Received: 6 December 2016 | Revised: 29 June 2017 | Accepted: 28 July 2017

DOI: 10.1002/rnc.3915

STOCHASTIC PREDICTIVE CONTROL

WILEY

Dynamic option hedging with transaction costs: A stochastic model predictive control approach

Mogens Graf Plessen[1] | Laura Puglia[1] | Tommaso Gabbriellini[2] | Alberto Bemporad[1]

[1]IMT School for Advanced Studies Lucca, Lucca, Italy

[2]Modelling & Quantitative Analysis, MPS Capital Services, Siena, Italy

Correspondence
Alberto Bemporad, IMT School for Advanced Studies Lucca, 55100 Lucca, Italy.
Email: alberto.bemporad@imtlucca.it

Summary

This paper proposes stochastic model predictive control as a tool for hedging derivative contracts (such as plain vanilla and exotic options) in the presence of transaction costs. The methodology combines stochastic scenario generation for the prediction of asset prices at the next rebalancing interval with the minimization of a stochastic measure of the predicted hedging error. We consider 3 different measures to minimize in order to optimally rebalance the replicating portfolio: a trade-off between variance and expected value of hedging error, conditional value at risk, and the largest predicted hedging error. The resulting optimization problems require solving at each trading instant a quadratic program, a linear program, and a (smaller-scale) linear program, respectively. These can be combined with 3 different scenario generation schemes: the lognormal stock model with parameters recursively identified from data, an identification method based on support vector regression, and a simpler scheme based on perturbation noise. The hedging performance obtained by the proposed stochastic model predictive control strategies is illustrated on real-world data drawn from the NASDAQ-100 composite, evaluated for a European call and a barrier option, and compared with delta hedging.

KEYWORDS

financial options, hedging techniques, scenario generation, stochastic model predictive control, stochastic programming, transaction costs

1 | INTRODUCTION

For a financial institution, hedging a derivative contract implies to dynamically rebalance a (self-financing) portfolio of underlying assets at periodic intervals so that, at the expiration date of the contract, the value of the portfolio is as close as possible to the payoff value to pay to the customer. In contrast, static hedging strategies do not involve rebalancing; thus, a replicating portfolio is formed initially and simply let evolve freely for the whole option life. For a general background on options and derivative contracts, see the work of Hull for instance.[1]

The most common derivative contracts are *plain vanilla* options: a European *call (put) option* gives the holder the right to buy (sell) the underlying asset at a given *expiration date* and at a determined *strike price*. A large number of other more complex derivative contracts, called *exotic* options, are nowadays traded, especially in the *over-the-counter* market. An example of an exotic option is the *barrier option*, a special kind of plain vanilla contract whose payoff drops to zero as soon as the price of the underlying asset reaches a certain barrier value. See the work of Graf Plessen and Bemporad[2] for optimization-based combinations of options.

The most common approach used in practice to dynamically rebalance the portfolio replicating the option is *delta hedging* (Δ-hedging), which was directly derived from the fundamental theory proposed by Black and Scholes.[3] In delta hedging, the replicating portfolio includes a cash position in the money market account and a quantity of stocks equal to the derivative of the option price with respect to the price of the underlying stock. The guiding notion of Δ-hedging is to make the portfolio insensitive to the stochastic evolution of the price of the underlying asset. In control theoretical words, this is equivalent to making the wealth of the portfolio tracking the option price while rejecting the disturbance induced by price fluctuations. Within the Black-Scholes theory, Δ-hedging makes the following (often unrealistic) assumptions: the underlying price follows the *lognormal* stock model, continuous-time hedging, static volatility, and the *absence of transaction costs*.

To handle transaction costs, Fedotov and Mikhailov[4] and Gondzio et al[5] proposed analytic methods based on stochastic optimization. In the work of Fedotov and Mikhailov,[4] the option price and the optimal trading strategy are jointly determined to reduce the total risk of writing the option. In the work of Gondzio et al,[5] a trinomial process was used for generating the scenarios required to set up a stochastic control problem in which the objective function is the expected value of a given performance index. To cope with transaction costs, in the work of Primbs,[6] the hedging problem was formulated as a linear quadratic regulation problem penalizing transaction costs in the objective function. As an alternative, a model predictive control approach is proposed to solve a quadratic program (QP) over a specified horizon, exploiting the linear quadratic regulation solution from the first approach in the cost function. In the work of Primbs,[7] transaction costs were taken into account in a finite-horizon constrained stochastic control problem formulation that is iteratively solved at each trading date by employing a semidefinite programming algorithm. Related ideas proposing the use of model predictive control for replicating portfolios appeared earlier in the works Dombrovskii et al[8] and Herzog et al.[9]

For the case *without* transaction costs, stochastic model predictive control (SMPC) approaches were proposed in the works of Bemporad et al.[10,11] Stochastic model predictive control can be seen as a suboptimal way of solving a stochastic multistage dynamic programming problem. Rather than solving the problem for the entire remaining time span of the option life, a smaller problem is solved repeatedly from the current time step t up to a certain number N of time steps in the future by suitably remapping the condition at the future expiration date into a value at the predicted time step $t + N$. Formulating the stochastic optimization problem requires enumerating a certain number of scenarios of future stock prices. A suitable stock price model is often not known a priori, and its parameters must be identified from data.

In this paper, we propose SMPC to solve dynamical option hedging problems with transaction costs. We consider different performance measures (a trade-off between variance and expected value of hedging error, conditional value at risk (CVaR), and the largest predicted hedging error) and show how the corresponding optimization problems can be easily solved via either quadratic or linear programming. A preliminary version of this work was presented in the conference paper of Bemporad et al,[12] which we largely extend here by considering real-world data in our results drawn from the NASDAQ-100 composite and by proposing suitable scenario generation schemes to construct the stochastic optimization problems.

This paper is structured as follows. In Section 2, we formulate the SMPC problem for option hedging based on enumeration of scenarios. In Section 3, we define transaction costs and describe how they affect the evolution of the portfolio. After formulating the SMPC problem, we focus on proportional transaction costs and propose the 3 different optimization strategies. In Section 4, we discuss 3 data-driven scenario generation methods for one-step-ahead stock and corresponding option price predictions. Simulation tests on real-world data are reported in Section 5 for a European call and a barrier option. Some concluding remarks are given in Section 6.

2 | DYNAMIC OPTION HEDGING

Consider the problem of hedging an option $\mathcal{O}$ defined over n underlying assets. We denote the time interval between 2 consecutive trading dates (the results of this paper can be easily generalized to nonuniform trading intervals T_s) by T_s; the trading instants, $t = 0, 1, \ldots, T$, by t; and the vector of spot prices of the assets by $s(t) = [s_1(t) \ldots s_n(t)]' \in \mathbb{R}^n$.

In general, the option price $p(t)$ of $\mathcal{O}$ at a generic instant t is the discounted expectation of the payoff $\mathcal{P}(m(T))$ at the expiration date in the risk-neutral measure, given the market state $m(t)$ at time t ($m(t) = s(t)$ for plain vanilla options). Denoting by T the maturity of $\mathcal{O}$ in terms of the number of sampling steps of duration T_s, the price of the hedged option at a generic intermediate date tT_s is $p(t) = (1 + r)^{t-N}\mathrm{E}\,[\mathcal{P}(m(T))|m(t)]$, where $\mathrm{E}[p(T)]$ is the expected value of the payoff in the risk-neutral measure. For European call options, the payoff is

$$\mathcal{P}\left(m(T)\right) = p(T) = \max\left\{s(T) - K, 0\right\}, \tag{1}$$

whereas for barrier options, it is

$$p(T) = \begin{cases} \max\left(s(T)-K,0\right), & \text{if} \quad s(t) < s_u, \forall t \le T \\ 0, & \text{otherwise} \end{cases}$$
$$= \begin{cases} \max\left(s(T)-K,0\right), & \text{if} \quad s_\ell(t) = 0 \\ 0, & \text{if} \quad s_\ell(t) = 1, \end{cases} \tag{2}$$

where s_u defines the upper barrier level and $s_\ell(t) \in \{0,1\}$ is a logic state with dynamics $s_\ell(t+1) = s_\ell(t)$ OR $[s(t) \ge s_u]$, $s_\ell(0) = 0$ (in this case, $m(t) = \{s(t), s_\ell(t)\}$).

Assume that there are no transaction costs and the standard *self-financing* condition holds, ie, the wealth $w(t)$ of the portfolio replicating option $\mathcal{O}$ is always totally reinvested. Then, the dynamics of the wealth $w(t)$ of the portfolio is

$$w(t+1) = (1+r)w(t) + \sum_{i=1}^{n} b_i(t)u_i(t), \tag{3}$$

where $u_i(t)$ is the quantity of asset i held at time t and $b_i(t) \triangleq s_i(t+1) - (1+r)s_i(t)$ is the excess return, ie, how much the asset gains (or loses) with respect to the risk-free rate. The initial condition $w(0)$ is the set equal to the price paid by the customer to purchase option $\mathcal{O}$, $w(0) = (1+r)^{-N}\mathrm{E}[p(T)|m(0)]$.

Dynamic hedging aims at making the final wealth $w(T)$ as close as possible to $p(T)$ for all possible market realizations. The hedging problem can be restated as a stochastic control problem. Using control systems jargon, wealth $w(t) \in \mathbb{R}$ represents the *state"* and the *regulated output* of the controlled process, the traded asset quantities $u(t) \in \mathbb{R}^n$ are the *inputs*, and the option price $p(t)$ is the *reference* for $w(t)$. By defining the *tracking error* $e(t) \triangleq w(t) - p(t)$, the objective can be restated as the one of minimizing $e(t)$ for all possible asset price realizations.

As shown in the works of Bemporad et al,[10,11] in the absence of transaction costs and under the lack of arbitrage, a way to achieve this is to minimize the variance of the hedging error

$$J\left(e(T)\right) = \mathrm{E}\left[\left(e(T) - \mathrm{E}\left[e(T)\right]\right)^2\right] \tag{4}$$

by solving the one-step-ahead minimum variance problem

$$\min_{\{u(t)\}} \quad \mathrm{Var}_{m(t+1)}\left[w\left(t+1, m(t+1)\right) - p\left(t+1, m(t+1)\right)\right] \tag{5a}$$

$$\text{s.t.} \quad w\left(t+1, m(t+1)\right) = (1+r)w(t) + \sum_{i=0}^{n} b_i\left(t, m(t+1)\right)u_i(t) \tag{5b}$$

with respect to the portfolio composition $u(t)$ at each trading date tT_s.

Note that expectations and variances are conditioned to the particular market realization $m(t)$ at time t; we omit here the conditional notation for simplicity and use the notation $w(t+1)$ from now on as a shortcut for the future wealth $w(t+1, m(t+1))$.

The formulation in Equation 5 is equivalent to an SMPC formulation with prediction horizon $N = 1$ under the terminal condition of perfect hedging between the prediction step $t+N$ and expiration step T. Problem (5) can be solved by enumerating a number M of scenarios, each one corresponding to a different realization of a certain sequence of prices, and optimizing the resulting sample variance. Each scenario j has a probability π_j of occurring, $j = 1, \ldots, M$, $\pi_j > 0$, $\pi_j \le 1$, $\sum_{k=1}^{M} \pi_j = 1$. Scenarios can be generated via Monte Carlo (MC) simulation,[10] where $\pi^j = \frac{1}{M}$, or by discretizing a given probability density function that describes the disturbance process generating the asset prices.[11] Note that, contrarily to multistage stochastic programming approaches that typically limit the number M of considered scenarios to only 2 or 3 to avoid the combinatorial explosion over the optimization horizon N, here M can be quite large without incurring into prohibitive computation efforts, as the prediction horizon is simply $N = 1$.

