AIAA Guidance, Navigation, and Control Conference and Exhibit
16 - 19 August 2004, Providence, Rhode Island

AIAA 2004-5251

Filter-Embedded UAV Task Assignment Algorithms for Dynamic Environments

Mehdi Alighanbari,* Luca F. Bertuccelli,† and Jonathan P. How‡

MIT Department of Aeronautics and Astronautics

Abstract

This paper presents a modified formulation of the classical task assignment problem that has recently been used to coordinate teams of UAVs. The main contribution is a version of the task assignment problem that can be used to tailor the control system to mitigate the effect of noise in the situational awareness on the solution. The net effect will be to limit the rate of change in the reassignment in a well-defined manner. The approach here is to perform reassignment at the rate that information is updated, which enables immediate reaction to any significant changes in the environment. We demonstrate that the modified formulation can be interpreted as a noise rejection algorithm that can be tuned to reduce the effect of variation in the uncertain parameters in the problem. Simulations are presented to demonstrate the effectiveness of this algorithm.

Keywords: UAV, Task Assignment, Sensing Noise, Churning.

I. Introduction

Future autonomous vehicles will be required to successfully operate in inherently dynamic and uncertain environments [1, 2]. The vehicles will be required to make both low-level control decisions, such as path planning, and high-level decisions, such as cooperative task assignment, based on uncertain and noisy information. While the impact of uncertainty on feedback control has been analyzed in detail in the controls literature, equivalent formulations to analyze this impact on the high-level planning processes have only recently been developed [3, 4]. Uncertainty will inherently propagate down from the high-level decisions to the lower-level ones, and thus it is very important to extend these tools and algorithms to provide new insights on the behavior of these real-time higher-level guidance and control algorithms in the face of uncertainty.

Task assignment *in the controls literature* has been generally viewed as an open-loop optimization with deterministic parameters. The optimization is generally done once (possibly made robust to uncertainty in the problem [5, 6, 7]), and task reassignment occurs only when substantial changes in the environment have been observed (*e.g.,* as a result of UAV loss or target re-classification [8, 9]). In reality, these information updates are continuously occurring throughout the mission due to vehicle sensing capabilities, adversarial strategies, and communicated updates of situational awareness (SA). The typical response to a change in the SA is to reassign the vehicles based on the most recent information. The problem of task reassignment due to changes in the optimization has been addressed by Kastner [10] in their use of incremental algorithms for

*Research Assistant, MIT Department of Aeronautics and Astronautics, mehdi_a@mit.edu
†Research Assistant, MIT Department of Aeronautics and Astronautics, lucab@mit.edu
‡Associate Professor, MIT Department of Aeronautics and Astronautics, jhow@mit.edu, Senior Member AIAA.

combinatorial auctions. They propose that the perturbed optimization problem should also include a term in the objective function that penalizes changes from the original solution. The work of Tierno and Khalak [3] also investigates the impact of replanning, with the objective function being a weighted sum of the mission goal and the difference between the previous plan and the current one. Both of these formulations rely on the plan generated prior to the current one as a reference, but they do not directly consider the impact of noise in the problem, nor do they develop techniques to mitigate its effect on the replanning.

The objective of this paper is to develop a modified formulation of the task assignment problem that mitigates the effect of noise in the SA on the solution. The approach taken here is to perform the reassignments at the rate the information is updated, which enables the planner to react immediately to any significant changes that occur in the environment. Also, rather than just limiting the rate of change of the plan, this new approach embeds a more sophisticated filtering operation in the task assignment algorithm. We demonstrate that this modified formulation can be interpreted as a noise rejection algorithm that reduces the effect of the high frequency noise on the planner. A key feature of this filter-embedded task assignment algorithm is that the coefficients of the filter are tuned online using the past information. Simulations are then presented to demonstrate the effectiveness of this algorithm.

II. Problem Statement

Consider the general weapon target assignment (WTA) problem expressed as a linear integer program (LIP). The following optimization can be solved to generate a plan, x_k at time k,

$$\begin{aligned} \max_{x_k} \quad & c_k^T x_k \\ \text{s.t.} \quad & x_k \in \mathcal{X}_k \\ & x_k \in \{0,1\}^N \end{aligned} \tag{1}$$

where $c_k \in \mathcal{R}^N$ is the cost vector and x_k is a vector of binary variables of size N. $x_k(i)$ is equal to one if target i is selected in the assignment at time k, and zero otherwise. Here $\mathcal{X}_k$ denotes the invariant feasible space for x_k. This space could represent general constraints such as limits on the total number of vehicles assigned to the mission.

Targets are assumed to have a value c_k, and the problem becomes one of selecting the "optimal" targets to visit subject to the afore mentioned constraints. In the deterministic formulation, the solution becomes a sorting problem, which can be solved in polynomial time. From a practical standpoint, these target values are uncertain as they could be the result of classification, battle situational awareness of the vehicle, and other a priori information. Furthermore, these uncertain values are likely to change throughout the course of the mission, and real-time task assignment algorithms must respond appropriately to these changes in information.

The most straightforward technique is to immediately react to this new information by reassigning the targets. In a deterministic sense, replanning proves to be beneficial since the parameters in the optimization are perfectly known; in a stochastic sense replanning may not be beneficial. For example, since the observations are corrupted by sensor noise, the key issue is that replanning *immediately* to this new information results in the task assignment equivalent of a "high bandwidth controller", making it susceptible to tracking the sensor noise. From the perspective of a human operator, continuous reassignments of the vehicles in the fleet may also prove to be undesirable, especially if this effect is due primarily to sensing errors. Furthermore, since the optimization is continuously responding to new information, it is likely that the integer constrained assignment will vary continuously in time, resulting in a "churning" effect in the assignment, as observed in 4. The noise tracking and churning features are undesirable both from a control and human operator perspective.

