Receding Horizon Control Methods in Financial Engineering

James A. Primbs
Management Science and Engineering
Stanford University

CRFMS Seminar, UCSB
May 21, 2007

Outline

Motivation and Background

Receding Horizon Control

Semi-Definite Programming Formulation

Theoretical Properties

Numerical Examples

Conclusions and Future Work

Asset Price and Wealth Dynamics in Discrete Time:

Let $S_i(k)$ denote the price of asset *i* at time *k*.

$$\boxed{S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1\ldots n}$$

Where $\left\{ \begin{array}{ll} \mu_i & \text{-- Expected Return} \\ w_i & \text{-- iid Noise Term} \end{array} \right.$ $\left[w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \quad \begin{array}{l} E[w_i] = 0 \\ E[ww^T] = \Sigma \end{array} \right]$

Asset Price and Wealth Dynamics in Discrete Time:

$$S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1 \ldots n$$

Assume that we can invest in $l \leq n$ of the $S_i(k)$.

Let $u_i(k)$ denote the dollar amount invested in $S_i(k)$

Let r_f be a risk free rate of interest.

Let $W(k)$ denote your wealth at time k.

$$W(k+1) = (1+r_f)W(k) + \sum_{i=1}^{l} \big((\mu_i - r_f) + w_i(k)\big)u_i(k)$$

Asset Price and Wealth Dynamics in Discrete Time:

$$S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1 \ldots n$$

$$W(k+1) = (1+r_f)W(k) + \sum_{i=1}^{l} \big((\mu_i - r_f) + w_i(k)\big) u_i(k)$$

Finance Problem #1: Index Tracking

$$\min_u E\left\{\sum_{k=0}^{\infty}\rho^{2k}(W(k)-I(k))^2\right\}$$

$$S_i(k+1)=(1+\mu_i+w_i(k))S_i(k) \qquad i=1\ldots n$$

$$W(k+1)=(1+r_f)W(k)+\sum_{i=1}^{l}\big((\mu_i-r_f)+w_i(k)\big)u_i(k)$$

An index is a weighted average of n stocks: $I(k)=\sum_{i=1}^{n}\alpha_i S_i(k)$

The index tracking problem is to trade $l<n$ of the stocks and a risk free bond in order to track the index as "closely as possible".

One possible measure of how closely we track the index is given by an infinite horizon discounted quadratic cost:

Finance Problem #1: Index Tracking

$$\min_u E\left\{\sum_{k=0}^{\infty} \rho^{2k}(W(k)-I(k))^2\right\}$$

$$S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1 \ldots n$$

$$W(k+1) = (1+r_f)W(k) + \sum_{i=1}^{l}\left((\mu_i - r_f) + w_i(k)\right)u_i(k)$$

Limits on short selling: $u_i(k) \geq 0$

Limits on wealth invested: $u_i(k) \leq \beta_i W(k)$

Value-at-Risk: $\Pr(W(k) \geq \gamma I(k)) \geq 1-\delta$

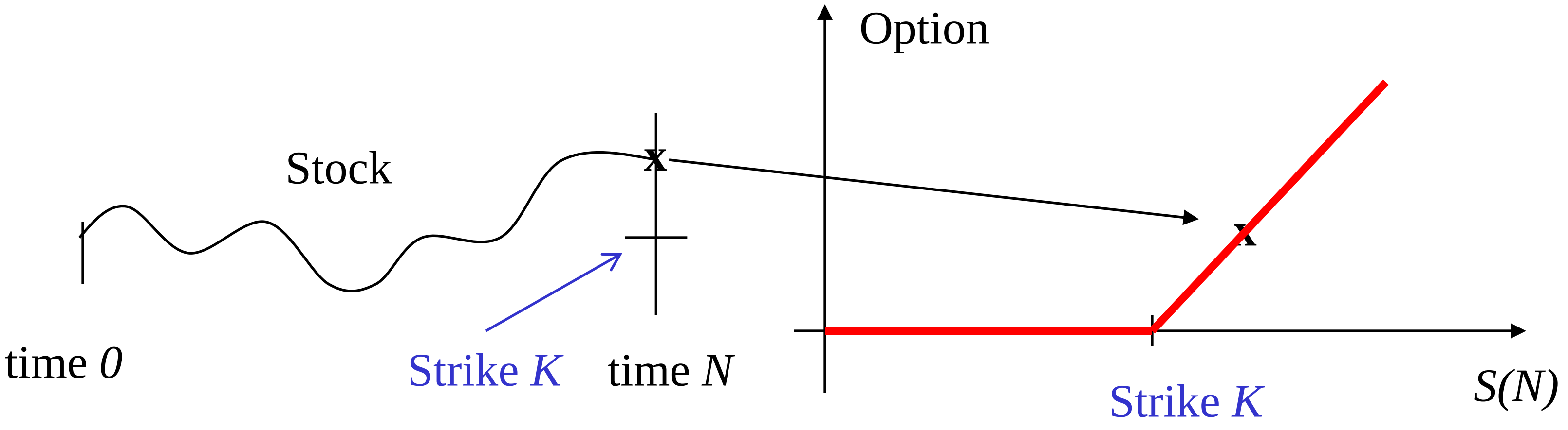
etc…

Finance Problem #2: Dynamic Hedging of Options

$$S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1 \ldots n$$

$$W(k+1) = (1+r_f)W(k) + \sum_{i=1}^{l} \big((\mu_i - r_f) + w_i(k)\big)u_i(k)$$

Trade a portfolio to replicate an option payoff $(S(N)-K)^+$

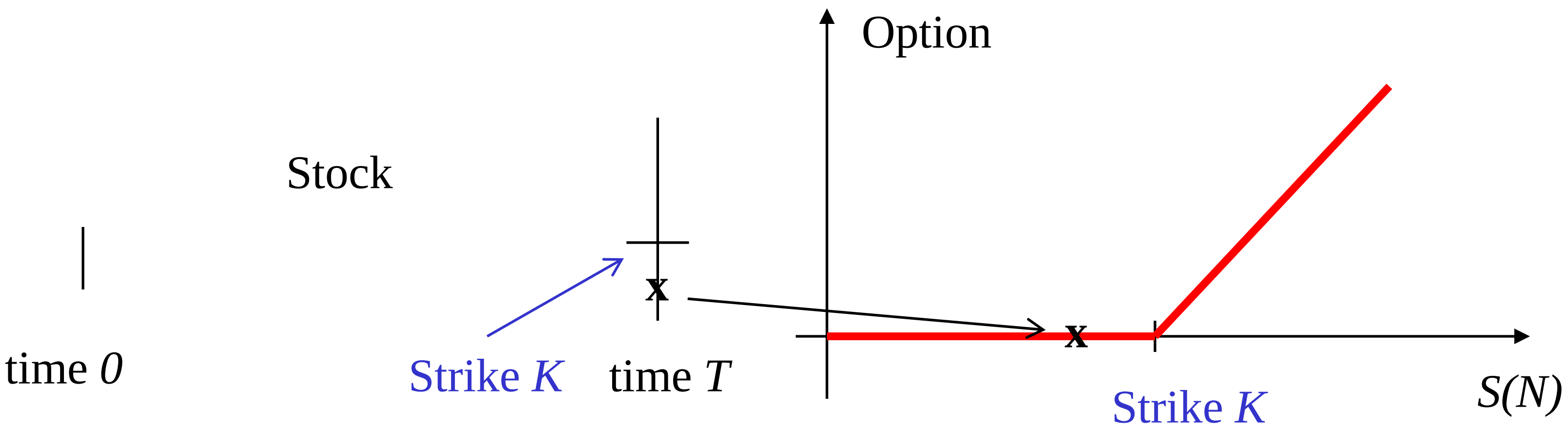


Finance Problem #2: Dynamic Hedging of Options

$$S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1 \ldots n$$

$$W(k+1) = (1+r_f)W(k) + \sum_{i=1}^{l}\Big((\mu_i - r_f) + w_i(k)\Big)u_i(k)$$

Trade a portfolio to replicate an option payoff $(S(N)-K)^+$



Finance Problem #2: Dynamic Hedging of Options

$$\max_u \delta$$

$$S_i(k+1) = (1 + \mu_i + w_i(k))S_i(k) \qquad i = 1 \ldots n$$

$$W(k+1) = (1 + r_f)W(k) + \sum_{i=1}^{l} \big((\mu_i - r_f) + w_i(k)\big)u_i(k)$$

$$E[W(N)] - \gamma Var[W(N)] \geq \delta$$

$$E[W(N) - (S(N) - K)] - \gamma Var[W(N) - (S(N) - K)] \geq \delta$$

for a well chosen γ !

It could also be subject to short selling constraints, transaction costs, etc.

Linear Systems with Multiplicative Noise

$$S_i(k+1) = (1+\mu_i + w_i(k))S_i(k) \qquad i = 1\ldots n$$

$$W(k+1) = (1+r_f)W(k) + \sum_{i=1}^{l}\big((\mu_i - r_f) + w_i(k)\big)u_i(k)$$

$$x(k) = \begin{bmatrix} S_1(k) \\ \vdots \\ S_n(k) \\ W(k) \end{bmatrix}$$

$$x(k+1) = Ax(k) + Bu(k) + \sum_{i=1}^{q}\big(C_i x(k) + D_i u(k)\big)w_i(k)$$

I will study an **expectation** constrained SLQ type problem

$$\left\{\begin{array}{l} \min_u E\left\{\sum_{k=0}^{\infty}\left(\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T M \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}\right)\right\} \quad \text{where} \quad \left\{\begin{array}{l} M = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \succeq 0 \\ Q \succ 0 \end{array}\right\} \\ \text{subject to:} \\ x(k+1) = Ax(k) + Bu(k) + \sum_{i=1}^{q}\left(C_i x(k) + D_i u(k)\right) w_i(k) \\ E\left(\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T H \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + f^T \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}\right) \le \beta \end{array}\right.$$

Ideally, I would also consider chance constraints

$$P(a^T x(k) + b^T u(k) \le c) \ge 1 - d$$

but these are too difficult computationally…

In this talk, I will present:

- ➡ Newly developed SDP based stochastic receding horizon control methods for constrained linear systems with multiplicative noise.
- ➡ Theoretical and practical challenges in stochastic formulations of constrained RHC.
- ➡ Numerical results for financial engineering problems.

References:

Unconstrained SLQ Problem

Willems and Willems, ’76, El Ghaoui, ’95, Ait Rami and Zhou, ’00, Yao et. al., ’01, Kleinman, ’69, Wonham, ’67, ’68, McLane, ’71

Stochastic Receding Horizon Control:

Control Theory (van Hessem and Bosgra, ’01,’02,’03,’04)
(Couchman et. al. ’06, Li et. al. ’02)
(Yan and Bitmead, ’05, Primbs ‘07)

Dynamic resource allocation: (Castanon and Wohletz, ’02)

Portfolio Optimization: (Herzog et al, ’06, Herzog ‘06)

Dynamic Hedging: (Meindl and Primbs, ’04)

Supply Chains: (Seferlis and Giannelos, ’04)

Stoch. Programming Approach: (Felt, ’03)

Operations Management: (Chand, Hsu, and Sethi, ’02)

Outline

Motivation and Background

$\boxed{\text{Receding Horizon Control}}$

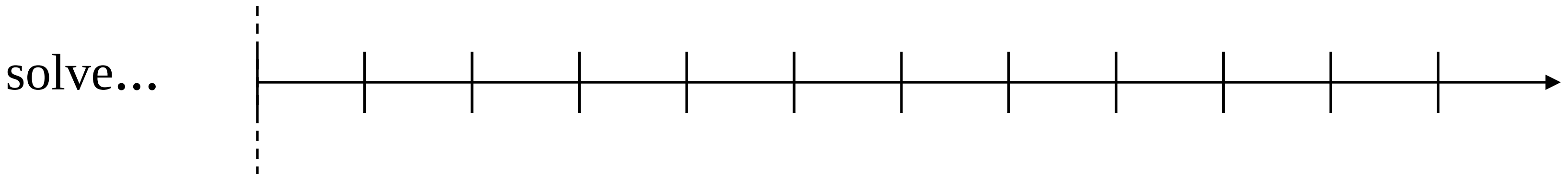
Semi-Definite Programming Formulation

Theoretical Properties

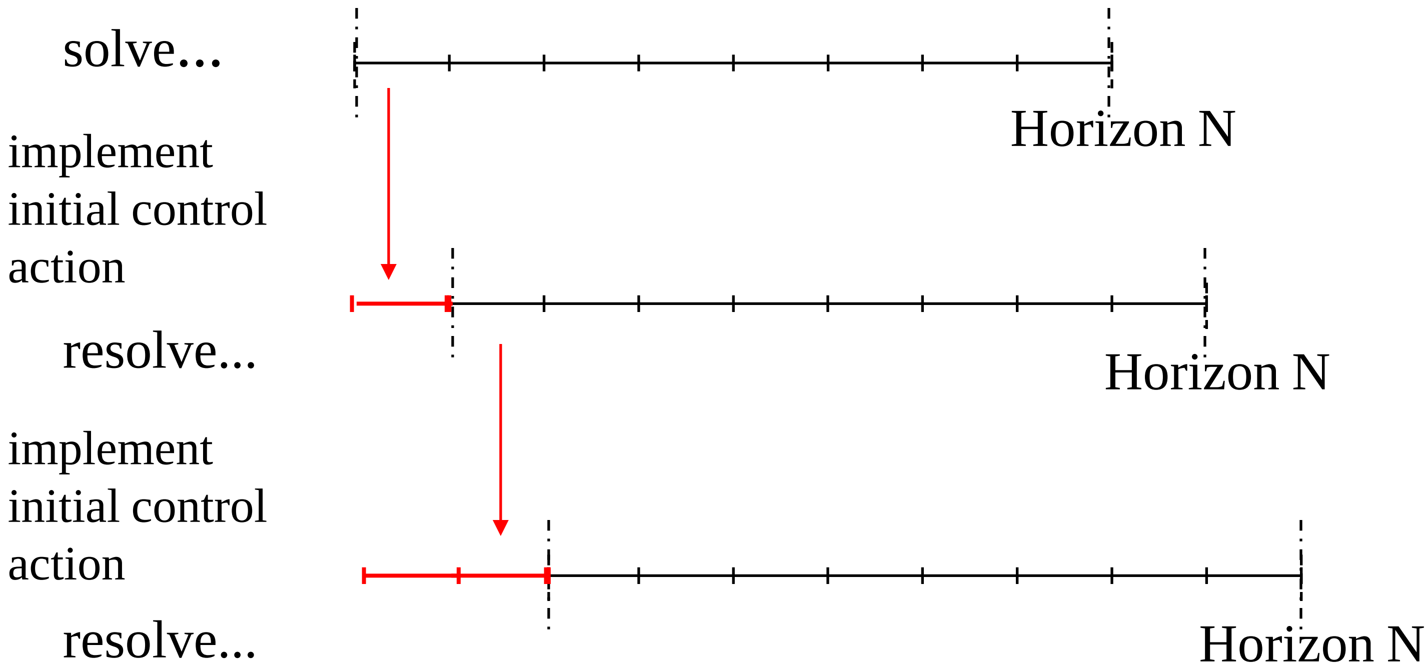
Numerical Examples

Conclusions and Future Work

Ideally, one would solve the optimal infinite horizon problem...



Instead, receding horizon control repeatedly solves finite horizon problems...

