Decentralized Fusion Planning for Scalability in Distributed Satellite Tracking

ASEN 6519: Optimization, Final Project

Nolan Stevenson

1 Introduction and Background

The purpose of this project is to develop an optimization algorithm that solves the decentralized fusion planning problem for scalability in distributed satellite tracking.
Consider the following situation:

- You have a network of LEO satellites, about 200+ total satellites. The satellite network must autonomously track targets, about 50+, on the Earth's surface.
- The satellite network is seperated into two different types of satellites, sensing and fusion satellites.
- The sensing satellites are able to collect measurements on targets using bearings only sensors.
- The sensing layer communicates the measurements to the fusion layer, which is responsible for fusing the measurements and tracking the targets, using extended kalman filters and distrbuted fusion algorithms.
- The fusion satellites are limited both computationally on the number of target tracks they can handle at a time, and in communication bandwidth between neighboring satellites.
- The entire satellite network must be able to autonomously operate to perform target tracking, this includes: tasking, sensing, fusion, communication, and control.

The following diagram shows the basic structure of the satellite network; a fusion layer, a sensing layer, and objective targets to track.

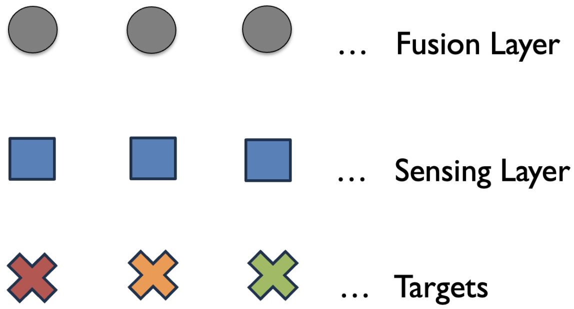

Figure 1: Satellite Network Diagram

In order to scalabily track this many targets in a satellite network, it is best to use a technique called "federated tracking". This is a system where each track is essentially treated as a centralized extended kalman filter (EKF). Thus, there is a single fusion node responsible for the EKF of each target. However, the challenge with this formulation is deciding which fusion node should be resposible for which target? I will introduce the following terms to help describe the problem:

- **Custody:** The fusion node that is responsible for the federated EKF for a particular target.
- **Bounty:** The protocol the sensing nodes use to send measurements about a target to the fusion layer.

With these two terms, there is an algorithm which assigns custody of targets to the fusion nodes. The following diagram represents this algorithm:

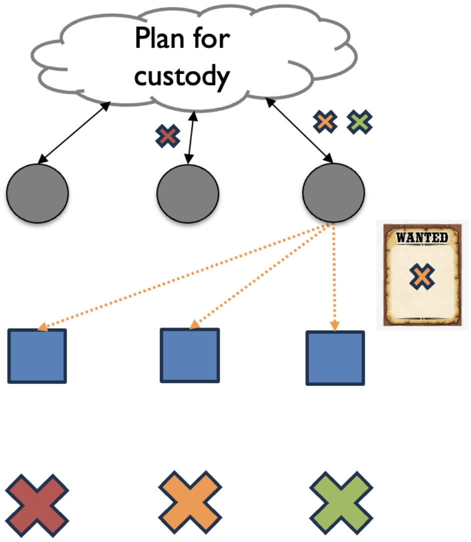

Figure 2: Custody Assignment Algorithm

This diagram shows the fusion layer nodes having a algorithm that decides who should have ”custody” of which targets. The fusion node with custody of a target is responsible for running the ”federated/centralized EKF” on that target for the next X amount of time. Once custody is assigned, the fusion nodes send out a ”bounty” on that target to the relavent sensing nodes. This bounty essentially says ”if you see this target, report back to me”.
This is the custody/bounty system I will refer to for the rest of the project. The goal of this project is to develop an optimization algorithm to plan for custody.

The following are goals of the optimization algorithm:

- All targets must be tracked for the next X amount of time.
- The computational limit of a fusion node cannot be exceeded (per node, maximum number of EKFs/custody‘’).
- The amount of communication throughout the network should be minimized (not hard limit, computation is limiting factor in this environment).
- Whenever the custody of a target is changed, track to track handoff must occur between old owner and new owner.

2 Problem Formulation

This problem can be formulated as a mixed-integer linear program (MILP), consisting of three key components:

1. Decision variables: These represent the possible choices in the optimization problem.
2. Objective function: This is the function to be maximized or minimized.
3. Constraints: These are the limitations or requirements that must be satisfied.

The objective function and constraints are expressed in linear combination of the decision variables, creating a mathematical formulation suitable for a linear program solver. In this case, we will use the Python package PuLP, which can efficiently solve MILPs using a branch and bound algorithm.

The problem can be written as:

$$\underset{x_{st}}{\text{minimize}} \sum_s \sum_t c_{st} x_{st} \tag{1}$$

subject to

$$x_{st} \in \{0, 1\} \tag{2}$$

$$\sum_s x_{st} \geq 1 \quad \forall t \tag{3}$$

$$\sum_t x_{st} \leq Comp(s) \quad \forall s \tag{4}$$

In the formulation above, s is the index for a fusion satellite, and t is the index for a target.
Equation (1) is the objective function, which is to minimize the cost of assigning custodies throughout the network (more dicussion on that cost below).
Equation (2) is the binary constraint, which ensures that the decision variable x_{st} is either 0 or 1. If $x_{st} = 1$, then fusion satellite s has custody of target t.
Equation (6) ensures that every target has at least one fusion satellite assigned to it. This means that each target is being tracked in the network.
Equation (4) ensures that no fusion satellite is assigned more targets than its computational limit, defined as $Comp(s)$.

The cost of assigning custody of a target to a fusion satellite is defined as c_{st}. A good cost function is one that minimizes the amount of communication throughout the network. Since the goal of the algorithm is to assign custodies for some short horizon into the future, the cost function needs to account for current and future communication needs. A simple choice for the cost function over some planning horizon T is:

$$c_{st} = \sum_{k=0}^{T} \|\mathbf{r}_k^{target} - \mathbf{r}_k^s\| \tag{5}$$

The cost function is the sum of the distance between the estimated target position (from the EKF track) and the fusion satellite position (known due to orbit params) over a discrete set of points from current time into the future horizon. This cost function is a good proxy for communication cost because it is a measure of how far the measurement data needs to travel from the sensing satellites to the fusion satellite.

Thus, this optimization problem is a valid way to plan for custody of targets over some short horizon in the fusion layer without violating computational constraints while minimizing communication costs.

3 NEW CONTENT AND EQS

$$\sum_s x_{st} == \sum_s Capacity(s) \quad \forall t \tag{6}$$