A simple example of churning is shown in Figure 1, where one vehicle is assigned to visit the target with the highest value. The original assignment of the vehicle (starting on the left) is to visit the bottom right target. At the next time step, due to simulated sensing noise, the assignment for the vehicle is switched to

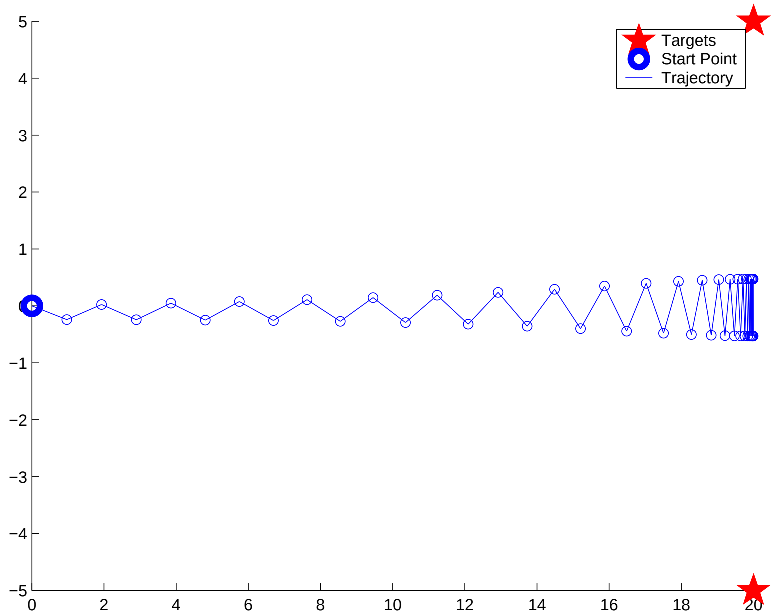

Fig. 1: Effect of churning on a simple assignment problem

the top right target. The vehicle changes direction towards that target, and then the assignment switches once again. The switching throughout the course of the mission is an extreme behavior of the churning phenomenon. In fact, it can be seen that as the mission progresses, the vehicle is still alternating the targets to visit, and never converges to an assignment that does not change. While this is a dramatic example of churning, it captures the notion that sensing noise alone could cause a vehicle to switch assignments throughout the course of the mission. Similar behavior was observed by Tierno[a] as a result of modeling errors in the cost calculations for the task assignment.

Clearly, Figure 1 shows an extreme situation. However, likely missions will involve multiple vehicles, each with unique information and independent sensors and noise sources. It might be quite difficult to identify and correct for churning behavior in a large fleet of UAVs. The subsequent sections in this chapter present methods of modifying the general task assignment problem to automatically avoid this phenomenon.

III. Frequency Domain Analysis

The general assignment problem was introduced in Eq. 1. Note that the main issue was that, even with small variations in the target values, the assignment problem results in highly differing solutions. Clearly, constraining the rate of change of the assignment would result in an attenuation of the churning phenomenon. As noted in Ref. [3], a key issue is which metric to use to measure the change in the assignment. There, the authors suggested a connection between time and frequency domains with regards to planning systems. In this section those ideas are extended by formally relating the time and frequency domain specifications for the assignment problem.

The approach proposed here is similar to that used in GPS (global positioning system) which uses correlation techniques to compare received satellite signals and locally generated Gold codes. A similar approach can be used if we consider the solution x_k as a length N binary "code". We restrict attention to a specific instance of Eq. 1, in which m vehicles are to assigned to N targets ($m < N$). Thus the following

[a]Personal communication, Sept 2003.

integer program is obtained:

$$\begin{aligned} \max_{x_k} \quad & c_k^T x_k \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_k(i) = m \\ & x_k \in \{0,1\}^N \end{aligned} \tag{2}$$

First note that each solution assigns m vehicles to N targets, so they have the same auto-correlation:

$$x_{k-1}^T x_{k-1} \equiv x_k^T x_k \equiv m \tag{3}$$

The two solutions are then compared using the *cross-correlation*

$$R_{k-1,k} = x_{k-1}^T x_k \tag{4}$$

where $R_{k-1,k}$ provides a direct measure of the changes that have been made to the plan from one time-step to the next. At time-step k, if a previous solution x_{k-1} already exists, then a constraint that limits the number of changes that are allowable in the current solution can be included:

$$\max_{x_k} \quad c_k^T x_k \tag{5}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_k(i) = m \tag{6}$$

$$x_{k-1}^T x_k \geq m - \alpha_k \tag{7}$$

$$x_k \in \{0,1\}^N \tag{8}$$

where the integer α_k indicates the number of changes that are allowed in the new plan.

One difficulty with this approach is that the new problem can become infeasible. This infeasibility can be avoided by converting the constraint in Eq. 7 to a soft constraint where α_k is a parameter that must be optimized

$$\max_{x_k,\alpha_k} \quad \begin{bmatrix} c_k^T & -\beta \end{bmatrix} \begin{bmatrix} x_k \\ \alpha_k \end{bmatrix} \tag{9}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_k(i) = m \tag{10}$$

$$\begin{bmatrix} -x_{k-1}^T & -1 \end{bmatrix} \begin{bmatrix} x_k \\ \alpha_k \end{bmatrix} \leq -m \tag{11}$$

$$0 \leq \alpha_k \leq \Gamma \tag{12}$$

$$x_k \in \{0,1\}^N \tag{13}$$

where Γ represents an upper bound on the rate of change of the plan and β represents the relative cost of making changes to the plan. The key point in this formulation is that Γ represents the largest possible decorrelation of the plan from one time-step to the next. Therefore

$$\begin{array}{rcl} R_{0,1} & \geq & m - \Gamma \\ R_{0,2} & \geq & m - 2\Gamma \\ & \vdots & \\ R_{0,k} & \geq & m - k\Gamma \end{array} \tag{14}$$