By optimizing the sample variance of $w(t+1)-p(t+1)$, in the absence of transaction costs, problem (5) can be rewritten as the following *least-squares* problem:

$$\min_{u(t)} \sum_{j=1}^{M} \pi^j \left(w^j(t+1) - p^j(t+1) - \left(\frac{1}{M} \sum_{i=1}^{M} w^i(t+1) - p^i(t+1) \right) \right)^2, \tag{6}$$

where $w^j(t+1) = (1+r)w(t) + \sum_{i=0}^{n} b_i^j(t)u_i(t)$ are the future values of portfolio wealth for each scenario $j = 1, \ldots, M$ and π^j is the corresponding probability, $\pi^j \ge 0$, $\sum_{i=1}^{M} \pi^j = 1$. The resulting SMPC algorithm is described by Algorithm 1.

Algorithm 1 SMPC algorithm for dynamic option hedging

1: Let t = current hedging date, $w(t)$ = current wealth of portfolio, $m(t)$ = current market state;
2: Generate M scenarios of future market states $m^1(t+1), \ldots, m^M(t+1)$, with corresponding probabilities $\pi^1, \ldots, \pi^M$;
3: Use a pricing engine to generate the corresponding future option prices $p^1(t+1), \ldots, p^M(t+1)$;
4: Solve the least-squares problem (6) to minimize the sample variance of $w(t+1) - p(t+1)$;
5: Rebalance the portfolio according of the optimal solution $u^*(t)$ of problem (6);
6: End.

An option pricing engine is needed to compute the future option prices $p^1(t+1), \ldots, p^M(t+1)$ over the generated scenarios. Unless simple analytical formulas for determining the option prices are available, this is the most time-consuming operation of the entire algorithm. In fact, numerical pricing engines must be used based on either MC simulation or other approximate methods such as the method described by Longstaff and Schwartz.[13] See the works of Bemporad et al[10,11] for a comparison of different pricing methods. In particular, Bemporad et al[11] showed that SMPC is superior to Δ-hedging when dealing with exotic options and quite robust also to errors in the dynamical model of the market.

3 | TRANSACTION COSTS

When trading assets on the market, one often suffers from friction due to transaction costs.[14] In mathematical terms, the investor pays a quantity $h_i(t)$ of wealth to change the number of assets in the portfolio from $u_i(t-1)$ at time $t-1$ to $u(t)$ at time t for each asset i. Such wealth $h_i(t)$ is taken away from the overall wealth $w(t)$ of the portfolio so that Equation 3 becomes (compare the work of Primbs and Yamada[15])

$$w(t+1) = (1+r)\left(w(t) - \sum_{i=1}^{n} h_i(t)\right) + \sum_{i=1}^{n} b_i(t)u_i(t). \tag{7}$$

In the simplest case, transaction costs $h_i(t)$ are proportional to the traded quantity of stock $|u_i(t) - u_i(t-1)|$

$$h_i(t) = \epsilon_i \, |u_i(t) - u_i(t-1)| \, s_i(t), \tag{8}$$

where the fixed quantity ϵ_i depends on commissions on trading asset i, $i = 1, \ldots, n$ (we assume that no costs are applied on transacting the risk-free asset).

Proposition 1. *The variance of the hedging error $e(t) = w(t) - p(t)$ is not affected by transaction costs.*

Proof. Let $\omega(t) = \sum_{i=1}^{n} h_i(t)$ be the total transaction cost paid at time t, which is a function of $u(t)$, $u(t-1)$, and $s(t)$. As $\omega(t)$ clearly does not depend on $s(t+1)$, the expected value of the hedging error $e(t+1) = w(t+1) - p(t+1)$ taken with respect to $s(t+1)$ is

$$\begin{aligned} \mathrm{E}\left[w(t+1) - p(t+1)\right] &= \mathrm{E}\left[(1+r)w(t) + \sum_{i=1}^{n} b_i(t)u_i(t) - p(t+1) - (1+r)\omega(t)\right] \\ &= \mathrm{E}\left[w_0(t+1) - p(t+1)\right] - (1+r)\omega(t), \end{aligned}$$

where $w_0(t+1)$ is the wealth at time $t+1$ in the absence of transaction costs. Therefore, while the expectation $\mathrm{E}[e(t+1)]$ of the hedging error $e(t+1)$ is affected by $\omega(t)$, its variance $\mathrm{Var}[e(t+1)]$ is clearly not, as

$$\begin{aligned} \mathrm{Var}\left[e(t+1)\right] &= \mathrm{E}\left[(e(t+1) - \mathrm{E}\left[e(t+1)\right])^2\right] \\ &= \mathrm{E}\left[\left(w_0(t+1) - p(t+1) - (1+r)\omega(t) - \mathrm{E}\left[w_0(t+1) - p(t+1)\right] + (1+r)\omega(t)\right)^2\right] \\ &= \mathrm{Var}\left[w_0(t+1) - p(t+1)\right]. \end{aligned}$$

□

Proposition 1 clearly shows that the minimum variance criterion (4) is insensitive to transaction costs and therefore potentially inadequate to handle them.

Constraints on how the quantities $u_i(t)$ are allocated can be additionally imposed. The formulation of optimization problems based on only Equation 7, in general, permits short selling, ie, $u_i(t) < 0$. *Short-selling constraints* can be included as $s_i(t)u_i(t) \geq -S_i^{\text{short}}$ or $\sum_{i=1}^{n} \min(s_i(t)u_i(t), 0) \geq -S^{\text{short}}$ for positive parameters S_i^{short} and S^{short}, respectively. *Diversification constraints* $s_i(t)u_i(t) \leq S_i^{\max}$ for some positive constant $S_i^{\max}$ or $s_i(t)u_i(t) \leq \rho_i w(t)$ for some fractional $\rho_i \in (0, 1]$ may also be imposed. Note that all of these constraints are *linear* in the control variables $u_i(t), \forall i = 1, \ldots, n$, a feature that is useful for the hedging formulations discussed in the next sections. Constraints on the *variance* or the *shortfall of risk*[16] would lead to convex constraints, although of *second-order cone* type.

3.1 | SMPC problem formulations

For SMPC for dynamic option hedging with transaction costs, we use again Algorithm 1 with the only difference that, in Step 4, an alternative optimization problem to the least-squares problem is solved. We introduce 3 possible SMPC formulations to account for transaction costs.

3.1.1 | Minimization of variance and expectation (QP-Var)

Let $x(t), y(t) \in \mathbb{R}^n$ be 2 vectors whose ith components are nonnegative and defined as

$$\begin{aligned} x_i(t) - y_i(t) &= u_i(t) - u_i(t-1) \\ x_i(t) \geq 0, y_i(t) &\geq 0, \quad \forall t = 0, \ldots, T. \end{aligned} \tag{9}$$

Accordingly, the proportional transaction cost $h_i(t)$ for trading a quantity $u_i(t) - u_i(t-1)$ of the ith asset is $h^i(t) = \epsilon^i |u_i(t) - u_i(t-1)| s_i(t) = \gamma_i(t)(x_i(t) + y_i(t))$, where $\gamma_i(t) \triangleq \epsilon^i s^i(t)$, $i = 1, \ldots, n$. The quantities $x_i(t)$ and $y_i(t)$ can be interpreted, respectively, as the amount of asset i *bought* at time t and the amount of asset i *sold* at time t. We can therefore introduce the new vector $v(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \in \mathbb{R}^{2n}$ of decision variables and replace $u(t) \in \mathbb{R}^n$ with

$$u(t) = u(t-1) + x(t) - y(t). \tag{10}$$

By letting

$$\mathbb{1} \triangleq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^M, \ \gamma(t) \triangleq \begin{bmatrix} \gamma_1(t) \\ \vdots \\ \gamma_n(t) \end{bmatrix},$$

from Equation 7, we can express the vector of future hedging errors $e(t+1) = w(t+1) - p(t+1)$ on the M different scenarios as

$$\begin{aligned} \begin{bmatrix} e^1(t+1) \\ \vdots \\ e^M(t+1) \end{bmatrix} &= B(t)u(t) + (1+r)\left(w(t) - \gamma'(t)\left(x(t) + y(t)\right)\right)\mathbb{1} - \begin{bmatrix} p^1(t+1) \\ \vdots \\ p^M(t+1) \end{bmatrix} \\ &= B(t)\left(u(t-1) + x(t) - y(t)\right) - (1+r)\mathbb{1}\gamma'(t)\left(x(t) + y(t)\right) + D(t) \\ &= A_v(t)v(t) + B_v(t) - \mathbb{1}G_v(t)v(t), \end{aligned} \tag{11}$$

where

$$B(t) \triangleq \begin{bmatrix} b_1^1(t) & \cdots & b_n^1(t) \\ \vdots & & \vdots \\ b_1^M(t) & \cdots & b_n^M(t) \end{bmatrix}, \quad D(t) \triangleq (1+r)\mathbb{1}w(t) - \begin{bmatrix} p^1(t+1) \\ \vdots \\ p^M(t+1) \end{bmatrix}$$

$$B_v(t) \triangleq B(t)u(t-1) + D(t), A_v(t) \triangleq \begin{bmatrix} B'(t) \\ -B'(t) \end{bmatrix}', \quad G_v(t) \triangleq (1+r)\begin{bmatrix} \gamma(t) \\ \gamma(t) \end{bmatrix}'.$$

The hedging error $e(t+1) = w(t+1) - p(t+1)$ has therefore the following empirical expectation:

$$\begin{aligned} \mathrm{E}\left[e(t+1)\right] &= \pi'\left(A_v(t)v(t) + B_v(t) - \mathbb{1}G_v(t)v(t)\right) \\ &= -G_v(t)v(t) + \pi'\left(A_v(t)v(t) + B_v(t)\right), \end{aligned} \tag{12}$$

where $\pi' = [\pi_1 \ldots \pi_M]' \in \mathbb{R}^M$, $\pi'\mathbb{1} = 1$. Note that, by Equation 12, we can rewrite $\mathrm{E}[e(t+1)] = K(t) - H(t)$, where $K(t) = \pi'(B(t)(x(t) - y(t)) + B_v(t))$ and $H(t) = (1+r)\gamma'(t)(x(t) + y(t))$. Therefore, $K(t)$ depends on the quantity $x(t) - y(t)$ (ie, on the net increment, $u(t) - u(t-1)$ of the underlying assets hold in portfolio from time $t-1$ to time t) and is independent of $\Lambda(t) = \min\{x(t), y(t)\}$ and of the transaction costs, whereas $H(t)$ depends on the actual number of transactions executed to rebalance the portfolio at time t, on $\Lambda(t)$, and, via $\gamma(t)$, on the transaction costs.

