Optimal control: discrete-time case

- Given a dynamical system
  \( x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K \)
- and a cost function \( J(x(k), u(k)) \)
- minimize \( J(x(k), u(k)) \), by choice of \( u(k), k \in [1, K] \), subject to the dynamics \( x(k + 1) = f(x(k), u(k)) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy a recurrence relation dependent on the choice \( u \).
- The recurrence relations
  \( x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K \) can be regarded as \( K \) equality constraints
- We can now reinterpret the optimal control problem as an optimization problem.
Optimal control: discrete-time case

- Given a dynamical system
  \[ x(k + 1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
- and a cost function \( J(x(k), u(k)) \)
- minimize \( J(x(k), u(k)) \), by choice of \( u(k), k \in [1, K] \), subject to the dynamics \( x(k + 1) = f(x(k), u(k)) \)
- Control \( u \) is a function; trajectory \( x \) must satisfy a recurrence relation dependent on the choice \( u \).
- The recurrence relations
  \[ x(k + 1) = f(x(k), u(k)), \quad k = 1, 2, \ldots, K \]
  can be regarded as \( K \) equality constraints
- We can now reinterpret the optimal control problem as an optimization problem.
Optimal control as mathematical optimization

- minimize $J(x(k), u(k))$, by choice of $u(k), k \in [1, K], x(k), k \in [1, K + 1]$
- subject to the $K$ equality constraints $x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K$
- This is a standard nonlinear programming problem, convex if $J$ is convex and $f$ is affine
- Easy to solve if convex, even if $J$ is nonlinear
- Convex constraints on $x$ and $u$ can be added without increasing the difficulty of solution!
- Price to be paid: $2K + 1$ decision variables.
Optimal control as mathematical optimization

- minimize $J(x(k), u(k))$, by choice of $u(k), k \in [1, K], x(k), k \in [1, K + 1]$
- subject to the $K$ equality constraints
  
  $x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K$

- This is a standard nonlinear programming problem, convex if $J$ is convex and $f$ is affine
- Easy to solve if convex, even if $J$ is nonlinear
- Convex constraints on $x$ and $u$ can be added without increasing the difficulty of solution!
- Price to be paid: $2K + 1$ decision variables.
Optimal control as mathematical optimization

- minimize $J(x(k), u(k))$, by choice of $u(k), k \in [1, K], x(k), k \in [1, K + 1]$
- subject to the $K$ equality constraints $x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K$
- This is a standard nonlinear programming problem, convex if $J$ is convex and $f$ is affine
  - Easy to solve if convex, even if $J$ is nonlinear
  - Convex constraints on $x$ and $u$ can be added without increasing the difficulty of solution!
  - Price to be paid: $2K + 1$ decision variables.
Optimal control as mathematical optimization

- minimize $J(x(k), u(k))$, by choice of $u(k), k \in [1, K], x(k), k \in [1, K + 1]$
- subject to the $K$ equality constraints $x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K$
- This is a standard nonlinear programming problem, convex if $J$ is convex and $f$ is affine
- Easy to solve if convex, even if $J$ is nonlinear
- Convex constraints on $x$ and $u$ can be added without increasing the difficulty of solution!
- Price to be paid: $2K + 1$ decision variables.
Optimal control as mathematical optimization

- minimize $J(x(k), u(k))$, by choice of $u(k), k \in [1, K], x(k), k \in [1, K + 1]$
- subject to the $K$ equality constraints $x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K$
- This is a standard nonlinear programming problem, convex if $J$ is convex and $f$ is affine
- Easy to solve if convex, even if $J$ is nonlinear
- **Convex constraints on $x$ and $u$ can be added without increasing the difficulty of solution!**
- Price to be paid: $2K + 1$ decision variables.
Optimal control as mathematical optimization

- minimize $J(x(k), u(k))$, by choice of $u(k), k \in [1, K], x(k), k \in [1, K + 1]$
- subject to the $K$ equality constraints $x(k + 1) = f(x(k), u(k)), k = 1, 2, \ldots, K$
- This is a standard nonlinear programming problem, convex if $J$ is convex and $f$ is affine
- Easy to solve if convex, even if $J$ is nonlinear
- **Convex constraints on $x$ and $u$ can be added without increasing the difficulty of solution!**
- Price to be paid: $2K + 1$ decision variables.
Krasovskii (1957): Impulse response → linear program formulation (not explicitly identified as such).
Zadeh (1962): made Krasovskii’s contribution explicit as linear program.
This talk

- Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.
- Not (necessarily) linear quadratic (LQ).
- Examples from Business Dynamics: index which takes both the management and financial aspects into account.
- Examples from numerical algorithm design, opinion dynamics & dynamic games.
- Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.
- Drawback: Approach is deterministic, although some statistical simulation is done
- Challenge (to this audience): Redo using probabilistic/stochastic approaches!
This talk

- Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.
- Not (necessarily) linear quadratic (LQ).
- Examples from Business Dynamics: index which takes both the management and financial aspects into account.
- Examples from numerical algorithm design, opinion dynamics & dynamic games.
- Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.
- Drawback: Approach is deterministic, although some statistical simulation is done.
- Challenge (to this audience): Redo using probabilistic/stochastic approaches!
Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.

Not (necessarily) linear quadratic (LQ).

Examples from Business Dynamics: index which takes both the management and financial aspects into account.

Examples from numerical algorithm design, opinion dynamics & dynamic games.

Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.

Drawback: Approach is deterministic, although some statistical simulation is done

Challenge (to this audience): Redo using probabilistic/stochastic approaches!
This talk

- Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.
- Not (necessarily) linear quadratic (LQ).
- Examples from Business Dynamics: index which takes both the management and financial aspects into account.
- Examples from numerical algorithm design, opinion dynamics & dynamic games.
- Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.
- Drawback: Approach is deterministic, although some statistical simulation is done
- Challenge (to this audience): Redo using probabilistic/stochastic approaches!
This talk

- Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.
- Not (necessarily) linear quadratic (LQ).
- Examples from Business Dynamics: index which takes both the management and financial aspects into account.
- Examples from numerical algorithm design, opinion dynamics & dynamic games.
- Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.
- Drawback: Approach is deterministic, although some statistical simulation is done.
- Challenge (to this audience): Redo using probabilistic/stochastic approaches!
This talk

- Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.
- Not (necessarily) linear quadratic (LQ).
- Examples from Business Dynamics: index which takes both the management and financial aspects into account.
- Examples from numerical algorithm design, opinion dynamics & dynamic games.
- Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.
- Drawback: Approach is deterministic, although some statistical simulation is done
- Challenge (to this audience): Redo using probabilistic/stochastic approaches!
This talk

- Optimization-based one step ahead optimal approach, discrete-time, implementable in real time.
- Not (necessarily) linear quadratic (LQ).
- Examples from Business Dynamics: index which takes both the management and financial aspects into account.
- Examples from numerical algorithm design, opinion dynamics & dynamic games.
- Contributions: No prediction required, close to optimal, effect of delays, Julia+JuMP open source code.
- Drawback: Approach is deterministic, although some statistical simulation is done
- Challenge (to this audience): Redo using probabilistic/stochastic approaches!
Omniscient optimal control

- **Dynamical system:**
  \[ x(k + 1) = F(x(k), u(k), e(k), k), \quad k = 1, \ldots, K_f \]
  - \( x(k) \): state variable.
  - \( u(k) \): control input chosen by the decision maker, usually subject to some constraints.
  - \( e(k) \): exogenous input known at time \( k \), but not known for future times, and not under the control of the decision maker.

- **Omniscient optimal control problem:** Given the exogenous input sequence \( \{e(k)\}_{k=1}^{K_f} \) and initial state \( x(1) \)

\[
\begin{align*}
\text{maximize} & \quad J_{om}(K_f + 1) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u(k), e(k), k) \\
\text{by choice of} & \quad u(k), \quad k = 1, \ldots, K_f
\end{align*}
\]
Omniscient optimal control

- Dynamical system:
  \[ x(k + 1) = F(x(k), u(k), e(k), k), \quad k = 1, \ldots, K_f \]

- \( x(k) \): state variable.

- \( u(k) \): control input chosen by the decision maker, usually subject to some constraints.

- \( e(k) \): exogenous input known at time \( k \), but not known for future times, and not under the control of the decision maker.

