Kanban-Controlled Production System

Simulation Analysis and Scenario Comparison Report

Model: DALGA.doe (Arena) • Experiment: Process Analyzer, 8 scenarios

Prepared by Hasan Çinar

Contents

1. Executive Summary

This report analyzes a five-station, kanban-controlled production line modeled in Arena. Eight scenarios were run in Process Analyzer, each changing a single kanban capacity from the baseline configuration (K1–K5 = 4, 3, 3, 5, 3) over a 10-day replication.

Key result: the system is remarkably robust. Increasing KANBAN 1, or decreasing KANBAN 3 or KANBAN 5, has no effect on total output (2,584 units). Only two cards act as constraints: reducing **KANBAN 4** from 5 to 4 causes the largest loss (−52 units, −2.0%), and any change to **KANBAN 2** also reduces output (−41 units when increased to 4, −16 when reduced to 2).

Recommendation: keep KANBAN 4 at 5 and KANBAN 2 at 3; KANBAN 1, 3, and 5 can likely be reduced to lower WIP at no throughput cost — pending confirmation with multiple replications.

2. System and Model Description

The Arena model (DALGA.doe, with Gerçek PROJE.doe as an earlier revision of the same model) represents a serial production line with the following elements:

- **Five stations** (STATION 1–5), each with a dedicated resource (Resource 1–5) and material movement between stations via routing and two conveyors.
- **Five kanban control loops** (KANBAN 1–5), each implemented with SEIZE/RELEASE logic: a part must acquire a kanban card before entering a stage and releases it downstream, capping the WIP allowed at each stage.
- **Match/Batch logic** to synchronize kanban cards with production orders.
- **Machine failures and a REPAIRMAN resource** (Failure 1), so stations experience downtime and repair delays.
- **Stochastic processing times**, e.g. TRIA(4, 6, 8), UNIF(8, 12), UNIF(16, 24), UNIF(20, 22), and EXPO(12) arrivals.

Because kanban capacities cap the number of parts in each stage, they are the natural experimental levers: too few cards starve downstream stations, too many allow WIP and blocking to build up.

3. Experimental Design

The experiment follows a **one-factor-at-a-time (OFAT)** design in Process Analyzer (PAN report dated Aug 19, 2025). Scenario 6 is the baseline; Scenarios 2–5, 7 and 8 each change exactly one kanban level. Each scenario ran 1 replication of 864,000 time units (10 days of continuous operation in seconds). The response variable is **System.NumberOut** — total finished units.

3.1 Scenario Settings and Results

S	Scenario	K1	K2	K3	K4	K5	Total Output	Δ vs Base
1	Baseline (not run)	4	3	3	5	3	—	—
2	K1 → 5	5	3	3	5	3	2,584	0
3	K2 → 4	4	4	3	5	3	2,543	−41
4	K2 → 2	4	2	3	5	3	2,568	−16
5	K3 → 2	4	3	2	5	3	2,584	0
6	**Baseline (run)**	4	3	3	5	3	2,584	ref
7	K4 → 4	4	3	3	4	3	2,532	−52
8	K5 → 2	4	3	3	5	2	2,584	0

Table 1. Kanban settings and total output per scenario (source: Process Analyzer report).

4. Results and Analysis

4.1 Total Output by Scenario

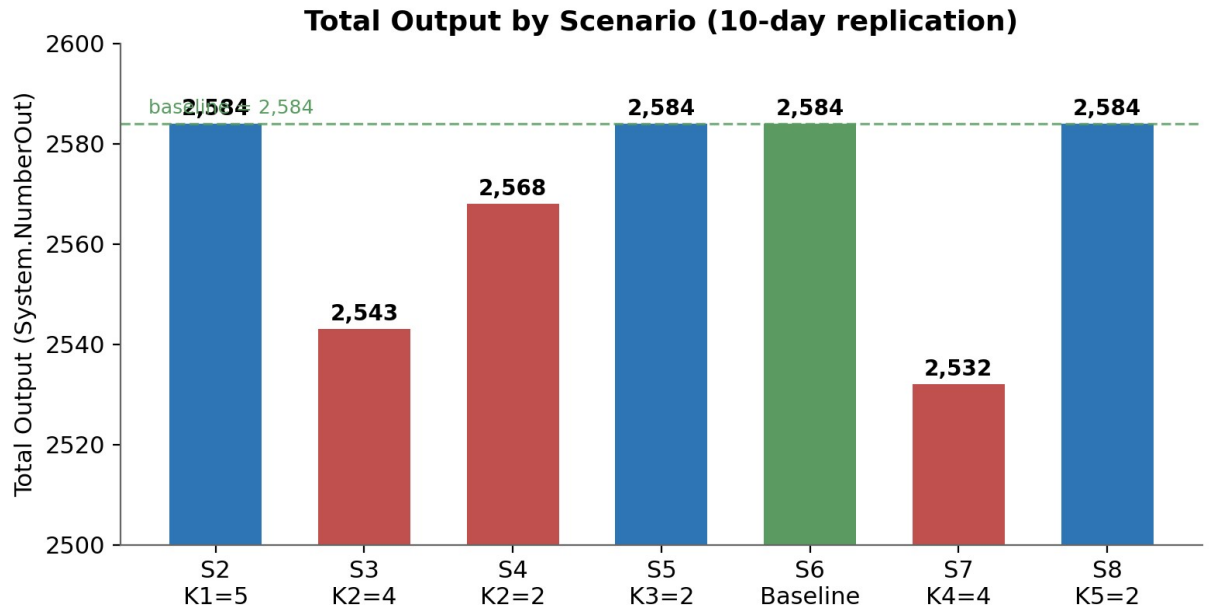

Figure 1. Total output per scenario. Four of seven runs match the baseline exactly.