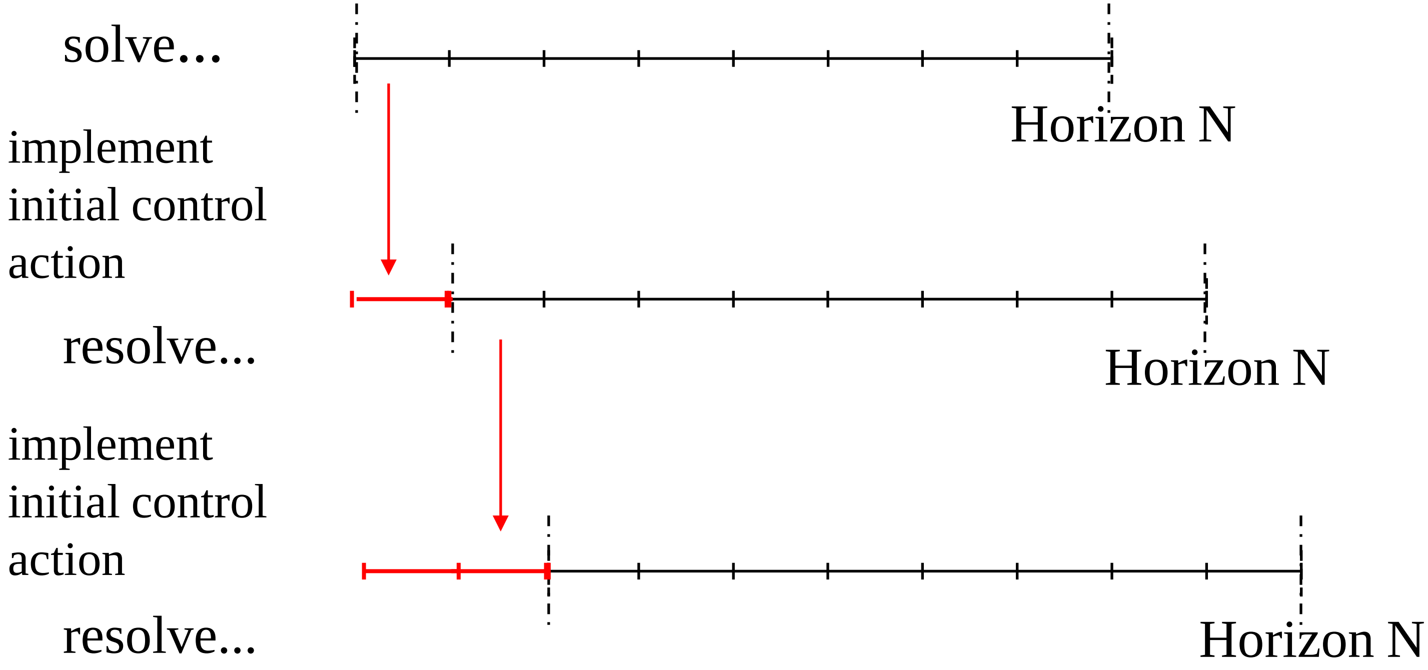


So, RHC involves 2 steps:

1) Solve finite horizon optimizations on-line ☐

2) Implement the initial control action ☐

Instead, receding horizon control repeatedly solves finite horizon problems...



Outline

Motivation and Background

Receding Horizon Control

$\boxed{\text{Semi-Definite Programming Formulation}}$

Theoretical Properties

Numerical Examples

Conclusions and Future Work

So, RHC involves 2 steps:

1) Solve finite horizon optimizations on-line

2) Implement the initial control action

Instead, receding horizon control repeatedly solved finite horizon problems...

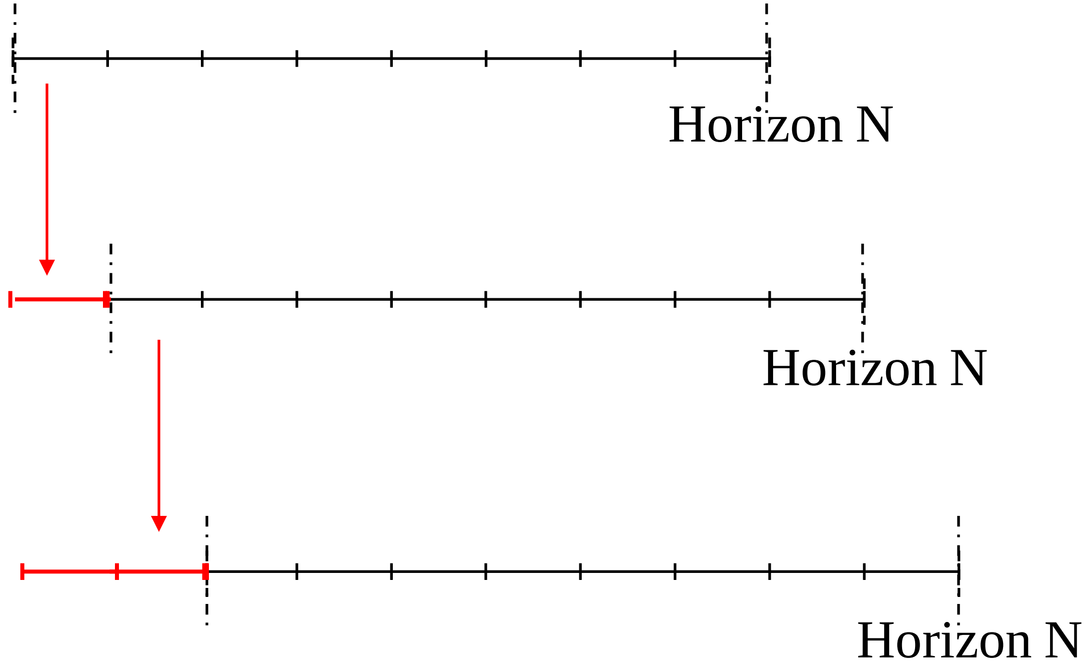
solve...

implement initial control action

resolve...

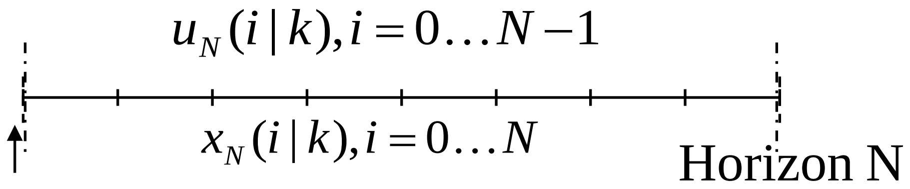
implement initial control action

resolve...



In receding horizon control, we consider a finite horizon optimal control problem.

solve...



From the current state *x(k)*, let:

$$\begin{cases} N \text{ denote the horizon length} \\ u_N(i|k), i = 0 \ldots N-1 \text{ denote the predicted control} \\ x_N(i|k), i = 0 \ldots N \text{ denote the predicted state} \\ \text{Note that: } x_N(0|k) = x(k) \end{cases}$$

We will impose an open loop plus linear feedback structure:

$$\begin{cases} u_N(0|k) = \bar{u}_N(0|k), \\ u_N(i|k) = \bar{u}_N(i|k) + K(i|k)[x_N(i|k) - \bar{x}_N(i|k)] \\ \text{where } \bar{x}_N(i|k) = E_{x(k)}[x_N(i|k)] \text{ and } \bar{u}_N(i|k) \in \Re^m \end{cases}$$

We will use quadratic expectation constraints (instead of probabilistic constraints) in the on-line optimizations.

Constraints will be in the form of quadratic expectation constraints

$$\boldsymbol{C}(x(k),N) \quad \begin{cases} E_{x(k)}\left(\begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}^T H \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix} + f^T \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix} \right) \le \beta & i = 0 \ldots N-1 \\ E_{x(k)}\left(x_N^T(i|k) H^X x_N(i|k) + (f^X)^T x_N(i|k) \right) \le \beta^X & i = 1 \ldots N \end{cases}$$

Note that state-only constraints are **not** imposed at time *i=0*.

Receding Horizon On-Line Optimization:

$$P(x(k),N)\begin{cases} V_N(x(k)) = \min_{\bar{u}_N(\cdot|k),K(\cdot|k)} E_{x(k)}\left\{\sum_{i=0}^{N-1}\left(\begin{bmatrix} x_N(i|k) \\ u_N(i|k)\end{bmatrix}^T M \begin{bmatrix} x_N(i|k) \\ u_N(i|k)\end{bmatrix}\right) + x_N^T(N|k)\Phi x_N(N|k)\right\} \\ \text{subject to:} \\ x_N(i+1|k) = Ax_N(i|k) + Bu_N(i|k) + \sum_{j=1}^{q}\left(C_j x_N(i|k) + D_j u_N(i|k)\right)w_j(i) \\ (x_N(\cdot|k),u_N(\cdot|k)) \in \mathbf{C}(x(k),N) \\ u_N(0|k) = \bar{u}_N(0|k),\ \ u_N(i|k) = \bar{u}_N(i|k) + K(i|k)[x_N(i|k) - \bar{x}_N(i|k)] \\ E_{x(k)}\left(x_N^T(N|k)\Phi x_N(N|k)\right) \le \alpha \end{cases}$$

➡ We denote the the optimizing predicted state and control sequence by $u_N^*(i|k)$ and $x_N^*(i|k)$.

➡ $\boxed{\text{The receding horizon control policy for state } x(k) \text{ is } u_N^*(0|k).}$

This problem has a linear-quadratic structure to it.

Hence, it essentially depends on the mean and variance of the controlled dynamics.

For convenience, let $u(i) = u_N(i \mid k), x(i) = x_N(i \mid k)$

$$\Sigma(i) = E_{x(k)}[(x(i) - \bar{x}(i))(x(i) - \bar{x}(i))^T]$$

➡ mean satisfies: $\bar{x}(i+1) = A\bar{x}(i) + B\bar{u}(i)$

➡ covariance satisfies:

$$\Sigma(i+1) = (A + BK(i))\Sigma(i)(A + BK(i))^T + \sum_{j=1}^{q}(C_j + D_jK(i))\Sigma(i)(C_j + D_jK(i))^T + \sum_{j=1}^{q}(C_j\bar{x}(i) + D_j\bar{u}(i))(C_j\bar{x}(i) + D_j\bar{u}(i))^T$$

Receding Horizon On-Line Optimization as an SDP (for *q=1*):

$$
RHC(x(k),N)\left\{
\begin{array}{l}
V_N(x(k)) = \min\limits_{\bar{u}(\cdot|k),K(\cdot|k)} \left\{ \sum\limits_{i=0}^{N-1} Tr(MP(i)) + Tr(\Phi P^X(N)) \right\} \\
\text{subject to:} \\
\quad \bar{x}(i+1) = A\bar{x}(i) + B\bar{u}(i) \\
\quad \begin{bmatrix} \Sigma(i+1) & A\Sigma(i)+BU(i) & C\Sigma(i)+DU(i) & C\bar{x}(i)+D\bar{u}(i) \\ (A\Sigma(i)+BU(i))^T & \Sigma(i) & 0 & 0 \\ (C\Sigma(i)+DU(i))^T & 0 & \Sigma(i) & 0 \\ (C\bar{x}(i)+D\bar{u}(i))^T & 0 & 0 & 1 \end{bmatrix} \succ 0 \\
\quad \begin{bmatrix} P(i) & * & * \\ \begin{bmatrix} \Sigma(i) & U^T(i) \end{bmatrix} & \Sigma(i) & * \\ \begin{bmatrix} x^T(i) & u^T(i) \end{bmatrix} & 0 & 1 \end{bmatrix} \succ 0 \qquad P(i) = \begin{bmatrix} P^X(i) & * \\ * & * \end{bmatrix} \\
\quad Tr\big(HP(i)\big) + f^T \begin{bmatrix} \bar{x}(i) \\ \bar{u}(i) \end{bmatrix} \le \beta \qquad i = 0 \ldots N-1 \\
\quad Tr\big(H^X P(i)\big) + (f^X)^T \bar{x}(i) \le \beta^X \qquad i = 1 \ldots N \\
\quad Tr\big(\Phi P^X(N)\big) \le \alpha
\end{array}
\right.
$$

By imposing structure in the on-line optimizations we are able to:

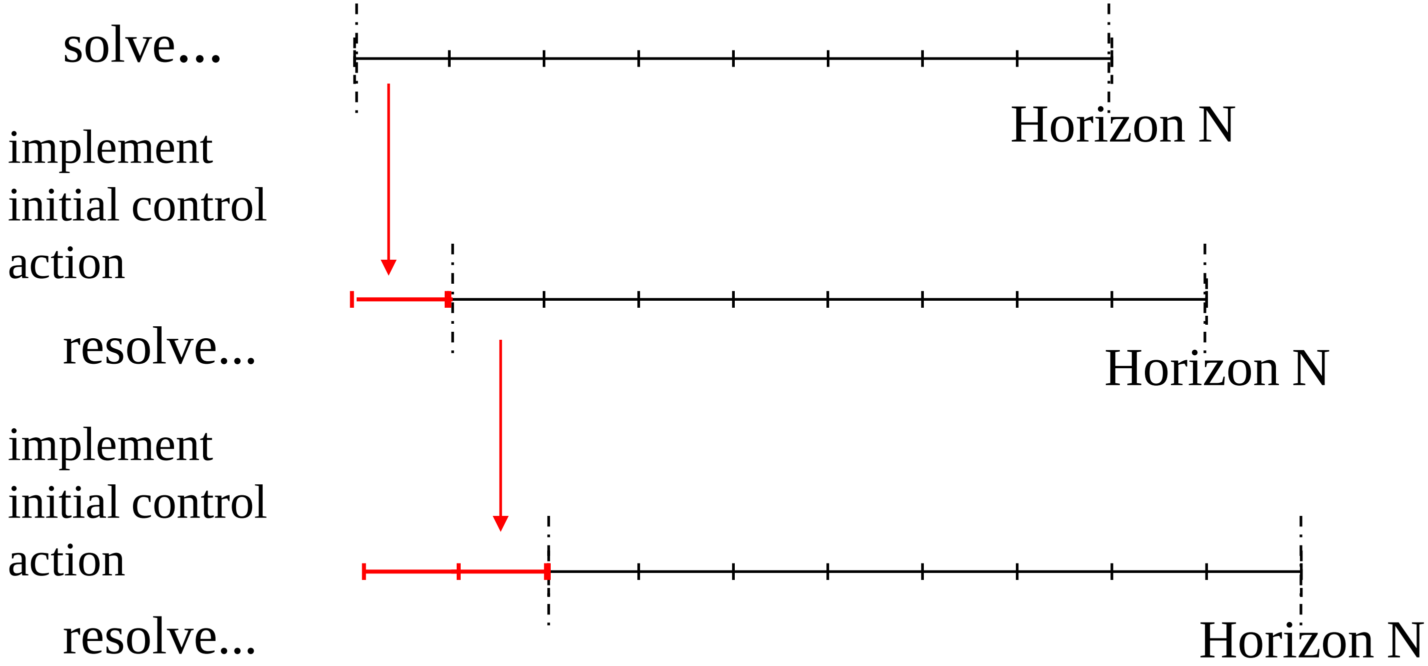
- ➡ Formulate them as semi-definite programs.
- ➡ Use closed loop feedback over the prediction horizon.
- ➡ Incorporate constraints in an expectation form.

So, RHC involves 2 steps:

1) Solve finite horizon optimizations on-line ☑

2) Implement the initial control action ☐

Instead, receding horizon control repeatedly solved finite horizon problems...