- **Omniscient optimal control problem:** Given the exogenous input sequence \( \{ e(k) \}_{k=1}^{K_f} \) and initial state \( x(1) \)

\[
\begin{align*}
\text{maximize} & \quad J_{om}(K_f + 1) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u(k), e(k), k) \\
\text{by choice of} & \quad u(k), \quad k = 1, \ldots, K_f 
\end{align*}
\]
Omniscient optimal control

- Dynamical system:
  \[ x(k + 1) = F(x(k), u(k), e(k), k), \ k = 1, \ldots, K_f \]
- \( x(k) \): state variable.
- \( u(k) \): control input chosen by the decision maker, usually subject to some constraints.
- \( e(k) \): exogenous input known at time \( k \), but not known for future times, and not under the control of the decision maker.

**Omniscient optimal control problem:** Given the exogenous input sequence \( \{e(k)\}_{k=1}^{K_f} \) and initial state \( x(1) \)

\[
\begin{align*}
\text{maximize} & \quad J_{om}(K_f + 1) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u(k), e(k), k) \\
\text{by choice of} & \quad u(k), \ k = 1, \ldots, K_f
\end{align*}
\]
Omniscient optimal control

Dynamical system:
\[ x(k + 1) = F(x(k), u(k), e(k), k), \quad k = 1, \ldots, K_f \]

- \( x(k) \): state variable.
- \( u(k) \): control input chosen by the decision maker, usually subject to some constraints.
- \( e(k) \): exogenous input known at time \( k \), but not known for future times, and not under the control of the decision maker.

Omniscient optimal control problem: Given the exogenous input sequence \( \{ e(k) \}_{k=1}^{K_f} \) and initial state \( x(1) \)

\[
\text{maximize} \quad J_{om}(K_f + 1) \\
\text{subject to} \quad x(k + 1) = F(x(k), u(k), e(k), k) \\
\text{by choice of} \quad u(k), \quad k = 1, \ldots, K_f
\]
Omniscient optimal control

- Dynamical system:
  \[ x(k + 1) = F(x(k), u(k), e(k), k), \quad k = 1, \ldots, K_f \]
- \( x(k) \): state variable.
- \( u(k) \): control input chosen by the decision maker, usually subject to some constraints.
- \( e(k) \): exogenous input known at time \( k \), but not known for future times, and not under the control of the decision maker.

**Omniscient optimal control problem:** Given the exogenous input sequence \( \{ e(k) \}_{k=1}^{K_f} \) and initial state \( x(1) \)

\[
\begin{align*}
\text{maximize} & \quad J_{om}(K_f + 1) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u(k), e(k), k) \\
\text{by choice of} & \quad u(k), \quad k = 1, \ldots, K_f
\end{align*}
\]
Omniscient optimal control

- Dynamical system:
  \[ x(k + 1) = F(x(k), u(k), e(k), k), \quad k = 1, \ldots, K_f \]
- \( x(k) \): state variable.
- \( u(k) \): control input chosen by the decision maker, usually subject to some constraints.
- \( e(k) \): exogenous input known at time \( k \), but not known for future times, and not under the control of the decision maker.

**Omniscient optimal control problem:** Given the exogenous input sequence \( \{e(k)\}_{k=1}^{K_f} \) and initial state \( x(1) \)

\[
\begin{aligned}
\text{maximize} & \quad J_{om}(K_f + 1) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u(k), e(k), k) \\
\text{by choice of} & \quad u(k), \quad k = 1, \ldots, K_f
\end{aligned}
\]
One step ahead optimal control

**One step ahead optimal control problem:** Given the current exogenous input \( e(k) \) and the current state \( x(k) \),

\[
\begin{align*}
\text{maximize} & \quad J_{os}(x(k + 1), u, e(k)) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u, e(k), k) \\
\text{by choice of} & \quad u
\end{align*}
\]

Also called *greedy control* (Lavretsky 2000), *one step lookahead control* (Bertsekas 2005),

In economics: *one period control* or *myopic control* (Mossin, 1968)
One step ahead optimal control

**One step ahead optimal control problem:** Given the current exogenous input $e(k)$ and the current state $x(k)$

\[
\begin{align*}
\text{maximize} & \quad J_{os}(x(k+1), u, e(k)) \\
\text{subject to} & \quad x(k+1) = F(x(k), u, e(k), k) \\
\text{by choice of} & \quad u
\end{align*}
\]

- Also called *greedy control* (Lavretsky 2000), *one step lookahead control* (Bertsekas 2005),
- In economics: *one period control* or *myopic control* (Mossin, 1968)
One step ahead optimal control

**One step ahead optimal control problem:** Given the current exogenous input $e(k)$ and the current state $x(k)$

\[
\begin{align*}
\text{maximize} & \quad J_{os}(x(k + 1), u, e(k)) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u, e(k), k) \\
\text{by choice of} & \quad u
\end{align*}
\]

- Also called *greedy control* (Lavretsky 2000), *one step lookahead control* (Bertsekas 2005),
- In economics: *one period control* or *myopic control* (Mossin, 1968)
One step ahead optimal control problem: Given the current exogenous input $e(k)$ and the current state $x(k)$

\[
\begin{align*}
\text{maximize} & \quad J_{os}(x(k + 1), u, e(k)) \\
\text{subject to} & \quad x(k + 1) = F(x(k), u, e(k), k) \\
\text{by choice of} & \quad u
\end{align*}
\]

Also called greedy control (Lavretsky 2000), one step lookahead control (Bertsekas 2005),

In economics: one period control or myopic control (Mossin, 1968)
Types of performance index/objective function & constraints

- **Linear**: linear combination of (nonnegative) states and controls.
- **Piecewise linear**: weighted combination of one- or infinity-norms of states and controls.
- **Quadratic**: sum of quadratic forms in “errors” and controls.
- **Linear/nonlinear, equality/inequality**: involving states and controls.
Types of performance index/objective function & constraints

- Linear: linear combination of (nonnegative) states and controls.
- Piecewise linear: weighted combination of one- or infinity-norms of states and controls.
- Quadratic: sum of quadratic forms in “errors” and controls.
- Linear/nonlinear, equality/inequality involving states and controls.
Types of performance index/objective function & constraints

- **Linear**: linear combination of (nonnegative) states and controls.
- **Piecewise linear**: weighted combination of one- or infinity-norms of states and controls.
- **Quadratic**: sum of quadratic forms in “errors” and controls.
- **Linear/nonlinear, equality/inequality involving states and controls.**
Types of performance index/objective function & constraints

- Linear: linear combination of (nonnegative) states and controls.
- Piecewise linear: weighted combination of one- or infinity-norms of states and controls.
- Quadratic: sum of quadratic forms in “errors” and controls.
- Linear/nonlinear, equality/inequality involving states and controls.
Debt amortization: problem statement

- Amortizing given initial debts with target date (planning horizon) to pay off.
- Upper bounds on withdrawals from investment accounts to make amortization payments.
- Additional debts incurred during planning horizon.
- Deposits and dividends in the investment accounts during planning horizon.
Debt amortization: problem statement

- Amortizing given initial debts with target date (planning horizon) to pay off.
- Upper bounds on withdrawals from investment accounts to make amortization payments.
- Additional debts incurred during planning horizon.
- Deposits and dividends in the investment accounts during planning horizon.
Debt amortization: problem statement

- Amortizing given initial debts with target date (planning horizon) to pay off.
- Upper bounds on withdrawals from investment accounts to make amortization payments.
- Additional debts incurred during planning horizon.
- Deposits and dividends in the investment accounts during planning horizon.
Debt amortization: problem statement

- Amortizing given initial debts with target date (planning horizon) to pay off.
- Upper bounds on withdrawals from investment accounts to make amortization payments.
- Additional debts incurred during planning horizon.
- Deposits and dividends in the investment accounts during planning horizon.
Debt amortization: mathematical model

\[ x_i(k+1) = (1 + \alpha_i) x_i(k) - \sum_{j=1}^{n} u_{ij}(k) + c_i(k), \forall i \]

\[ y_j(k+1) = (1 + \beta_j) y_j(k) - \sum_{i=1}^{m} u_{ij}(k) - \gamma_j \sum_{i=1}^{m} u_{ij}(k) + d_j(k), \forall j \]