Note that since variables are binary, the cross correlation must remain positive semi-definite, and we must ensure that $R_{0,j} \geq 0 \; \forall j$. To demonstrate the above statements (Eq. 14), define the plan $P_k = x_k$ and the change in the plan, ΔP_{k+1} as the difference between P_{k+1} and P_k

$$P_1 = P_0 + \Delta P_1 \tag{15}$$
$$P_2 = P_1 + \Delta P_2 \tag{16}$$

Then, by taking the cross-correlation between plans P_0 and P_1,

$$\begin{aligned} P_0^T P_1 = P_0^T (P_0 + \Delta P_1) &\geq m - \Gamma \\ \Rightarrow P_0^T (\Delta P_1) &\geq -\Gamma \end{aligned} \tag{17}$$

Likewise, for the cross-correlation between plans P_0 and P_2, by substituting the above result,

$$P_0^T P_2 = P_0^T (P_0 + \Delta P_1 + \Delta P_2) \geq m - \Gamma + P_0^T \Delta P_2. \tag{18}$$

Now consider the case with $\Delta P_1^T \Delta P_2 = 0$ (implying that the difference between two plans is orthogonal), therefore

$$P_1^T \Delta P_2 = (P_0 + \Delta P_1)^T \Delta P_2 = P_0^T \Delta P_2 \geq -\Gamma \tag{19}$$

Combining these results,

$$P_0^T P_2 \geq m - \Gamma + P_0^T \Delta P_2 \geq m - 2\Gamma \tag{20}$$

The above can be extended by induction to any value of k, giving the result that

$$R_{0,k} \equiv P_0^T P_k \geq m - k\Gamma \tag{21}$$

This corresponds to a triangular correlation plot that is of the form of the *Bartlett Window* typically used in lag windows [12]. Insights into the frequency content of the equivalent controller can be developed by converting the linear correlation plot into the frequency domain via the Fourier transform. This conversion is straightforward since the correlation plot is equivalent to the Bartlett window

$$w_{\tau,n} = \begin{cases} 1 - |\tau|/n, & |\tau| < n \\ 0, & |\tau| \geq n \end{cases} \tag{22}$$

Here n is the *window parameter*; we then have

$$W_n(f) = \Delta t \sum_{\tau=-n}^{n} w_{\tau,n} e^{-i2\pi f \tau \Delta t} = \frac{\Delta t}{n} \left(\frac{\sin(n\pi f \Delta t)}{\sin(\pi f \Delta t)} \right)^2 \tag{23}$$

There are various measures of the bandwidth of $W_n(f)$, one being $\beta_W = 1.5/(n\Delta t)$ [12]. In this case, $n = \text{ceil}(m/\Gamma)$ and $\Delta t = T_s$, so

$$\beta_W \approx \frac{1.5}{((m/\Gamma)T_s)} = \frac{1.5\Gamma}{mT_s}$$

which clearly shows that increasing Γ (the maximum decorrelation rate) increases the effective bandwidth of the controller, as might be expected. A typical plot for this conversion is shown in Figure 2. This analysis establishes an explicit link between the time and frequency domains for the task assignment problem, a relation that has been exploited in many other areas of control design and analysis because it is often insightful to discuss the control problem in the time domain, and modeling errors and sensing noise in the frequency domain.

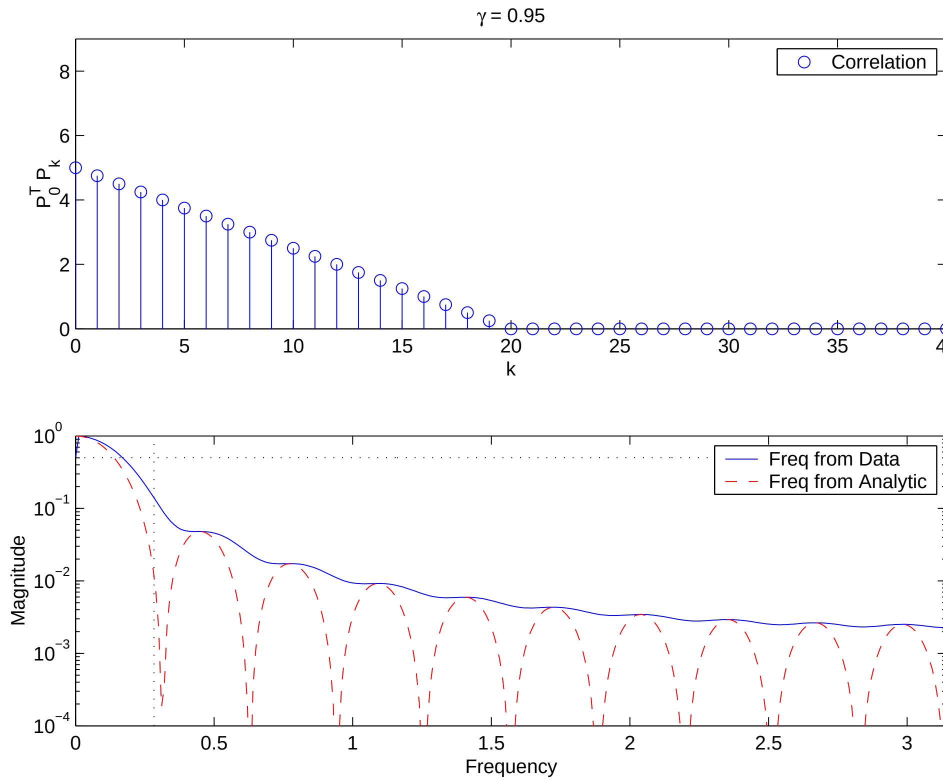

Fig. 2: Equivalent frequency response of the planning system with a constraint on the correlation with the previous plan. The control system is effectively a low-pass filter, with a bandwidth proportional to $\Gamma/(mT_s)$, shown by the vertical dashed line.