Outline

Motivation and Background

Receding Horizon Control

Semi-Definite Programming Formulation

$\boxed{\text{Theoretical Properties}}$

Numerical Examples

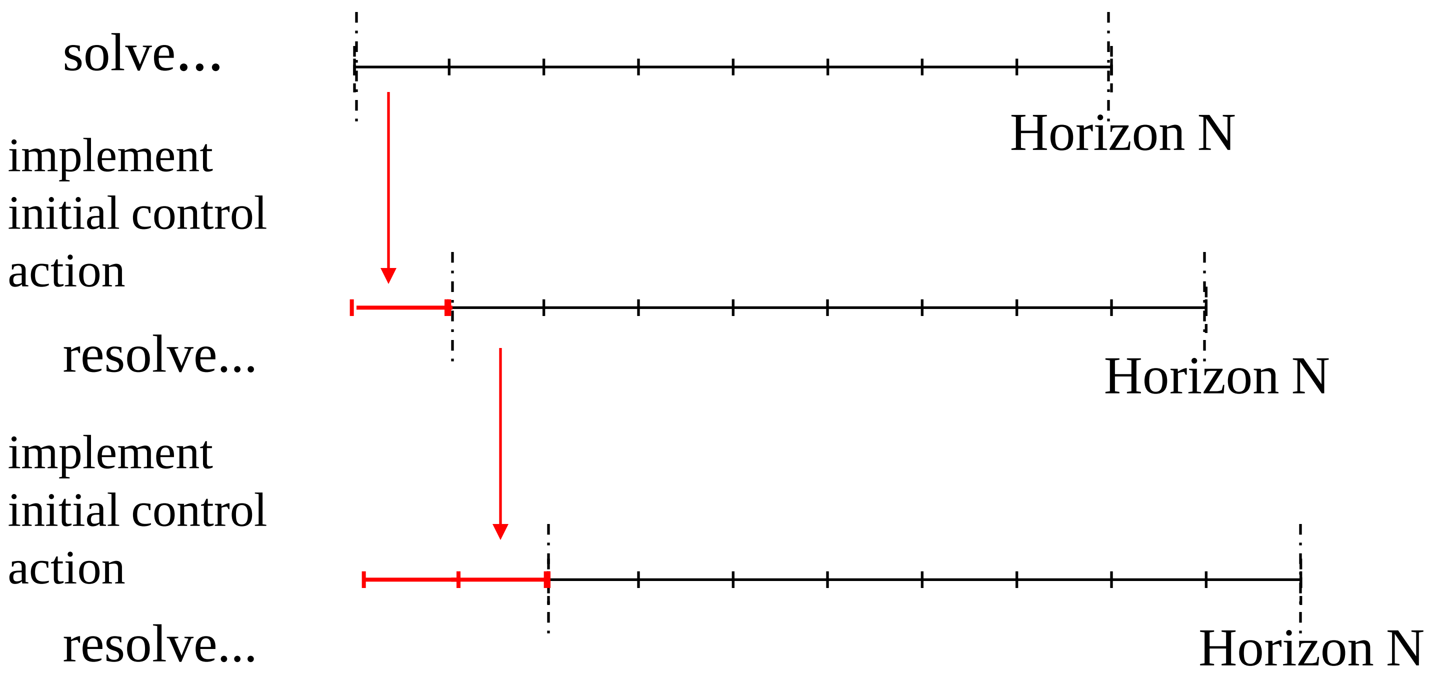
Conclusions and Future Work

So, RHC involves 2 steps:

1) Solve finite horizon optimizations on-line

2) Implement the initial control action

Instead, receding horizon control repeatedly solved finite horizon problems...



Three important questions:

Stability:

Does the receding horizon approach guarantee asymptotic stability?

Performance:

What can be said about the performance of the receding horizon strategy versus the optimal infinite horizon strategy?

Constraint Satisfaction:

What can be said about the satisfaction of constraints over the infinite horizon under the receding horizon strategy?

To characterize stability properties for **hard constraints**, we

1) Impose a terminal cost at the end of the finite horizon.

2) Impose a terminal constraint at the end of the horizon.

$$V_N(x(k)) = \min_{u_N(\cdot|k)} E_{x(k)}\left\{\sum_{i=0}^{N-1}\left(\begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}^T M \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}\right) + x_N^T(N|k)\Phi x_N(N|k)\right\}$$

$x^T \Phi x$ is a Stochastic Lyapunov Function

Terminal Constraint: $E_{x(k)}\left(x_N^T(N|k)\Phi x_N(N|k)\right) \leq \alpha$

3) Address feasibility issues…

A feasibility issue.

Since we have stochastic dynamics, there is some probability that we may encounter infeasible states for the on-line optimization problems.

Two ways to deal with this infeasibility:

- ➡ Stop the system when/if it goes infeasible.
- ➡ Define a control policy when infeasible state are encountered.

We take the second route...

Consider the following policy:

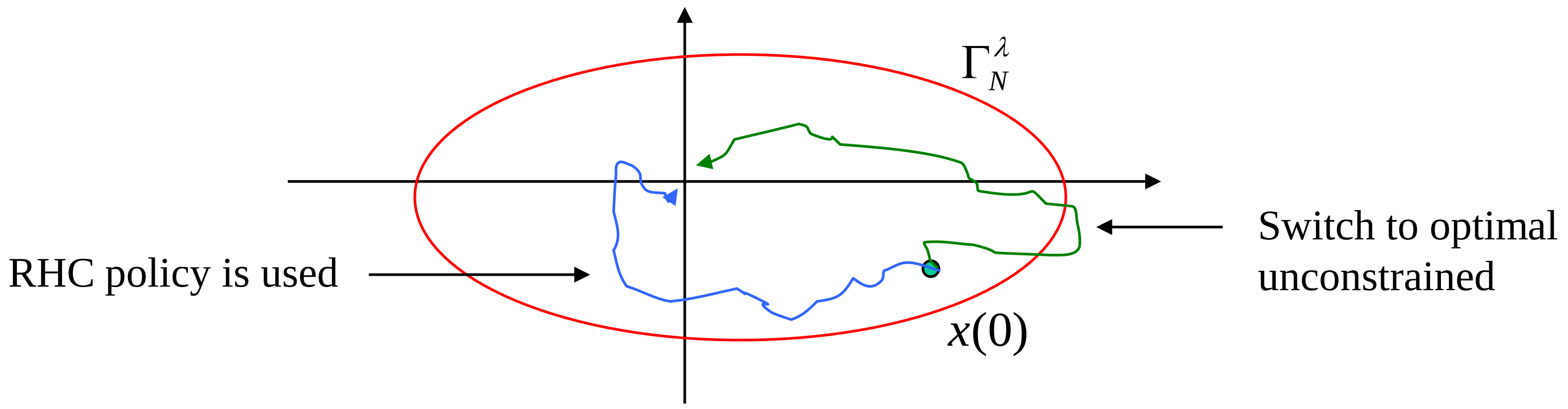
Define: $\Gamma_N^{\lambda} = \{ x \mid V_N(x) < \lambda \}$

Let τ be the stopping time: $\tau = \min\{ k \mid x(k) \notin \Gamma_N^{\lambda} \}$

And define the control policy: $\tilde{u}^{\lambda}(k) = u_N^*(0 \mid k) 1_{\{\tau > k\}} + u_\infty(k) 1_{\{\tau \le k\}}$

where $u_N^*(0 \mid k)$ is the receding horizon based policy with finite horizon cost $V_N(x)$

$u_\infty(k)$ is the optimal *unconstrained* policy with infinite horizon cost. $V_\infty^{un}(x)$



More Feasibility Issues

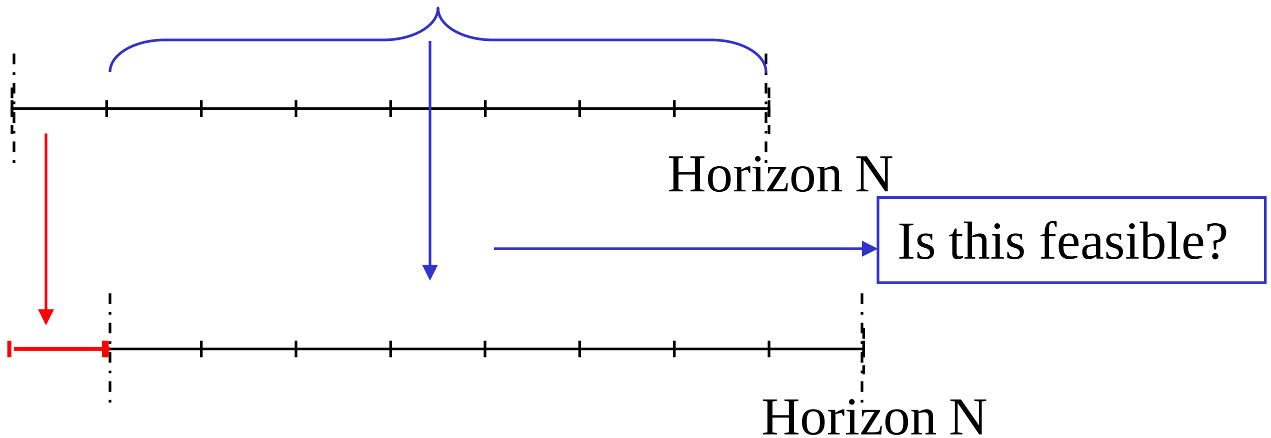
$$[u_N^*(i\,|\,k), i = 1 \ldots N-1]$$

time

$k \longrightarrow$ solve...

implement
initial control
action

$k+1 \longrightarrow$ resolve...



$$F_x = \left\{ y \;\middle|\; \begin{array}{l} [u_N^*(i\,|\,0), i = 1 \ldots N-1] \\ \text{corresponding to } RHC(x, N) \\ \text{is feasible for } RHC(y, N-1) \end{array} \right\}$$

F_x is the describes the set of states where the solution at x will be feasible at the next time step. The probability of this set will play a role in our stability results!

Theorems:

Stability:

Assume that for all $x \in \Gamma_N^\lambda$ we have

$$x^T Q x - \lambda P_x(G_x^c) \geq \rho(\| x \|)$$

where ρ is a class K function and $G_x = F_x \cup (\Gamma_N^\lambda)^c$.

Then under the control policy $\tilde{u}^\lambda$, $x(k) \to 0$ with probability one.

The proof involves showing that *V(k)* given by

$$V(k) = V_N(x(k))1_{\{\tau > k\}} + V_\infty^{un}(x(k))1_{\{\tau \leq k\}}$$

is a stochastic Lyapunov function (i.e. is a supermartingale).

Theorems:

Performance:

For all $x \in \Gamma_N^\lambda$

$$V_N(x) + \lambda \sum_{k=0}^{\infty} P_x(G_{x(k)}^c \cap \{\tau > k\}) \geq V_\infty^{rhc}(x) \geq V_\infty^{un}(x)$$

where $V_\infty^{rhc}(x)$ is the infinite horizon cost associated with $\tilde{u}^\lambda$

This result follows from the stability result.

Note that when feasibility is not an issue (i.e. $P_x(F_x)=1$), then the finite horizon cost is a bound on the infinite horizon performance.

Theorems:

Constraint Satisfaction:

Let λ be such that for $x \in \Gamma_N^\lambda$ we have $x^T Qx - \lambda P_x(F_x^c) > 0, \ \forall x \neq 0.$

Then

$$P_x\left(\sup_{k\geq 0} V_N(x(k)) \geq \lambda\right) \leq \frac{V_N(x)}{\lambda}$$

Thus, the random paths stay in Γ_N^λ with probability at least $1 - \dfrac{V_N(x)}{\lambda}$ and hence the receding horizon policy is applied over the infinite horizon with at least this probability. Furthermore, under the pure receding horizon policy $u^*(0|k)$ we have that

$$P_x\left(\begin{bmatrix} x(k) \\ u^*(0|k) \end{bmatrix}^T H \begin{bmatrix} x(k) \\ u^*(0|k) \end{bmatrix} + f^T \begin{bmatrix} x(k) \\ u^*(0|k) \end{bmatrix} \leq \beta\right) \geq 1 - \frac{V_N(x)}{\lambda}$$

A similar result can be stated for the state-only constraints, but with respect to modified feasibility sets that impose state-only constraints at $i=0$.

To circumvent these feasibility issues, one may use a **soft constraint** approach.

Replace hard constraint

$$E_{x(k)}\left(\begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}^T H \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix} + f^T \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}\right) \le \beta$$

with a soft constraint

$$E_{x(k)}\left(\begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}^T H \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix} + f^T \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}\right) \le \beta + \delta$$

and penalize δ in objective

$$\min_{u_N(\cdot|k)} E_{x(k)}\left\{\sum_{i=0}^{N-1}\left(\begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}^T M \begin{bmatrix} x_N(i|k) \\ u_N(i|k) \end{bmatrix}\right) + \delta^T \Psi \delta\right\}$$

This can be solved as an SDP, and an *always* feasible terminal constraint can be used to guarantee stability with probability *1*.

(See Primbs ‘07 CDC submission)

Outline

Motivation and Background

Receding Horizon Control

Semi-Definite Programming Formulation

Theoretical Properties

Numerical Examples

Conclusions and Future Work

Example Problem:

Dynamics

$$x(k+1) = Ax(k) + Bu(k) + \big(Cx(k) + Du(k)\big)w(k)$$

$$A = \begin{bmatrix} 1.02 & -0.1 \\ 0.1 & 0.98 \end{bmatrix} \qquad B = \begin{bmatrix} 0.1 & 0 \\ 0.05 & 0.01 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix} \qquad D = \begin{bmatrix} 0.04 & 0 \\ -0.04 & 0.008 \end{bmatrix}$$

Cost Parameters

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 5 & 0 \\ 0 & 20 \end{bmatrix}$$

Example Problem:

Optimal Unconstrained Cost to go:

$$V(x(k)) = x^T(k)\Phi x(k)$$

$$\Phi = \begin{bmatrix} 41.0331 & -5.7929 \\ -5.7929 & 54.3889 \end{bmatrix}$$

Terminal Constraint

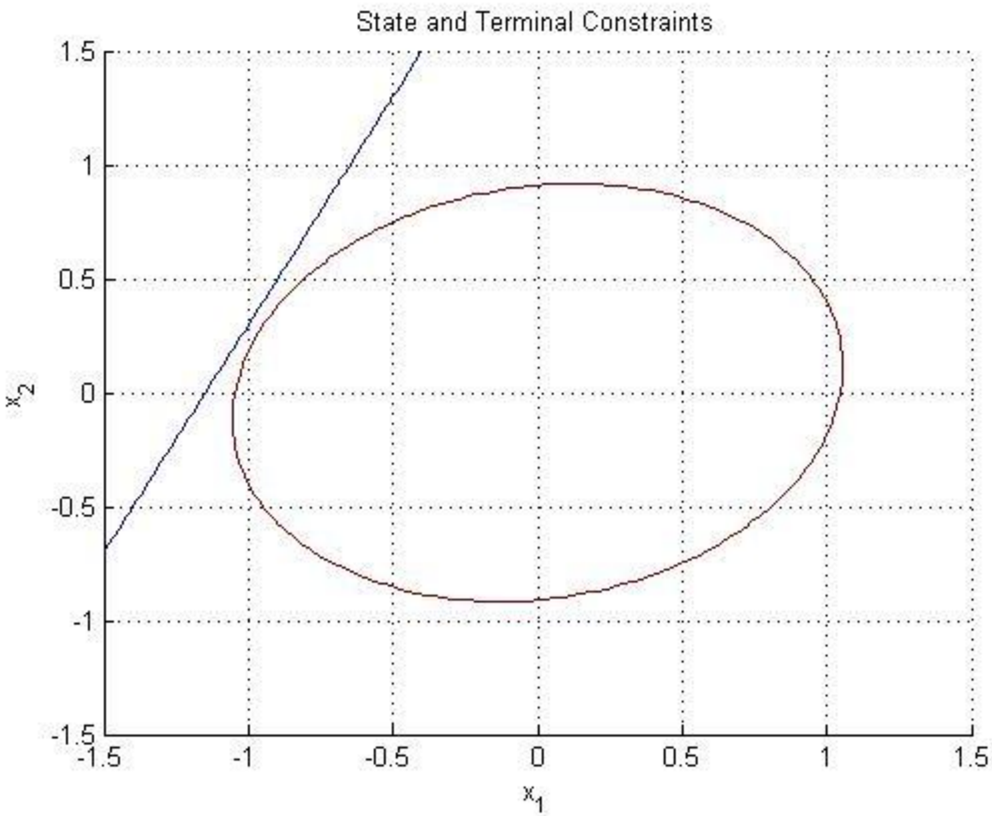
$$E_{x(k)}[x^T(N|k)\Phi x(N|k)] \le 45$$

