Coding #03: Z-Transforms

2025/9/17

10 Points Possible

Attempt 1

In Progress

NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed

2025/9/10 to 2025/9/24

Details

🗃️ Case File #3: *The Trail*

Situation:

The suspects have gone dark — but our sensors caught a faint signal trail during the car chase. Unfortunately, the readings are corrupted by noise and dropouts. If we don't track this trail in real-time, we lose them.

Our analysts have extracted the measurements using `get_trail.p`.

```
z = get_trail('#########')
```

Replace `########` with your **UFID** to retrieve your mission data:

The data we receive (`z`) is a noisy, dropout-ridden observation of the *true hidden trail* (`x`). Our job is to design **real-time filters** to estimate the true path as it unfolds.

Key Constraint:
All solutions must be implemented as **difference equations** (real-time updates). Do **not** use built-in convolution or filtering functions (`conv`, `filter`, `movmean`, etc.).

You will implement three separate filters, each as its own MATLAB function.

Required Tools

`get_trail` function: [get_trail.p (https://ufl.instructure.com/courses/540008/files/99726592?wrap=1)](https://ufl.instructure.com/courses/540008/files/99726592?wrap=1) (https://ufl.instructure.com/courses/540008/files/99726592/download)

Problem 1: Running Average Filter

You will implement a 3-point running average filter.

Function signature:

```
function y = running_average_filter(x0,x1,x2)
```

- **Inputs**
  - `x0` : Most recent input sample (current).
  - `x1` : Previous input sample.
  - `x2` : Input sample two steps ago.
- **Output**
  - `y` : Filtered output (scalar).

Notes:

This filter estimates the current trail point by averaging the most recent three observations. It is the simplest real-time smoother.

Problem 2: Integrator-Type Filter

You will implement a first-order recursive filter.

Function signature:

```
function y = integrator_filter(x,y0,a2)
```

- **Inputs**
  - `x` : Current input sample (scalar).
  - `y0` : Previous output sample (scalar).
  - `a2` : Filter parameter, chosen by you (0 < a2 < 1).
- **Output**
  - `y` : Current output sample (scalar).

Notes:

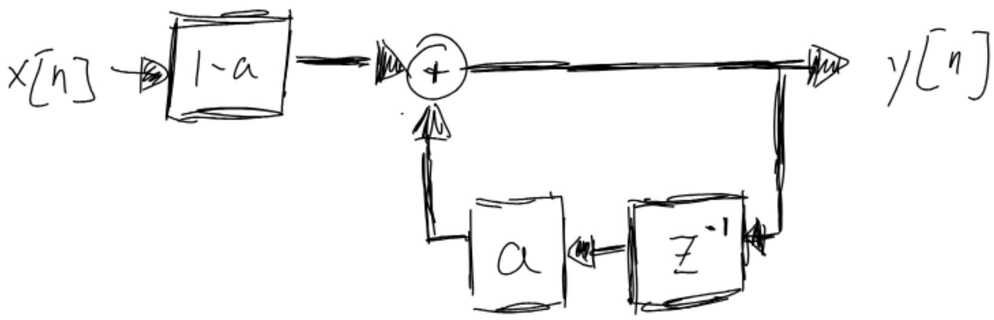
- The value of `a2` is chosen by “guess and check” to make the filtered signal look good when plotted.
- Think of this filter as a *weighted integrator* that blends past estimates with new information.

Block diagram (conceptual):

- Input `x` goes into a weighted summer with the delayed output `y0`.

- The summer produces the new `y`.

Building blocks in the z-domain:



You must construct the filter's update equation using only these building blocks.

Problem 3: Adaptive Filter

You will implement an adaptive filter that learns to correct itself in real-time.

Function signature:

```
function [y,s] = adaptive_filter(x,y0,s0,a3)
```

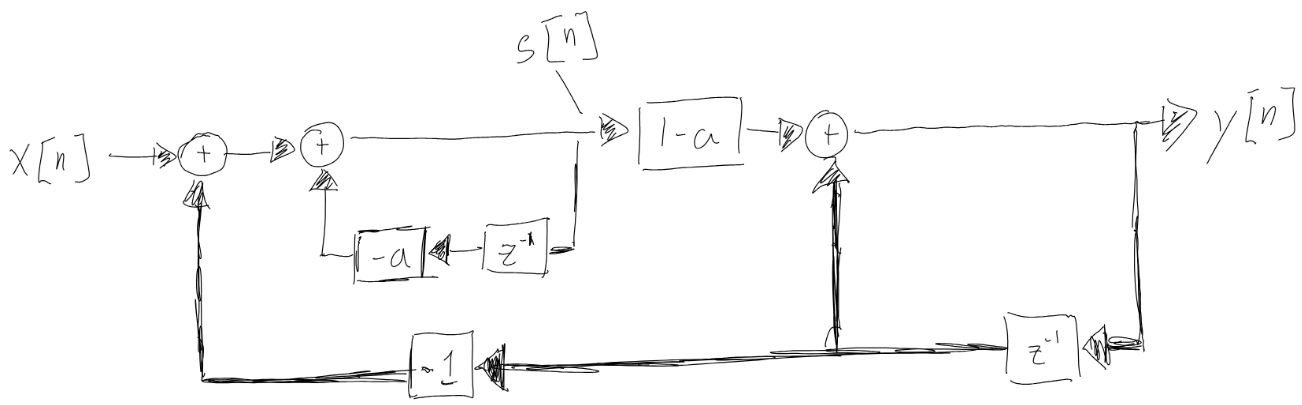
- **Inputs**
  - `x` : Current input sample (scalar).
  - `y0` : Previous output sample (scalar).
  - `s0` : Previous slope estimate (scalar).
  - `a3` : Filter parameter, chosen by you (0 < a3 < 1).
- **Outputs**
  - `y` : Current output sample (scalar).

- `s` : Updated slope estimate (scalar).

Notes:

- This filter predicts the trail's next value, compares it against the actual measurement, and then corrects itself using an adaptive slope term.
- Again, `a3` should be tuned by visual inspection of your plots.

Building blocks in the z-domain:



Your task is to connect these blocks to form the adaptive filter update.

Submission Instructions

- Submit your three functions as:
  - `running_average_filter.m`
  - `integrator_filter.m`
  - `adaptive_filter.m`
- Submit your results in a single `.mat` file (filename is not important, but for consistency, you may use `case3_results.mat`).

This `.mat` file should contain the variables:

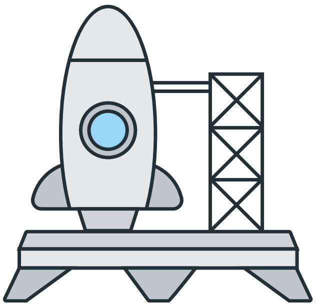
- `y_avg` : Output of the running average filter.
- `y_int` : Output of the integrator filter.
- `y_adapt` : Output of the adaptive filter.
- `a2` : a value chosen for Problem 2.
- `a3` : a value chosen for Problem 3.

Choose a submission type

Upload

Google Drive

More



Choose a file to upload

File permitted: MAT, M

or

Canvas Files