IV. Filter Design

Section III introduced time and frequency domain interpretations of the task assignment problem. In this section, the correlation concept introduced in the Section III is extended to develop a filter for the assignment problem. This filter rejects the effect of noise in the information (parameter) vector and can be tuned to capture different noise frequencies.

A. Binary Filter

Similar to section III, assume that the plan is a binary vector of size N. Also assume that the *length* of this vector stays constant in each replanning. Define a binary filter of size r, a system whose binary output changes at most with the rate of once every r time steps. Figures 3 shows the input and output to two binary scalar filters with lengths $r = 3$, $r = 5$. As illustrated in Figure 3-top, the input is a signal with the maximum rate of change (change at each time step). The output is a binary signal with the maximum rate of one change every 3 steps (Figure 3-middle), and every 5 steps (Figure 3-bottom). These figures show the filter for a single binary value, but the same idea can be extended to a binary vector.

Now we present equations that result in a binary filter of size r. Let us define a single binary input signal to the filter at time k, x_k, and a filtered output signal y_k. Define $\delta y_{k,j}$ as the changes in the value of y in

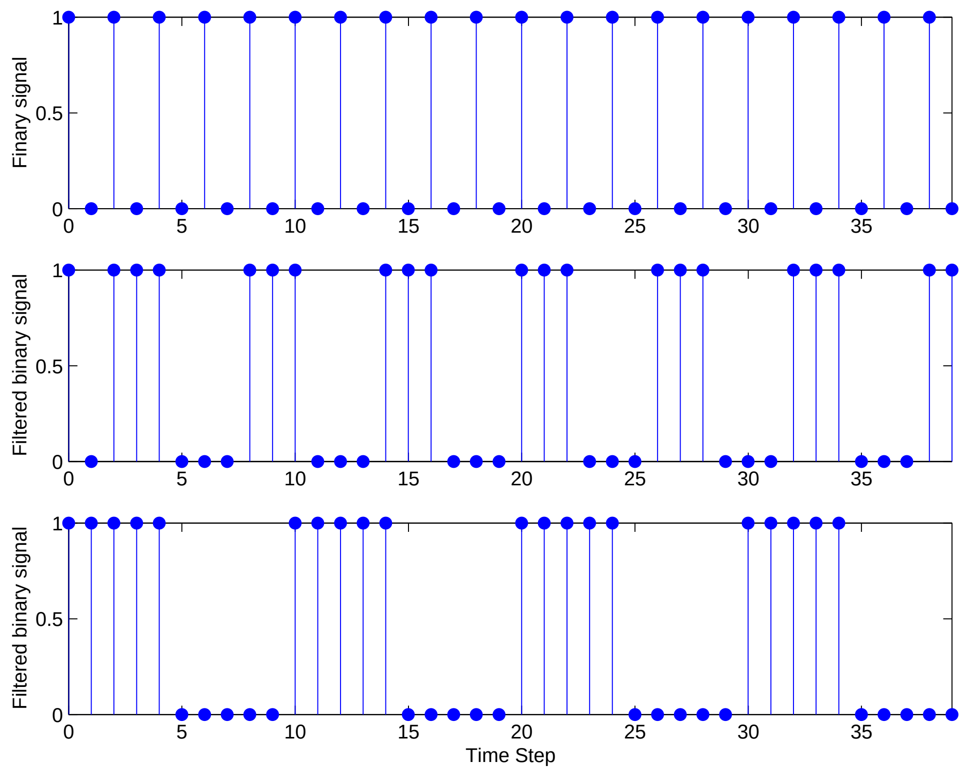

Fig. 3: (top): Input to the binary filter; (middle): Filtered signal ($r = 3$); (bottom): Filtered signal ($r = 5$)

the past iterations:

$$\delta y_{k,j} = \begin{cases} x_k \oplus y_{k-1} & j = 1 \\ y_{k-j} \oplus y_{k-j+1} & j = 2, \ldots, r \end{cases} \tag{24}$$

where $\oplus$ represents the exclusive OR (XOR) operation. Having these differences, define z_k as follows:

$$z_k = \delta y_{k,1} + \frac{1}{r} \sum_{j=2}^{r} \delta y_{k,j} \tag{25}$$

z_k is thus a weighted summation of the difference in plans from time $k - r$ to k.

Now if $z_k > 1$, then y has changed at least once in the previous r steps and x_k is different from y_{k-1}; therefore if $z_k > 1$, y_k should equal $y_{k-1} = (\sim x_k)$ and $y_k = x_k$ otherwise. y_k can be calculated as follows:

$$y_k = \begin{cases} x_k & \text{if} \quad z_k \leq 1 \\ \sim x_k & \text{otherwise} \end{cases} \tag{26}$$

where $\sim$ denotes the *NOT* operation. Having defined the binary filter, the following sections demonstrate how to implement it inside the planning algorithm.

B. Assignment With Filtering: Formulation

Starting with the simple assignment problem of Eq. 1; the idea here is to replan for the same system in each time step and suppress the effect of parameter noise (uncertainty) on the output of the system (generated plan). If the above problem is solved in each iteration, the optimization parameters will be directly impacted by the noise and the assignment can be completely different at each step. To avoid large variations in the solution, a binary filter is integrated into the problem to limit the rate of change in the assignment problem.