State Constraint

$$E[-2x_1(k) + x_2(k)] \le 2.3$$

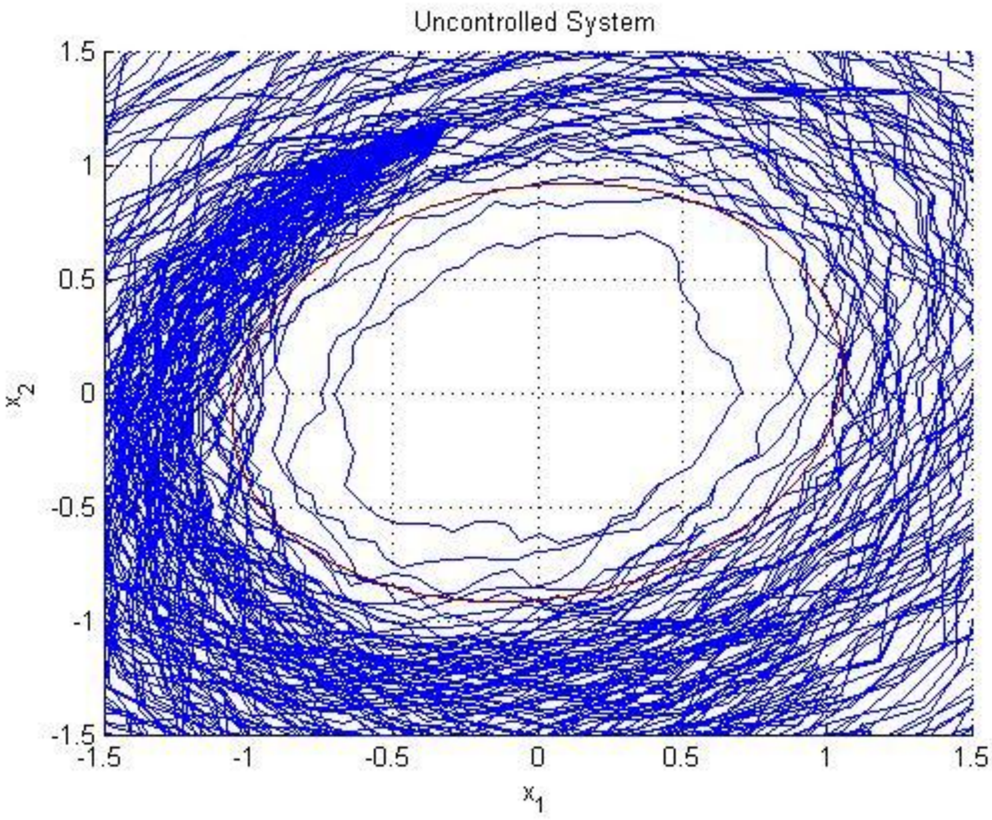
Horizon

$$N = 10$$



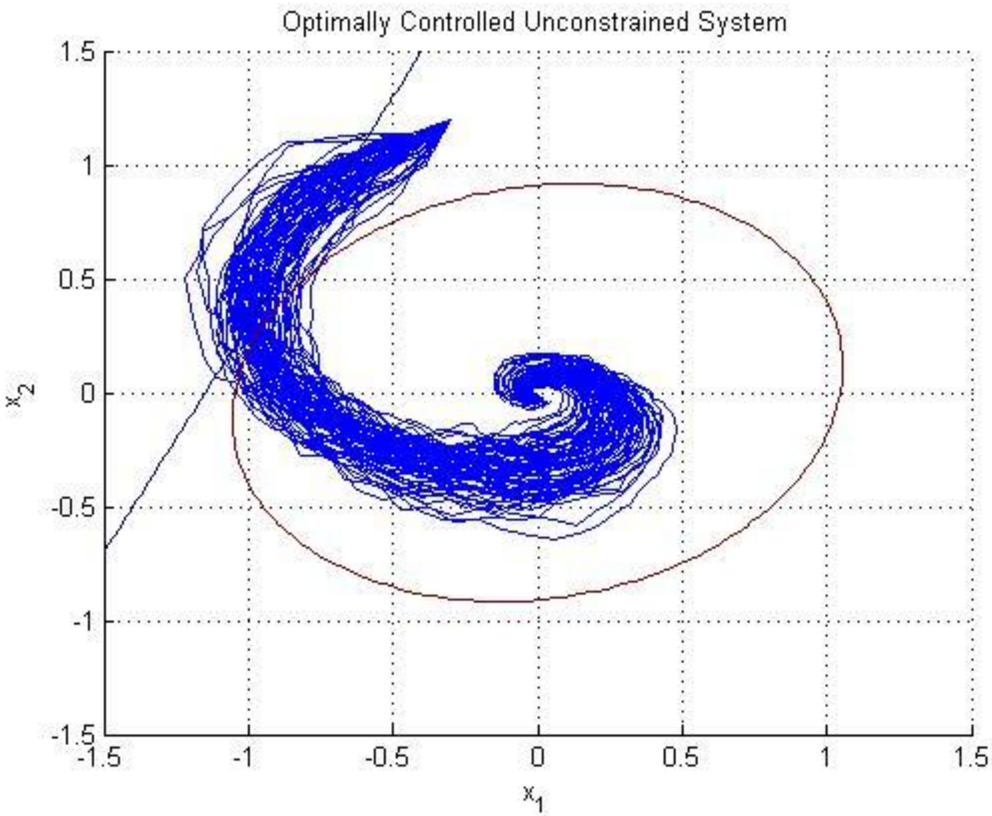
Level Sets of the State and Terminal Constraints.

Uncontrolled Dynamics



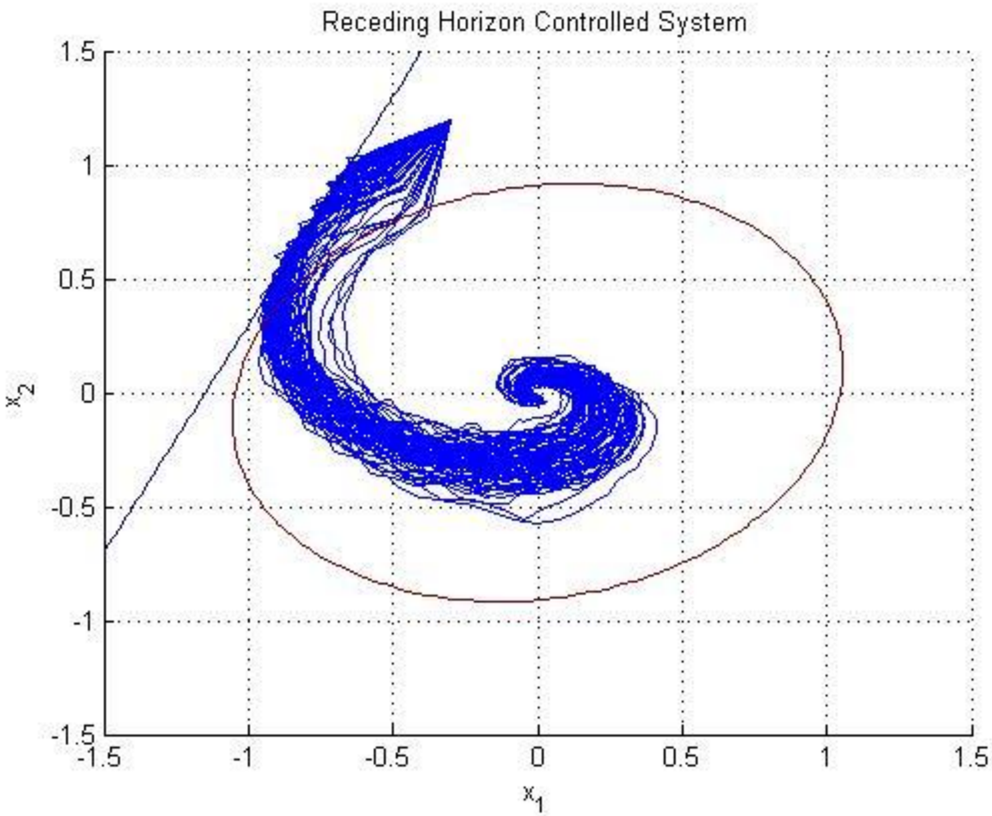
75 Random Simulations from Initial Condition [-0.3,1.2].

Optimal Unconstrained Dynamics



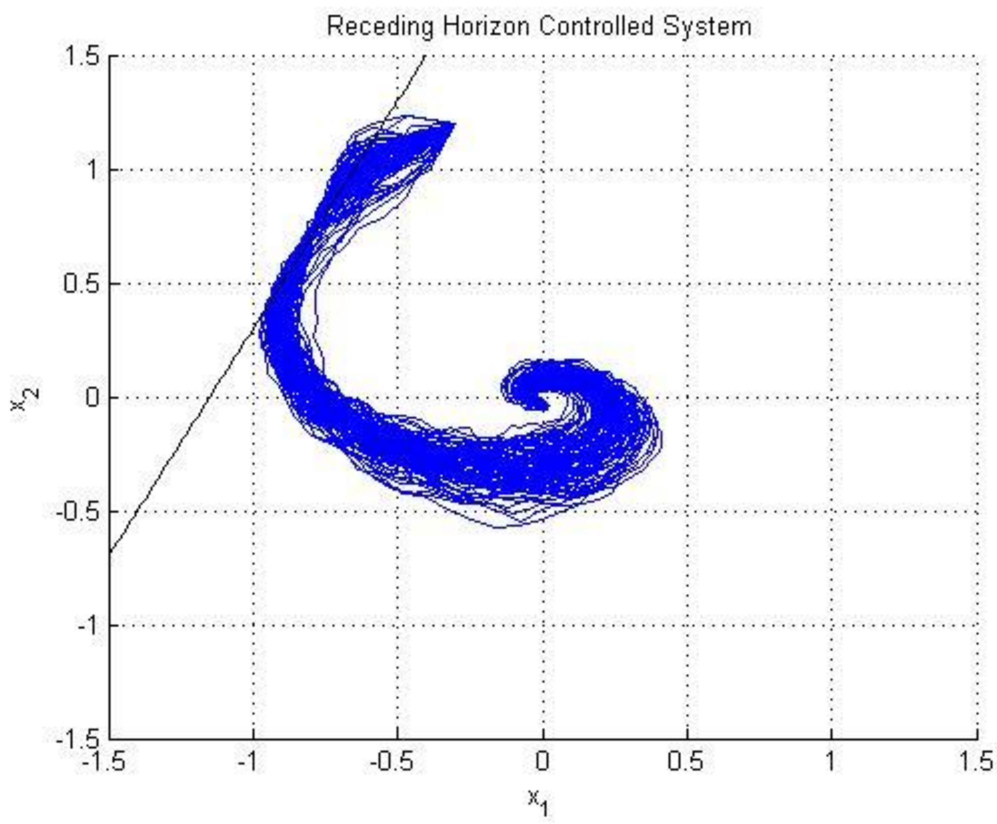
75 Random Simulations from Initial Condition [-0.3,1.2].

RHC Dynamics (Hard Constraint)



75 Random Simulations from Initial Condition [-0.3,1.2].

RHC Dynamics (Soft Constraint)



75 Random Simulations from Initial Condition [-0.3,1.2].

Ex #1: Index Tracking Example:

Five Stocks: IBM, 3M, Altria, Boeing, AIG

Means, variances and covariances estimated from 15 years of weekly data beginning in 1990 from Yahoo! finance.

Risk free rate of 5%

Initial prices and wealth assumed to be $100.

Horizon of $N = 5$ with time step equal to 1 month.

Ex #1: Index Tracking Example:

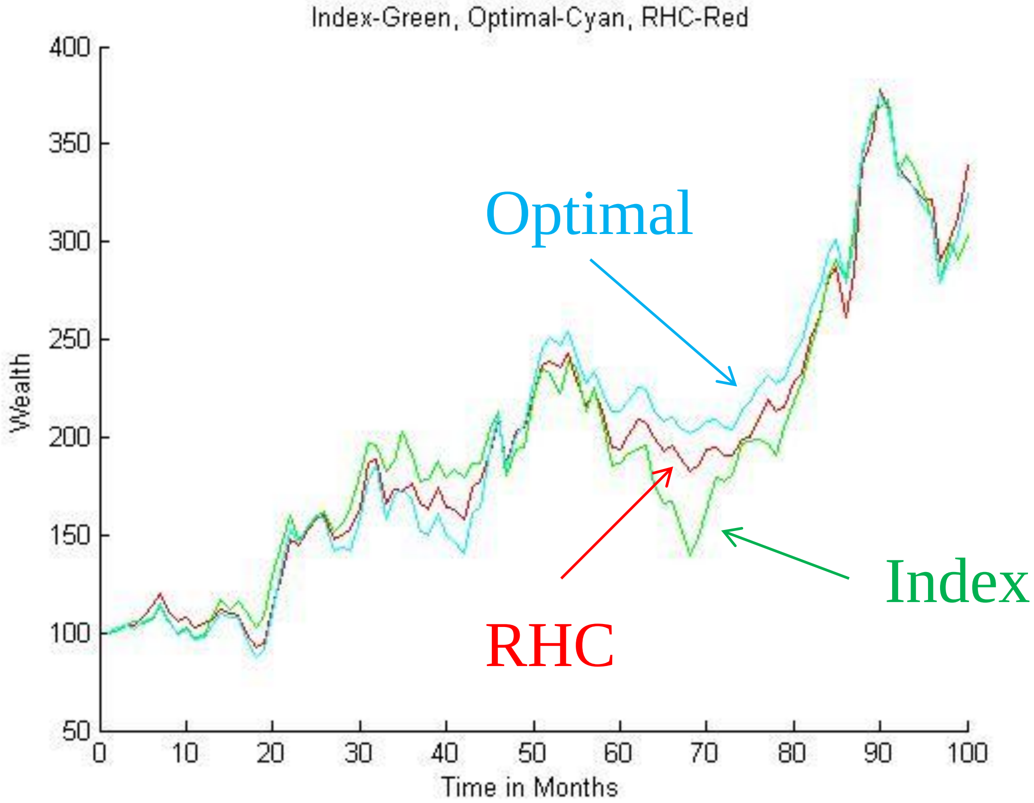
Five Stocks: IBM, 3M, Altria, Boeing, AIG

The index is an equally weighted average of the 5 stocks.

