ISSN 1022-8594 © 2024 Jahangirnagar University Journal of Science
JUJS Volume 44, No 2, December 2024

An Efficient Approach for Estimating Unknown Constants in an Adaptive Inventory Control System under Uncertainty.

Sushanta K. Roy[1],Mohammad I. Khan[2].
[1]Jahangirnagar University, Savar, Dhaka - 1342, Bangladesh.
[2]Hamilton College, Clinton, NY 13323, USA.
Email: sushi.math14819@juniv.edu, mkhan@hamilton.edu.

Abstract

This study focuses on inventory control in a manufacturing system for a typical machine building enterprise, involving machine building, transport, storage bunker, and assembly line. The storage bunker faces varying disturbances from the assembly line, necessitating consistent product flow from machining and transport to prevent operational failures. However, the lack of an exact machining model and uncertainties related to machine failures demand an adaptive decision-making system, which has already been developed. In this paper, we propose a modified, more realistic approach, to estimate some properties of the machining model more accurately, accounting for uncertainties due to machine failures and predictable, inconsistent disturbances from the assembly line. The effectiveness of this mo*dified approach is demonstrated through simulation experiments.

Key words: Storage Bunker, Adaptive decision-making system, uncertainties.

1. Introduction

The in-process inventory control problem, initially introduced decades ago in Buchan and Koenigsberg's work in 1963, continues to be a subject of considerable interest in both theoretical and practical contexts. Since the pioneering studies of Simon (1952) and Yokoyama (1955), classical control theory has been employed as a valuable tool for managing manufacturing systems that involve in-process inventories. Significant advancements in this research area have been achieved by Axsater (1985), Kuntsevich (1973), Shin *et al.* (2008), Skurikhin (1972), Wiendahl and Breithaupt (2000), and further extended by Azarskov *et al.* (2006) and Zhiteckii *et al.* (2007), who investigated dynamic processes in typical production control systems.

In recent times, novel results from modern control theory have inspired various approaches to address manufacturing control problems. These

*Author for correspondance Email: sushi.math14819@juniv.edu

DOI: https://doi.org/10.59185/jos.v44i02.185

approaches encompass linear programming and dynamic programming, robust and adaptive control concepts, genetic algorithms, Li-optimization, and more, as discussed in

the works of Aharon *et al.* (2009), Azarskov *et al.* (2013), Bauso *et al.* (2006), Boukas (2006), Grubbstrom and Wilmer (1996), Hennet (2003), Hoberg *et al.* (2007), Ignaciuk and Bartoszewic (2010), Kostio (2009), Rodrigues and Boukas (2006), Taleizadeh *et al.* (2009), and Towill *et al.* (1997).

Achieving a perfect inventory control for manufacturing systems requires an exact mathematical model concerning machining (Skurikhin, 1972). However, in practice, only an approximate model of machining is available for decision-making systems, and the possibility of machine failures further introduces uncertainty into the order (reorder) policy formation (Azarskov *et al.*, 2006; Zhiteckii *et al.*, 2007).

In dealing with this uncertainty, modern control theory offers two primary approaches: the nonadaptive robust approach, as proposed by Sanchez-Pena and Szanier in 1998, and the adaptive approach introduced by Landau *et al.* in 1997. These methods aim to address the challenges posed by uncertainty and contribute to the development of effective inventory control strategies in manufacturing systems.

In this paper, the algorithm to estimate γ, an unknown constant that appears in the adaptive inventory control model by Azarskov *et al.* (2017), is modified, addressing the ambiguities and inconsistencies that appear in the original paper. On top of that, the proposed approach requires a supplementary estimation algorithm for $\gamma_{\min}$, another unknown constant in the adaptive inventory control model. The main contribution of this paper is that the proposed estimation algorithms can be used in other inventory control models that use these unknown constants.

2. Inventory Control System

2.1 The Basics

The in-process inventory control system, as developed in Buchan and Koenigsberg (1962, chapter 22), of a typical machine building enterprise which includes machining, transportation, storage bunker and assembly line operates as follows. At the start of each interval, $t=t_n=nT_0$, where T_0 is the duration of each interval, the decision making system requests the current product stock level, $H(t)$, that is, $H(t_n)=H_n$. Then it determines the deviation of H_n from the safety stock level r^0 as expressed by

$$e_n=r^0-H_n \text{ \(in units\)}.$$

Then the decision making system places an order (reorder) θ_n, which defines the volume of product that has to be produced during the planning interval $t_n \le t \le t_{n+1}$ according to the rule