The modified assignment problem can be written as follow:

$$\max_{y_k, z_k} c_k^T y_k \tag{27}$$

$$\text{s.t.} \qquad y_k \in \mathcal{Y}_k \tag{28}$$

$$z_k = y_k \oplus y_{k-1} \tag{29}$$

$$t_{k,j} = z_k^T \delta y_{k,j} = 0 \qquad j = 2, \ldots, r \tag{30}$$

where $\delta y_{k,j}$ represents the changes in the assignment in previous plans and functions as an input to the optimization problem:

$$\delta y_{k,j} = y_{k-j} \oplus y_{k-j+1} \quad j = 2, \ldots, r \tag{31}$$

Note that $\delta y_{k,i}$ can be calculated prior to optimization, as the previous plans have been stored. Also note that the constraint in Eq. 30 restricts changes in the current plan from the previously generated plans.

C. Assignment With Filtering: Implementation

This section presents the results for a simple, but general, assignment problem and compares the results of the unfiltered and filtered formulations. The objective of this problem is to pick m targets from N existing targets ($m < N$) in order to maximize the total value of the selected targets. Each target has a value associated with it that is affected by noise:

$$c_k = c + \delta c_k \tag{32}$$

where c is the nominal target value and δc_k is the noise added to these values. The nominal value for all targets is set to 5. Solving this problem for $N = 4$, $m = 2$ for 30 iterations results in 30 different plans which are directly affected by the noise. Figure 4 shows the result of this simulation. In these figures, $\bullet$ represents 1 and $\circ$ represents 0. Thus, targets 1 and 2 are selected in assignment 1, targets 1 and 4 are selected in assignment 2.

The filter is implemented by converting the constraint in Eq. 29 to a linear form for LIP implementation

$$z_k(i) \geq -y_{k-1}(i) + y_k(i) \tag{33}$$

$$z_k(i) \geq y_{k-1}(i) - y_k(i) \tag{34}$$

$$z_k(i) \leq y_k(i) + y_{k-1}(i) \tag{35}$$

$$z_k(i) \leq 2 - y_k(i) - y_{k-1}(i) \tag{36}$$

$$z_k(i) \in \{0, 1\} \qquad i = 1, \ldots, N \tag{37}$$

One potential issue with this formulation, is that it might make the problem infeasible. This means that if the basic problem has a solution, a solution is not guaranteed in the filtered formulation. The constraint that is capable of making the filtered formulation infeasible is $t_{k,j} = 0$, $j = 2, \ldots, r$. This limits the number of bit changes in the plan compared to previous plan and can make the problem infeasible. To avoid this difficulty, this set of constraints can be relaxed and added to the cost function as a penalty function. The new cost function can be written as

$$\max_{y_k, t_{k,j}} c_k^T y_k + \sum_{j=1}^{r} d_j t_{k,j} \tag{38}$$

Figures 5 and 6 show the results of applying filtering to the previous example, with values of $r = 3$, $r = 5$ used for the filter length. Comparing these results with the result of the unfiltered formulation (Figure 4) clearly shows the noise mitigation in the filtered formulation. Although this result helps attenuate the noise

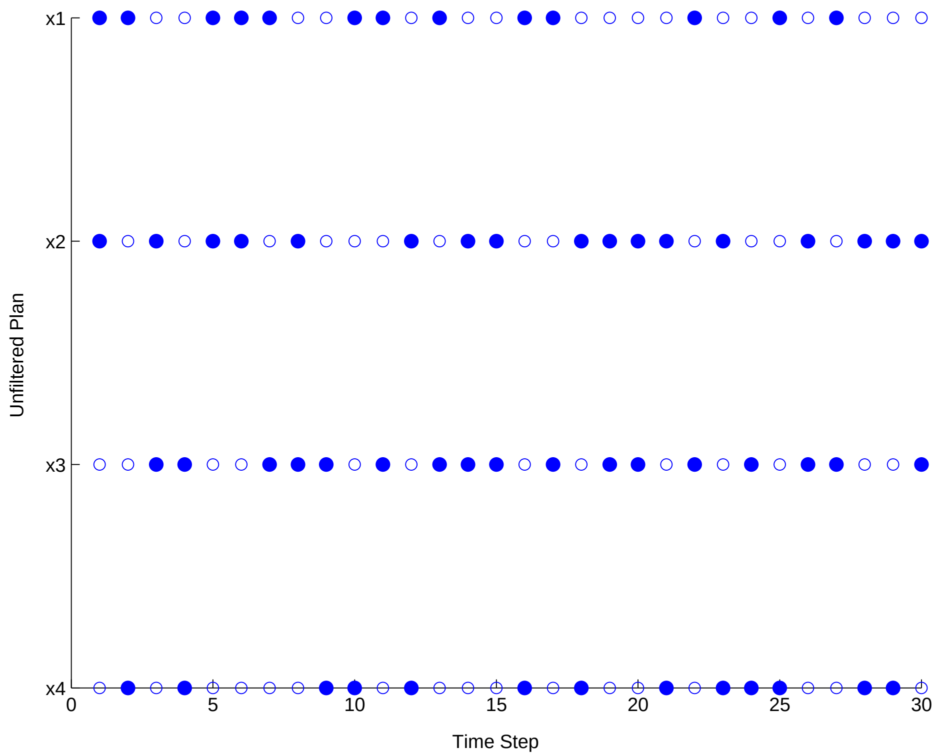

Fig. 4: Unfiltered plan (*i.e.*, optimal plan) showing which two of the four targets are chosen at each time step (• represents 1 and ∘ represents 0).

effect and reduce the rate of change in the plan, it does not completely reject the noise. The filter clearly prevents sudden changes in the plan; a change in the i^{th} element of x_k, $x_k(i)$, can happen only after being in its current state for r time steps. The ultimate goal is to make the filter respond to low frequency changes and suppress high frequency changes, which will help to reject the high frequency noise while responding to low frequency changes in the environment (system). This will reduce churning, which is the result of noise and/or the changes in the environment that occur too rapidly to track, and may therefore be treated as noise as well.