By letting i_j be the jth vector of the canonical basis of $\mathbb{R}^M$, ie, $i_j = [\underbrace{0 \dots 0}_{j-1}\ 1\ \underbrace{0 \dots 0}_{M-j}]'$, and omitting the dependence of t for ease of notation, we get

$$\begin{aligned} \mathrm{E}\left[e^2(t+1)\right] &= \sum_{j=1}^{M} \pi_j \left(i_j'(A_v v + B_v - \mathbb{1} G_v v)\right)^2 \\ &= v' G_v' G_v v + (A_v v + B_v)' \mathrm{diag}(\pi)(A_v v + B_v) - 2\pi'(A_v v + B_v) G_v v \end{aligned} \tag{13}$$

$$\begin{aligned} \mathrm{E}^2\left[e(t+1)\right] &= \left(\sum_{j=1}^{M} \pi_j i_j'(A_v v + B_v - \mathbb{1} G_v v)\right)^2 = \left(\pi'(A_v v + B_v) - G_v v\right)^2 \\ &= v' G_v' G_v v + (A_v v + B_v)' \pi\pi'(A_v v + B_v) - 2\pi'(A_v v + B_v) G_v v. \end{aligned} \tag{14}$$

Hence, the variance of $e(t+1)$ is

$$\begin{aligned} \mathrm{Var}\left[e(t+1)\right] &= \mathrm{E}\left[(e(t+1) - \mathrm{E}\left[e(t+1)\right])^2\right] \\ &= \mathrm{E}\left[e^2(t+1)\right] - \mathrm{E}^2\left[e(t+1)\right] \qquad (15a) \\ &= (A_v(t)v(t) + B_v(t))' \left(\mathrm{diag}(\pi) - \pi\pi'\right)(A_v(t)v(t) + B_v(t)). \qquad (15b) \end{aligned}$$

Note that Equation 15b does not depend on $\gamma(t)$, in accordance with Proposition 1, and that $\mathrm{diag}(\pi) - \pi\pi'$ is a positive semidefinite matrix.* Note also that $\mathrm{Var}[e(t+1)]$ does not depend on $x(t) - y(t)$ and, therefore, on $\Lambda(t)$, which confirms what was observed earlier about $\Lambda(t)$ only affecting transaction costs that are deterministic.

In order to minimize both the variance and the expected value of the one-step-ahead hedging error $e(t+1)$, we solve the following optimization problem:

$$\begin{aligned} \min_{v(t)} \quad & \mathrm{Var}\left[e(t+1)\right] + \alpha \mathrm{E}^2\left[e(t+1)\right] \\ \text{s.t.} \quad & v(t) \geq 0, \end{aligned} \tag{16}$$

where α is a fixed scalar, $\alpha \geq 0$. Problem (16) is a QP problem with $2n$ variables subject to nonnegativity constraints.

The hedging strategy defined by Equation 16 might lead to choosing optimal quantities $x_i(t)$ and $y_i(t)$ that are both positive, ie, $\Lambda_i(t) \triangleq \min\{x_i(t), y_i(t)\} > 0$. This amounts to allow the trader to simultaneously buy and sell the same quantity $\Lambda_i(t)$ of asset i at the same trading instant t (compare page 290 of the work of Cornuejols and Tutuncu[17]) or, in alternative, to violate the self-financing condition (3) by subtracting the wealth $\Lambda_i(t)\gamma_i(t)$ from the total portfolio wealth and rebalancing $u_i(t) = u_i(t-1) + \bar{x}_i(t) - \bar{y}_i(t)$, where $\bar{x}_i(t) = x_i(t) - \Lambda_i(t)$, $\bar{y}_i(t) = y_i(t) - \Lambda_i(t)$, $\bar{x}_i(t) - \bar{y}_i(t) = x_i(t) - y_i(t)$, and either $\bar{x}_i(t) = 0$ or $\bar{y}_i(t) = 0$. Constraining $\Lambda_i(t) = 0$ would make Equation 16 a nonconvex problem and, therefore, more complicated to solve numerically; however, leaving $\Lambda_i(t)$ unconstrained does not lead to undesired effects from a hedging viewpoint. In fact, having $x_i(t)$ and $y_i(t)$ both positive ($\Lambda_i(t) > 0$) might be a good choice to avoid super-replication without altering the variance of the hedging error. On the other hand, if, at optimality, $\mathrm{E}[e(t+1)] \leq 0$, ie, one is under-replicating the option price at time t, then, necessarily, $\Lambda_i(t) = 0$, otherwise $\bar{x}_i(t), \bar{y}_i(t)$ would be a solution with the same variance and a lower $\mathrm{E}^2[e(t+1)]$, thus providing a lower value of the objective function in Equation 16 than $x(t)$ and $y(t)$.

Note also that one could minimize $\mathrm{Var}[e(t+1)] + \alpha \mathrm{E}[e(t+1)]$ instead of Equation 16, therefore not penalizing super-replication. In this setting, either $x_i(t) = 0$ or $y_i(t) = 0$ spontaneously at optimality (ie, $\Lambda_i(t) = 0$ always holds at optimality) because, as observed earlier, a positive quantity $\Lambda_i(t)$ would only increase the term $H(t)$ because of transaction costs without altering $K(t)$ and $\mathrm{Var}[e(t+1)]$.

An alternative formulation based on mixed-integer quadratic programming, related to the approach of Glen[18] but based on the theory of hybrid dynamical systems,[19] that can handle more general transaction costs than proportional costs is reported in the Appendix.

*Matrix $\mathrm{diag}(\pi) - \pi\pi'$ is positive semidefinite by definition: $v'(\mathrm{diag}(\pi) - \pi\pi')v = \sum_{i=1}^{M} \pi_i v_i^2 - (\sum_{i=1}^{M} \pi_i v_i)(\sum_{j=1}^{M} \pi_j v_j) = \sum_{i=1}^{M} \pi_i (v_i^2 - 2v_i \sum_{j=1}^{M} \pi_j v_j + v_i \sum_{j=1}^{M} \pi_j v_j) = (\sum_{i=1}^{M} \pi_i (v_i^2 - 2v_i \sum_{j=1}^{M} \pi_j v_j)) + (\sum_{i=1}^{M} \pi_i v_i)(\sum_{j=1}^{M} \pi_j v_j)(\sum_{i=1}^{M} \pi_i) = \sum_{i=1}^{M} \pi_i (v_i - \sum_{j=1}^{M} \pi_j v_j)^2 \geq 0, \forall v \in \mathbb{R}^M$, which is the sampled version of Equation 15a.

Before stating the second SMPC formulation in Section 3.1.2, we motivate Equation 16 differently. Starting from a minimum-variance objective that alone is not suitable to account for transaction costs according to Proposition 1, we extend it by a probabilistic *chance constraint*[20] as follows:

$$\min_{v(t)\geq 0} \quad \mathrm{Var}\left[e(t+1)\right] \tag{17a}$$

$$\text{s.t.} \quad p\left(e(t+1)\leq e_{\text{low}}\right)\leq \eta, \tag{17b}$$

where we treat $e(t+1)$ as a random variable defined on some probability space and parameterized by $v(t)$, $p(e_{t+1}\leq e_{\text{low}})$ denotes the probability of event $e_{t+1}\leq e_{\text{low}}$, and e_{low} and $\eta\in[0,1]$ are 2 parameters. Then, under the assumption that $e(t+1)$ is Gaussian distributed, $e(t+1)\sim\mathcal{N}(\mu(t+1),\sigma(t+1))$, we obtain $p(e(t+1)\leq e_{\text{low}}) = N\left(\frac{e_{\text{low}}-\mu(t+1)}{\sqrt{\sigma(t+1)}}\right)$, where $N(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. Distinguishing between 2 cases, $\eta\in[0.5,1]$ and $\eta\in[0,0.5)$, we obtain for Equation 17b $e_{\text{low}}-\mu(t+1)\leq N^{-1}(\eta)\sqrt{\sigma(t+1)}$ and $\mu(t+1)-e_{\text{low}}\geq N^{-1}(1-\eta)\sqrt{\sigma(t+1)}$, respectively. Thus, we can formulate equivalently

$$\min_{v(t)\geq 0} \quad \mathrm{Var}\left[e(t+1)\right] \tag{18a}$$

$$\text{s.t.} \quad (e_{\text{low}}-\mu(t+1))^2\leq \left(N^{-1}(\eta)\right)^2\sigma(t+1), \tag{18b}$$

$$(e_{\text{low}}-\mu(t+1))\geq 0 \tag{18c}$$

for the first case and, similarly, for the second case. Here, a remark needs to be made. Suppose we parameterize $e(t+1)$ according to Equation 12 and make the additional (simplistic) assumption that $b_i(t)=s_i(t+1)-(1+r)s_i(t)$ is Gaussian distributed; then, $e(t+1)$ is likewise Gaussian (by linearity of Equation 12) and thus fits above the framework. However, then, Equation 18 is generally *not* a quadratically constrained *convex* program. This is because the Hessian of Equation 18b, which is here considered as a second-order condition for convexity, is generally not positive semidefinite. Instead, we consider the softened version of Equation 18 as $\min_{v(t)\geq 0}\{\mathrm{Var}[e(t+1)]+\lambda((e_{\text{low}}-\mu(t+1))^2\leq(N^{-1}(\eta))^2\sigma(t+1))+\xi(\mu(t+1)-e_{\text{low}})\}$ with Lagrangian multipliers $\lambda,\xi\in\mathbb{R}$. This resembles Equation 16 for $e_{\text{low}}=0$ and $\xi=0$. In fact, after a scaling step (not affecting the minimizer), it translates to

$$\min_{v(t)\geq 0} \quad \mathrm{Var}\left[e(t+1)\right]+\frac{\lambda}{1-\lambda\left(N^{-1}(\eta)\right)^2}\mathrm{E}^2\left[e(t+1)\right], \tag{19}$$

with $0\leq\lambda<\frac{1}{(N^{-1}(\eta))^2}\in[1,4]$ (derived from the equivalent $\alpha\geq 0$ in Equation 16) for $\eta\in[0.5,1]$. Similarly, the case for $\eta\in[0,0.5]$ can be obtained with $\alpha\geq 0$ in Equation 16 being represented by $\frac{-\lambda}{1+\lambda(N^{-1}(1-\eta))^2}$, and thus, $0\geq\lambda>-\frac{1}{(N^{-1}(1-\eta))^2}\in[-1,-4]$ to have $\alpha\geq 0$.

To summarize, we motivated Equation 16 starting from a minimum-variance objective and added chance constraint (17b) to account for transaction costs. It is stressed that only under the assumption of $e(t+1)$ following a Gaussian distribution and a relaxation of Equation 17b, a (loose) relation to Equation 16 could be established. Note that the chance-constraint formulation (17) permits to formulate *general* probabilistic constraints (without making the assumption of a Gaussian distribution of $e(t+1)$). Then, following Nemirovski and Shapiro,[21] *convex approximations* of general chance constraints (17b) can be formulated by means of different *generating functions* that place different penalties on how tight the original chance constraints are approximated.