- \( x_i \): \( i \)th debt, \( y_j \): \( j \) th investment account balance.
- \( \alpha_i, \beta_j \): \( i \)th debt and \( j \) th investment account interest rates, respectively.
- \( \gamma_j \): transaction cost per unit of cash transferred from investment account.
- Box constraints on \( x_i, y_j \), upper bounds on \( u_i(k) := \sum_{j=1}^{n} u_{ij}(k) \) for each \( i \).
Debt amortization: mathematical model

\[ x_i(k + 1) = (1 + \alpha_i)x_i(k) - \sum_{j=1}^{n} u_{ij}(k) + c_i(k), \forall i \]

\[ y_j(k + 1) = (1 + \beta_j)y_j(k) - \sum_{i=1}^{m} u_{ij}(k) - \gamma_j \sum_{i=1}^{m} u_{ij}(k) + d_j(k), \forall j \]

- \( x_i \): \( i \)th debt, \( y_j \): \( j \) th investment account balance.
- \( \alpha_i, \beta_j \): \( i \)th debt and \( j \) th investment account interest rates, respectively.
- \( \gamma_j \): transaction cost per unit of cash transferred from investment account.
- Box constraints on \( x_i, y_j \), upper bounds on \( u_i(k) := \sum_{j=1}^{n} u_{ij}(k) \) for each \( i \).
Debt amortization: mathematical model

\[
x_i(k+1) = (1 + \alpha_i)x_i(k) - \sum_{j=1}^{n} u_{ij}(k) + c_i(k), \forall i
\]

\[
y_j(k+1) = (1 + \beta_j)y_j(k) - \sum_{i=1}^{m} u_{ij}(k) - \gamma_j \sum_{i=1}^{m} u_{ij}(k) + d_j(k), \forall j
\]

- \(x_i\): \(i\)th debt, \(y_j\): \(j\) th investment account balance.
- \(\alpha_i, \beta_j\): \(i\)th debt and \(j\) th investment account interest rates, respectively.
- \(\gamma_j\): transaction cost per unit of cash transferred from investment account.
- Box constraints on \(x_i, y_j\), upper bounds on \(u_i(k) := \sum_{j=1}^{n} u_{ij}(k)\) for each \(i\).
Debt amortization: mathematical model

\[ x_i(k + 1) = (1 + \alpha_i)x_i(k) - \sum_{j=1}^{n} u_{ij}(k) + c_i(k), \forall i \]

\[ y_j(k + 1) = (1 + \beta_j)y_j(k) - \sum_{i=1}^{m} u_{ij}(k) - \gamma_j \sum_{i=1}^{m} u_{ij}(k) + d_j(k), \forall j \]

- \( x_i \): \( i \)th debt, \( y_j \): \( j \)th investment account balance.
- \( \alpha_i, \beta_j \): \( i \)th debt and \( j \)th investment account interest rates, respectively.
- \( \gamma_j \): transaction cost per unit of cash transferred from investment account.
- Box constraints on \( x_i, y_j \), upper bounds on \( u_i(k) := \sum_{j=1}^{n} u_{ij}(k) \) for each \( i \).
Debt amortization: mathematical model

\[ x_i(k + 1) = (1 + \alpha_i)x_i(k) - \sum_{j=1}^{n} u_{ij}(k) + c_i(k), \forall i \]

\[ y_j(k + 1) = (1 + \beta_j)y_j(k) - \sum_{i=1}^{m} u_{ij}(k) - \gamma_j \sum_{i=1}^{m} u_{ij}(k) + d_j(k), \forall j \]

- \( x_i \): 1st debt, \( y_j \): jth investment account balance.
- \( \alpha_i, \beta_j \): 1st debt and jth investment account interest rates, respectively.
- \( \gamma_j \): transaction cost per unit of cash transferred from investment account.
- Box constraints on \( x_i, y_j \), upper bounds on \( u_i(k) := \sum_{j=1}^{n} u_{ij}(k) \) for each \( i \).
Weighted debt amortization with wealth maximization

\[ J_{os}(k+1) = \sum_{j=1}^{n} \kappa_j y_j(k+1) - \sum_{i=1}^{m} \mu_i x_i(k+1) \]

\[ J_{om} = \sum_{j=1}^{n} \kappa_j y_j(K_f + 1) - \sum_{i=1}^{m} \mu_i x_i(K_f + 1) \]
Debt amortization: Performance indices

- Weighted debt amortization with wealth maximization

\[
J_{os}(k + 1) = \sum_{j=1}^{n} \kappa_j y_j(k + 1) - \sum_{i=1}^{m} \mu_i x_i(k + 1)
\]

\[
J_{om} = \sum_{j=1}^{n} \kappa_j y_j(K_f + 1) - \sum_{i=1}^{m} \mu_i x_i(K_f + 1)
\]
Weighted debt amortization with wealth maximization

- $J_{os}(k + 1) = \sum_{j=1}^{n} \kappa_j y_j(k + 1) - \sum_{i=1}^{m} \mu_i x_i(k + 1)$
- $J_{om} = \sum_{j=1}^{n} \kappa_j y_j(K_f + 1) - \sum_{i=1}^{m} \mu_i x_i(K_f + 1)$
Debt amortization + wealth maximization: LP formulation

- maximize $J_{om}$ (resp. $J_{os}$) [linear objective function]
- subj. to dynamics for all $k$ (resp. at $k$) [linear equality constraints]
- and subj. to box and upper bound constraints for all $k$ (resp. at $k$) [linear inequality constraints].
Debt amortization + wealth maximization: LP formulation

- maximize $J_{om}$ (resp. $J_{os}$) [linear objective function]
- subj. to dynamics for all $k$ (resp. at $k$) [linear equality constraints]
- and subj. to box and upper bound constraints for all $k$ (resp. at $k$) [linear inequality constraints].
Debt amortization + wealth maximization: LP formulation

- maximize $J_{om}$ (resp. $J_{os}$) [linear objective function]
- subj. to dynamics for all $k$ (resp. at $k$) [linear equality constraints]
- and subj. to box and upper bound constraints for all $k$ (resp. at $k$) [linear inequality constraints].
Debt amortization + wealth maximization: example

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expenditures $c_1$</td>
<td>565</td>
<td>311</td>
<td>Random</td>
</tr>
<tr>
<td>Expenditures $c_2$</td>
<td>501</td>
<td>288</td>
<td>Random</td>
</tr>
<tr>
<td>Deposits $d_1$</td>
<td>345</td>
<td>147</td>
<td>Random</td>
</tr>
<tr>
<td>Deposits $d_2$</td>
<td>320</td>
<td>231</td>
<td>Random</td>
</tr>
</tbody>
</table>
Example: evolution of debt sequences

Debt sequences 1 and 2 for OSA and omniscient optimal controls are similar, but not identical.
Example: amortization payments

\( \alpha_1 = 0.05, \alpha_2 = 0.07, \beta_1 = 0.08, \beta_2 = 0.11, \gamma_1 = 0.02, \gamma_2 = 0.01, \kappa_1 = 0, \kappa_2 = 0 \)

\( \mu_1 = 1, \mu_2 = 1, \mu_1, \text{max} = 2500, \mu_2, \text{max} = 2500, \gamma_1, \text{min} = 1000, 0, \gamma_2, \text{min} = 1000.0 \)

\[ C \left( P_{\text{in}} \right) \]

(a) One step ahead amortization payments for debt 1, maximizing \( J_{\text{os}}^{md, \omega} \).

(b) Omniscient amortization payments for debt 1, maximizing \( J_{\text{om}, f}^{md, \omega} \).

Total amortization payment \( u_1 = u_{11} + u_{12} \) tracks expenditure \( c_1 \), after initial period at \( u_{1, \text{max}} \).
Example: investment account balances

OSA: Investment account 1 $\rightarrow$ minimum. Omniscient: Investment account 2 $\rightarrow$ minimum.
Debt amortization: choice of weights

<table>
<thead>
<tr>
<th>Row</th>
<th>Index</th>
<th>$\kappa_1$</th>
<th>$\kappa_2$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>Final debt</th>
<th>Final wealth</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>93011</td>
<td>$x_1, x_2 \downarrow 0, y_2 \downarrow y_2, \text{min}$</td>
</tr>
<tr>
<td>2</td>
<td>$J_{os}^{md,\omega}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>215305</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>3</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22703</td>
<td>322604</td>
<td>$y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>4</td>
<td>$J_{os}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20000</td>
<td>160026</td>
<td>$y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>5</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>319202</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>6</td>
<td>$J_{os}^{md,\omega}$</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>194998</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>7</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>319206</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>8</td>
<td>$J_{os}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>92731</td>
<td>$x_1, x_2 \downarrow 0, y_2 \downarrow y_2, \text{min}$</td>
</tr>
</tbody>
</table>

