**Задание 1**

*Дано*

Данные хранятся в блоках памяти. Пользователь отправляет команды чтения данных из блоков в некотором порядке. Некоторые из этих команд можно объединить в кластеры, т.е. в группы команд в которых блоки часто читаются последовательно (друг за другом). Пусть уже известны 2 алгоритма кластеризации команд, но неизвестно как сравнить какой из алгоритмов для множества последовательностей команд.

*Пример*

Последовательность команд: 2 5 7 2 5 3 8 7 2 5 7 2 3 5 7 2 5 7

Алгоритм А1 может выделить кластер 2 5 7

Алгоритм А2 может выделить кластер 5 7 2 5

*Найти*

предложить метрику для оценки качества алгоритмов кластеризации в двух случаях:

a) ограничения на память и вычислительные ресурсы отсутствуют,

b) есть ограничения на память и вычислительные ресурсы.

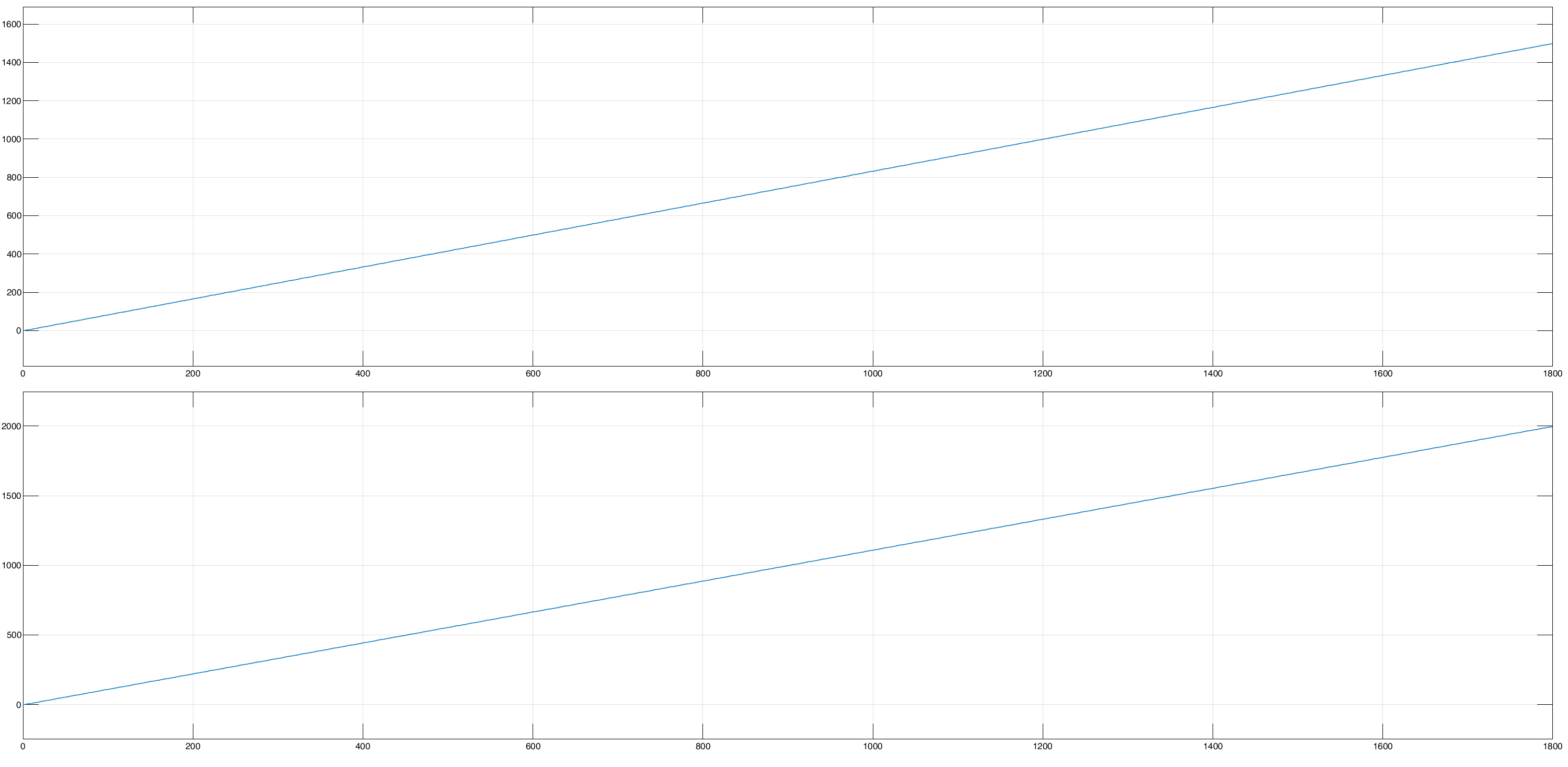
(написать явно формулу и/или алгоритм вычисления метрики)