Initial value of the index is assumed to be $100.

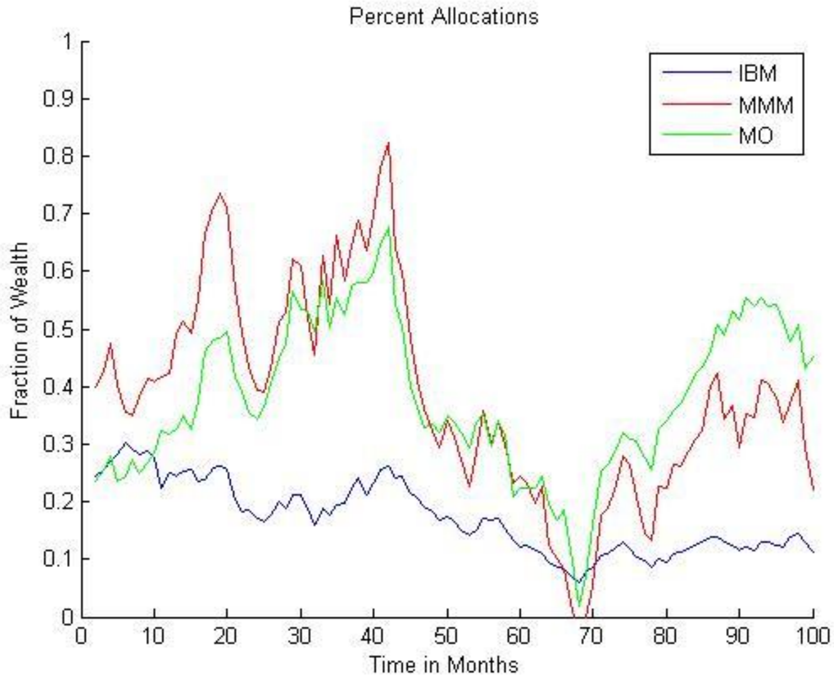
The index will be tracked with a portfolio of the first 3 stocks: IBM, 3M, and Altria.

We place a constraint that the fraction invested in 3M cannot exceed 10%.

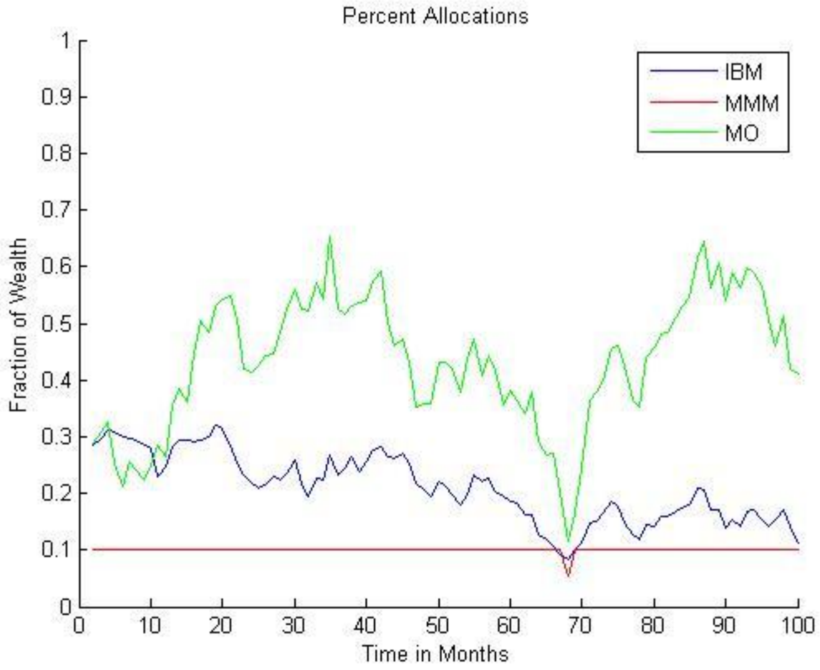


Index – Green, Optimal Unconstrained - Cyan, RHC - Red

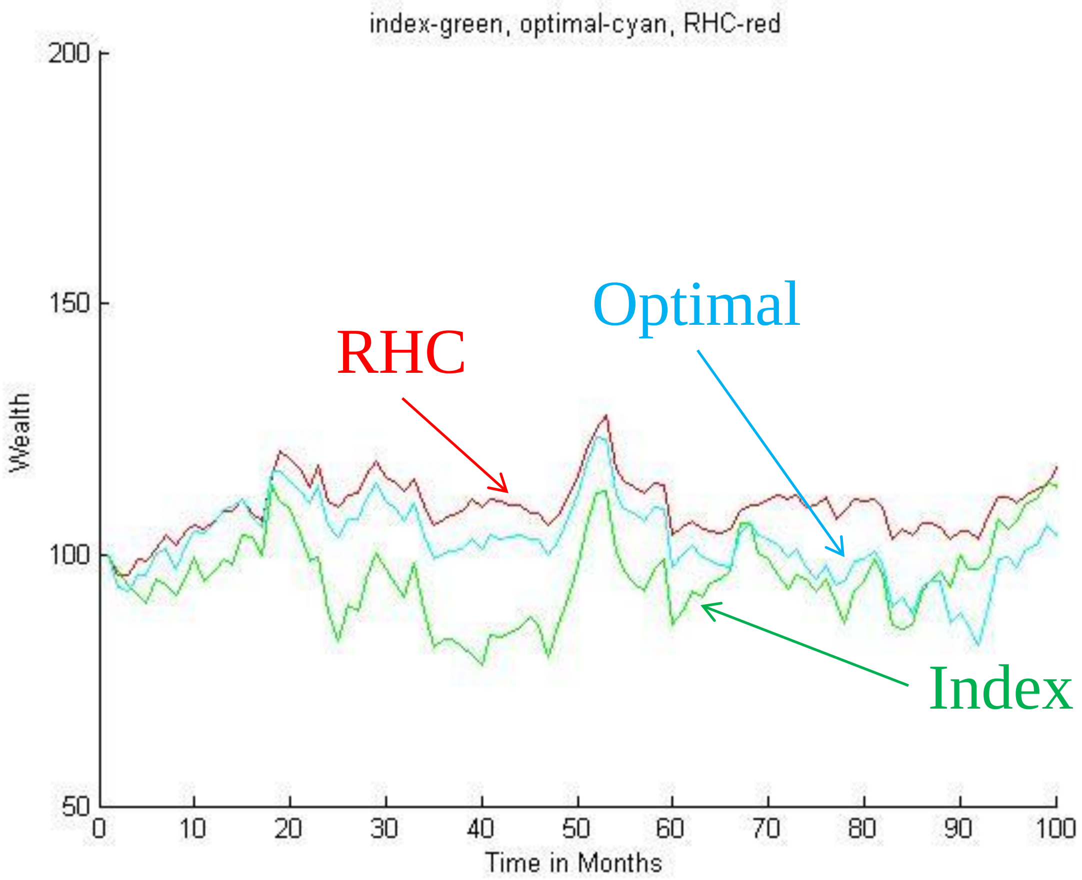
Optimal Unconstrained Allocations



RHC Allocations

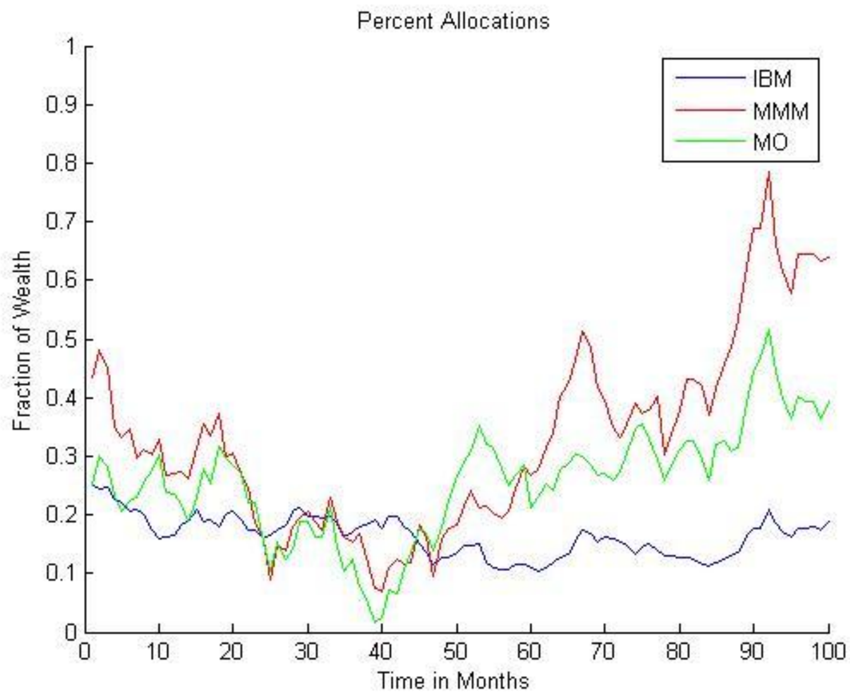


Blue – IBM, Red – 3M, Green - Altria

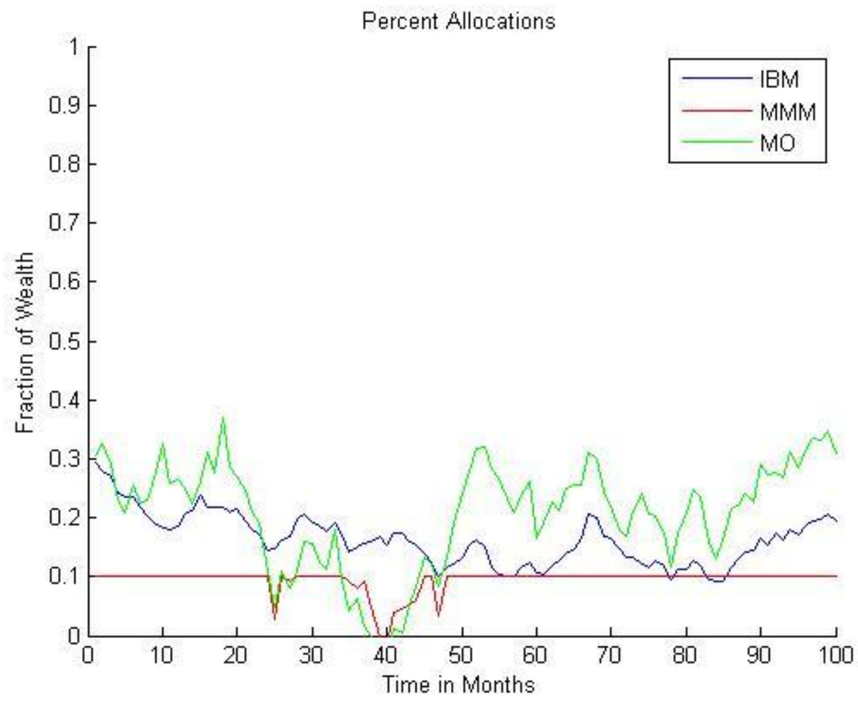


Index – Green, Optimal Unconstrained - Cyan, RHC - Red

Optimal Unconstrained Allocations



RHC Allocations



Blue – IBM, Red – 3M, Green - Altria

Ex #2: Dynamic Hedging

Dynamic Hedging of a European Call Option under Trans. Costs

Underlying asset follows geometric Brownian motion

$$dS = \mu S dt + \sigma S dz$$

$\mu=9.16\%$ and $\sigma=30.66\%$ (estimated from IBM)

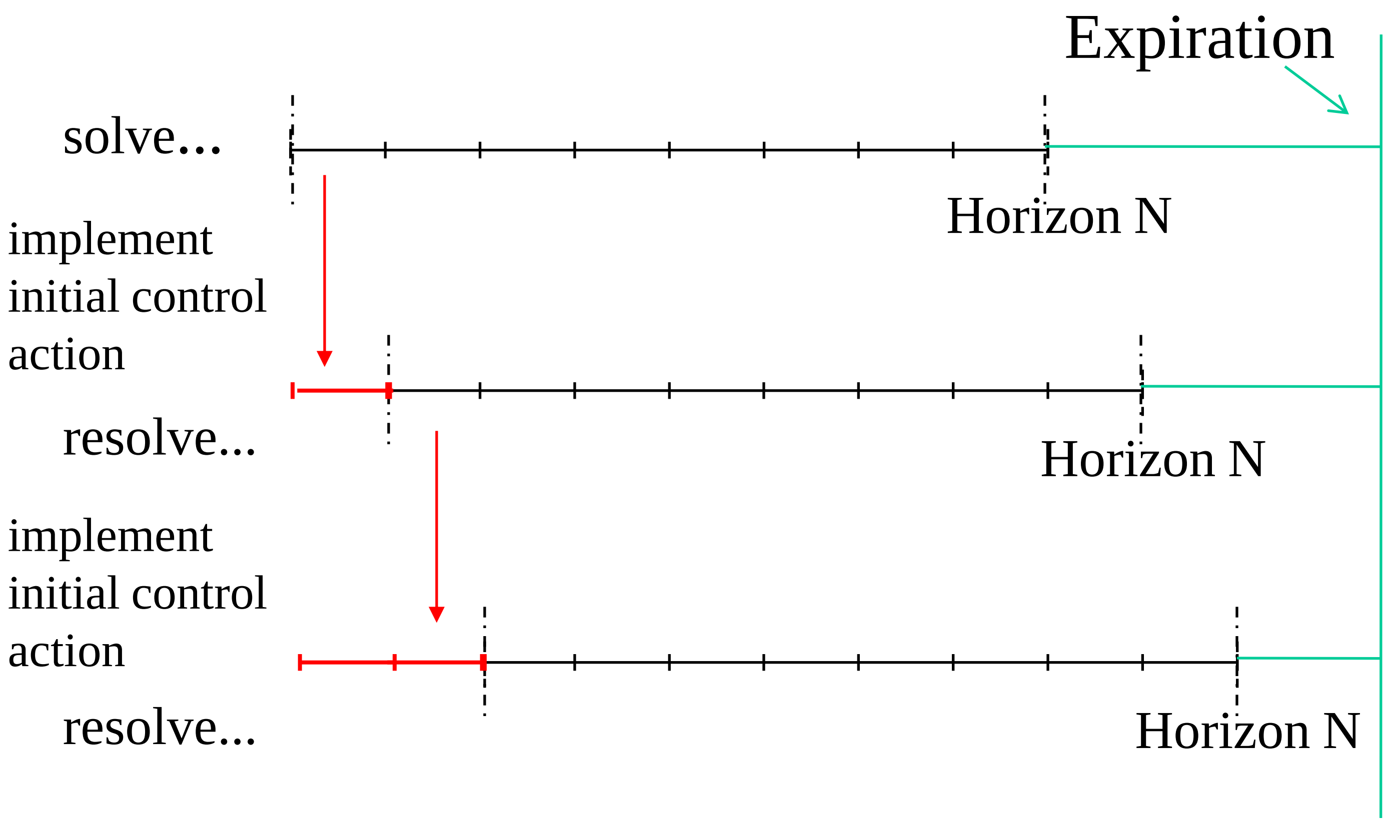
Risk free rate of 5%. Expiration in 2 weeks.

Initial price of stock is \$10. Strike price of \$10.

Initial wealth equal to Black-Scholes price.