$$\theta_n = \begin{cases} \theta_{\max} & \text{if } \theta_n^c > \theta_{\max}, \\ \theta_n^c & \text{if } 0 \le \theta_n^c \le \theta_{\max}, \backslash(\text{in units}\backslash) \\ 0 & \text{if } \theta_n^c < 0, \end{cases}$$

where $\theta_{\max}$ is the maximum order volume that can be produced during the time interval $t_n \le t \le t_{n+1}$ with all available manufacturing resources at its maximum capacity, and θ_n^c defined by a given order policy. The simplest order policy (Kuntsevich, 1973; Shurikhin, 1972) is given by

$$\theta_n^c = e_n \backslash(\text{in units}\backslash).$$

Then the decision making system determines the production capacity necessary to produce the order volume θ_n, as expressed by

$$q_n = q(\theta_n),$$

where q is a vector-valued operator, and formally gives an operation schedule for each machine.

At the end of the time interval $[t_n, t_{n+1}]$, the product fabricated by the machining can be represented as

$$Q_{n+1} = P_{n,n+1}(q_n) - \xi_{n,n+1} \backslash(\text{in units}\backslash)$$

where $P_{n,n+1}$ is a time varying operation and $\xi_{n,n+1}$ is an additive non-negative noise $(\xi_{n,n+1} \ge 0)$ introduced due to machine failure during the time interval $t_n \le t \le t_{n+1}$.

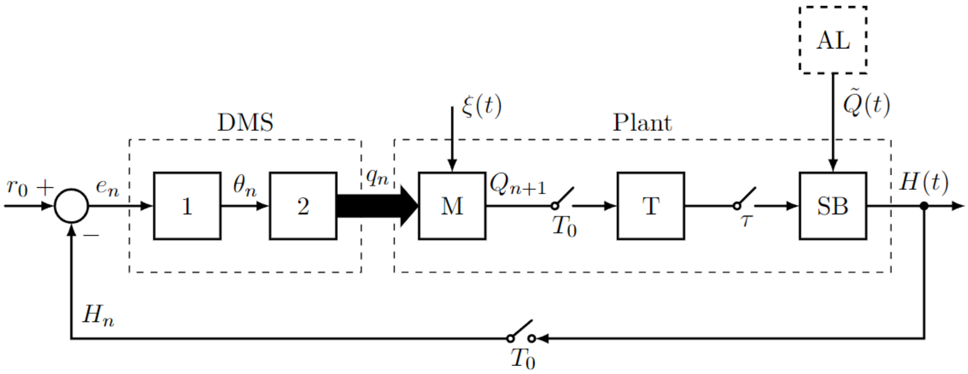


Fig. 1. Block diagram of inventory control system where the controller parts 1 and 2 form the decision making system. The plant consists of machining (M), transportation (T) and storage bunker (SB). The storage bunker is subjected to the external disturbance coming from the assembly line (AL).

As in Azarskov *et al.* (2006), Shurikhin (1972) and Zhiteckii *et al.* (2007), it is assumed that the product is then transported to the storage bunker with a time delay $\tau < T_0$ at the time instant $t = t_{n+1} + \tau$. Then the product is taken from the storage bunker to the assembly line based on the demands coming from there, with a rate $k(t) \geq 0$. So, the product stock level $H(t)$ slowly decreases at all times until the lot size of Q_{n+1} arrives from the machining and $H(t)$ increases step-wise.

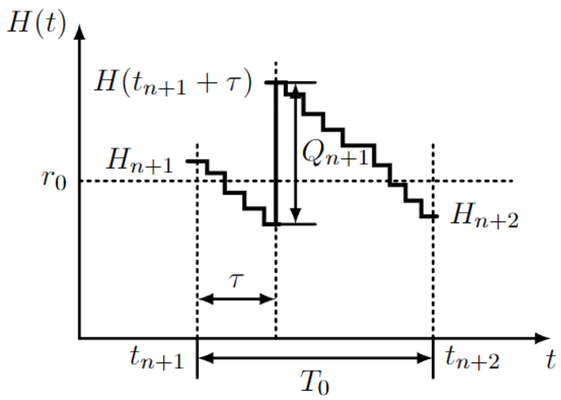


Fig. 2. A typical inventory history over the time interval $[t_{n+1}, t_{n+2}]$.