- **Similar outcomes**: Row 2 OS (implementable)/Row 7 Omni (unimplementable) = 67%.
- **Overall conclusion**: choice of weights $\kappa_i = 0, \mu_j = 1, \forall i, j$ (debt amortization) is good candidate for real time debt management.
Debt amortization: choice of weights

<table>
<thead>
<tr>
<th>Row</th>
<th>Index</th>
<th>$\kappa_1$</th>
<th>$\kappa_2$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>Final debt</th>
<th>Final wealth</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>93011</td>
<td>$x_1, x_2 \downarrow 0, y_2 \downarrow y_2, \text{min}$</td>
</tr>
<tr>
<td>2</td>
<td>$J_{os}^{md,\omega}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$215305$</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>3</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22703</td>
<td>322604</td>
<td>$y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>4</td>
<td>$J_{os}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20000</td>
<td>160026</td>
<td>$y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>5</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>319202</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>6</td>
<td>$J_{os}^{md,\omega}$</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>194998</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>7</td>
<td>$J_{om,f}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>$319206$</td>
<td>$x_1, x_2 \downarrow 0, y_1 \downarrow y_1, \text{min}$</td>
</tr>
<tr>
<td>8</td>
<td>$J_{os}^{md,\omega}$</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>92731</td>
<td>$x_1, x_2 \downarrow 0, y_2 \downarrow y_2, \text{min}$</td>
</tr>
</tbody>
</table>

- Similar outcomes: Row 2 OS (implementable)/Row 7 Omni (unimplementable) = 67%.
- **Overall conclusion:** choice of weights $\kappa_i = 0, \mu_j = 1, \forall i, j$ (debt amortization) is good candidate for real time debt management.
Every firm/financial institution has cash current account to meet daily transaction requirements.

Cash balance: large enough for requirements, not much larger.

Extra cash earns larger returns in an investment account.

**Cash flows**, both deposits and withdrawals, into *current accounts with low interest rates*.

**Transfers from investment accounts** to make negative current account balances positive.

Transfer excess balance in current accounts to investment accounts (better returns).

All transfers are subject to transaction costs, always deducted from current accounts.

**Determining the best way to make these transfers**, with cash flow sequences unknown, is the **cash balance problem**.
Cash balance problem: description

- Every firm/financial institution has cash current account to meet daily transaction requirements.
- Cash balance: large enough for requirements, not much larger.
- Extra cash earns larger returns in an investment account.
- **Cash flows**, both deposits and withdrawals, into current accounts with low interest rates.
- **Transfers from investment accounts** to make negative current account balances positive.
- Transfer excess balance in current accounts to investment accounts (better returns).
- All transfers are subject to transaction costs, always deducted from current accounts.
- Determining the best way to make these transfers, with cash flow sequences unknown, is the **cash balance problem**.
Cash balance problem: description

- Every firm/financial institution has cash current account to meet daily transaction requirements.
- Cash balance: large enough for requirements, not much larger.
- Extra cash earns larger returns in an investment account.
- **Cash flows**, both deposits and withdrawals, into current accounts with low interest rates.
- **Transfers from investment accounts** to make negative current account balances positive.
- Transfer excess balance in current accounts to investment accounts (better returns).
- All transfers are subject to transaction costs, always deducted from current accounts.
- Determining the best way to make these transfers, with cash flow sequences unknown, is the **cash balance problem**.
Cash balance problem: description

- Every firm/financial institution has cash current account to meet daily transaction requirements.
- Cash balance: large enough for requirements, not much larger.
- Extra cash earns larger returns in an investment account.
- **Cash flows**, both deposits and withdrawals, into *current accounts with low interest rates*.
- Transfers from investment accounts to make negative current account balances positive.
- Transfer excess balance in current accounts to investment accounts (better returns).
- All transfers are subject to transaction costs, always deducted from current accounts.
- Determining the best way to make these transfers, with cash flow sequences unknown, is the **cash balance problem**.
Cash balance problem: description

- Every firm/financial institution has cash current account to meet daily transaction requirements.
- Cash balance: large enough for requirements, not much larger.
- Extra cash earns larger returns in an investment account.
- **Cash flows**, both deposits and withdrawals, into current accounts with low interest rates.
- **Transfers from investment accounts** to make negative current account balances positive.
- Transfer excess balance in current accounts to investment accounts (better returns).
- All transfers are subject to transaction costs, always deducted from current accounts.
- Determining the best way to make these transfers, with cash flow sequences unknown, is the cash balance problem.
Cash balance problem: description

- Every firm/financial institution has cash current account to meet daily transaction requirements.
- Cash balance: large enough for requirements, not much larger.
- Extra cash earns larger returns in an investment account.
- **Cash flows**, both deposits and withdrawals, into *current accounts with low interest rates*.
- **Transfers from investment accounts** to make negative current account balances positive.
- Transfer excess balance in current accounts to investment accounts (better returns).
- All transfers are subject to transaction costs, always deducted from current accounts.
- Determining the best way to make these transfers, with cash flow sequences unknown, is the **cash balance problem**.
Every firm/financial institution has cash current account to meet daily transaction requirements.

Cash balance: large enough for requirements, not much larger.

Extra cash earns larger returns in an investment account.

Cash flows, both deposits and withdrawals, into current accounts with low interest rates.

Transfers from investment accounts to make negative current account balances positive.

Transfer excess balance in current accounts to investment accounts (better returns).

All transfers are subject to transaction costs, always deducted from current accounts.

Determining the best way to make these transfers, with cash flow sequences unknown, is the cash balance problem.
Cash balance problem: description

- Every firm/financial institution has cash current account to meet daily transaction requirements.
- Cash balance: large enough for requirements, not much larger.
- Extra cash earns larger returns in an investment account.
- **Cash flows**, both deposits and withdrawals, into current accounts with low interest rates.
- **Transfers from investment accounts** to make negative current account balances positive.
- Transfer excess balance in current accounts to investment accounts (better returns).
- All transfers are subject to transaction costs, always deducted from current accounts.
- **Determining the best way to make these transfers**, with cash flow sequences unknown, is the cash balance problem.
Cash balance: notation
Cash balance dynamics

\[ x_i^+ = (1 + \alpha_i) x_i + \sum_{j=1}^{M} u_{ij} - \sum_{j=1}^{M} \gamma_i |u_{ij}| - d_i + \sum_{m \neq i} c_{im}, \forall i, \]

\[ y_j^+ = (1 + \beta_j) y_j - \sum_{i=1}^{N} u_{ij}, \forall j, \]

\[ 0 = c_{im}(k) + c_{mi}(k), \forall i, \forall m, \]

\[ 0 < x_{i,\text{min}} \leq x_i(k) \leq x_{i,\text{max}}, \forall i, \]

\[ 0 < y_{j,\text{min}} \leq y_j(k), \forall j. \]

Total wealth: \[ w(k) = \sum_{i=1}^{N} x_i(k) + \sum_{j=1}^{M} y_j(k) \]
Cash balance problem: performance indices

- \( J_{os}(k+1) = \sum_j \kappa_j y_j(k+1) - \sum_i \mu_i x_i(k+1) \).
- \( J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1) \).
- \( \kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow \text{maximization of total wealth.} \)
- \( \kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow \text{maximizing investment account balance, while minimizing current account balances.} \)
- \( \kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow \text{minimizing sum of current account balances.} \)

Wealth gain = \( \frac{w(K_f) - w(1)}{w(1)} \), where \( w(k) = \sum_i x_i(k) + \sum_j y_j(k) \), for a single demand.

- Performance metric := Average wealth gain over a set of demands.
Cash balance problem: performance indices

- $J_{os}(k + 1) = \sum_j \kappa_j y_j(k + 1) - \sum_i \mu_i x_i(k + 1)$.
- $J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1)$.