*Решение*

Очевидно, что более эффективный кластер тот, который имеет наибольшую плотность совпадений в реальной последовательности команд. Выявить частоту таких совпадений можно используя метрику Хемминга – поскольку в данном случае важным является полное совпадение команд, а не их близость в численном эквиваленте. Теоретически длина последовательности команд ***x*** бесконечна, а кластер ***c*** имеет конечную длину *L* представляющего его числового вектора – значит измерение расстояния Хэмминга является непрерывным процессом на каждом шагу поступления новой команды. Блок измерения расстояний должен быть инвариантен ко времени и по сути является структурой с конечным импульсным откликом каждый *n*-й выход которого определяется выражением:

Также стоит учитывать, что частичные совпадения (т.е. не нулевые расстояния не равные максимальному значению *L*) также имеют значения, а значит более эффективный кластер тот, который имеет большую площадь под графиком процесса на выходе блока измерения расстояний. Если нет ограничения на память и вычислительные ресурсы, то можно интегрировать (накапливать) выход блока измерения расстояний бесконечно и иметь таким образом оценку эффективности того или иного кластера:

Так для примера в задании можем установить, что кластер А2 более эффективен:

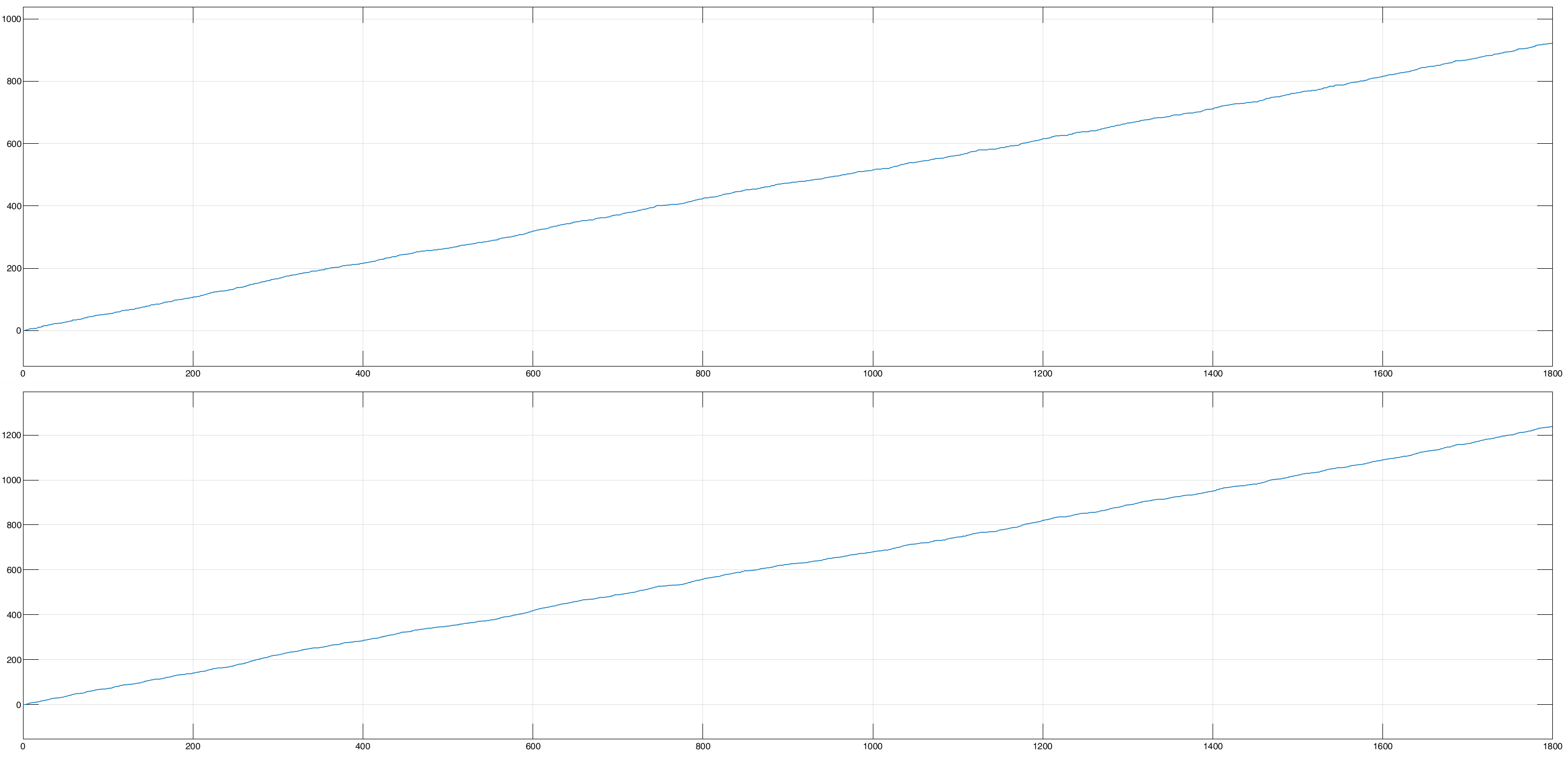


А2

А1

Рисунок 1 – Эксперимент для примера в задании для 1800 шагов (последовательность команд циклически повторяется)