During the time period $[t_{n+1}, t_{n+2}]$, the lot size taken from the storage bunker on the demand of assembly line can be written as

$$\Delta \widetilde{Q}_{n+1,n+2} = \int_{n+1}^{n+2} k(t)\, dt \text{ (in units)}$$

where, $k(t) = 0$ if and only if $H(t) = 0$ since $H(t)$ cannot be negative. The inventory level at the time instant $t_{n+2} = (n+2) T_0$ can then be given by

$$H_{n+2} = H_{n+1} - \Delta \widetilde{Q}_{n+1,n+2} + Q_{n+1}.$$

Note that $\Delta \widetilde{Q}_{n+1,n+2}$ and $\Delta \widetilde{Q}_{n+2,n+3}$ can exactly be predicted for some variables $\Delta \widetilde{Q}[n+i, n+i+1]$ $(i = 1, 2)$ at each n utilizing a technique from Azarskov *et al.* (2006). Also note that it can be assumed that $\Delta \widetilde{Q}_{n+1,n+2}$ can vary with n.

The mathematical model of the in-process inventory control system is defined by the equations (1)-(7).

2.2 Features of the System

Similar to Azarskov *et al.* (2017), equation (5) together with equation (4) yields

$$Q_{n+1} = P_{n,n+1}(q_n(\theta_n)) - \xi_{n,n+1},$$

which means $Q_{n+1} \neq \theta_n$ even when $\xi_{n,n+1} = 0$ because the machining model is not exact. Define

$$\gamma_n = \frac{P_{n,n+1}\left(q_n\left(\theta_n\right)\right)}{\theta_n} \leq 1.$$

Using (9), (8) can be rewritten as

$$Q_{n+1} = \gamma_n \theta_n - \xi_{n,n+1}.$$

Ideally, $\gamma_n = 1$ and $\xi_{n,n+1} = 0$, which yields $Q_{n+1} = \theta_n$.

Suppose γ_n is a random coefficient with possibly nonstochastic nature (Zhiteckii, 1996) and changes within the interval

$$\gamma_{\min} \leq \gamma_n \leq 1,$$

where $\gamma_{\min}$ is the unknown lower bound to γ_n. Further, let

$$0 \leq \xi_{n,n+1} \leq \acute{\xi},$$

where $\acute{\xi}$ is the known upper bound to $\xi_{n,n+1}$.

Accoding to Azarskov *et. al.* (2017), the production at the end of the interval $\left[t_n, t_{n+1}\right]$ can be written as

$$Q_{n+1} = \gamma\,\theta_n - \acute{\xi}/2 + v_{n,n+1},$$

where γ is a unknown constant and $v_{n,n+1}$ is an equivalent virtual symmetrical noise satisfying

$$\left|v_{n,n+1}\right| \leq \varepsilon$$

and

$$\varepsilon \leq \left[\left(1-\gamma_{\min}\right)\theta_{\max} + \acute{\xi}\right]/2.$$

However, the use of the constant term $\acute{\xi}/2$ in (1) is ambiguous. Besides, the assumption that upper bound of ε in (3) for all θ_n is a constant is, although logically correct, unrealistic. We will try to address these issues.

Modifying (1), we get,

$$Q_{n+1} = \gamma\,\theta_n + v_{n,n+1},$$

where, $\left|v_{n,n+1}\right| \leq \varepsilon$, and $\varepsilon = \left[\left(1-\gamma_{\min}\right)\theta_n + \acute{\xi}\right]/2$.

Since $\gamma_{\min}$ is unknown, ε is also unknown. (16) and (17) are visually depicted in Fig. 3.

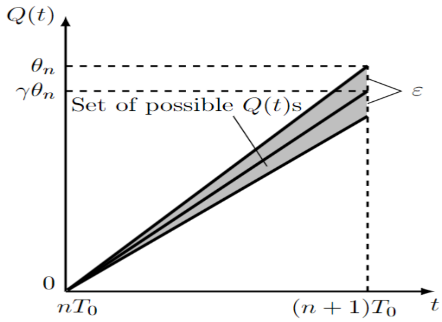


Fig. 5. Illustration of the modified production process. Note that the visual interpretation of the production process is identical to the one proposed in Azarskov *et al.* (2017).

2.3 Control Objective

The control performance index,

$$J = \lim_{n\to\infty} |e_n| ¿$$

evaluates the ultimate behavior of the control system (1) - (7) for all sufficiently large *n*.

The aim of the control system is to form the reorder policy yielding $\theta_n = \theta_1, \theta_2, \theta_3, \ldots$ minimizing J according to $J \to \min_{\{\theta_n\}}$

with the uncertainties of the form (10) and (11) present.