- $\kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow$ maximization of total wealth.
- $\kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow$ maximizing investment account balance, while minimizing current account balances.
- $\kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow$ minimizing sum of current account balances.
- Wealth gain $= \frac{w(K_f) - w(1)}{w(1)}$, where $w(k) = \sum_i x_i(k) + \sum_j y_j(k)$, for a single demand.
- Performance metric := Average wealth gain over a set of demands.
Cash balance problem: performance indices

- $J_{os}(k + 1) = \sum_j \kappa_j y_j(k + 1) - \sum_i \mu_i x_i(k + 1)$.
- $J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1)$.
- $\kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow$ maximization of total wealth.
- $\kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow$ maximizing investment account balance, while minimizing current account balances.
- $\kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow$ minimizing sum of current account balances.
- Wealth gain $= \frac{w(K_f) - w(1)}{w(1)}$, where $w(k) = \sum_i x_i(k) + \sum_j y_j(k)$, for a single demand.
- Performance metric := Average wealth gain over a set of demands.
Cash balance problem: performance indices

- \( J_{os}(k+1) = \sum_j \kappa_j y_j(k+1) - \sum_i \mu_i x_i(k+1). \)
- \( J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1). \)
- \( \kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow \text{maximization of total wealth.} \)
- \( \kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow \text{maximizing investment account balance, while minimizing current account balances.} \)
- \( \kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow \text{minimizing sum of current account balances.} \)
- Wealth gain = \( \frac{w(K_f) - w(1)}{w(1)} \), where \( w(k) = \sum_i x_i(k) + \sum_j y_j(k) \), for a single demand.
- Performance metric := Average wealth gain over a set of demands.
Cash balance problem: performance indices

- $J_{os}(k + 1) = \sum_j \kappa_j y_j(k + 1) - \sum_i \mu_i x_i(k + 1)$.
- $J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1)$.
- $\kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow$ maximization of total wealth.
- $\kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow$ maximizing investment account balance, while minimizing current account balances.
- $\kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow$ minimizing sum of current account balances.
- Wealth gain $= \frac{w(K_f) - w(1)}{w(1)}$, where $w(k) = \sum_i x_i(k) + \sum_j y_j(k)$, for a single demand.
- Performance metric := Average wealth gain over a set of demands.
Cash balance problem: performance indices

- $J_{os}(k + 1) = \sum_j \kappa_j y_j(k + 1) - \sum_i \mu_i x_i(k + 1)$.
- $J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1)$.
- $\kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow$ maximization of total wealth.
- $\kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow$ maximizing investment account balance, while minimizing current account balances.
- $\kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow$ minimizing sum of current account balances.
- Wealth gain $= \frac{w(K_f) - w(1)}{w(1)}$, where $w(k) = \sum_i x_i(k) + \sum_j y_j(k)$, for a single demand.
- Performance metric := Average wealth gain over a set of demands.
Cash balance problem: performance indices

- $J_{os}(k+1) = \sum_j \kappa_j y_j(k+1) - \sum_i \mu_i x_i(k+1)$.
- $J_{om} = \sum_j \kappa_j y_j(K_f + 1) - \sum_i \mu_i x_i(K_f + 1)$.
- $\kappa_j = 1, \forall j, \mu_i = -1, \forall i \Rightarrow$ maximization of total wealth.
- $\kappa_j = 1, \forall j, \mu_i = 1, \forall i \Rightarrow$ maximizing investment account balance, while minimizing current account balances.
- $\kappa_j = 0, \forall j, \mu_i = -1, \forall i \Rightarrow$ minimizing sum of current account balances.
- Wealth gain $= \frac{w(K_f) - w(1)}{w(1)}$, where $w(k) = \sum_i x_i(k) + \sum_j y_j(k)$, for a single demand.
- Performance metric := Average wealth gain over a set of demands.
In practice, demand estimates $\hat{d}_i(k) = d_i(k - 1)$ work well.

If transfers are end-of-the-day operations, $\hat{d}_i(k) = d_i(k)$. 
In practice, demand estimates $\hat{d}_i(k) = d_i(k - 1)$ work well.
If transfers are end-of-the-day operations, $\hat{d}_i(k) = d_i(k)$. 

Feedback control with dead zone, no explicit use of demand.

\[ u_i(k) = \begin{cases} 
  z_i - x_i(k) & \text{if } x_i(k) \geq z_{iH}, \\
  0 & \text{if } z_{iL} \leq x_i(k) \leq z_{iH}, \\
  z_i - x_i(k) & \text{if } x_i(k) \leq z_{iL}. 
\end{cases} \]

Miller & Orr derived optimal thresholds for Bernoulli (equiprobable \( \pm 1 \)) demands.

Most real demands are not Bernoulli.

Thresholds need to be tuned: optimal control problem? ML approach!

OSA version would imply time-varying thresholds. Not practical!
Cash balance: classical Miller-Orr scheme (1966)

- Feedback control with dead zone, no explicit use of demand.
  \[
  u_i(k) = \begin{cases} 
  z_i - x_i(k) & \text{if } x_i(k) \geq z_iH, \\
  0 & \text{if } z_iL \leq x_i(k) \leq z_iH, \\
  z_i - x_i(k) & \text{if } x_i(k) \leq z_iL. 
  \end{cases}
  \]

- Miller & Orr derived optimal thresholds for Bernoulli (equiprobable \( \pm 1 \)) demands.
- Most real demands are not Bernoulli.
- Thresholds need to be tuned: optimal control problem? ML approach!
- OSA version would imply time-varying thresholds. Not practical!
Feedback control with dead zone, no explicit use of demand.

\[ u_i(k) = \begin{cases} 
    z_i - x_i(k) & \text{if } x_i(k) \geq z_{iH}, \\
    0 & \text{if } z_{iL} \leq x_i(k) \leq z_{iH}, \\
    z_i - x_i(k) & \text{if } x_i(k) \leq z_{iL}. 
\end{cases} \]

Miller & Orr derived optimal thresholds for Bernoulli (equiprobable ±1) demands.

Most real demands are not Bernoulli.

Thresholds need to be tuned: optimal control problem? ML approach!

OSA version would imply time-varying thresholds. Not practical!
Cash balance: classical Miller-Orr scheme (1966)

- Feedback control with **dead zone**, no explicit use of demand.
  \[
  u_i(k) = \begin{cases} 
  z_i - x_i(k) & \text{if } x_i(k) \geq z_{iH}, \\
  0 & \text{if } z_{iL} \leq x_i(k) \leq z_{iH}, \\
  z_i - x_i(k) & \text{if } x_i(k) \leq z_{iL}.
  \end{cases}
  \]

- Miller & Orr derived optimal **thresholds** for Bernoulli (equiprobable $\pm 1$) demands.

- Most real demands are not Bernoulli.

- Thresholds need to be tuned: optimal control problem? ML approach!

- OSA version would imply time-varying thresholds. Not practical!
Cash balance: classical Miller-Orr scheme (1966)

- Feedback control with **dead zone**, no explicit use of demand.
  
  \[ u_i(k) = \begin{cases} 
  z_i - x_i(k) & \text{if } x_i(k) \geq z_{iH}, \\
  0 & \text{if } z_{iL} \leq x_i(k) \leq z_{iH}, \\
  z_i - x_i(k) & \text{if } x_i(k) \leq z_{iL}. 
  \end{cases} \]

- Miller & Orr derived optimal **thresholds** for Bernoulli (equiprobable ±1) demands.

- Most real demands are not Bernoulli.

- Thresholds need to be tuned: optimal control problem? ML approach!

- OSA version would imply time-varying thresholds. Not practical!
Cash balance: classical Miller-Orr scheme (1966)

- Feedback control with dead zone, no explicit use of demand.
  \[ u_i(k) = \begin{cases} 
  z_i - x_i(k) & \text{if } x_i(k) \geq z_iH, \\
  0 & \text{if } z_iL \leq x_i(k) \leq z_iH, \\
  z_i - x_i(k) & \text{if } x_i(k) \leq z_iL. 
\end{cases} \]

- Miller & Orr derived optimal thresholds for Bernoulli (equiprobable ±1) demands.

- Most real demands are not Bernoulli.

- Thresholds need to be tuned: optimal control problem? ML approach!

- OSA version would imply time-varying thresholds. Not practical!
Cash balance: average performance of OSAOC

- Comparison of average and relative wealth gain (AWG,RWG).
- Current demand is assumed unknown: OSAOC uses
  \[ \hat{d}_i(k) = d_i(k - 1) \]
- Four methods OS_NI/WI, OM_NI/WI compared.
- Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).
- 20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.
- Transfers between all pairs of CA, IA allowed, intra-account only for CA.
- OSA current account balance \( \rightarrow \min \), Omni final wealth \( \rightarrow \max \)
- Mean and std. dev. of AWG & RWG
  \( (=AWG_{OS}/AWG_{OM}) \)
Comparison of average and relative wealth gain (AWG, RWG).

Current demand is assumed unknown: OSAOC uses

\[ \hat{d}_i(k) = d_i(k - 1) \]

Four methods OS_NI/WI, OM_NI/WI compared.

Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).

20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.

Transfers between all pairs of CA, IA allowed, intra-account only for CA.