RHC horizon of $N = 5$ with time step equal to 1 day.

Different levels of proportional transaction costs.



Expiration
solve...
Horizon N
implement
initial control
action
resolve...
Horizon N
implement
initial control
action
resolve...
Horizon N

To implement transaction costs, adjust wealth dynamics:

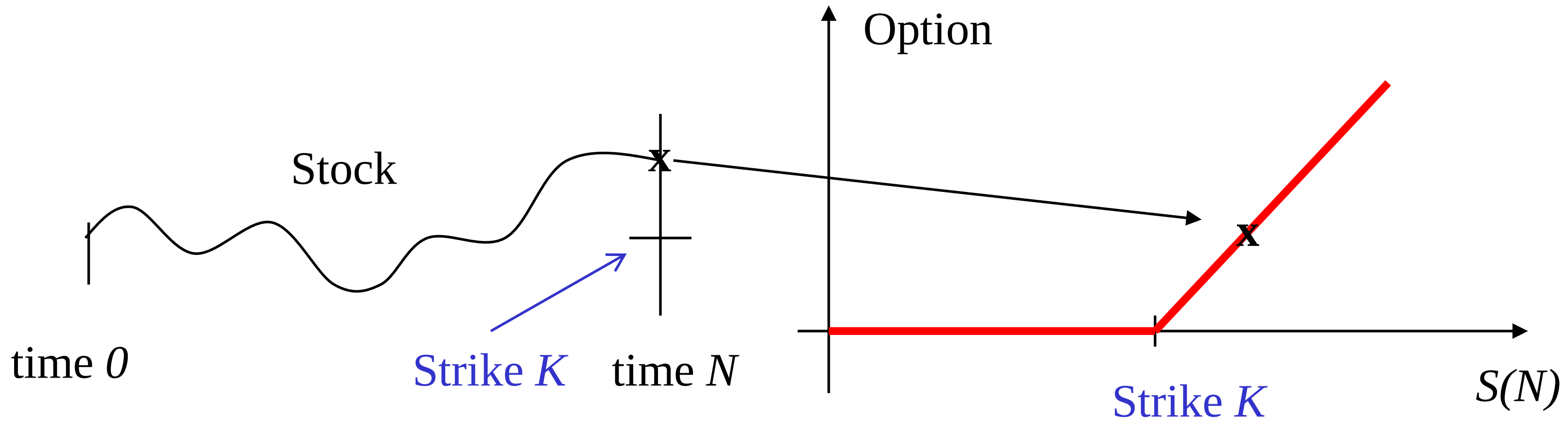
$$W(k+1) = (1+r_f)\left(W(k) - \sum_{i=1}^{l} \tau_i(k)\right) + \sum_{i=1}^{l} \left((\mu_i - r_f) + w_i(k)\right) u_i(k)$$

Transaction costs

where $\tau_i(k) \geq \left| \bar{u}_i(k) - (1+\mu_i)\bar{u}_i(k-1) \right|$

is an approximation of the expected transaction cost.

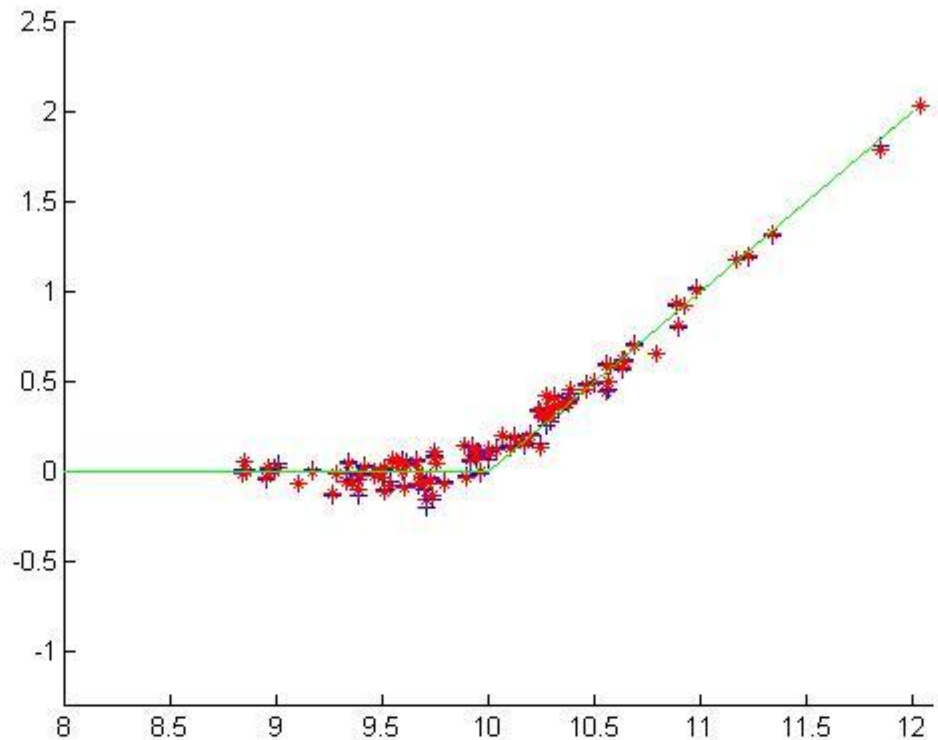
Recall Dynamic Hedging Picture



Black-Scholes★ vs. RHC ✚

0% Transaction Cost

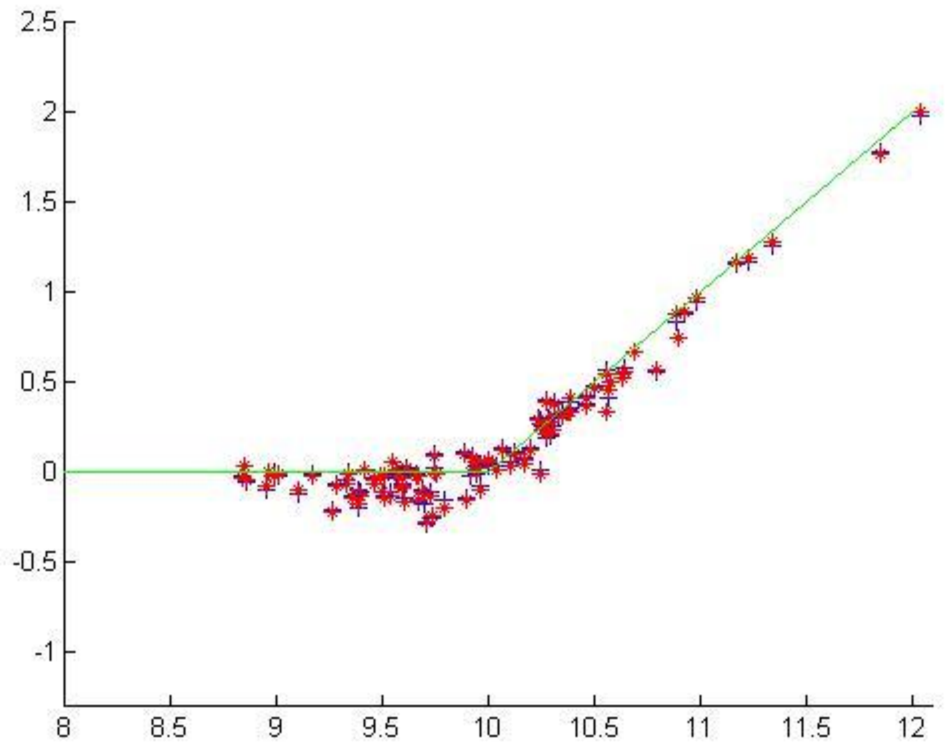
$W(N)$



$S(N)$

Black-Scholes★ vs. RHC ✚

1% Transaction Cost

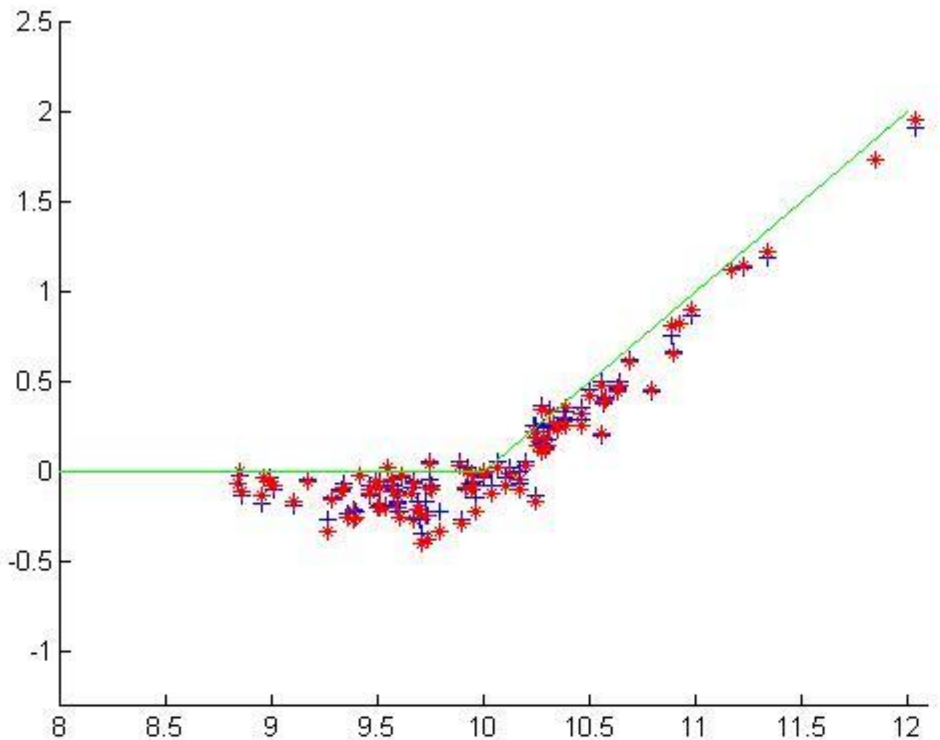


W(N)

S(N)

Black-Scholes★ vs. RHC ✚

2% Transaction Cost

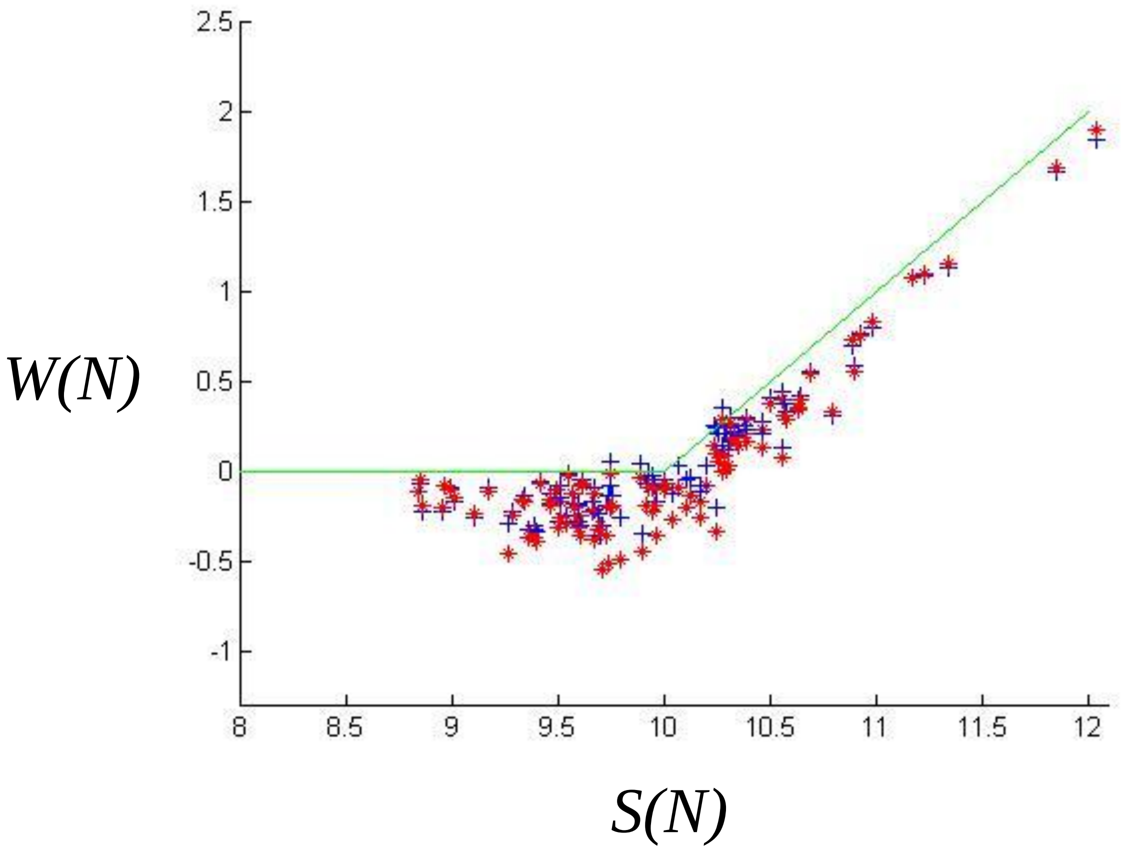


W(N)

S(N)