3. Main Result

3.1 Modified Adaptive Estimation Algorithm

The modified adaptive estimation algorithm proposed here for estimating the unknown constants γ and $\gamma_{\min}$ is directly inspired by the adaptive algorithm advanced in Azarskov *et al.* (2017). Similar to that approach, expanding the inequality in (17) yields

$$|Q_{n+1} - \gamma\theta_n| \leq \varepsilon,$$

and recursively solving (21) yields

$$\frac{(Q_{n+1} - \varepsilon)}{\theta_n} \leq \gamma \leq \frac{(Q_{n+1} + \varepsilon)}{\theta_n}$$,which produces the set membership estimation procedure of the form

$$\underline{\gamma}(n+1)=\begin{cases}\underline{\gamma}(n) & \text{for } \theta_n<0 \text{ or } \underline{\gamma}(n)\ge(Q_{n+1}-\varepsilon)/\theta_n,\\(Q_{n+1}-\varepsilon)/\theta_n & \text{otherwise,}\end{cases}$$
$$\acute{\gamma}(n+1)=\begin{cases}\acute{\gamma}(n) & \text{for } \theta_n<0 \text{ or } \acute{\gamma}(n)\le(Q_{n+1}+\varepsilon)/\theta_n,\\(Q_{n+1}+\varepsilon)/\theta_n & \text{otherwise}\end{cases}$$

when $\underline{\gamma}(n+1)\le\acute{\gamma}(n+1)$. If (23) causes $\underline{\gamma}(n+1)>\acute{\gamma}(n+1)$,

$$\underline{\gamma}(n+1)=\underline{\gamma}(0),\acute{\gamma}(n+1)=\acute{\gamma}(0).$$

Since $\gamma_{\min}$ is unknown, define

$$\gamma_{\min}'(n+1)=\begin{cases}\gamma_{\min}(n) & \text{for } \gamma_{\min}(n)\le(Q_{n+1}+\acute{\xi})/\theta_n\\(Q_{n+1}+\acute{\xi})/\theta_n & \text{otherwise},\end{cases}$$

where $\gamma_{\min}(n)$ is the current best estimation of $\gamma_{\min}$. Furthermore,

$$\gamma_{\min}''(n+1)=\begin{cases}\gamma_{\min}(n) & \text{if } \underline{\gamma}(n+1)\le\acute{\gamma}(n+1)\\\gamma_{\min}(n)-\Delta & \text{otherwise},\end{cases}$$

where the constant $\Delta>0$ is a small enough positive number chosen by the designer. Also define

$$\gamma_{\min}(n+1)=\begin{cases}\gamma_{\min}'(n+1) & \text{if } \gamma_{\min}(n)\ge\gamma_{\min}''(n+1)\ge\gamma_{\min}'(n+1),\\\gamma_{\min}''(n+1) & \text{if } \gamma_{\min}(n)\ge\gamma_{\min}'(n+1)\ge\gamma_{\min}''(n+1)\\\gamma_{\min}(n) & \text{otherwise.}\end{cases}$$

Inspired by Zhiteckii (1996), the point estimation procedure used for deriving $\gamma(n)$ is

$$\gamma(n+1)=\begin{cases}\gamma(n) & \text{if } |S_n|\le\varepsilon'(n)\\\gamma(n)-(S_n-\varepsilon(n))/\theta_n & \text{if } S_n>\varepsilon'(n)\\\gamma(n)-(S_n+\varepsilon(n))/\theta_n & \text{if } S_n\leftarrow\varepsilon'(n)\end{cases}$$

where, $S_n=\gamma(n)\theta_n-Q_{n+1}$ is the current identification error and $\varepsilon'(n)=\varepsilon(n)+\Delta'$

Here, $\Delta'>0$ is a sufficiently small positive constant chosen by the designer as in Zhiteckii (1996).

3.2 Adaptive Reorder Policy

The reorder policy developed by Azarskov *et al.* (2013) as given by

$$\theta_n^c=(e_n+\Delta\widetilde{Q}[n,n+1]+\Delta\widetilde{Q}[n+1,n+2]-\gamma\theta_{n-1})/\gamma$$

is used here, where $\Delta\tilde{Q}[n,n+1]\equiv\Delta\tilde{Q}_{n,n+1}$ and $\Delta\tilde{Q}[n+1,n+2]\equiv\Delta\tilde{Q}_{n+1,n+2}$. The difference is that γ is replaced by $\gamma(n)$ forming

$$\theta_n^c=\left(e_n+\Delta\tilde{Q}[n,n+1]+\Delta\tilde{Q}[n+1,n+2]-\gamma(n)\theta_{n-1}\right)/\gamma(n).$$

Contrary to Azarskov *et al.* (2013), θ_n is defined by (2).