Finally, let us also draw a relation between the β-VaR (value at risk) and the corresponding chance-constraint optimization problem formulation, as well as state the associated linear program (LP) in our scenario-based SMPC framework. Note that in the next Section 3.1.2, the β-VaR is used to derive the β-CVaR (conditional value at risk) and defined by a minimization problem after the definition of a loss function. For contrast, we here directly work with $e(t+1)$ and consider a more intuitive maximization problem formulation. We here define the β-VaR as e_{low} being the solution of the *chance-constraint* optimization problem as follows:

$$\max_{v(t)\geq 0,e_{\text{low}}} \quad e_{\text{low}} \tag{20a}$$

$$\text{s.t.} \quad p\left(e_{t+1}\leq e_{\text{low}}\right)\leq 1-\beta, \tag{20b}$$

where β is a parameter, typically $\beta = 90\%$, 95%, or 99%. For our *scenario-based* approach of Equation 11, we can approximate Equation 20 as

$$\max_{v(t)\geq 0,e_{\text{low}}} \quad e_{\text{low}} \tag{21a}$$

$$\text{s.t.} \quad \left\|\max\left(e_{\text{low}}\cdot\mathbb{1}-\left[e^1(t+1),\ \dots,e^M(t+1)\right],\mathbb{0}\right)\right\|_0\leq(1-\beta)M, \tag{21b}$$

where $\mathbb{0}$ indicates a vector of zeros, $\|\cdot\|_0$ denotes the ℓ_0-norm, and the max operator is acting elementwise. After convexifying constraint (21b) by substituting the ℓ_0- with the ℓ_1-norm and exploiting the natural nonnegativity of the formulation permitting us to drop absolute values, we obtain

$$\max_{v(t)\geq 0, e_{\text{low}}} \quad e_{\text{low}} \tag{22a}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} \max\left(e_{\text{low}} - e^i(t+1), 0\right) \leq (1-\beta)M. \tag{22b}$$

Ultimately, from Equation 22, we obtain the final LP formulation as follows:

$$\max_{v(t)\geq 0, e_{\text{low}}, \{y_i\}_{i=1}^{M}} \quad e_{\text{low}} \tag{23a}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} y_i \leq (1-\beta)M, \tag{23b}$$

$$y_i \geq e_{\text{low}} - e^i(t+1), \tag{23c}$$

$$y_i \geq 0. \tag{23d}$$

As motivated by Rockafellar and Uryasev,[22] the CVaR can be considered to be a more consistent measure of risk than VaR. Therefore, in the evaluation in Section 5, we only consider the CVaR-based SMPC formulation introduced in Section 3.1.2 rather than Equation 23.

For very recent work on chance constraints with applications to portfolio optimization, see also the works of Sun et al[23] and Sengupta and Kumar.[24]

3.1.2 | Minimization of conditional value at risk (LP-CVaR)

A drawback of the QP formulation (16) is that it requires calibrating the scalar α that achieves the best trade-off between variance (=risk) and expectation (=lack of hedging accuracy because of transaction costs). *Conditional value at risk* can be used as an alternative performance measure to penalize the hedging error $e(t+1)$ and is defined as follows.

Let $f(u,s) : \mathbb{R}^{n+k} \to \mathbb{R}$ be a loss function associated with the decision vector $u \in \mathbb{R}^n$ and with the random vector $s \in \mathbb{R}^k$. In our case, $u = u(t)$, $s = m(t+1)$, $f(u,s) = |e(t+1)|$ (in case super-replication of the option price is not penalized, $f(u,s) = -e(t+1)$). Let $p(s)$ be the probability density function of s. With respect to a given probability β, $0 \leq \beta \leq 1$, the β-VaR is defined as the lowest value ℓ such that, with probability β, the loss will not exceed ℓ. The number β is a fixed value, typically $\beta = 90\%$, 95%, or 99%. The main drawback of VaR is that the amount of loss occurring with probability $(1-\beta)$ is not taken into account directly. To avoid this, β-CVaR was introduced, ie, the conditional expectation of the loss function above ℓ, quantifying what the *average loss* is when one loses *more than* ℓ, with probability $1-\beta$.[22] The probability of $f(u,s)$ not exceeding the threshold ℓ is

$$\psi(u,\ell) = \int_{f(u,s)\leq \ell} p(s)ds.$$

The β-VaR and the β-CVaR are defined, respectively, as

$$\ell_\beta(u) = \min\{\ell \in \mathbb{R} : \ \psi(u,\ell) \geq \beta\}$$

and

$$\phi_\beta(u) = \frac{1}{1-\beta}\int_{f(u,s)\geq \ell_\beta(u)} f(u,s)p(s)ds.$$

Rockafellar and Uryasev[22] showed that the β-CVaR of the loss associated with any u can be determined by the formula

$$\phi_\beta(u) = \min_{\ell\in\mathbb{R}} F_\beta(u,\ell),$$

where

$$F_\beta(u,\ell) = \ell + \frac{1}{1-\beta}\int_{s\in\mathbb{R}^m} \left[f(u,s)-\ell\right]^+ p(s)ds \tag{24}$$

and $[\cdot]^+$ denotes the positive part of its argument, $[f]^+ = \max\{f, 0\}$. The integral in Equation 24 can be approximated by sampling the distribution of s according to the density function $p(s)$. If the sampling generates a collection of M vectors $s^1, \dots, s^M$, each of which has probability π_j of occurring, $j = 1, \dots, M$, then the corresponding approximation $\tilde{F}_\beta(u,\ell)$ is

$$\tilde{F}_\beta(u,\ell) = \ell + \frac{1}{1-\beta}\sum_{j=1}^{M} \pi_j \left[f(u,s^j)-\ell\right]^+.$$

Finally, we use CVaR to formulate the following SMPC problem for dynamic hedging:

$$\min_{v(t),\ell(t),\{z_j(t)\}_{j=1}^{M}} \quad \ell(t) + \frac{1}{1-\beta}\sum_{j=1}^{M} \pi_j z_j(t) \tag{25a}$$

$$\text{s.t.} \quad z_j(t) \geq w^j(t+1) - p^j(t+1) - \ell \tag{25b}$$

$$z_j(t) \geq -w^j(t+1) + p^j(t+1) - \ell \tag{25c}$$

$$z_j(t) \geq 0 \tag{25d}$$

$$j = 1, \ldots, M$$

$$v(t) \geq 0 \tag{25e}$$

for the given fixed value of β, where $w^j(t+1) - p^j(t+1)$ is given by Equation 11. Problem (25) is an LP problem with $M+2n+1$ variables and $3M+2n$ constraints. Note that, by removing constraint (25b), one does not penalize super-replication of the option price, as the loss function becomes $\max\{-e(t+1), 0\}$.

3.1.3 | Minimization of worst-case error (LP-MinMax)

A simpler approach than CVaR is to penalize the worst-case loss over the set of M generated scenarios, which is the largest absolute value $|e(t+1)|$ of the hedging error. The resulting formulation is the following LP problem:

$$\min_{v(t),\ell(t)} \quad \ell(t) \tag{26a}$$

$$\text{s.t.} \quad \ell(t) \geq w^j(t+1) - p^j(t+1) \tag{26b}$$

$$\ell(t) \geq -w^j(t+1) + p^j(t+1) \tag{26c}$$

$$j = 1, \ldots, M$$

$$\ell(t) \geq 0 \tag{26d}$$

$$v(t) \geq 0, \tag{26e}$$

where $w^j(t+1) - p^j(t+1)$ is given by Equation 11. Note that the LP (26) is simpler than Equation 25 as it only involves $2n + 1$ variables and $2(M + n) + 1$ constraints (they are identical for $M = 1$). In contrast, it is clear that the MinMax formulation (26) does not exploit the available information about the probability distribution of the stochastic variables that affect the evolution of the portfolio.

Finally, alternative performance measures to penalize the hedging error $e(t+1)$ are possible, such as the average of the maximum *shortfall*, with shortfall for scenario j defined as $\max\{-(w^j(t+1) - p^j(t+1)), 0\}$. In addition, terms penalizing transactions may be added to the objective functions, and, for example, transaction-rate constraints may be introduced.

4 | SCENARIO GENERATION

The closed-loop performance of SMPC heavily depends on the way the scenarios of both $s^j(t+1)$ and $p^j(t+1), j = 1, \ldots, M$, are generated. This is the focus of this section.

4.1 | Stock models

We propose 3 scenario generation methods for stock prices.

1. logn. The most widely used model to describe the dynamics of stock prices is the lognormal (logn) model. Its discrete-time form is

$$s_i(t+1) = s_i(t)e^{\left(\mu_i - \frac{1}{2}\sigma_i^2\right)T_s + \sigma_i\sqrt{T_s}\eta_i(t)}, \tag{27}$$

with T_s being the sampling interval (eg, 1 day), $\eta_i(t) \sim \mathcal{N}(0,1), \forall i = 1, \ldots, n$. Parameters μ_i and σ_i must be estimated from data, typically, as the *maximum likelihood* (ML) estimates from $\mathcal{T} + 1$ past stock prices using

$\left\{\ln\left(\frac{s_i(t-\mathcal{T}+1)}{s_i(t-\mathcal{T})}\right), \dots, \ln\left(\frac{s_i(t)}{s_i(t-1)}\right)\right\}$ and exploiting the Gaussian distribution of $\eta_i(t)$. We tested 3 methods. First, after ML identification on the training data, we maintained estimates μ_i and σ_i constant throughout the option's life. Second, we recursively re-estimated them using ML on a time-shifted data set (up until the current hedging date) of constant window length $\mathcal{T}+1$. Third, after initialization at $t=0$ using the ML estimate, we applied a *discrete extended Kalman filter* (dEKF) to adapt online. However, for artificially generated toy examples with underlying price $s(t)$ following a logn model, dEKF estimates converged to the true parameters; for real-world data it was not the case (real-world data do not follow the lognormal stock model). An additional disadvantage of the dEKF solution is the difficulty to select suitable tuning parameters (trading off model predictions and actual measurements). The recursive ML estimation performed overall best and is our preferred method when using the logn method. After identification and given $s_i(t), \forall i=1,\dots,n$, at current hedging date t, the M scenarios are generated by drawing $\eta_i^j(t) \sim \mathcal{N}(0,1)$ before evaluating Equation 27 to obtain $s_i^j(t+1)$ and $\pi_i^j = \frac{1}{M}, \forall j=1,\dots,M$.

2. SVR. This second model for stock price predictions is based on *support vector regression* (SVR) using Vapniks's ϵ-insensitive loss function for 1-dimensional outputs (see the work of Smola and Schölkopf[25]). The guiding motivation is to derive a parametric *nonlinear* fit to past stock data with input signal being the time instances $\{t-\mathcal{T}_{\text{SVR}},\dots,t\}$, and the output signal being the corresponding stock prices $\{s_i(t-\mathcal{T}_{\text{SVR}}),\dots,s_i(t)\}$. SVR can generate excellent nonlinear fits to past stock data. This motivated us to use the identified model for a one-step-ahead prediction. The prediction model has the form $\hat{s}_i(t+1) = W^T\varphi(t+1)+q$ with parameters $W \in \mathbb{R}^{n_f\times 1}$, where n_f denotes a high-dimensional feature space dimension, $q \in \mathbb{R}$ and $\varphi(\cdot): \mathbb{R} \to \mathbb{R}^{n_f\times 1}$. According to *Mercer's theorem*, $\varphi(t)^T\varphi(\tilde{t}) = K(t,\tilde{t})$ with symmetric and positive definite kernel function, eg, for the radial basis function (RBF) kernel; $K(t,\tilde{t}) = e^{(-\|t-\tilde{t}\|)/\sigma_{\text{RBF}}^2}$. The tuning parameters of the method are the positive scalars $\mathcal{T}_{\text{SVR}}$, σ_{RBF}, C_{SVR}, and ϵ_{SVR}, optimized as follows:

$$\min_{\substack{W,q,\xi_\tau,\xi_\tau^*\\ \tau=0,\dots,\mathcal{T}_{\text{SVR}}}} \quad \frac{1}{2}\|W\|_2^2 + C_{\text{SVR}}\mathbb{1}^T\xi_\tau + C_{\text{SVR}}\mathbb{1}^T\xi_\tau^* \tag{28a}$$

$$\text{s.t.} \quad \text{given data: } \{t-\tau, s_i(t-\tau)\}_{\tau=0}^{\mathcal{T}_{\text{SVR}}}, \tag{28b}$$

$$\xi_\tau, \xi_\tau^* \geq 0, \tag{28c}$$

$$s_i(t-\tau) - W^T\varphi(t-\tau) - q \leq \epsilon_{\text{SVR}} + \xi_\tau, \tag{28d}$$

$$-s_i(t-\tau) + W^T\varphi(t-\tau) + q \leq \epsilon_{\text{SVR}} + \xi_\tau^*, \tag{28e}$$

where $\mathbb{1}$ denotes a column vector of ones. We solve Problem (28) in the standard way by first formulating and solving its *dual* problem, which is a QP, and then determining $q \in \mathbb{R}$ via the Karush-Kuhn-Tucker conditions. The prediction is then conducted using the dual optimization variables, training input data, and applying Mercer's theorem. We then generate the M scenarios from

$$s_i^j(t+1) = \hat{s}_i(t+1) + 2\delta_{\text{SVR}}\eta_i^j(t),\ \eta_i^j(t) \sim \mathcal{N}(0,1),\ \ j=1,\dots,M \tag{29}$$

for all $i=1,\dots,n$ with $\delta_{\text{SVR}} = \frac{1}{\mathcal{T}_{\text{SVR}}}\sum_{\tau=1}^{\mathcal{T}_{\text{SVR}}} |s_i(0+1-\tau) - s(0-\tau)|$, ie, identified from the offline training data set. The coefficient 2 in Equation 29 was determined from closed-loop experiments. We found that a relatively high value was required for improved robustness and consistent solution quality (see also Section 5.2 for a related discussion).