Вообще алгоритм А2 оказывается более эффективным и при случайном поступлении команд с равномерным распределением между 2 и 8:



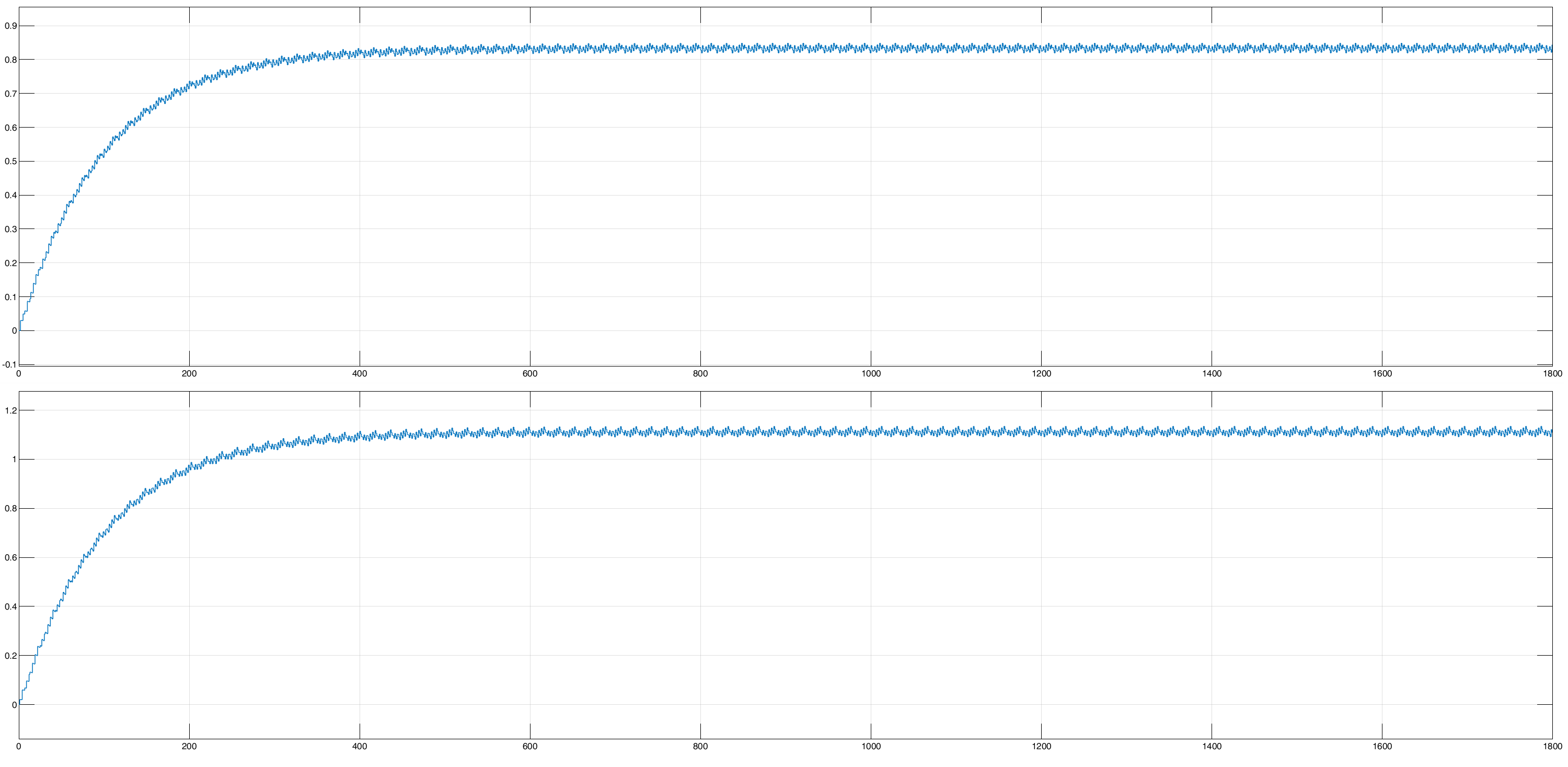
А2

А1

Рисунок 2 – Эксперимент для примера в задании для 1800 шагов (последовательность команд случайна между 2 и 8)