3.3 Post Adaptation Stage

After the adaptation stage defined by (23) - (30) and (32) is complete, the estimate values for γ and $\gamma_{\min}$ can be determined by taking the mode of their respective values obtained during the adaptation stage for sufficiently large n. The assumption is that the model for estimating γ and $\gamma_{\min}$ is accurate enough that the deviations from the mode can be considered outliers and can be explained by the absence of exact machining model and uncertainties due to the likelihood of machine failure. The algorithm will produce more accurate estimations for γ and $\gamma_{\min}$ for larger values of n.

4. Simulation

A simulation was run to demonstrate the adaptive stage defined by (23) - (30) and (32) as well as the post adaptation inventory control. The simulation program (see Appendix A for source code) was set up by taking $r^0=40$, $\theta_{\max}=50$, $\acute{\xi}=10$ and $10\leq\Delta\tilde{Q}_{n,n+2},\Delta\tilde{Q}_{n+1,n+2}\leq 20$.
For the adaptive estimation algorithm to run most effectively, the values for some of the starting variables were set to their suitable bounds, that is, $\underline{\gamma}(0)=0$, $\acute{\gamma}(0)=1$ and $\gamma_{\min}(0)=1$.
The other relevant starting variables were $\gamma(0)=1$, $\theta_0=40$, $\Delta=0.05$ and $\Delta'=0.0001$. The exact value of $\gamma_{\min}$ for this simulation was set to 0.7.

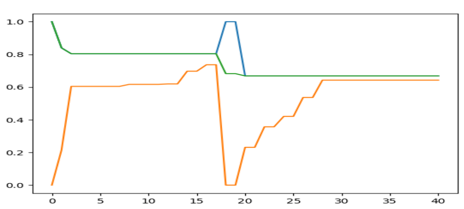


Fig. 6: $\gamma(n)$ vs n. Adaptive estimation for γ. The blue, green and orange lines represent $\acute{\gamma}(n)$, $\gamma(n)$ and respectively. The algorithm yields $\gamma=0.67$, which is a reasonable estimation based on the exact value of $\gamma_{\min}$.

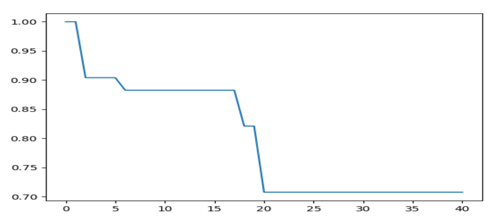
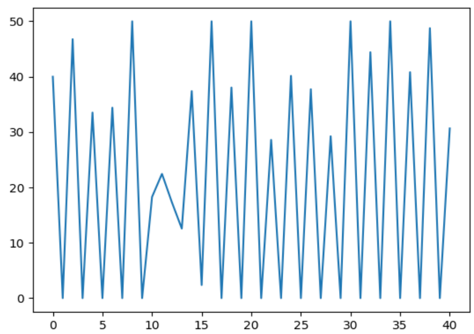


Fig. 7: $\gamma_{\min}(n)$ vs n. Adaptive estimation for $\gamma_{\min}$. The algorithm yields $\gamma_{\min}=0.71$, which is a reasonably accurate estimation since it deviates only slightly from the exact value of $\gamma_{\min}$.



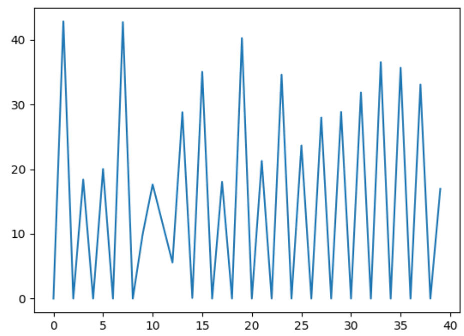


Fig. 9. Q_n vs n. Post adaptation production volume.

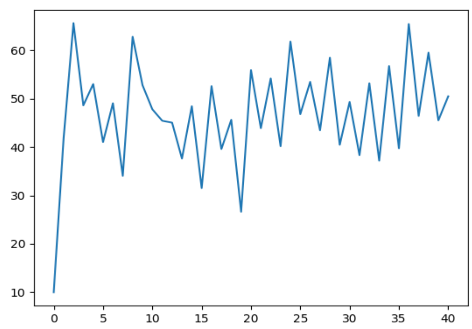


Fig. 10. H_n vs n. Post adaptation stock level.