In the above formulation, the filter has memory which allows it to use the previous outputs to generate the current output. The solution can be written as

$$y_k = f(x_k, y_{k-1}, y_{k-2}, \dots, y_{k-r}) \tag{39}$$

where x_k is the current optimal solution (interpreted as the input to the system) and $y_{k-1}, \dots, y_{k-r}$ are the previous plans (outputs of the system). In addition to these values, the previous unfiltered plans can also be used as input to the filter. At each iteration, both filtered, y_k, and unfiltered, x_k, plans can be generated. Having generated x_k, a filter of the following form can then be designed

$$y_k = f(x_k, x_{k-1}, \dots, x_{k-q}, y_{k-1}, y_{k-2}, \dots, y_{k-r}) \tag{40}$$

Figure 7 gives a block diagram representation of assignment with filtering. Here, FTA and $UFTA$ represent the filtered and unfiltered task assignments respectively. TA represents the overall task assignment and Z^{-1} represents a bank of unit delays.

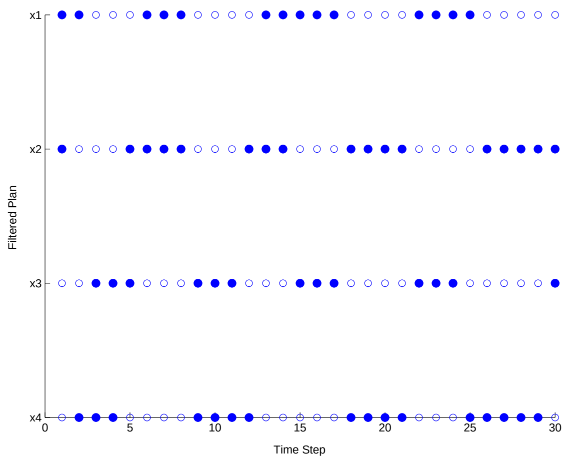

Fig. 5: Filtered plan with $r = 3$

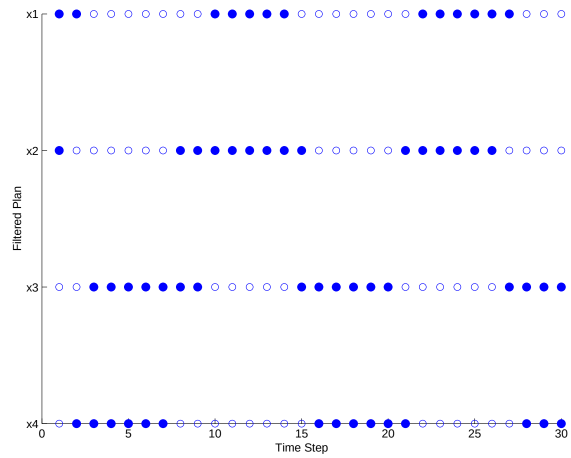

Fig. 6: Filtered plan with $r = 5$

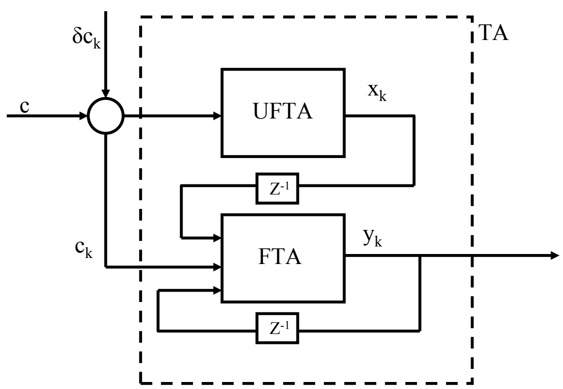

Fig. 7: Block diagram representation of assignment with filtering

To explain how this can reduce the impact of high frequency noise in the parameters, suppose the objective is to reject noise with frequency 1 but also track changes occurring with frequency of 0.5. This means the effect of noise makes the system parameters change at each time step, while the changes in the parameters are occurring every two time steps. Now consider at time k the unfiltered solution to two previous plans: x_{k-1} and x_{k-2}. In order to calculate y_k given x_k and these values, the difference between x_{k-2}, x_{k-1}, and x_k are used

$$\delta x_k = x_{k-1} \oplus x_k \quad \text{and} \quad \delta x_{k-1} = x_{k-2} \oplus x_{k-1}$$

Then calculate the output as

$$y_k(i) = \begin{cases} x_k(i) & \text{if} \quad \delta x_k(i)\delta x_{k-1}(i) = 0 \\ y_{k-1}(i) & \text{if} \quad \delta x_k(i)\delta x_{k-1}(i) = 1 \end{cases} \tag{41}$$

If the value of the bit $x_{k-1}(i)$ had changed from its previous value,$x_{k-2}(i)$ and the value of the same bit is changed from $x_{k-1}(i)$ to $x_k(i)$, then this change is a change with frequency of 1, which is intended to be canceled. Therefore the change is ignored and $y_k(i) = y_{k-1}(i)$.