3. pert. The proposed third model for stock price predictions takes the current stock price as the mean estimate (*Martingale process*) and generates scenarios by adding white perturbation noise, ie,

$$s_i^j(t+1) = s_i(t) + \sigma_{\text{pert}}\eta_i^j(t),\ \eta_i^j(t) \sim \mathcal{N}(0,1),\ \ j=1,\dots,M \tag{30}$$

for all $i=1,\dots,n$.

For final closed-loop simulations, we considered 1 year (252 trading days) of past real-world stock prices, which we partitioned into $\mathcal{T} = 125$ days of training data for initialization of μ and σ estimates (logn model). The coefficient $\sigma_{\text{pert}} = 0.3$ for the pert model was determined experimentally from both artificially generated and real-world data in closed-loop hedging experiments. We likewise determined $\sigma_{\text{RBF}} = 100$, $C_{\text{SVR}} = 1$, and $\epsilon_{\text{SVR}} = 0.01$. For the SVR model, an interesting finding was that the very short time period $\mathcal{T}_{\text{SVR}} = 10$ in combination with relatively large perturbation variance $(2\delta_{\text{SVR}})^2$ yielded the best closed-loop hedging results. Even if the presented SVR scheme permits in practice arbitrarily accurate nonlinear fits to *past* stock price data, the correspondingly identified model does *not* enable correct one-step-ahead stock price *predictions* with the same accuracy (also not even by sign). For the 3 methods, the average

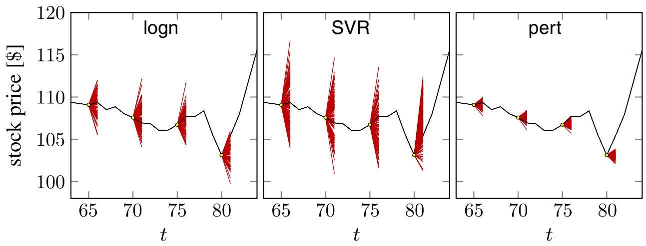

FIGURE 1 Comparison of the identified 3 scenario generation methods for the prediction of $s(t+1)$. The black line indicates the true stock price. The red lines indicate the scenarios $s^j(t+1), j = 1, \ldots, M = 100$ at times $t \in \{65, 70, 75, 80\}$. SVR, support vector regression [Colour figure can be viewed at wileyonlinelibrary.com]

computation time for the generation of all of $s^j(t+1), j = 1, \ldots, M = 100$ at each trading date was 0.14 ms, 0.4 ms (including the time for building and solving of the dual QP), and 0.018 ms. As expected, the SVR solution requires by far the most computations. Figure 1 quantitatively visualizes the 3 final scenario generation methods for the prediction of $s(t+1)$.

4.2 | Option pricing engine

An option pricing engine is needed at Step 3 of Algorithm 1 to estimate future option prices $p^j(t+1) = (1+r)^{-(T-(t+1))}\mathrm{E}[p^j(T)|s^j(t+1)], \forall j = 1, \ldots, M$. By employing MC simulations and the lognormal stock model, estimates for a European call option can be computed from

$$s_{i,k}^j(t+l) = s_{i,k}^j(t+l-1)e^{\left(\mu_i - \frac{\sigma_i^2}{2}\right)Ts + \sigma_i\sqrt{T_s}\eta_i(t+l-1)}, \tag{31}$$

$$\mathrm{E}\left[p^j(T)|s^j(t+1)\right] = \frac{1}{N_{\mathrm{sim}}}\sum_{k=1}^{N_{\mathrm{sim}}}\max\left(s_{i,k}^j(T) - K, 0\right), \tag{32}$$

with $i = 1$, $\eta_i(t+l-1) \sim \mathcal{N}(0,1)$, $l = 2, \ldots, T-t$, $s_{i,k}^j(t+1) = s_i^j(t+1), \forall k = 1, \ldots, N_{\mathrm{sim}}$. Note that $i = 1$ since there is 1 stock underlying both an European call and a barrier option. Even for a replicating portfolio holding $n-1$ other assets, the option underlying stock shall always be identified with $i = 1$. Thus, starting from $s_1^j(t+1)$, additional N_{sim} scenarios (eg, 100) up until expiration date T are generated. For the *path-dependent* barrier option, Equation 32 is replaced according to Equation 2.

As an alternative, we tested the following simpler pricing scheme for European call option prices

$$p^j(t+1) = (1+r)^{-(T-(t+1))}\max\left(s_i^j(t+1) - K, 0\right), \tag{33}$$

$i = 1, j = 1, \ldots, M$, and, similarly, for the barrier option. Thus, in comparison to the first method, we implicitly assumed $s_{1,k}^j(t+l) = s_{1,k}^j(t+1), \forall l = 2, \ldots, T-t, k = 1, \ldots, N_{\mathrm{sim}}$.

In fact, in addition to being much faster,† the second pricing engine (33) yielded significantly better and more consistent closed-loop hedging results. Intuitively, this has the following reason. Real-world data do not follow a lognormal stock model. Thus, Equation 31 can only very crudely predict $s_i(T)$. This is especially the case for a large difference between the current hedging date t and the expiration date T.

5 | HEDGING RESULTS

We test the 3 SMPC formulations for dynamic hedging defined, respectively, by Equations 16, 25, and 26 on a European plain vanilla call option and a barrier option with scenarios of stock and option prices generated according to Section 4. For the QP-Var approach, we select $\alpha = 0.25$, as it was calibrated in the works of Bemporad et al[12] using simulations of a lognormal stock model and assuming real market generating prices according to the same model (idealized nominal case). For the LP-CVaR approach, we use $\beta = 95\%$ in Equation 25.

†The average computation time was 0.6 seconds and 6.6e−5 seconds, respectively, for the generation of $p^j(t+1), j = 1, \ldots, M = 100$. Computation time is relevant since it permits to increase the number of scenarios M.

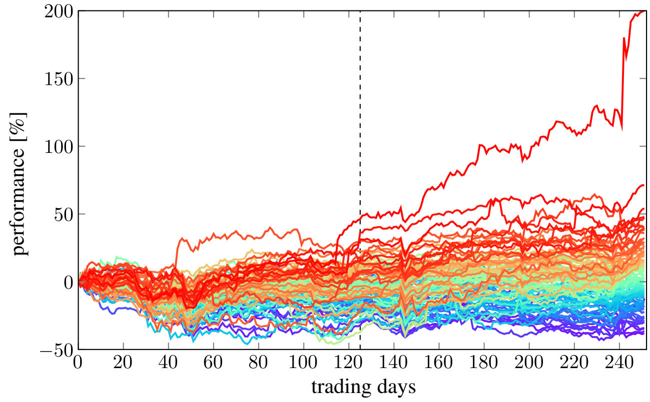

FIGURE 2 Normalized 1-year evolution of all 105 stocks held in NASDAQ-100 between November 27, 2015, and November 25, 2016. The 252 trading days are partitioned by the black-dashed vertical line into training and option evaluation data, respectively. Thus, $t = 0$ is initialized at trading day 125 [Colour figure can be viewed at wileyonlinelibrary.com]

For the solution of the QPs and LPs, we employ the domain-specific language CVXPY for optimization embedded in Python (see the work of Diamond and Boyd[26]). For completeness, QP and LP solution times are reported. Since they are in the millisecond range, the typical QP and LP complexities are not limiting factors for daily rebalancing. For more frequent rebalancing, however, they may become relevant. Note that the original Δ-hedging theory[3] is based on a *continuous-time* rebalancing assumption. Numerical experiments throughout this paper were run in Python 2.7 on a laptop running Ubuntu 14.04 equipped with an Intel Core i7 CPU @2.80 GHz×8, 15.6 GB of memory.

Throughout this section, we present results for real-world price data of all 105 stocks held in the NASDAQ-100 composite between November 27, 2015, and November 25, 2016. Data were obtained from finance.yahoo.com[‡] (see Figure 2 for visualization). We initialize $t = 0$ at trading day 125 (May 27, 2016). The option was simulated to expire after $T = 127$ rebalancing intervals (on November 25, 2016 with daily rebalancing). We assume proportional transaction costs of 1% and an effective annual risk-free rate $r_a = 1\%$. For each stock $s_i(t), i = 1, \ldots, 105$, we assume a European call option with strike price $K = s_i(0)$ and initialize $w(0) = 0.01 s_i(0)$. For the up-and-out option, the barrier is set as $s_u = 1.1 s_i(0)$. In all cases, the replicating portfolio is composed of the underlying stock and a cash position in the money market account (a setup similar to common Δ-hedging). We therefore drop subscripts i in the following.

A standard option contract typically covers 100 shares. Thus, $u(t) = 1$ implies a portfolio such that, at the end of the rebalancing interval, 100 shares of the underlying asset are held. Throughout the plots of this section, the average one-step-ahead predicted option price is denoted by $\bar{p}(t+1) = \frac{1}{M}\sum_{j=1}^{M} p^j(t+1)$ and the true option price is denoted by $p^\star(t) = (1+r)^{-(T-t)} \max(s(T) - K, 0)$ for a European call option and, similarly, for the path-dependent barrier option. For both option scenarios, we initialize $w(0) = 0.01 s(0)$. This simplistic wealth initialization is used for the reason that it permits good evaluation of tracking capabilities of the controllers. Since the second option pricing engine in Section 4.2 is our preferred choice, we obtain $\bar{p}(0) = 0$. This is because, in experiments, as outlined above, we initialize $K = s(0)$. Since stock prices typically cost much less than 1000$, our $w(0)$ choice implies a small initial estimated hedging error $w(0) - \bar{p}(0)$. Tracking capabilities of the controller can then be evaluated when proceeding with $t = 0, \ldots, T$. For visualization, see also Figures 3 and 4 (and accordingly $w(0) - \bar{p}(0)$) *uniformly* for all NASDAQ-100 stocks. This is not done as a means of testing robustness. In practice, $w(0)$ is initialized ideally as the true option price (which is unknown in a causal setting at $t = 0$) plus a premium. In general, it may be very difficult to select an appropriate $w(0)$, beyond classical cases, especially in the presence of transaction costs. The results of this paper offer a practical tool to financial institutions to *simulate* the effect of different initial prices $w(0)$ and choose a proper one.