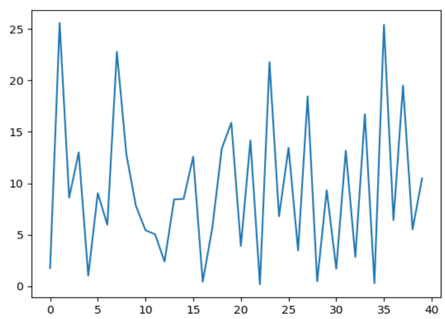


Fig. 11. e_n vs n. Deviation of product stock level from safety stock value post adaptation.

Conclusion

The adaptive estimation algorithm for γ and $\gamma_{\min}$ can be used for developing inventory control systems that exploits these unknown constants. The proposed algorithm is expected to work with any reorder policy, however more research has to be put into this matter.

References

[1] Aharon, Ben-Tal, Golany Boaz, and Shtern Shimrit. ‘Robust Multi-Echelon Multi-Period Inventory Control’. European Journal of Operational Research 199, no. 3 (2009): 922–35. https://doi.org/10.1016/j.ejor.2009.01.058.

[2] Axsater, Sven. ‘Control Theory Concepts in Production and Inventory Control’. International Journal of Systems Science 16, no. 2 (1985): 161–69. https://doi.org/10.1080/00207728508926662.

[3] Azarskov, Valerii N., Leonid S. Zhiteckii, Klavdia Yu Solovchuk, Olga A. Sushchenko, and Roman O. Lupoi. ‘Inventory Control for a Manufacturing System under Uncertainty: Adaptive Approach’. IFAC-PapersOnLine 50, no. 1 (2017): 10154–59. https://doi.org/10.1016/j.ifacol.2017.08.1762.

[4] Azarskov, V. N., L. S. Zhiteckii, R. O. Lupoi, and O. O. Oliinyk. ‘On a Decision-Making Problem within the Man-Machine Control System of an Enterprise’. Kibernetika i Vychislitel’naya Tekhnika 151 (2006): 49–59.

[5] Azarskov, Valerii N., Vladimir I. Skurikhin, Leonid S. Zhiteckii, and Roman O. Lypoi. ‘Modern Control Theory Applied to Inventory Control for a Manufacturing System’. IFAC Proceedings Volumes 46, no. 9 (2013): 1200–1205. https://doi.org/10.3182/20130619-3-ru-3018.00463.

[6] Bauso, Dario, Franco Blanchini, and Raffaele Pesenti. ‘Robust Control Strategies for Multi–Inventory Systems with Average Flow Constraints’. Automatica 42, no. 8 (2006): 1255–66. https://doi.org/10.1016/j.automatica.2005.12.006.

[7] Boukas, E. K. ‘Manufacturing Systems: LMI Approach’. IEEE Transactions on Automatic Control 51, no. 6 (2006): 1014–18. https://doi.org/10.1109/tac.2006.876945.

[8] Buchan, Joseph, and Ernest Koenigsberg. Scientific Inventory Management. Prentice-Hall, 1963.

[9] Grubbström, Robert W., and Joakim Wikner. ‘Inventory Trigger Control Policies Developed in Terms of Control Theory’. International Journal of Production Economics 45, no. 1–3 (1996): 397–406. https://doi.org/10.1016/0925-5273(96)00018-7.

[10] Hennet, Jean-Claude. ‘A Bimodal Scheme for Multi-Stage Production and Inventory Control’. Automatica 39, no. 5 (2003): 793–805. https://doi.org/10.1016/s0005-1098(03)00026-8.

[11] Hoberg, Kai, James R. Bradley, and Ulrich W. Thonemann. ‘Analyzing the Effect of the Inventory Policy on Order and Inventory Variability with Linear Control Theory’. European Journal of Operational Research 176, no. 3 (2007): 1620–42. https://doi.org/10.1016/j.ejor.2005.10.040.

[12] Ignaciuk, Przemysław, and Andrzej Bartoszewicz. ‘Linear–Quadratic Optimal Control Strategy for Periodic-Review Inventory Systems’.

Automatica 46, no. 12 (2010): 1982–93. https://doi.org/10.1016/j.automatica.2010.09.010.

[13] Kostić, Konstantin. ‘Inventory Control as a Discrete System Control for the Fixed-Order Quantity System’. Applied Mathematical Modelling 33, no. 11 (2009): 4201–14. https://doi.org/10.1016/j.apm.2009.03.004.

[14] Kuntsevich, V. M. ‘On a Problem of Control in the Automated Control Systems of Enterprises’. Upravlyayushchie Sistemy i Mashiny 5 (1973): 14–15.