This idea of filtering noise is used to implement an assignment algorithm that is robust to rapid changes in the situational awareness (*i.e.,* noise). In other words the effect of noise in the information vector is suppressed in the assignment. To avoid the difficulties of adding hard constraints to the assignment problem, the constraints are relaxed and added to the objective function. A filter that rejects noise with frequency 1 (this is the highest frequency, meaning that the signal can change in each time step) can be formulated as

$$\max \quad c_k^T y_k - \beta_k^T (y_k \oplus y_{k-1}) \tag{42}$$

$$\text{s.t.} \quad y_k \in \mathcal{Y}_k \tag{43}$$

$$\delta x_j = x_{k-j} \oplus x_{k-j+1} \tag{44}$$

$$\beta_k = \sum_{j=1}^{q} b_j \delta x_j \tag{45}$$

where b_j is the weighting coefficient and x_j is the unfiltered solution at step j and is given as an input to the optimization. The second term in the objective function (Eq. 42) is the penalty associated with changes that are made in the current plan compared to the previous plan. Contrary to existing approaches, in this approach penalty coefficients are not constant. These coefficient are calculated dynamically using previous plans to selectively reject the changes associated with the high frequency noise. β_k is the coefficient vector that penalizes each change (each bit in the assignment vector). Each element of this coefficient vector is a weighted summation of previous changes in the optimal plan. Since the optimal plan will track environmental noise, measuring changes in the optimal plan results in a good metric to identify and suppress such disturbances. This is implemented in Eqs. 44 and 45. q here defines how far in the past is included in calculating the coefficients. The coefficients b_j tune the effect of previous plans in the current plan. Setting these coefficients defines the importance of the changes in the previous plans and so the *effective bandwidth* of the filter. A good candidate for b_j is

$$b_j = \frac{b}{2^j} \tag{46}$$

where b is a constant that can be set based on the problem. This set of coefficients will attenuate the effect of the changes that happened in the far past (larger j), compared to the more recent changes in the plan. As j increases then, the weighting on the past plans is decreased.

The formulation above is a special case of the general filter in Eq. 40 with $r = 1$, which was described for simplicity. A more comprehensive form of this filter ($r > 1$) can also be implemented to obtain very general filtering properties. For example, the formulation given above can be combined with the formulation in the previous section to create a filter with the properties of both. This filter will reject noise with high frequency while limiting the rate of change of the plans. For instance a filter that rejects noise with frequencies greater than 0.5 will have an additional term in the cost function in Eq. 42

$$\max c_k^T y_k - \beta_k^T (y_k \oplus y_{k-1}) - \gamma_k^T (y_k \oplus y_{k-2}) \tag{47}$$

where γ is calculated as

$$\delta^2 x_j = x_{k-j} \oplus x_{k-j+2} \tag{48}$$

$$\gamma_k = \sum_{j=2}^{q} b_j \delta^2 x_j \tag{49}$$

D. Assignment With Filtering: Simulation Results

1. Example 1:

Figure 8 presents the results of the unfiltered and filtered formulations for the example introduced in Section C. The cost coefficients change randomly with time (top plot of Figure 8), and the noise is uniformly distributed in the interval $[-0.5, 0.5]$. The middle plot shows the two selected targets (•) out of the four choices (x_1–x_4). Note that as the costs change in the top plot from one time period to the next, the unfiltered plan changes as well. However, the filtered solution (bottom plot) remains unchanged for much longer periods of time. Thus it is clear that the unfiltered solution tracks the noise in the cost coefficients to a much greater extent than the filtered plan. To demonstrate that the filtered plan is only rejecting the noise, the coefficient c_2 is changed at time step 7 by increasing it by 0.7 and at time step 16 by decreasing it by 1.4. The results in the bottom plot show that the filtered plans are modified to follow these lower frequency changes.

2. Example 2:

Figure 9 makes another comparison between the unfiltered and filtered assignments. An example with 40 targets and 20 vehicles was simulated for 100 time steps. At each time step the current plan was correlated with the previous plan, and the results are shown in the figure. Figure 9 shows that adding our filtering tends to increase the correlation between plans from one time-step to the next. This means that the task assignment is returning the same solution even though the data in the problem is changing slightly due to noise/disturbances/uncertainty in our cost estimates. The unfiltered results show lower correlation, which means the plans are changing and the vehicles would be re-assigned to new tasks (each plan might be optimal at that time-step, but this can lead to a "churning" type of behavior wherein the vehicles flip back and forth between assignments.)

3. Example 3:

This set of simulations compares the result of filters with two different cut-off frequencies that were introduced in previous section. In these examples, there are two targets and one UAV. The UAV can go to and hit only one target, and the objective is to maximize the score. The values of the targets are very close (both 10 plus/minus 1). The simulation was run for the two formulations in Eqs. 42 and 47 and the results are shown in Figures 10 and 11, respectively. The top-left figure in each four figure block shows the score for one of the targets. The score of the other target is kept constant at 10. Uncertainty in the target classification is introduced as a noisy score. The frequency of this noise is 1 up to $t = 20$ and then decreases to 0.5 for $t > 20$. The top-right figure shows the task assignment result without using any filtering. A value of 1 here means that the UAV picks the upper target, and 0 means that it picks the lower one. The bottom-right figure shows the assignment using the filtered assignment algorithm. The trajectory of the UAV for both filtered and unfiltered assignments is shown in the bottom-left (× for the unfiltered case and ∘ for the filtered case).

The results in Figure 10, clearly show that when the assignment is done with a filter with cut-off frequency of 1, the churning seen in the unfiltered solution is eliminated when the noise frequency is 1. However, at $t = 20$, when the noise frequency changes to 0.5 this filter fails to reject the noise and the churning phenomenon starts. Figure 11 shows that by adding the γ term in the cost function (Eq. 47) we can adjust the cut-off frequency of the filter to also reject lower frequency noise so that the churning phenomenon is eliminated.