OSA current account balance \( \rightarrow \) min, Omni final wealth \( \rightarrow \) max

Mean and std. dev. of AWG & RWG 

\( (=AWG_{OS}/AWG_{OM}) \)
Cash balance: average performance of OSAOC

- Comparison of average and relative wealth gain (AWG, RWG).
- Current demand is assumed unknown: OSAOC uses
  \[ \hat{d}_i(k) = d_i(k - 1) \]
- Four methods OS_NI/WI, OM_NI/WI compared.
  - Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).
  - 20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.
  - Transfers between all pairs of CA, IA allowed, intra-account only for CA.
  - OSA current account balance \( \rightarrow \) min, Omni final wealth \( \rightarrow \) max
- Mean and std. dev. of AWG & RWG
  \((= \text{AWG}_{OS}/\text{AWG}_{OM})\)
Cash balance: average performance of OSAOC

- Comparison of average and relative wealth gain (AWG,RWG).
- Current demand is assumed unknown: OSAOC uses \( \hat{d}_i(k) = d_i(k - 1) \)
- Four methods OS_NI/WI, OM_NI/WI compared.
- Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).
  - 20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.
  - Transfers between all pairs of CA, IA allowed, intra-account only for CA.
  - OSA current account balance \( \rightarrow \) min, Omni final wealth \( \rightarrow \) max
- Mean and std. dev. of AWG & RWG (=AWG_OS/AWG_OM)
Cash balance: average performance of OSAOC

- Comparison of average and relative wealth gain (AWG,RWG).
- Current demand is assumed unknown: OSAOC uses \( \hat{d}_i(k) = d_i(k - 1) \)
- Four methods OS_NI/WI, OM_NI/WI compared.
- Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).
- 20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.
- Transfers between all pairs of CA, IA allowed, intra-account only for CA.
- OSA current account balance \( \rightarrow \) min, Omni final wealth \( \rightarrow \) max
- Mean and std. dev. of AWG & RWG (=AWG_OS/AWG_OM)
Cash balance: average performance of OSAOC

- Comparison of average and relative wealth gain (AWG, RWG).
- Current demand is assumed unknown: OSAOC uses \( \hat{d}_i(k) = d_i(k-1) \)
- Four methods OS_NI/WI, OM_NI/WI compared.
- Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).
- 20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.
- Transfers between all pairs of CA, IA allowed, intra-account only for CA.
- OSA current account balance \( \rightarrow \text{min} \), Omni final wealth \( \rightarrow \text{max} \)
- Mean and std. dev. of AWG & RWG \( (=\text{AWG}_{OS}/\text{AWG}_{OM}) \)
Comparison of average and relative wealth gain (AWG,RWG).

Current demand is assumed unknown: OSAOC uses
\[ \hat{d}_i(k) = d_i(k-1) \]

Four methods OS_NI/WI, OM_NI/WI compared.

Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).

20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.

Transfers between all pairs of CA, IA allowed, intra-account only for CA.

OSA current account balance \( \rightarrow \) min, Omni final wealth \( \rightarrow \) max

Mean and std. dev. of AWG & RWG
\( (=\text{AWG}_\text{OS}/\text{AWG}_\text{OM}) \)
Comparison of average and relative wealth gain (AWG, RWG).

Current demand is assumed unknown: OSAOC uses
\[ \hat{d}_i(k) = d_i(k - 1) \]

Four methods OS_NI/WI, OM_NI/WI compared.

Experiment design: 25 pairs \( N_{CA}, N_{IA} \in \{2, 4, 6, 8, 10\} \).

20 normally distributed 30 day cash demands, zero mean, unit std. dev. for each pair.

Transfers between all pairs of CA, IA allowed, intra-account only for CA.

OSA current account balance \( \rightarrow \) min, Omni final wealth \( \rightarrow \) max

Mean and std. dev. of AWG & RWG
\( (=\text{AWG}\_\text{OS}/\text{AWG}\_\text{OM}) \)
Cash balance: average performance of OSAOC

(a) Average wealth gain

(b) Relative average wealth gain

(a) Average wealth gain

(b) Relative average wealth gain
Hierarchy \( \text{OSA\_NI} \leq \text{OSA\_WI} \leq \text{OM\_NI} \leq \text{OM\_WI} \)

Variance of AWG decreases as \( N_{CA}, N_{IA} \uparrow \) in both cases (current demand known/unknown).

\( \text{AWG} \approx 40\%, \text{RWG} \approx 70\% \) (normal demand), \( \text{AWG} \approx 30\%, \text{RWG} \approx 60\% \) (uniform, Bernoulli)

Compute time, memory requirements: scale linearly (OSAOC), scale exponentially (Omni)
Inventory control and management

\[ s_e(k) = \begin{cases} 
  d(k), & \text{if } d(k) \leq s_t(k) - w + x_1(k) \\
  s_t(k) - w + x_1(k), & \text{otherwise}
\end{cases} \]

A

\[
\begin{align*}
\text{OSA Optimizer} & \quad o(k) \quad \text{Shipping pipeline} \quad (= \text{delay of } D \text{ units}) \quad x_1(k) \\
 & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
& \quad s_t(k) \quad s_h(k) \quad \text{Shipping} \quad s_t(k) \\
& \quad \hat{d}(k) \quad \text{Estimated Demand} \quad d(k) \quad \text{Dem} \\
& \quad o(k) \quad o(k-1) \quad o(k-2) \quad \ldots \quad o(k-D+1) \quad o(k-D) \\
\end{align*}
\]

Safety stock \( w \)

B

\[
\begin{align*}
\text{Shipping pipeline} \\
& x_D(k) \quad x_{D-1}(k) \quad \ldots \quad x_2(k) \quad x_1(k)
\end{align*}
\]
Inventory control and management

Diagram:

- Demand estimates $\hat{d}_i(k)$
- Upper bound on delays
- OSA optimizer at time $k$ (contains copy of SESC dynamics, constraints and uses fixed upper bound on delays)
- OSA optimal orders $o^*(k)$
- Actual delays
- Actual demands $d_i(k)$
- SESC dynamics
- State $z(k)$
- $z(k+1)$
Inventory control: robustness to demand uncertainty

Efficiency Surfaces

APIOBPCS
OSAO
(s,S)

Mean ($\mu$)

Std Deviation ($\sigma$)

Efficiency ($n_x$)
Inventory control: robustness to lead time uncertainty

<table>
<thead>
<tr>
<th>Controller</th>
<th>Variable delay efficiency</th>
<th>Constant delay efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIOBPCS</td>
<td>63.65 (± 8.38)%</td>
<td>83.44%</td>
</tr>
<tr>
<td>OSAO</td>
<td>70.32 (± 3.68)%</td>
<td>85.69%</td>
</tr>
<tr>
<td>(s, S)</td>
<td>48.44 (± 8.96)%</td>
<td>66.06%</td>
</tr>
</tbody>
</table>
OSAO inventory control: summary

- Novel scheme: no prediction, robust to demand, delay uncertainties.
- Uses Economic Value Added as objective function.
- Efficient, attaining 80% of (unattainable) omniscient global optimum.
- Computationally efficient: LP in small number of decision variables.
- Candidate for cheap and universal inventory control scheme.
OSAO inventory control: summary

- Novel scheme: no prediction, robust to demand, delay uncertainties.
- Uses Economic Value Added as objective function.
- Efficient, attaining 80% of (unattainable) omniscient global optimum.
- Computationally efficient: LP in small number of decision variables.
- Candidate for cheap and universal inventory control scheme.
OSAO inventory control: summary

- Novel scheme: no prediction, robust to demand, delay uncertainties.
- Uses Economic Value Added as objective function.
- Efficient, attaining 80% of (unattainable) omniscient global optimum.
- Computationally efficient: LP in small number of decision variables
- Candidate for cheap and universal inventory control scheme.
OSAO inventory control: summary

- Novel scheme: no prediction, robust to demand, delay uncertainties.
- Uses Economic Value Added as objective function.
- Efficient, attaining 80% of (unattainable) omniscient global optimum.
- Computationally efficient: LP in small number of decision variables
- Candidate for cheap and universal inventory control scheme.
OSAO inventory control: summary

- Novel scheme: no prediction, robust to demand, delay uncertainties.
- Uses Economic Value Added as objective function.
- Efficient, attaining 80% of (unattainable) omniscient global optimum.
- Computationally efficient: LP in small number of decision variables
- Candidate for cheap and universal inventory control scheme.
Opinion dynamics