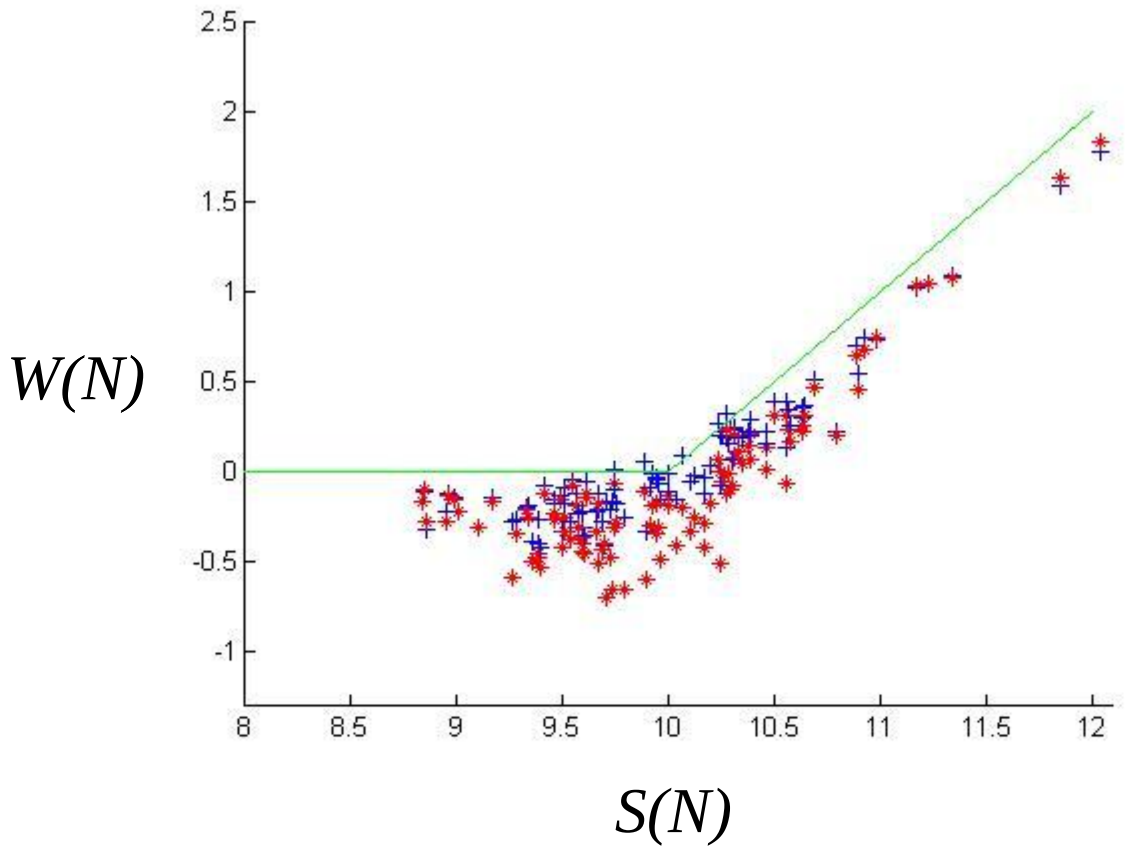
Black-Scholes★ vs. RHC ✚

3% Transaction Cost



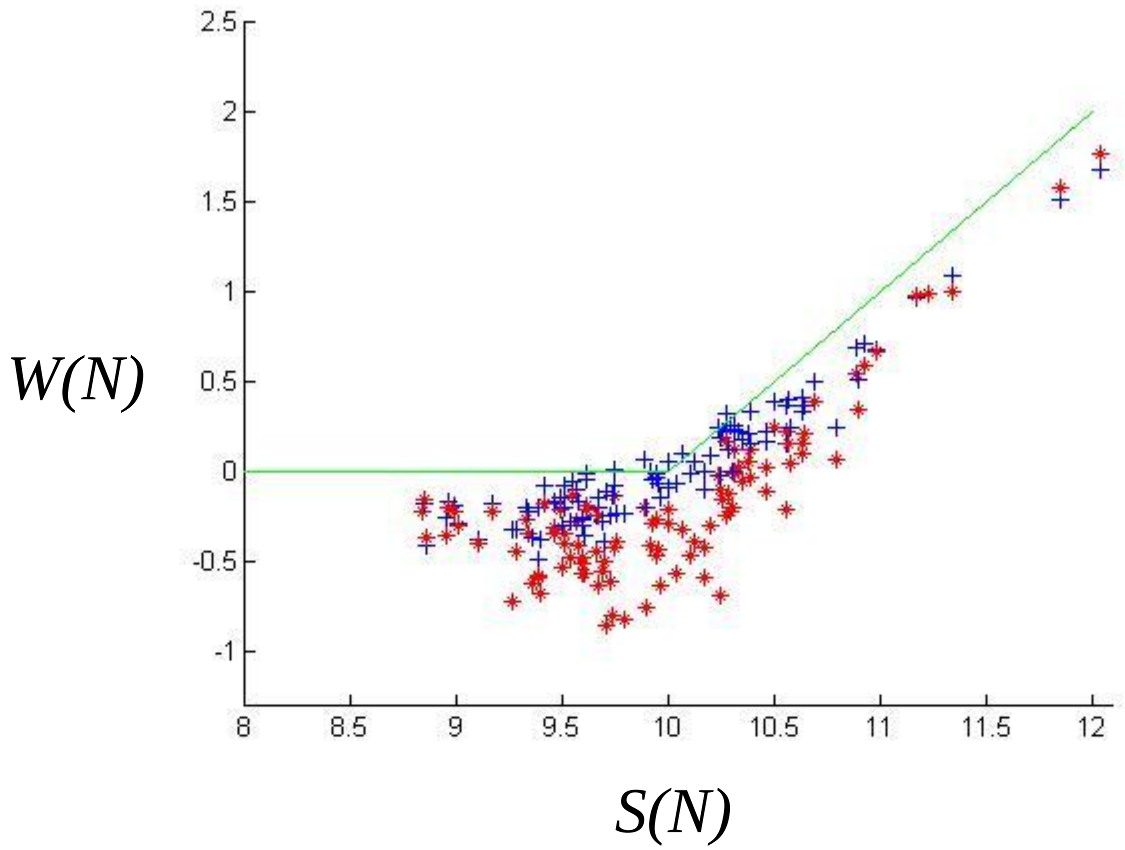
Black-Scholes★ vs. RHC ✚

4% Transaction Cost



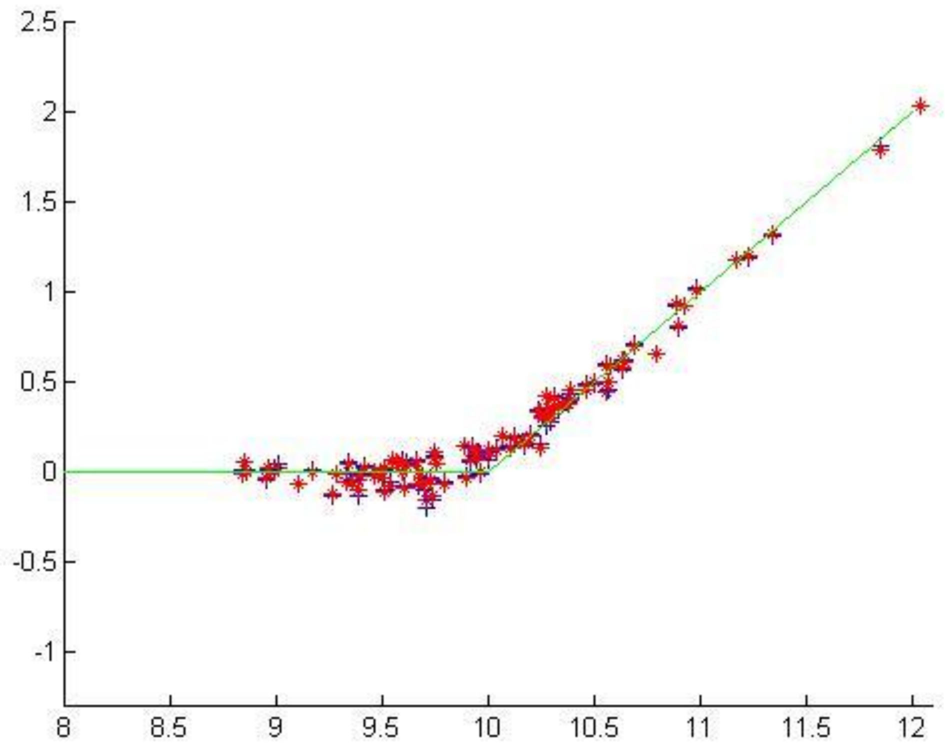
Black-Scholes★ vs. RHC ✚

5% Transaction Cost



Leland★ vs. RHC ✚

0% Transaction Cost

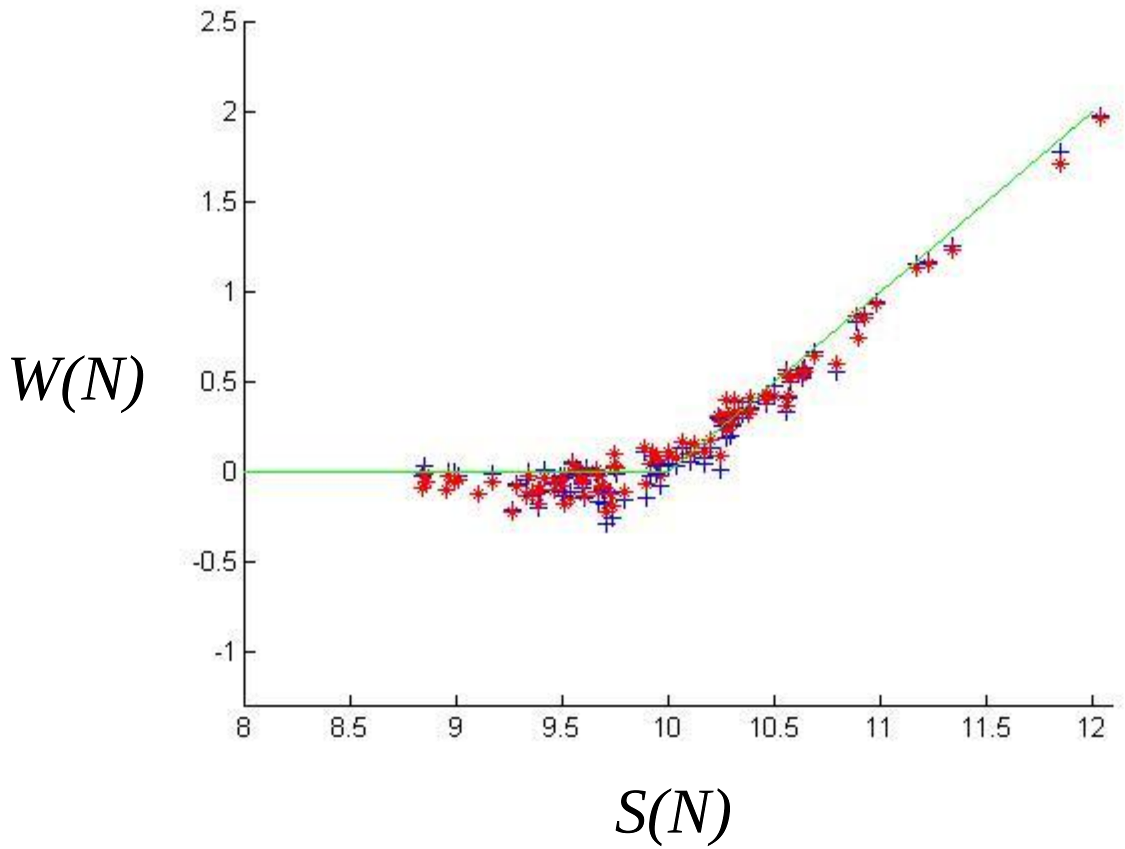


W(N)

S(N)