The objective of the reported simulations is to understand what is the most suitable SMPC algorithm and scenario generation scheme, how they perform in comparison to Δ-hedging, and how perfect one-step-ahead knowledge affect results. For the last issue, we will assume that, at a given time t, we know $s(t+1)$ (but not $s(t+2), s(t+3), \ldots$). Finally, we want to assess, in general, whether SMPC can have a significant practical application for dynamic option-hedging with transaction costs.

[‡]The stock split (ratio two for one) that took place on December 2, 2015 for Ctrip.com International Ltd was not accounted for in the retrieved data.

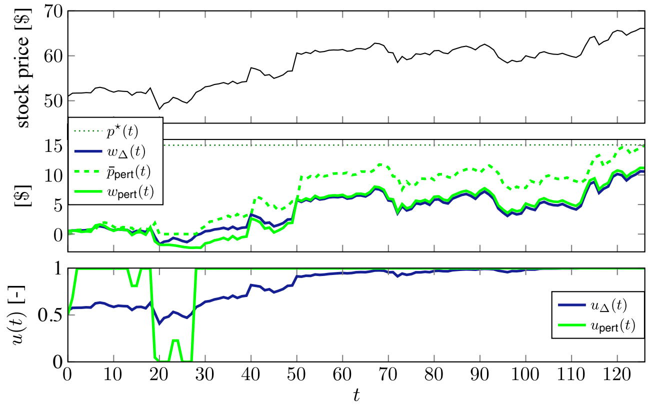

FIGURE 3 Causal setting. Comparison of Δ-hedging and linear program conditional value at risk (LP-CVaR) for a European call option. The underlying stock price (top frame) is of Microchip Technology Inc (May 27 until November 25, 2016) [Colour figure can be viewed at wileyonlinelibrary.com]

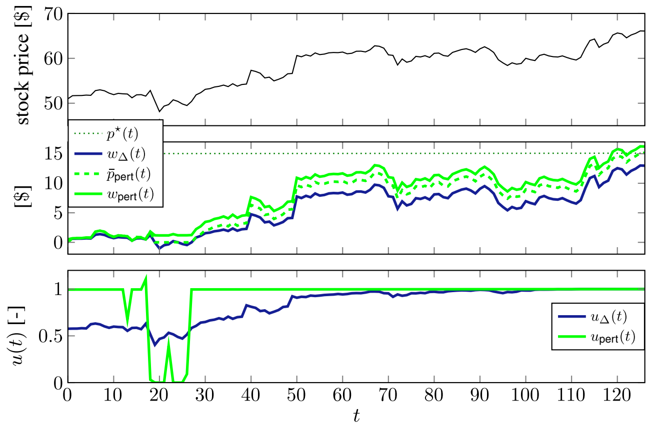

FIGURE 4 Noncausal setting. Comparison of Δ-hedging and linear program conditional value at risk (LP-CVaR) for a European call option. The underlying stock price (top frame) is of Microchip Technology Inc (May 27 until November 25, 2016) [Colour figure can be viewed at wileyonlinelibrary.com]

5.1 | European call option

We first test the SMPC algorithm on a European call option. Table 1 summarizes the expected and most negative final hedging error $e(T)$ and its variance for the considered stock data (see also Figure 5). A zero (or even positive) $\min(e(T))$ is desired, as it indicates the wealth shortfall at expiration. For Δ-hedging, we employed the analytical hedging formula for $u(t)$ from[3]

$$u(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d_1(t)} e^{-\frac{\xi^2}{2}} d\xi = N\left(d_1(t)\right), \tag{34}$$

with $d_1(t) = (\ln(\frac{s(t)}{K}) + (r_c + \frac{\sigma^2}{2}^{2})(T-t))/(\sigma\sqrt{T-t})$, where $r_c = \frac{\ln(1+r_a)}{T}$ is the continuously compounded interest rate (and r_a being the effective *annual* risk-free rate) and σ recursively estimated at each t is the ML as done for the logn scenario generation method, and with $N(\cdot)$ denoting the cumulative distribution function of the standard normal distribution. All Δ-hedging results in Section 5.1 refer to above method. Δ-hedging results in Table 1 refer to this method. We also tested the control law $u_t = \frac{p(t)-p(t-1)}{s(t)-s(t-1)}$ employing only known price data at t by computing $p(t) = (1+r)^{-(T-t)} \max(s(t) - K, 0)$ and, similarly, $p(t-1)$. In addition, we tested generating M scenarios $s^j(t+1), j = 1, \ldots, M$, before computing an average and the derivative approximation to account for the *step-ahead* nature of $\frac{\partial p(t+1)}{\partial s(t+1)}$. This, however, did not yield improvements. Figure 3 illustrates typical rebalancing trajectories for Δ-hedging and for an SMPC-based algorithm, here, using LP-CVaR and pert for scenario generation. A characteristic for Δ-hedging is rebalancing at almost every sampling time until reaching saturation (see Equation 34). The control command associated with SMPC is much less jagged often of stepwise nature and displaying variations more sparsely. These properties could be observed in multiple experiments.

TABLE 1 Results for a European call option without knowledge of the exact stock price $s(t+1)$ at time t. The CPU time (in ms) for building and solving of the QP and LPs for Δ-hedging is given by $\bar{\tau}$. For LP-CVaR and $M = 1000$, no solution could be returned for some of the stocks considered

Controller	M	$\mathbf{E}[e(T)]$ logn/SVR/pert	min(e(T)) logn/SVR/pert	$\mathbf{Var}[e(T)]$ logn/SVR/pert	$\bar{\tau}$, ms
QP-Var	100	−9.6/ − 19.3/ − 7.2	−124.7/ − 139.0/ − 193.2	216.7/634.3/382.5	3.3
LP-CVaR	100	−14.7/ − 14.0/ − 7.1	−134.0/ − 117.0/ − 45.4	387.9/306.9/51.0	4.3
LP-MinMax	100	−16.2/ − 16.5/ − 9.1	−150.6/ − 131.5/ − 94.8	458.8/459.2/132.3	3.2
QP-Var	1000	−9.2/ − 17.4/ − 7.1	−137.9/ − 146.0/ − 165.1	226.5/558.2/281.7	12.7
LP-CVaR	1000	−/ − /−	−/ − /−	−/ − /−	−
LP-MinMax	1000	−15.7/ − 13.9/ − 10.6	−179.3/ − 89.8/ − 189.3	522.0/309.5/361.5	10.6
Δ-hedging		−12.0	−114.6	282.4	6e − 2

Abbreviations: LP, linear program; LP-CVaR, linear program conditional value at risk; QP-Var, quadratic program variance; SVR, support vector regression.

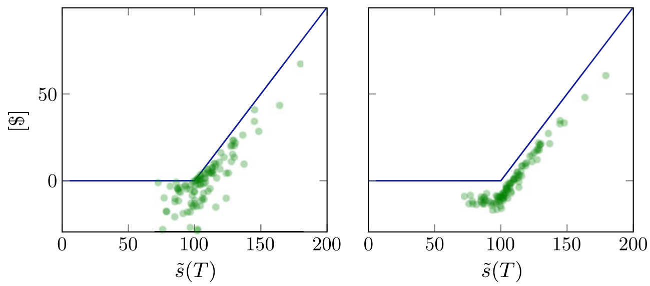

FIGURE 5 Results for a European call option in the causal setting. The normalized portfolio wealth $\tilde{w}(T) = 100w(T)/K$ and corresponding $\tilde{s}(T) = 100s(T)/K$ for all 105 stocks (green dots) is compared to the normalized payoff $\tilde{p}(T) = \max(\tilde{s}(T) - 100, 0)$ (solid blue) at expiration date T. (Left) Solution for linear program conditional value at risk (LP-CVaR), $M = 100$, and pert (see the second row from the top in Table 1). (Right) Solution for Δ-hedging (see the last row from the top in Table 1) [Colour figure can be viewed at wileyonlinelibrary.com]

Secondly, we tested the SMPC in a *noncausal* setting; at every time t, we assumed a perfect *one-step-ahead* knowledge of price $s(t+1)$. Consequently, we replaced the 3 stock price scenario generation schemes from Section 4.1 and instead used

$$s^j(t+1) = s(t+1) + \sigma_{\text{pert}}\eta^j(t), \eta^j(t) \sim \mathcal{N}(0,1),\ j = 1, \dots, M, \tag{35}$$

which is identical to Equation 30, except that $s(t+1)$ replaces $s(t)$ as the mean. The perturbation parameter σ_{pert} is set to 0.3 as for the causal setting. The reason for maintaining perturbation noise is to robustify the SMPC algorithm and is discussed in greater detail in the next section. For Δ-hedging in the noncausal setting, we still employ Equation 34 and replace, however, $d_1(t)$ by $d_1(t+1)$ to make use of $s(t+1)$ knowledge. Figure 4 illustrates a typical dynamic hedging result. Notice that the final hedging error $e(T)$ is positive for SMPC but negative for Δ-hedging. This behavior could be observed frequently (see Figure 6 and Table 2 for the hedging results for all 105 stock prices considered). By the definition of Equation 34, Δ-hedging always constrains $u(t)$ to lie between 0 and 1. For the SMPC formulation, this is not the case. Short selling and unconstrained ($u(t) > 1$) buying commands spontaneously result from solving the SMPC optimization problems (see Figure 4 around $t = 15$ for an illustration). Note that, for a practical implementation of the SMPC algorithm, it is recommended to add constraints such as $u_{\min}(t) \leq u(t) \leq u_{\max}(t)$, whereby the bounds have to be determined according to the requirements of the party writing the option. In the simplest case, $0 \leq u(t) \leq 1, \forall t$. As Table 2 indicates, when analyzing all 105 components of the NASDAQ-100 composite and assuming perfect knowledge of the one-step-ahead underlying price $s(t+1)$, a significant wealth shortfall of $\min(e(T)) = -45.7$ resulted for Δ-hedging, whereas $\min(e(T)) = -0.6$ resulted for LP-CVaR.

To summarize, when comparing the 3 stock price scenario generation methods (logn, SVR and pert) we found the pert scheme to perform best within our SMPC setting. Moreover, the combination of perfect $s(t+1)$ information and a SMPC algorithm results in consistently excellent final hedging errors and significantly outperforms common Δ-hedging. This finding encourages the employment of the presented SMPC algorithms and emphasizes the importance of accurately predict $s(t+1)$ at time t, as one would expect.

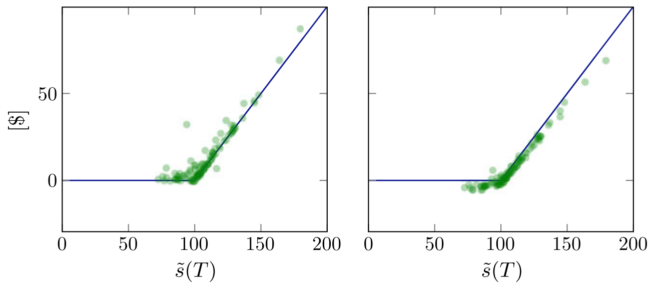

FIGURE 6 Results for a European call option in the noncausal setting. The normalized portfolio wealth $\tilde{w}(T) = 100w(T)/K$ and corresponding $\tilde{s}(T) = 100s(T)/K$ for all 105 stocks (green dots) is compared to the normalized payoff $\tilde{p}(T) = \max(\tilde{s}(T) - 100, 0)$ (solid blue) at expiration date T. (Left) Solution for linear program conditional value at risk (LP-CVaR), $M = 100$, and pert (see the second row from the top in Table 2). (Right) Solution for Δ-hedging (see the last row from the top in Table 2) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Results for a European call option with perfect knowledge of exact stock price $s(t+1)$at time t. In accordance with the scenario generation in Section 5.1, we set $M = 100$

Controller	E[$e(T)$]	min(e(T))	Var[$e(T)$]	τ, ms
QP-Var	3.3	−0.6	109.8	3.3
LP-CVaR	3.4	−0.6	123.8	3.8
LP-MinMax	3.3	−0.9	123.8	3.0
Δ-hedging	−2.8	−45.7	37.9	$6e-2$

Abbreviations: LP, linear program; LP-CVaR, linear program conditional value at risk; QP-Var, quadratic program variance.