[15] Landau, Yoan D., R. Lozano, Mohammed M’Saad, and Alireza Karimi. Adaptive Control. Springer, 2013.

[16] Rodrigues, Luis, and El-Kebir Boukas. ‘Piecewise-Linear Controller Synthesis with Applications to Inventory Control of Switched Production Systems’. Automatica 42, no. 8 (2006): 1245–54. https://doi.org/10.1016/j.automatica.2006.04.004.

[17] Qu, Zhihua. Robust Control of Nonlinear Uncertain Systems. Wiley, 1998.

[18] Shin, Joonho, Jongku Lee, Seungyoung Park, Kee-Kahb Koo, and Moonyong Lee. ‘Analytical Design of a Proportional-Integral Controller for Constrained Optimal Regulatory Control of Inventory Loop’. Control Engineering Practice 16, no. 11 (2008): 1391–97. https://doi.org/10.1016/j.conengprac.2008.04.006.

[19] Simon, Herbert A. ‘On the Application of Servomechanism Theory in the Study of Production Control’. Econometrica 20, no. 2 (1952): 247. https://doi.org/10.2307/1907849.

[20] Taleizadeh, Ata Allah, Seyed Taghi Niaki, and Mir-Bahador Aryanezhad. ‘A Hybrid Method of Pareto, TOPSIS and Genetic Algorithm to Optimize Multi-Product Multi-Constraint Inventory Control Systems with Random Fuzzy Replenishments’. Mathematical and Computer Modelling 49, no. 5–6 (2009): 1044–57. https://doi.org/10.1016/j.mcm.2008.10.013.

[21] Towill, D. R., G. N. Evans, and P. Cheema. ‘Analysis and Design of an Adaptive Minimum Reasonable Inventory Control System’. Production Planning & Control 8, no. 6 (1997): 545–57. https://doi.org/10.1080/095372897234885.

[22] Wiendahl, Hans-Peter, and Jan-Wilhelm Breithaupt. ‘Automatic Production Control Applying Control Theory’. International Journal of Production Economics 63, no. 1 (2000): 33–46. https://doi.org/10.1016/s0925-5273(98)00253-9.

[23] Yokoyama, T. ‘A Note on Discrete Servo Theory in Inventory Control’. Industrial Administration Research Memorandum 1 (1955): 1–3.

[24] Skurikhin, V. I. ‘On Dynamical Processes Arising in the Automated Control Systems of Enterprises’. Upravlyayushchie Sistemy i Mashiny 1 (1972): 35–41.

[25] Zhiteckii, L. S. ‘Adaptive Control of Systems Subjected to Bounded Disturbances’. In Bounding Approaches to System Identification, 383–407. Plenum Press, 1996.

[26] Zhiteckii, L. S., R. O. Lupoi, and O. O. Oliinyk. ‘On Dynamical Processes in Controlling a Typical Junction of Manufacturing Network of a Factory with Discrete Production’. Upravlyayushchie Sistemy i Mashiny 2 (2007): 39–47.