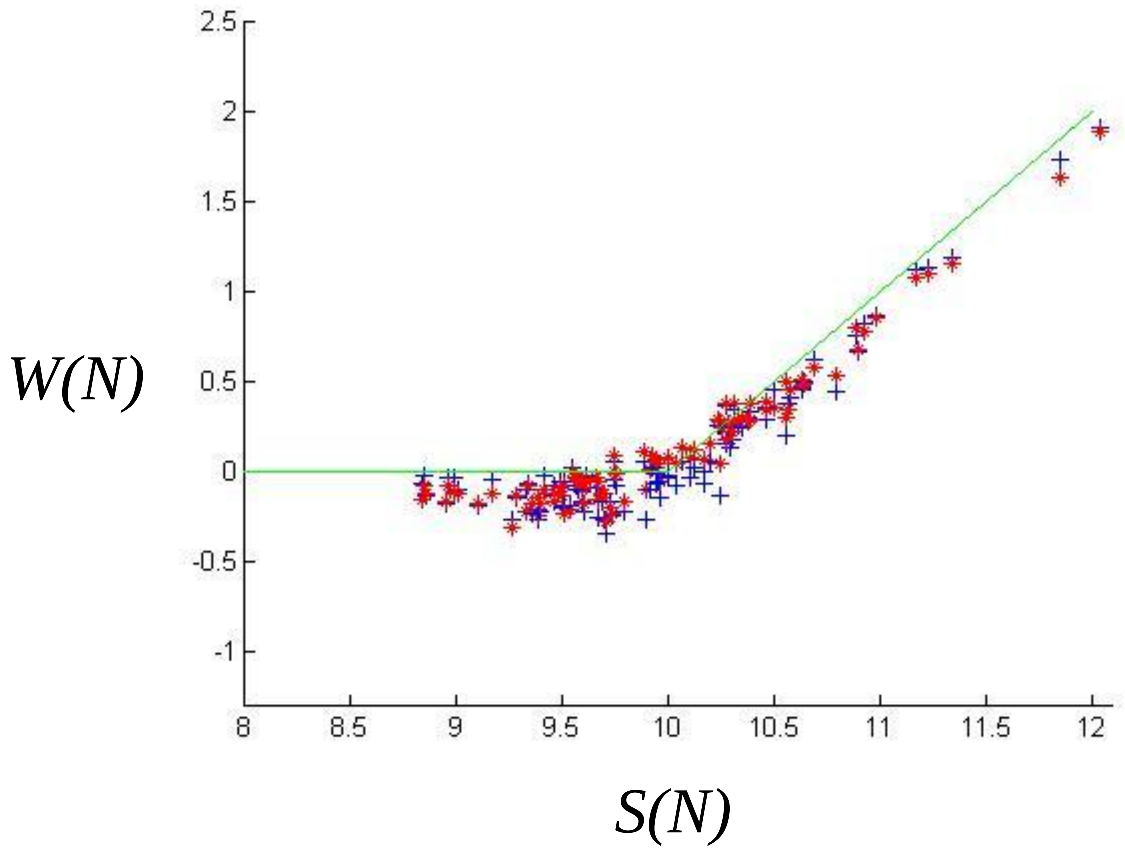
Leland★ vs. RHC ✚

1% Transaction Cost



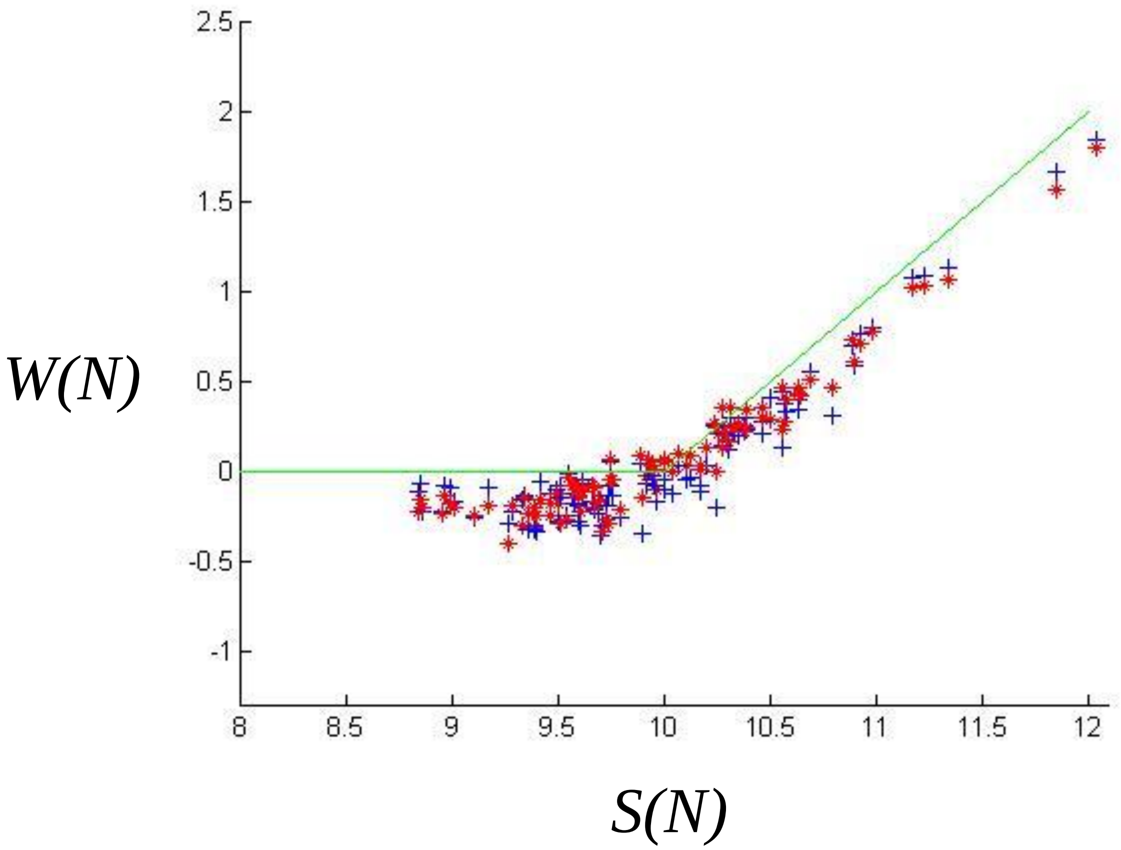
Leland★ vs. RHC ✚

2% Transaction Cost



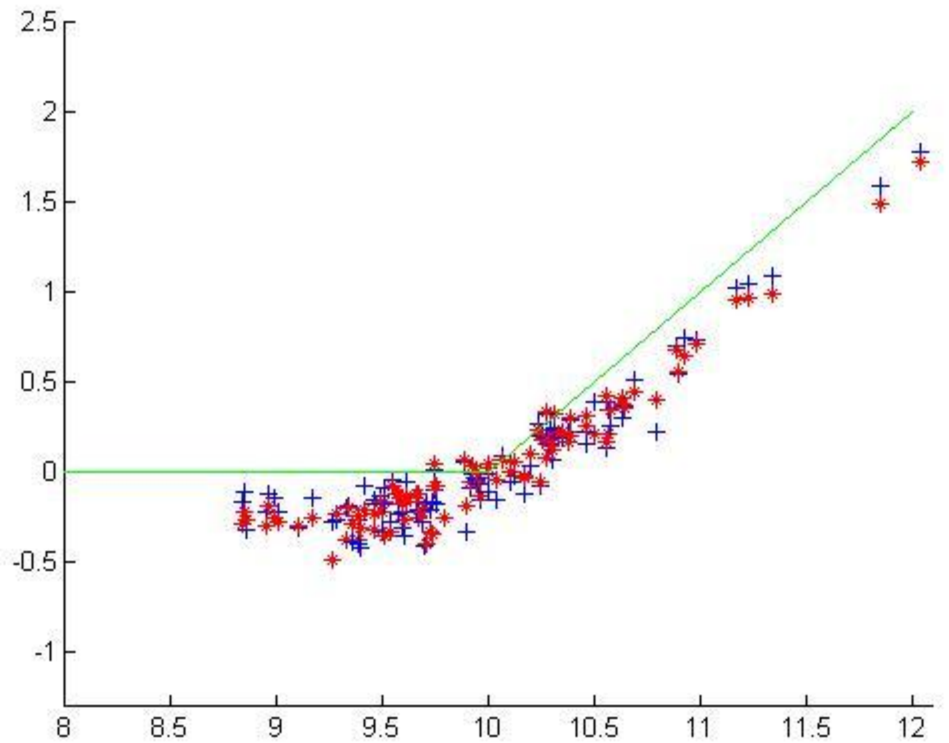
Leland★ vs. RHC ✚

3% Transaction Cost



Leland★ vs. RHC ✚

4% Transaction Cost

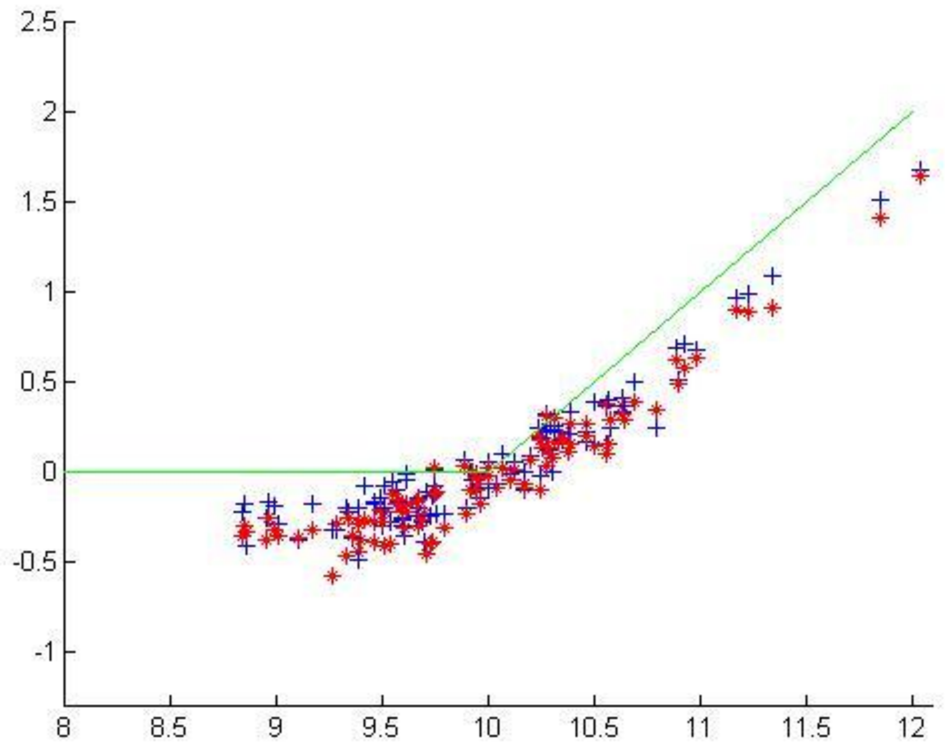


W(N)

S(N)

Leland★ vs. RHC ✚

5% Transaction Cost



W(N)

S(N)

This can easily be applied to…

a 5 dimensional basket option with

transaction costs,

short selling constraints,

restrictions on which assets can be traded,

etc.

Ex #2: Dynamic Hedging of a Basket Option:

Five Stocks: IBM, 3M, Altria, Boeing, AIG

The basket is an equally weighted average of the 5 stocks.

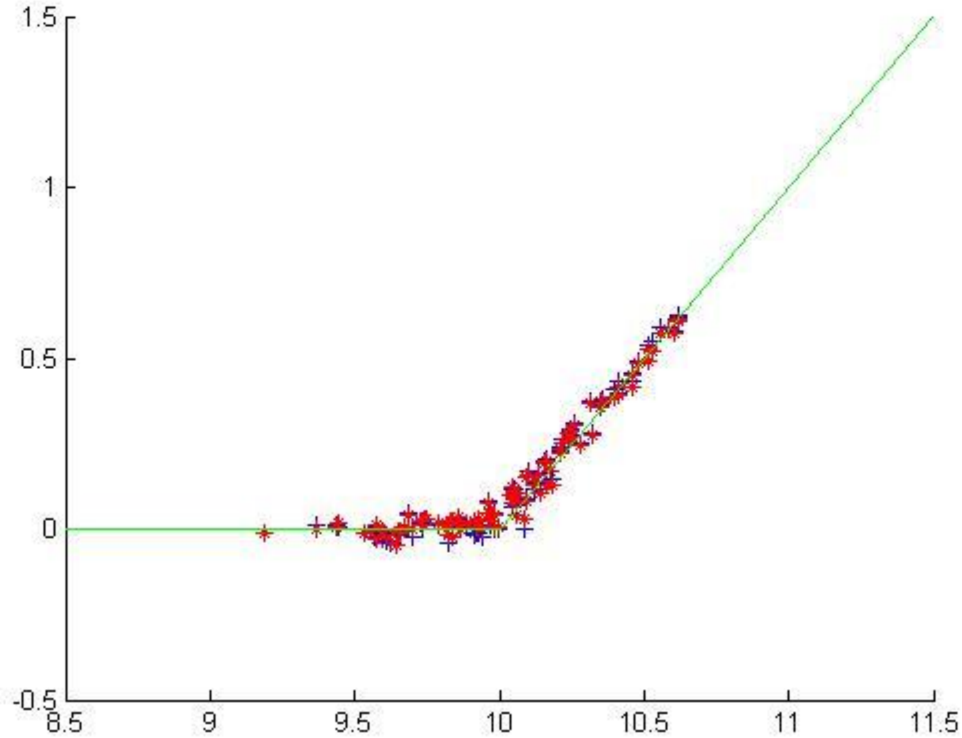
Initial value of all stocks and strike is assumed to be $10.

Initial wealth of hedging portfolio is Method of Moments price (uses first two moments of basket in Black-Scholes formula).

Different levels of proportional transaction costs.

Method of Moments ★ vs. RHC ✚

0% Transaction Cost



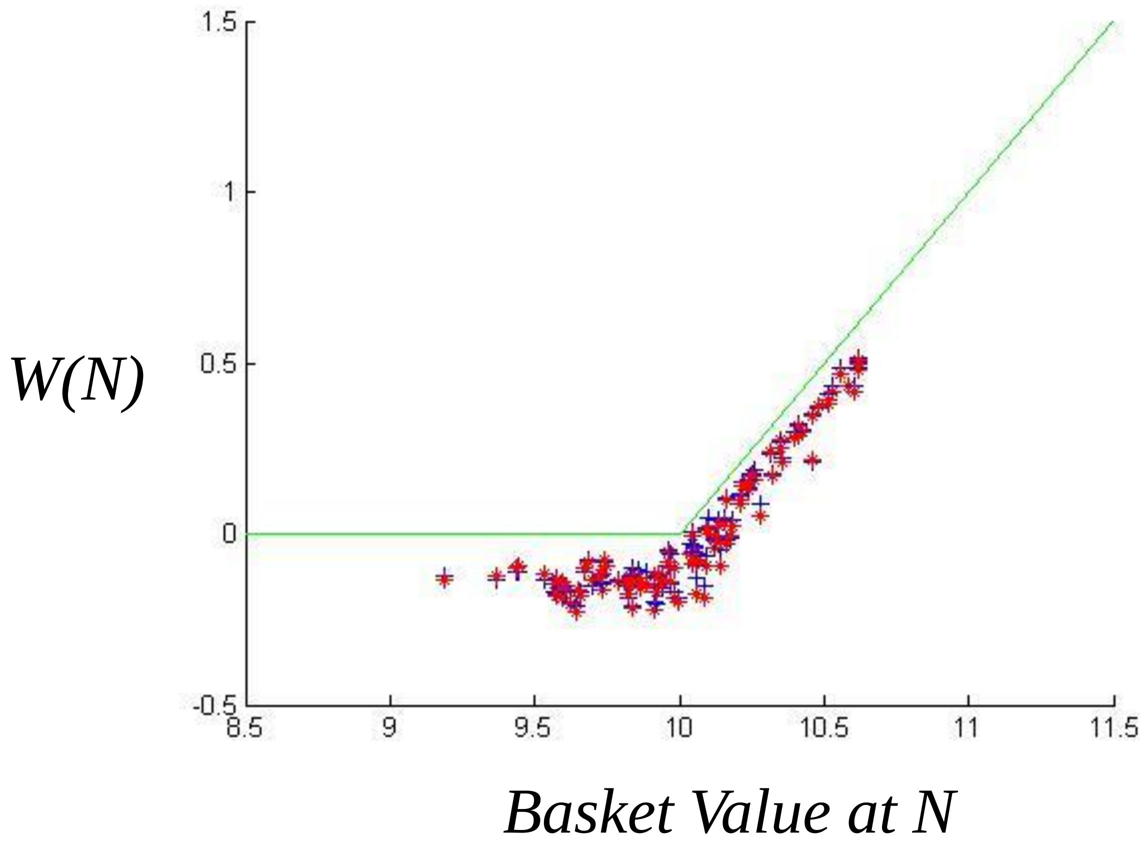
W(N)

Basket Value at N

5 Dimensional Basket Option

Method of Moments ★ vs. RHC ✚

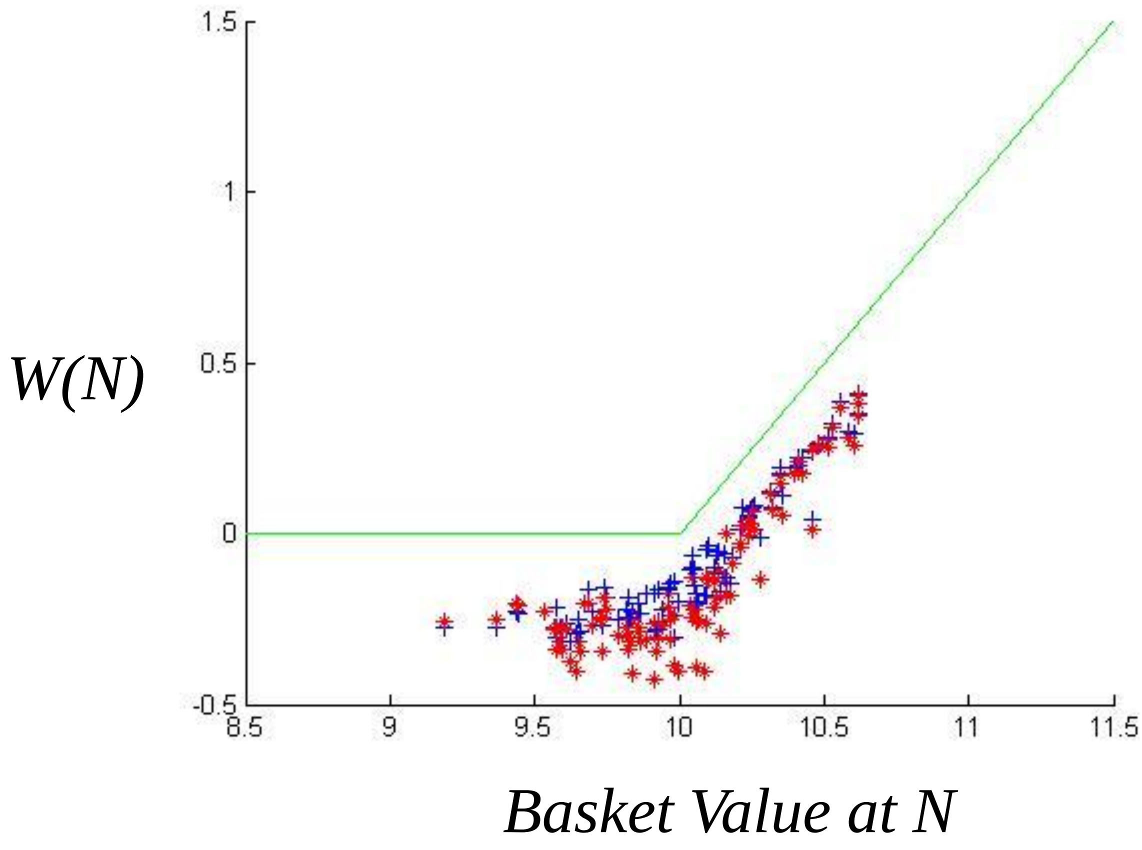
1% Transaction Cost



5 Dimensional Basket Option

Method of Moments ★ vs. RHC ✚

2% Transaction Cost



5 Dimensional Basket Option

Outline

Conclusions

By exploiting and imposing problem structure, constrained stochastic receding horizon control can be implemented in a computationally tractable manner for a number of problems.

Preliminary results suggest that stochastic receding horizon control can address a number of financial engineering problems with success.

As future work, we are looking at formulations to address a wider class of dynamics, and still remain computationally tractable.

References

J. Primbs, “ Portfolio Optimization Applications of Stochastic Receding Horizon Control”, To appear ACC 2007.

J. Primbs, “ Stochastic Receding Horizon Control of Constrained Linear Systems with State and Control Multiplicative Noise”, To appear ACC 2007.

J. Primbs, “ A Soft Constraint Approach to Stochastic Receding Horizon Control”, Submitted to CDC 2007.

J. Primbs, S. Mudchanatongsuk, and W. Wong “ A Receding Horizon Control Formulation of European Basket Option Hedging”, Submitted to FEA2007.