Когда имеет место ограничение на память и вычислительные ресурсы мы не можем накапливать бесконечно, тогда необходимо либо периодически сбрасывать аккумулятор, либо можно применить усреднение, например, вычисление экспоненциально среднего значения:

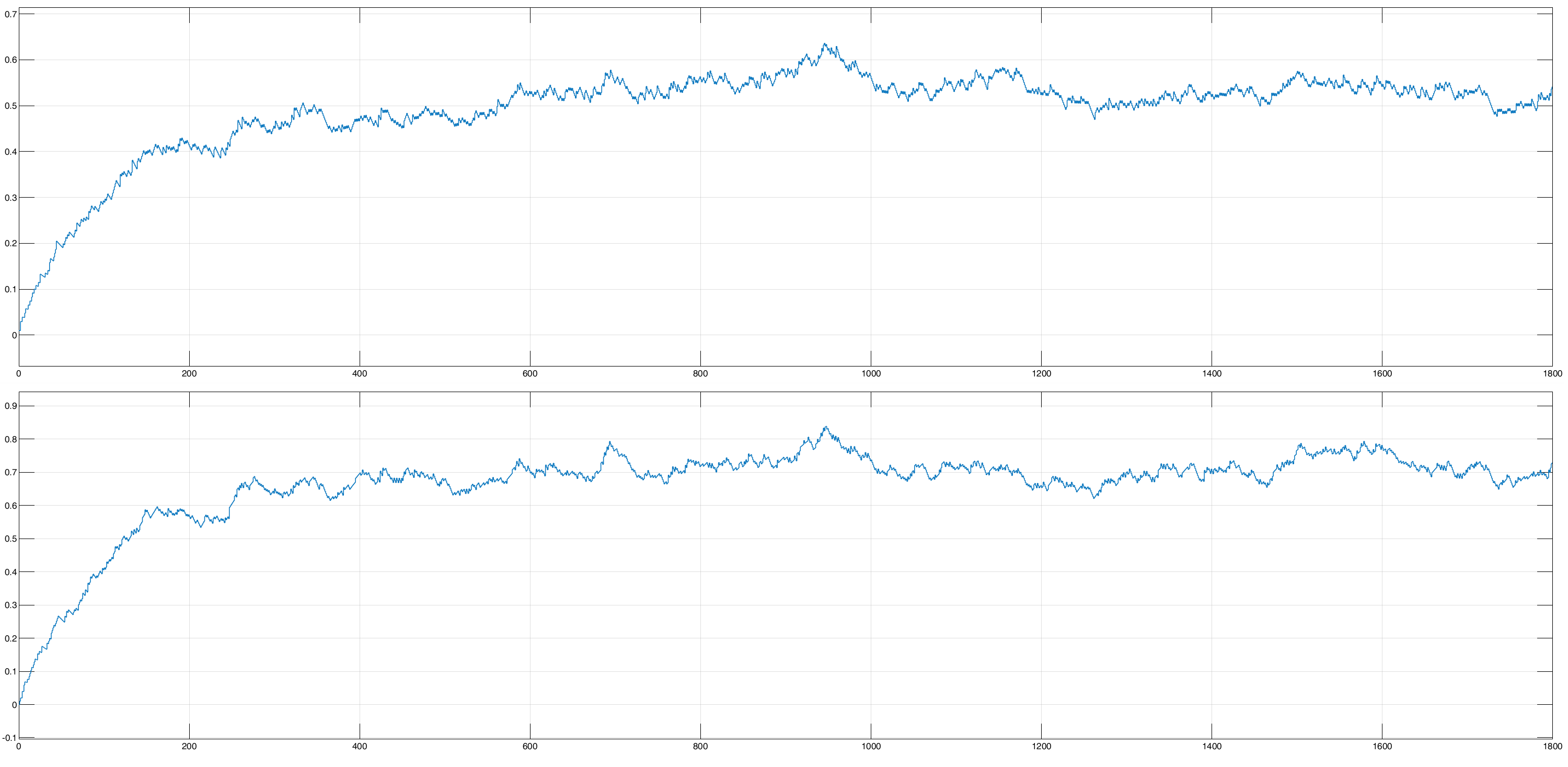
В таком варианте видно, что более эффективным оказался кластер А2 – как для последовательности из примера, так и для случайной последовательности с коэффициентом экспоненциального фильтра 0,01:



А2

А1

Рисунок 1 – Эксперимент для примера в задании для 1800 шагов (последовательность команд циклически повторяется)



А2

А1

Рисунок 2 – Эксперимент для примера в задании для 1800 шагов (последовательность команд случайна между 2 и 8)