Appendix A Simulation Source Code

```
import random
from matplotlib import pyplot as plt
from statistics import mode


def thetan(r0, h_n, qn_n1, qn1_n2, theta_max, theta_n1,
gamma_n):
    e_n = r0 - h_n
    theta_c = (e_n + qn_n1 + qn1_n2 - gamma_n *
theta_n1)/gamma_n
    # theta_c = (e_n + 30 - gamma_n * theta_n1) /
gamma_n
    if theta_c > theta_max:
        return theta_max
    elif theta_c < 0:
        return 0
    else:
        return theta_c


def gammanLower(gamman_l, theta_n, epsilon_n, q_n):
    if theta_n != 0:
        beta = (q_n - epsilon_n)/theta_n
        if gamman_l >= beta:
            return gamman_l
        else:
```

```
            return beta
    else:
        return gamman_l


def gammanUpper(gamman_u, theta_n, epsilon_n, q_n):
    if theta_n != 0:
        beta = (q_n + epsilon_n)/theta_n
        if gamman_u <= beta:
            return gamman_u
        else:
            return beta
    else:
        return gamman_u


def gammaMin1(gamma_min, q_n1, xi_max, theta_n):
    if theta_n != 0:
        if gamma_min <= (q_n1 + xi_max)/theta_n:
            return gamma_min
        else:
            return (q_n1 + xi_max)/theta_n
    else:
        return gamma_min


def gammaMin2(gamma_min, gamman_l, gamman_u, delta):
    if gamman_l <= gamman_u:
        return gamma_min
    else:
        return gamma_min - delta


def gamman(gamma_n, theta_n, q_n1, epsilon, delta):
    if theta_n != 0:
        s_n = gamma_n*theta_n - q_n1
        epsilon_prime = epsilon + delta
        if abs(s_n) <= epsilon_prime:
            return gamma_n
        elif s_n > epsilon_prime:
            return gamma_n - (s_n - epsilon)/theta_n
        else:
            return gamma_n - (s_n + epsilon)/theta_n
```

```
    else:
        return gamma_n


def main():
    r0 = 40
    theta_max = 50
    xi_max = 10
    theta_n = 40
    dq_lower = 10
    dq_upper = 20
    qn_n1 = random.randint(dq_lower, dq_upper)
    gamma_min_n = 1
    real_gamma_min = 0.7
    gamman_l = 0
    gamman_l0 = gamman_l
    gamman_u = 1
    gamman_u0 = gamman_u
    gamman_n = 1
    delta = 0.05
    delta_prime = 0.0001
    upper = [gamman_u]
    lower = [gamman_l]
    gamma_n = [gamman_n]
    theta_n_list = [theta_n]
    q_list = []
    ht_0 = 10
    ht_list = [ht_0]
    e_list = []
    gamma_min_list = [gamma_min_n]
    for i in range(40):
        qn1_n2 = random.randint(dq_lower, dq_upper)
        theta_n = thetan(r0, ht_0, qn_n1, qn1_n2,
theta_max,
                         theta_n, gamman_n)
        # theta_n_list.append(theta_n)
        epsilon_n = ((1 - gamma_min_n) * theta_n +
xi_max)/2
        if theta_n != 0:
            noise = random.randint(0, xi_max)
            q_n1 = theta_n *
random.uniform(real_gamma_min, 1)
            while q_n1 - noise < 0:
                noise = random.randint(0, xi_max)
```

```
            q_n1 = q_n1 - noise
        else:
            q_n1 = 0
        # q_list.append(q_n1)
        gamman_u = gammanUpper(gamman_u, theta_n,
epsilon_n, q_n1)
        gamman_l = gammanLower(gamman_l, theta_n,
epsilon_n, q_n1)
        if gamman_l > gamman_u:
            gamman_l = gamman_l0
            gamman_u = gamman_u0
        upper.append(gamman_u)
        lower.append(gamman_l)
        gamma_min_n1 = gammaMin1(gamma_min_n, q_n1,
xi_max, theta_n)
        gamma_min_n2 = gammaMin2(gamma_min_n, gamman_l,
gamman_u, delta)
        if gamma_min_n >= min(gamma_min_n1,
gamma_min_n2):
            gamma_min_n = min(gamma_min_n1,
gamma_min_n2)
        gamma_min_list.append(gamma_min_n)
        gamman_n = gamman(gamman_n, theta_n, q_n1,
                          epsilon_n, delta_prime)
        gamma_n.append(gamman_n)
        ht_0 = ht_0 - qn1_n2 + q_n1
        # ht_list.append(ht_0)
        # e_list.append(abs(r0 - ht_0))
        # qn_n1 = qn1_n2

    gamman_n = mode(gamma_n)

    for i in range(40):
        qn1_n2 = random.randint(dq_lower, dq_upper)
        theta_n = thetan(r0, ht_0, qn_n1, qn1_n2,
                         theta_max, theta_n, gamman_n)
        theta_n_list.append(theta_n)
        if theta_n != 0:
            noise = random.randint(0, xi_max)
            q_n1 = theta_n *
random.uniform(real_gamma_min, 1)
            while q_n1 - noise < 0:
                noise = random.randint(0, xi_max)
            q_n1 = q_n1 - noise
```

```
        else:
            q_n1 = 0
        q_list.append(q_n1)
        ht_0 = ht_0 - qn1_n2 + q_n1
        ht_list.append(ht_0)
        e_list.append(abs(r0 - ht_0))
        qn_n1 = qn1_n2

    print(upper)
    print(lower)
    print(gamma_n)
    print(gamman_n)
    print(mode(gamma_min_list))

    plt.plot(upper)
    plt.plot(lower)
    plt.plot(gamma_n)
    plt.show()
    plt.plot(theta_n_list)
    plt.show()
    plt.plot(q_list)
    plt.show()
    plt.plot(ht_list)
    plt.show()
    plt.plot(e_list)
    plt.show()
    plt.plot(gamma_min_list)
    plt.show()


if __name__ == "__main__":
    main()
```