4.2 Sensitivity to Individual Kanban Changes

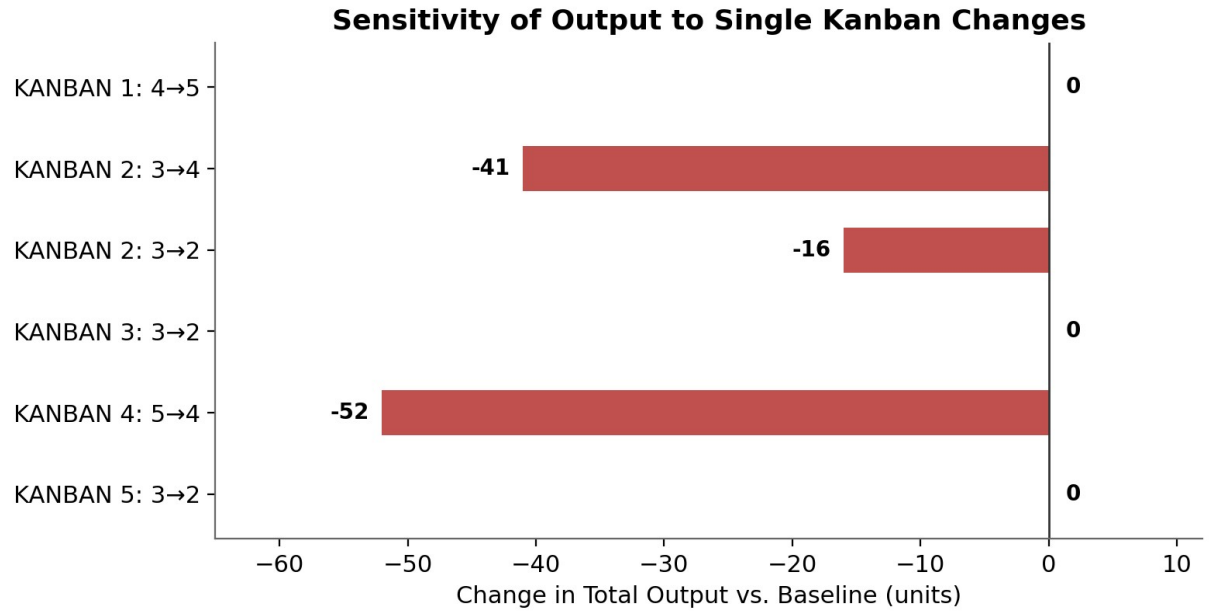

Figure 2. Change in output relative to baseline for each single-kanban perturbation.

4.3 Interpretation

- **KANBAN 4 is the binding constraint.** Removing one card (5 → 4) costs 52 units (−2.0%), the largest single effect. Stage 4 is the bottleneck loop: its card count directly limits flow, so capacity there should not be reduced.
- **KANBAN 2 is tuned near its optimum.** Both directions hurt: raising it to 4 lost 41 units, cutting it to 2 lost 16. The loss from adding a card is counterintuitive and suggests extra WIP at stage 2 creates blocking upstream of the bottleneck — although with a single replication, random variation cannot be ruled out.
- **KANBAN 1, 3, and 5 have slack.** Adding a card at stage 1, or removing one at stages 3 or 5, changed output by exactly 0 units. These loops are not constraining the system at current levels.
- **Overall stability.** The full range across all scenarios is only 2,532–2,584 units (2.0%), meaning the kanban configuration is already close to a local optimum for throughput.

5. Recommendations

- **Protect the bottleneck:** keep KANBAN 4 at 5 cards (or test 6 to see whether output improves further).
- **Hold KANBAN 2 at 3:** both tested deviations reduced output.
- **Trim WIP where it is free:** KANBAN 3 and KANBAN 5 can likely drop to 2 cards with no output loss, reducing inventory and cycle time. KANBAN 1 does not need to increase.

- **Candidate configuration to test next:** K1–K5 = 4, 3, 2, 6, 2 — combining the free WIP reductions with one extra bottleneck card.
- **Add replications:** rerun the experiment with at least 10 replications per scenario and compare confidence intervals; single-replication differences of 16–52 units may partly reflect randomness. Also record WIP, cycle time, and resource utilization as responses, not just output.

6. Limitations

Each scenario was run once, so no statistical significance can be attached to the observed differences. Scenario 1 was configured but not executed (0 replications); Scenario 6 serves as the baseline reference. The experiment varies kanban levels only — resource capacities, failure/repair parameters, and arrival rates were held constant. Earlier draft reports accompanying this project contained illustrative placeholder figures; all numbers in this report come exclusively from the actual Process Analyzer output.