- Key ideas: vector \( x(k) \in \mathbb{R}^n \) of \( n \) individual agent opinions; Network or Graph: nodes = agents, edges = connect agents
- Update at time \( k \): each agent updates its opinion by taking weighted average of connected neighbors’ opinions.
- de Groot used a row stochastic matrix (one-step transition probabilities of a Markov chain) & limit theorems.
- Huge boom in this area in the last decade due to interest in social networks.
- Many alternative models (sociologists, physicists, control theorists, ...): Friedkin-Johnson, Hegselmann-Krause, Galam, Altafini
Opinion dynamics

- Key ideas: vector $x(k) \in \mathbb{R}^n$ of $n$ individual agent opinions;
  Network or Graph: nodes = agents, edges = connect agents
- Update at time $k$: each agent updates its opinion by taking weighted average of connected neighbors’ opinions.
- de Groot used a row stochastic matrix (one-step transition probabilities of a Markov chain) & limit theorems.
- Huge boom in this area in the last decade due to interest in social networks.
- Many alternative models (sociologists, physicists, control theorists, ...): Friedkin-Johnson, Hegselmann-Krause, Galam, Altafini
Opinion dynamics

- Key ideas: vector $x(k) \in \mathbb{R}^n$ of $n$ individual agent opinions; Network or Graph: nodes = agents, edges = connect agents
- Update at time $k$: each agent updates its opinion by taking weighted average of connected neighbors’ opinions.
- de Groot used a row stochastic matrix (one-step transition probabilities of a Markov chain) & limit theorems.
- Huge boom in this area in the last decade due to interest in social networks.
- Many alternative models (sociologists, physicists, control theorists,...): Friedkin-Johnson, Hegselmann-Krause, Galam, Altafini
Opinion dynamics


- Key ideas: vector $x(k) \in \mathbb{R}^n$ of $n$ individual agent opinions; Network or Graph: nodes = agents, edges = connect agents

- Update at time $k$: each agent updates its opinion by taking weighted average of connected neighbors’ opinions.

- de Groot used a row stochastic matrix (one-step transition probabilities of a Markov chain) & limit theorems.

- Huge boom in this area in the last decade due to interest in social networks.

- Many alternative models (sociologists, physicists, control theorists,...): Friedkin-Johnson, Hegselmann-Krause, Galam, Altafini
Opinion dynamics

- Key ideas: vector $x(k) \in R^n$ of $n$ individual agent opinions;
  Network or Graph: nodes = agents, edges = connect agents
- Update at time $k$: each agent updates its opinion by taking weighted average of connected neighbors’ opinions.
- de Groot used a row stochastic matrix (one-step transition probabilities of a Markov chain) & limit theorems.
- Huge boom in this area in the last decade due to interest in social networks.
- Many alternative models (sociologists, physicists, control theorists,...): Friedkin-Johnson, Hegselmann-Krause, Galam, Altafini
Opinion dynamics

- Key ideas: vector $x(k) \in R^n$ of $n$ individual agent opinions; Network or Graph: nodes = agents, edges = connect agents
- Update at time $k$: each agent updates its opinion by taking weighted average of connected neighbors’ opinions.
- de Groot used a row stochastic matrix (one-step transition probabilities of a Markov chain) & limit theorems.
- Huge boom in this area in the last decade due to interest in social networks.
- Many alternative models (sociologists, physicists, control theorists,...): Friedkin-Johnson, Hegselmann-Krause, Galam, Altafini
Opinion dynamics with control

- Recent interest: adding control to an opinion dynamics model.
  - Additive (affine) control term affects opinions after averaging process (Veetaseveera 2021, Mazalov 2023)
  - Additive (affine) control term affects opinions before averaging process (Barabanov 2010, Bullo 2023)
- Agent opinion is affected by an entity called player.
- Players wish to drive agent opinions to target values
- Targets could be the same (cooperative players) or different (competitive players)
Opinion dynamics with control

- Recent interest: adding control to an opinion dynamics model.
- Additive (affine) control term affects opinions after averaging process (Veetaseveera 2021, Mazalov 2023)
- Additive (affine) control term affects opinions before averaging process (Barabanov 2010, Bullo 2023)
- Agent opinion is affected by an entity called player.
- Players wish to drive agent opinions to target values
- Targets could be the same (cooperative players) or different (competitive players)
Opinion dynamics with control

- Recent interest: adding control to an opinion dynamics model.
- Additive (affine) control term affects opinions after averaging process (Veetaseveera 2021, Mazalov 2023)
- Additive (affine) control term affects opinions before averaging process (Barabanov 2010, Bullo 2023)

Agent opinion is affected by an entity called player.
Players wish to drive agent opinions to target values.
Targets could be the same (cooperative players) or different (competitive players)
Opinion dynamics with control

- Recent interest: adding control to an opinion dynamics model.
- Additive (affine) control term affects opinions after averaging process (Veetaseveera 2021, Mazalov 2023)
- Additive (affine) control term affects opinions before averaging process (Barabanov 2010, Bullo 2023)
- Agent opinion is affected by an entity called player.
  - Players wish to drive agent opinions to target values
  - Targets could be the same (cooperative players) or different (competitive players)
Opinion dynamics with control

- Recent interest: adding control to an opinion dynamics model.
- Additive (affine) control term affects opinions after averaging process (Veetaseveera 2021, Mazalov 2023)
- Additive (affine) control term affects opinions before averaging process (Barabanov 2010, Bullo 2023)
- Agent opinion is affected by an entity called **player**.
- Players wish to drive agent opinions to target values
  - Targets could be the same (cooperative players) or different (competitive players)
Opinion dynamics with control

- Recent interest: adding control to an opinion dynamics model.
- Additive (affine) control term affects opinions after averaging process (Veetaseveera 2021, Mazalov 2023)
- Additive (affine) control term affects opinions before averaging process (Barabanov 2010, Bullo 2023)
- Agent opinion is affected by an entity called player.
- Players wish to drive agent opinions to target values
- Targets could be the same (cooperative players) or different (competitive players)
Ingredients of OD games with control

- OD model on graph for agents.
- Each player acts on a set of agent nodes and has a cost function (involving target and control).
- Game playing procedure: sequential ([randomized] Gauss-Seidel), parallel (Jacobi) or simultaneous.
- Players are assumed to be rational, i.e., optimize cost one step ahead (in each round of play).
- Questions: Targets attained? Nash equilibrium? Effect of game-playing procedure?
- Results: Published on TechArXiv (IEEE), currently under review.
Opinion dynamics games with OSAO control

Ingredients of OD games with control

- OD model on graph for agents.
- Each player acts on a set of agent nodes and has a cost function (involving target and control).
- Game playing procedure: sequential ([randomized] Gauss-Seidel), parallel (Jacobi) or simultaneous.
- Players are assumed to be rational, i.e., optimize cost one step ahead (in each round of play).
- Questions: Targets attained? Nash equilibrium? Effect of game-playing procedure?
- Results: Published on TechArXiv (IEEE), currently under review.
Ingredients of OD games with control

- OD model on graph for agents.
- Each player acts on a set of agent nodes and has a cost function (involving target and control).
- Game playing procedure: sequential ([randomized] Gauss-Seidel), parallel (Jacobi) or simultaneous.
- Players are assumed to be rational, i.e., optimize cost one step ahead (in each round of play).
- Questions: Targets attained? Nash equilibrium? Effect of game-playing procedure?
- Results: Published on TechArXiv (IEEE), currently under review.
Opinion dynamics games with OSAO control

Ingredients of OD games with control

- OD model on graph for agents.
- Each player acts on a set of agent nodes and has a cost function (involving target and control).
- Game playing procedure: sequential ([randomized] Gauss-Seidel), parallel (Jacobi) or simultaneous.
- Players are assumed to be rational, i.e., optimize cost one step ahead (in each round of play).

Questions: Targets attained? Nash equilibrium? Effect of game-playing procedure?