5.2 | Barrier option

For the up-and-out option, we assumed a low barrier of $s_u = 1.1s(0)$. This resulted in 62.9% of all stock trajectories that the barrier was reached for at least one time instant before expiration date T. Simulation results are summarized in Tables 3 and 4. See also Figures 7 and 8. As for the European call option, the combination of perfect $s(t+1)$ information and a SMPC algorithm outperformed Δ-hedging. Most importantly, mainly positive final hedging errors could be recorded. As visualized by Figure 8, when analyzing all 105 stocks, the percentage of positive final hedging error was 90.5% and 62.9% for LP-CVaR and Δ-hedging, respectively. The rebalancing policy for Δ-hedging follows the method in Section 5.1 and sets $u(t+\tau) = 0, \forall \tau \in [0, T-t]$ as soon as the underlying stock price reaches the barrier at time t. Consequently, at time t, all wealth is transferred toward the risk-free asset.

Before concluding, we report 2 additional experiments. The first is motivated by the nature of barrier options. By definition, as soon as the underlying stock exceeds the barrier limit, the option value drops to 0. A corresponding wealth and control trajectory is displayed in Figure 9 with a characteristic $w(T) > 0$. Suppose for a theoretical reason one may

TABLE 3 Results for a path-dependent barrier option in the causal case without knowledge of exact stock price $s(t+1)$at time t. For LP-CVaR and $M = 1000$, no solution could be returned for some of the stocks considered

Controller	M	E[$e(T)$] logn/SVR/pert	min(e(T)) logn/SVR/pert	Var[$e(T)$] logn/SVR/pert	τ, ms
QP-Var	100	−0.34/ − 4.8/3.0	−33.2/ − 95.0/ − 27.3	421.5/178.1/152.9	3.0
LP-CVaR	100	−2.1/ − 2.5/1.9	−38.9/ − 29.0/ − 25.9	119.7/81.6/148.6	3.8
LP-MinMax	100	−2.9/ − 3.8/0.5	−45.9/ − 49.2/ − 23.0	123.4/85.2/158.6	3.0
QP-Var	1000	0.2/ − 3.1/3.3	−29.1/ − 84.1/ − 27.2	447.4/153.4/172.9	12.6
LP-CVaR	1000	−/ − /−	−/ − /−	−/ − /−	−
LP-MinMax	1000	−2.1/ − 1.4/ − 0.7	−46.3/ − 25.5/ − 35.0	124.5/75.2/142.2	9.8
Δ-hedging		0.7	−33.3	111.7	$6e-2$

Abbreviations: LP, linear program; LP-CVaR, linear program conditional value at risk; QP-Var, quadratic program variance; SVR, support vector regression.

TABLE 4 Results for a path-dependent barrier option in the noncausal case with perfect knowledge of exact stock price $s(t+1)$at time t. In accordance with the scenario generation in Section 5.1, we set $M = 100$

Controller	$\mathbf{E}[e(T)]$	min(e(T))	$\mathbf{Var}[e(T)]$	τ, ms
QP-Var	6.3	−5.1	373.6	3.1
LP-CVaR	7.0	−3.1	400.3	3.8
LP-MinMax	7.0	−3.4	401.8	3.2
Δ-hedging	5.8	−7.8	192.1	4e−2

Abbreviations: LP, linear program; LP-CVaR, linear program conditional value at risk; QP-Var, quadratic program variance.

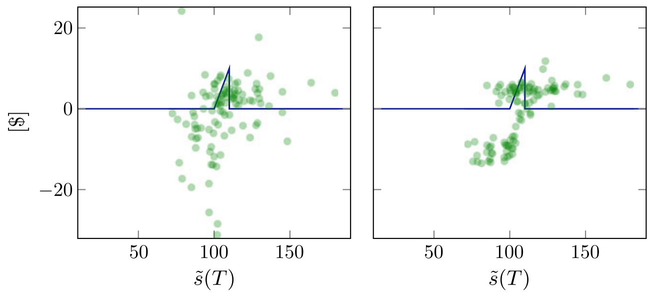

FIGURE 7 Results for a path-dependent barrier option in the causal case. The normalized portfolio wealth $\tilde{w}(T) = 100w(T)/K$ and corresponding $\tilde{s}(T) = 100s(T)/K$ for all 105 stocks (green dots) is compared to the normalized payoff $\tilde{p}(T)$ (solid blue) at expiration date T. (Left) Solution for linear program conditional value at risk (LP-CVaR), $M = 100$, and pert (see the second row from the top in Table 3). (Right) Solution for Δ-hedging (see the last row from the top in Table 3) [Colour figure can be viewed at wileyonlinelibrary.com]

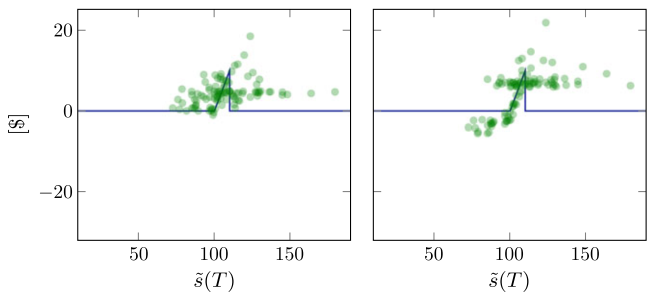

FIGURE 8 Results for a path-dependent barrier option in the noncausal case. The normalized portfolio wealth $\tilde{w}(T) = 100w(T)/K$ and corresponding $\tilde{s}(T) = 100s(T)/K$ for all 105 stocks (green dots) is compared to the normalized payoff $\tilde{p}(T)$ (solid blue) at expiration date T. (Left) Solution for linear program conditional value at risk (LP-CVaR), $M = 100$, and pert (see the second row from the top in Table 4). (Right) Solution for Δ-hedging (see the last row from the top in Table 4) [Colour figure can be viewed at wileyonlinelibrary.com]

still want to decrease the final hedging error $e(T)$ close to zero, thereby frequently reducing excess wealth accumulated at the time of barrier reaching. Then, for both the causal and noncausal settings, this can be achieved by adding perturbation noise, ie, by setting $p^j(t+1) = 0 + 0.3\eta^j(t), \eta^j(t) \sim \mathcal{N}(0,1)$ for option price scenarios, where coefficient 0.3 was chosen from experiments. As Figure 9 shows, the resulting control signal $u_{\text{pert},0}(t)$ is very jagged and requires short selling. Nevertheless, it is capable of reaching $e(T) = 0$ even in a *causal* setting. This behavior displays the powerful tracking capabilities of the SMPC algorithm, which may not only be exploited for dynamic hedging but also for index replication[27] and target performance tracking.[28]

The concluding experiment is to stress the necessity of stochasticity and a sufficiently large number of generated scenarios M, even in the case of perfect *one-step-ahead* knowledge of stock prices. We consider Equation 35 for scenario generation of stock prices and vary both M and the perturbation noise parameter σ_{pert}. Simulation results are summarized in Table 5. For visualization, see Figure 10. Note the sensitivity of control trajectories for $\sigma_{\text{pert}} = 0.01$ and the resulting temporary catastrophic tracking accuracy despite perfect one-step-ahead knowledge.

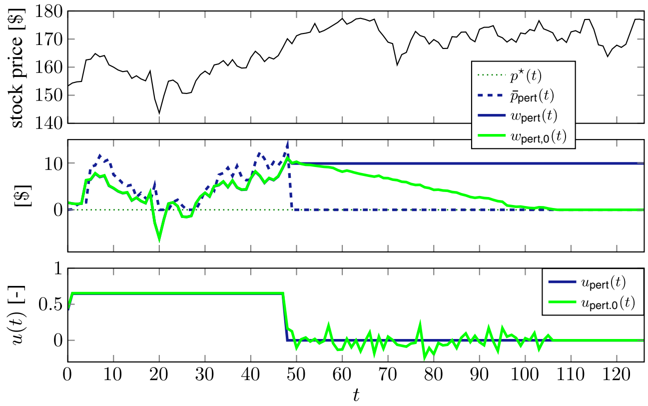

FIGURE 9 Employing quadratic program variance (QP-Var) as the stochastic model predictive control algorithm for a path-dependent barrier option in the causal setting. See Section 5.2 for the reduction of the final hedging error by means of perturbation noise. The underlying stock price (top frame) is of Broadcom Ltd (May 27 until November 25, 2016) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Results for a European call option in the noncausal case with perfect knowledge of exact stock price $s(t+1)$at time t. The average results for the entire NASDAQ-100 are reported. LP-MinMax is employed as the SMPC algorithm. For $\sigma_{\text{pert}} < 0.06$, no solution could be returned anymore for some of the stocks considered (the solver failed to find a solution)

$(M, \sigma_{\text{pert}})$	$\text{E}[e(T)]$	min(e(T))	$\text{Var}[e(T)]$	τ, ms
(2, 0.06)	−0.56	−339.2	1264.2	2.3
(2, 0.3)	−6.0	−1749.4	36547.3	2.3
(100, 0.06)	3.5	−0.8	121.8	3.1
(100, 0.3)	3.3	−0.9	123.8	3.0

Abbreviations: LP, linear program; SMPC, stochastic model predictive control.

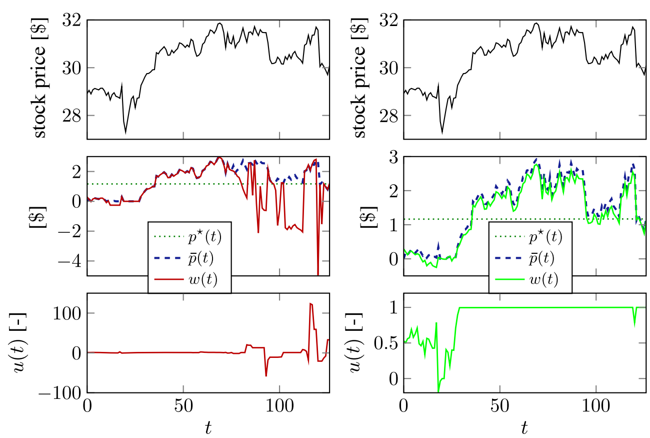

FIGURE 10 Noncausal case. Illustration of 2 different levels of perturbation in Equation 35: $\sigma_{\text{pert}} = 0.01$ (left frame) and $\sigma_{\text{pert}} = 0.3$ (right frame). Linear program (LP)-MinMax is employed as the stochastic model predictive control algorithm. The underlying stock price (top frame) is of Cisco Systems Inc (May 27 to November 25, 2016) [Colour figure can be viewed at wileyonlinelibrary.com]

Finally, we remark some success ratios reported in the literature for correct *sign* predictions of step-ahead price difference $s(t+1)s(t)$. They are meant to underline the difficulty in generating continuously accurate step-ahead predictions in practice. In the work of Kim,[29] support vector machines in combination with 12 technical indicators (such as Williams %R, stochastic %K, disparity, etc) are used to predict the direction of change in the *daily* Korea composite stock price index (KOSPI). For validation data and their best tuning parameter choices, they report a prediction performance between 50.0861% and 57.8313%. The same author mentioned similar results in an earlier work.[30]

6 | CONCLUSIONS

Stochastic model predictive control has been a suitable trading strategy for dynamic option hedging in the presence of transaction costs. The simple scenario generation method according to the pert scheme outperformed both the SVR-based and logn models. For the noncausal and theoretical case of perfect one-step-ahead knowledge of stock price data, consistently excellent hedging performance could be observed for the SMPC algorithms, especially the LP-based LP-CVaR, significantly surpassing common Δ-hedging. These findings encourage future efforts on improving short-term stock price predictions. For a practical implementation, constraints on $u(t)$ need to be added in the simplest case $u(t) \in [0, 1]$. Stochastic model predictive control can handle such input constraints naturally. In fact, as the evaluations on real-world stock prices of the NASDAQ-100 empirically show, accurate *one-step-ahead* predictions in combination with the presented SMPC framework are sufficient to achieve quasi-perfect hedging. This is in stark contrast to Δ-hedging, which is not able to benefit in the same manner from perfect one-step-ahead stock price predictions. The results importantly also imply that not more than accurate *one-step-ahead* (instead of multiple-step-ahead) predictions are required to achieve quasi-perfect hedging.

In this view, we reported hedging results based on real-world data from the NASDAQ-100, first, in the *realistic* and causal setting and, second, in the *optimal* setting with perfect one-step-ahead stock price knowledge. As success ratios reported in the literature show, continuously good step-ahead predictions are tremendously difficult or impossible to achieve. Nevertheless, the 2 discussed settings let one interpolate the potential of SMPC for different step-ahead stock price prediction qualities that improve upon the discussed pert scheme.

An additional practical benefit of the SMPC approach is its ability to easily incorporate a variety of constraints in the optimization problem formulations. Within the SMPC framework, we discussed the importance of scenario generation and perturbation noise even in the case of perfect one-step-ahead knowledge of stock prices.

ORCID

Alberto Bemporad http://orcid.org/0000-0001-6761-0856

REFERENCES

1. Hull J. *Options, Futures and Other Derivatives*. 6th ed. Upper Saddle River, NJ: Prentice Hall; 2006.
2. Graf Plessen M, Bemporad A. Parallel investments in multiple call and put options for the tracking of desired profit profiles. Paper presented at: Proceedings of American Control Conference; 2017; Seattle, WA.
3. Black F, Scholes M. Pricing of options and corporate liabilities. *J Polit Econ*. 1973;81(3):637-654.
4. Fedotov S, Mikhailov S. Option pricing for incomplete markets via stochastic optimization: transaction costs, adaptive control and forecast. *Int J Theor Appl Finance*. 1999;4(1):179-195.
5. Gondzio J, Kouwenbergb R, Vorst T. Hedging options under transaction costs and stochastic volatility. *J Econ Dyn Control*. 2003;27(6):1045-1068.
6. Primbs JA. LQR and receding horizon approaches to multi-dimensional option hedging under transaction costs. Paper presented at: Proceedings of American Control Conference; 2010; Baltimore, MD, USA.
7. Primbs JA. Dynamic hedging of basket options under proportional transaction costs using receding horizon control. *Int J Control*. 2009;82(10):1841-1855.
8. Dombrovskii VV, Dombrovskii DV, Lyashenko EA. Predictive control of random-parameter systems with multiplicative noise. Application to investment portfolio optimization. *Autom Remote Control*. 2005;66(4):583-595.
9. Herzog F, Dondi G, Geering HP. Stochastic model predictive control and portfolio optimization. *Int J Theor Appl Finance*. 2007;10(2):203-233.
10. Bemporad A, Bellucci L, Gabbriellini T. Dynamic option hedging via stochastic model predictive control based on scenario simulation. *Quant Finan*. 2014;14(10):1739-1751.
11. Bemporad A, Gabbriellini T, Puglia L, Bellucci L. Scenario-based stochastic model predictive control for dynamic option hedging. Paper presented at: Proceedings of 49th IEEE Conference on Decision and Control; 2010; Atlanta, GA, USA.
12. Bemporad A, Puglia L, Gabbriellini T. A stochastic model predictive control approach to dynamic option hedging with transaction costs. Paper presented at: Proceedings of American Control Conference; 2011; San Francisco, CA, USA.
13. Longstaff FA, Schwartz ES. Valuing American options by simulation: A simple least-squares approach. *Rev Financ Stud*. 2001;114(1):113-147.
14. Edirisinghe C, Naik V, Uppal R. Optimal replication of options with transactions costs and trading restrictions. *J Financ Quant Anal*. 1993;28(1):117-138.
15. Primbs JA, Yamada Y. A new computational tool for analyzing dynamic hedging under transaction costs. *Quant Finan*. 2008;8(4):405-413.

16. Lobo MS, Fazel M, Boyd S. Portfolio optimization with linear and fixed transaction costs. *Ann Oper Res*. 2007;152(1):341-365.
17. Cornuejols G, Tutuncu R. *Optimization Methods in Finance*. New York, USA: Cambridge University Press; 2007.
18. Glen JJ. Mean-variance portfolio rebalancing with transaction costs and funding changes. *J Oper Res Soc*. 2011;62:667-676.
19. Bemporad A, Morari M. Control of systems integrating logic, dynamics, and constraints. *Automatica*. 1999;35(3):407-427.
20. Charnes A, Cooper WW. Chance-constrained programming. *Manag Sci*. 1959;6(1):73-79.
21. Nemirovski A, Shapiro A. Convex approximations of chance constrained programs. *SIAM J Optim*. 2006;17(4):969-996.
22. Rockafellar RT, Uryasev S. Optimization of conditional value at risk. *J Risk*. 2000;2(3):21-42.
23. Sun Y, Aw G, Loxton R, Teo KL. Chance-constrained optimization for pension fund portfolios in the presence of default risk. *Eur J Oper Res*. 2017;256(1):205-214.
24. Sengupta RN, Kumar R. Robust and reliable portfolio optimization formulation of a chance constrained problem. *Found Comput Decis Sci*. 2017;42(1):83-117.
25. Smola AJ, Schölkopf B. A tutorial on support vector regression. *Stat Comput*. 2004;14(3):199-222.
26. Diamond S, Boyd S. CVXPY: A python-embedded modeling language for convex optimization. *J Mach Learn Res*. 2016;17(83):1-5.
27. Beasley JE, Meade N, Chang T-J. An evolutionary heuristic for the index tracking problem. *Eur J Oper Res*. 2003;148(3):621-643.
28. Gaivoronski AA, Krylov S, Van der Wijst N. Optimal portfolio selection and dynamic benchmark tracking. *Eur J Oper Res*. 2005;163(1):115-131.
29. Kim K. Financial time series forecasting using support vector machines. *Neurocomputing*. 2003;55(1):307-319.
30. Kim K, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl*. 2000;19(2):125-132.
31. IBM, Inc. IBM ILOG CPLEX Optimization Studio 12.4 - User Manual; 2012.
32. Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual; 2012.
33. Bemporad A. Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Trans Autom Control*. 2004;49(5):832-838.

How to cite this article: Graf Plessen M, Puglia L, Gabbriellini T, Bemporad A. Dynamic option hedging with transaction costs: A stochastic model predictive control approach. *Int J Robust Nonlinear Control*. 2019;29:5058-5077. https://doi.org/10.1002/rnc.3915

APPENDIX

DYNAMIC HEDGING BASED ON MIXED-INTEGER PROGRAMMING

Piecewise affine transaction costs as in Equation 8 can be also handled by introducing binary variables. Let $x^u(t) \triangleq u(t-1) \in \mathbb{R}^n$ be the composition of the portfolio immediately before trading at time t and introduce auxiliary variables $\delta_i(t) \in \{0,1\}$

$$[\delta_i(t)=1] \leftrightarrow \left[u_i(t)-x_i^u(t)\geq 0\right] \tag{A1}$$

and $q_i(t) \in \mathbb{R}$

$$q_i(t)=\begin{cases} u_i(t)-x_i^u(t), & \text{if} \quad \delta_i(t)=1 \\ 0, & \text{otherwise.}\end{cases} \tag{A2}$$

By using the so-called *big-M* technique, Equation A1 can be translated into the mixed-integer linear inequalities

$$u_i(t)-x_i^u(t) \geq -M_i(1-\delta_i(t)) \tag{A3a}$$

$$u_i(t)-x_i^u(t) \leq M_i\delta_i(t)-\epsilon \tag{A3b}$$

and Equation A2 into

$$q_i(t) \leq u_i(t)-x_i^u(t)+M_i\,(1-\delta_i(t)) \tag{A4a}$$

$$q_i(t) \geq u_i(t)-x_i^u(t)-M_i\,(1-\delta_i(t)) \tag{A4b}$$

$$q_i(t) \leq M_i\delta_i(t), \tag{A4c}$$

$$q_i(t) \geq -M_i\delta_i(t) \tag{A4d}$$

where M_i is an upper bound on $|u_i(t) - x_i^u(t)|$, that is, the maximum allowed asset reallocation and $\epsilon > 0$ is a small scalar (eg, the machine precision). Equation 7 can therefore be interpreted as the evolution of a *hybrid dynamical system* that is expressed in the following *mixed logical dynamical* form[19]:

$$w(t+1) = (1+r)\left(u_0(t) - \sum_{i=1}^{n} q_i(t) - 2\left(u_i(t) - x_i^u(t)\right)\right) + \sum_{i=1}^{n} s_i(t+1)u_i(t) \tag{A5a}$$

$$x^u(t+1) = u(t) \tag{A5b}$$

$$\text{s.t. Equations A3 and A4} \tag{A5c}$$

with states $w(t), x^u(t)$, input $u(t)$, auxiliary vector $\delta(t) = [\delta_1(t) \dots \delta_n(t)]' \in \{0,1\}^n$ of binary variables, and vector $q(t) = [q_1(t) \dots q_n(t)]' \in \mathbb{R}^n$ of auxiliary continuous variables. Note that, from a system theoretical viewpoint, transaction costs introduce a unit delay (A5b) in the dynamics because of the additional state variable $x^u(t)$.

By using the stochastic hybrid dynamical model (A5), problem (16) can be recast as a mixed-integer quadratic programming problem (see the work of Bemporad and Morari[19] for details) to be minimized with respect to vector $u(t) \in \mathbb{R}^n$, for which very efficient solvers are available.[31,32] See also the work of Glen[18] for a related approach. For options involving a single stock, the number n of assets is usually very small ($n = 1$ or $n = 2$) so that the minimum variance problem with transaction costs can be solved also by enumerating the possible 2^n instances of vector $\delta(t)$ (ie, in system theoretical terms, by transforming the mixed logical dynamics (A5c) into an equivalent piecewise affine form[33] and enumerating the *modes* of the resulting piecewise affine dynamics) and by solving the corresponding QPs (6) subject to $u_i(t) \geq x_i^u(t)$ if the corresponding $\delta_i(t) = 1$ or $u_i(t) \leq x_i^u(t)$ if $\delta_i(t) = 0$ for all $i = 1, \dots, n$.

While the method in Section 3.1.1 is generally more efficient from a numerical viewpoint, in that it completely avoids introducing integer variables to handle *proportional* transaction costs, the mixed-integer quadratic programming method of this section is more general; for example, it can be easily extended to handle transaction costs of the form $h_i(u_i(t) - u_i(t-1)) = \min\{c_0, \epsilon^i s_i(t)|u_i(t) - u_i(t-1)|\}$, where c_0 is a given minimum fixed cost to be paid in each transaction.