Results: Published on TechArXiv (IEEE), currently under review.
Ingredients of OD games with control

- OD model on graph for agents.
- Each player acts on a set of agent nodes and has a cost function (involving target and control).
- Game playing procedure: sequential ([randomized] Gauss-Seidel), parallel (Jacobi) or simultaneous.
- Players are assumed to be rational, i.e., optimize cost one step ahead (in each round of play).
- Questions: Targets attained? Nash equilibrium? Effect of game-playing procedure?
- Results: Published on TechArXiv (IEEE), currently under review.
Opinion dynamics games with OSAO control

Ingredients of OD games with control

- OD model on graph for agents.
- Each player acts on a set of agent nodes and has a cost function (involving target and control).
- Game playing procedure: sequential ([randomized] Gauss-Seidel), parallel (Jacobi) or simultaneous.
- Players are assumed to be rational, i.e., optimize cost one step ahead (in each round of play).
- Questions: Targets attained? Nash equilibrium? Effect of game-playing procedure?
- Results: Published on TechArXiv (IEEE), currently under review.
ODG+OSAOC: main results

- **Closed-form expression for OSAOC.**
- OSAOC results in computable Nash equilibria under J, GS and RGS game-playing procedures.
- For each player, OSAOC control = approximate projection of “residual error” onto “control direction”.
- Randomized Gauss-Seidel leads to (small) oscillations around target opinions.
- OSAOC approach works equally well for linear and nonlinear opinion dynamics models.
- OSAOC approach can be reformulated to find best agents to influence (= “controls”).
ODG + OSAOC: main results

- Closed-form expression for OSAOC.
- OSAOC results in computable Nash equilibria under J, GS and RGS game-playing procedures.
  - For each player, OSAOC control = approximate projection of “residual error” onto “control direction”.
- Randomized Gauss-Seidel leads to (small) oscillations around target opinions.
- OSAOC approach works equally well for linear and nonlinear opinion dynamics models
- OSAOC approach can be reformulated to find best agents to influence (= “controls”).
ODG+OSAOC: main results

- Closed-form expression for OSAOC.
- OSAOC results in computable Nash equilibria under J, GS and RGS game-playing procedures.
- For each player, OSAOC control = approximate projection of “residual error” onto “control direction”.
- Randomized Gauss-Seidel leads to (small) oscillations around target opinions.
- OSAOC approach works equally well for linear and nonlinear opinion dynamics models
- OSAOC approach can be reformulated to find best agents to influence (= “controls”).
ODG+OSAOC: main results

- Closed-form expression for OSAOC.
- OSAOC results in computable Nash equilibria under J, GS and RGS game-playing procedures.
- For each player, OSAOC control = approximate projection of “residual error” onto “control direction”.
- Randomized Gauss-Seidel leads to (small) oscillations around target opinions.
- OSAOC approach works equally well for linear and nonlinear opinion dynamics models
- OSAOC approach can be reformulated to find best agents to influence (= “controls”).
ODG + OSAOC: main results

- Closed-form expression for OSAOC.
- OSAOC results in computable Nash equilibria under J, GS and RGS game-playing procedures.
- For each player, OSAOC control = approximate projection of “residual error” onto “control direction”.
- Randomized Gauss-Seidel leads to (small) oscillations around target opinions.
- OSAOC approach works equally well for linear and nonlinear opinion dynamics models
- OSAOC approach can be reformulated to find best agents to influence (= “controls”).
ODG+OSAOC: main results

- Closed-form expression for OSAOC.
- OSAOC results in computable Nash equilibria under J, GS and RGS game-playing procedures.
- For each player, OSAOC control = approximate projection of “residual error” onto “control direction”.
- Randomized Gauss-Seidel leads to (small) oscillations around target opinions.
- OSAOC approach works equally well for linear and nonlinear opinion dynamics models.
- OSAOC approach can be reformulated to find best agents to influence (= “controls”).
ODG under OSAOC: flipping polarization

Figure 11. Directed graph of ten polarized agents in 2 clusters: agents \( \{1, 2, 3, 4, 5\} \) have initial opinions close to 1 and agents \( \{6, 7, 8, 9, 10\} \) have initial opinions close to \(-1\). Two players target these clusters with the objective of flipping the polarization.
Concluding remarks

- One step ahead optimal control is simple and useful!
- Descriptive name (OSAOC) avoids terms like “greedy”, “myopic”, “simple MPC”.
- Discrete-time setting allows immediate use of powerful optimization tools, enabling real time control.
- Suboptimal, but often not too far from omniscient optimal, as seen in many applications.
- Many models are piecewise linear, indices are often linear or quadratic: OSAOC should work!
- Applications in pipeline: biological pest control, ant colony migration, debt stabilization game ...
Concluding remarks

- One step ahead optimal control is simple and useful!
- Descriptive name (OSAOC) avoids terms like “greedy”, “myopic”, “simple MPC”.
- Discrete-time setting allows immediate use of powerful optimization tools, enabling real time control.
- Suboptimal, but often not too far from omniscient optimal, as seen in many applications.
- Many models are piecewise linear, indices are often linear or quadratic: OSAOC should work!
- Applications in pipeline: biological pest control, ant colony migration, debt stabilization game ...
Concluding remarks

- One step ahead optimal control is simple and useful!
- Descriptive name (OSAOC) avoids terms like “greedy”, “myopic”, “simple MPC”.
- Discrete-time setting allows immediate use of powerful optimization tools, enabling real time control.
- Suboptimal, but often not too far from omniscient optimal, as seen in many applications.
- Many models are piecewise linear, indices are often linear or quadratic: OSAOC should work!
- Applications in pipeline: biological pest control, ant colony migration, debt stabilization game ...
Concluding remarks

- One step ahead optimal control is simple and useful!
- Descriptive name (OSAOC) avoids terms like “greedy”, “myopic”, “simple MPC”.
- Discrete-time setting allows immediate use of powerful optimization tools, enabling real time control.
- Suboptimal, but often not too far from omniscient optimal, as seen in many applications.
- Many models are piecewise linear, indices are often linear or quadratic: OSAOC should work!
- Applications in pipeline: biological pest control, ant colony migration, debt stabilization game ...
Concluding remarks

- One step ahead optimal control is simple and useful!
- Descriptive name (OSAOC) avoids terms like “greedy”, “myopic”, “simple MPC”.
- Discrete-time setting allows immediate use of powerful optimization tools, enabling real time control.
- Suboptimal, but often not too far from omniscient optimal, as seen in many applications.
- Many models are piecewise linear, indices are often linear or quadratic: OSAOC should work!
- Applications in pipeline: biological pest control, ant colony migration, debt stabilization game ...
Concluding remarks

- One step ahead optimal control is simple and useful!
- Descriptive name (OSAOC) avoids terms like “greedy”, “myopic”, “simple MPC”.
- Discrete-time setting allows immediate use of powerful optimization tools, enabling real time control.
- Suboptimal, but often not too far from omniscient optimal, as seen in many applications.
- Many models are piecewise linear, indices are often linear or quadratic: OSAOC should work!
- Applications in pipeline: biological pest control, ant colony migration, debt stabilization game ...
Acknowledgments

- Eugenius Kaszkurewicz (coauthor on all work except OD), Gabriel Gentil (PhD student working on OD)
- CAPES, CNPq, FAPERJ for partial support of the research reported here.
- NACAD, PEE/COPPE/UFRJ
Acknowledgments

- Eugenius Kaszkurewicz (coauthor on all work except OD), Gabriel Gentil (PhD student working on OD)
- CAPES, CNPq, FAPERJ for partial support of the research reported here.
- NACAD, PEE/COPPE/UFRJ
Eugenius Kaszkurewicz (coauthor on all work except OD), Gabriel Gentil (PhD student working on OD)
CAPES, CNPq, FAPERJ for partial support of the research reported here.
NACAD, PEE/COPPE/UFRJ
References

- Gabriel Gentil, AB, Opinion Dynamics Games Under OSAOC, Preprint under review, September 2023 TechRxiv IEEE
References


- Gabriel Gentil, AB, Opinion Dynamics Games Under OSAOC, Preprint under review, September 2023 TechRxiv IEEE


Gabriel Gentil, AB, Opinion Dynamics Games Under OSAOC, Preprint under review, September 2023 TechRxiv IEEE.
References

- Gabriel Gentil, AB, Opinion Dynamics Games Under OSAOC, Preprint under review, September 2023 TechRxiv IEEE
Last slide!

- Thank you!
- Questions?

Business Dynamics Models
Optimization-Based
One Step Ahead Optimal Control

Eugenius Kaszkurewicz
Amit Bhaya

https://www.amazon.com/Business-
Last slide!

- Thank you!
- Questions?


https://www.amazon.com/Business-Dynamics-Models-Optimization-Based/dp/1611977304
Last slide!

- Thank you!
- Questions?

Business Dynamics Models
Optimization-Based
One Step Ahead Optimal Control

Eugenius Kaszkurewicz
Amit Bhaya

https://www.amazon.com/Business-