V. Conclusions

This paper has formulated a modification of the classical task assignment under noisy conditions. We have extended the frequency domain interpretation as originally formulated using correlation as a metric. We have developed a formulation that performs the reassignments at the rate the information is updated, which enables the planner to react immediately to any significant changes that occur in the environment. Also, rather than just limiting the rate of change of the plan, this new approach embeds a more sophisticated filtering operation in the task assignment algorithm that relies on modifying the weighting coefficients based on the plan changes. We demonstrate that this modified formulation can be interpreted as a noise rejection

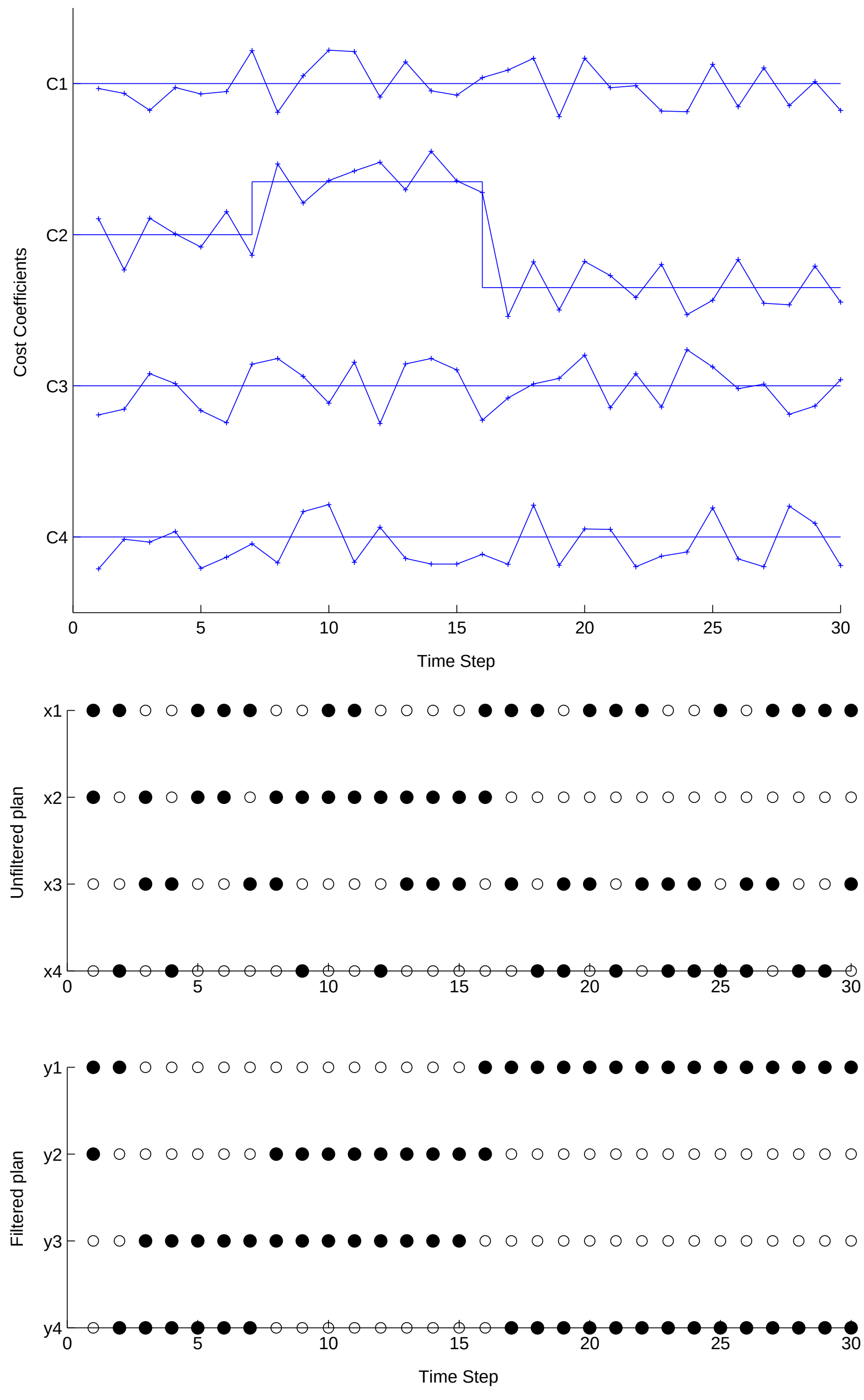

Fig. 8: (top): Noisy cost coefficients; (middle): Plans with no filter; (bottom): Filtered plan

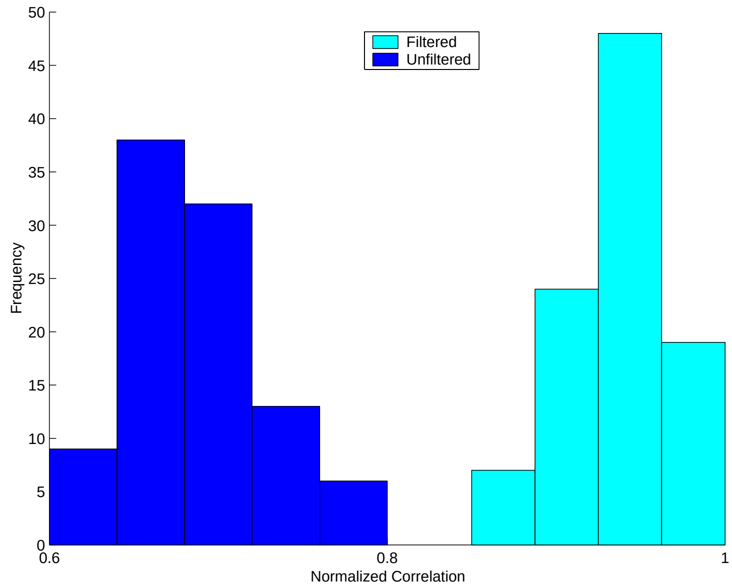

Fig. 9: Histogram showing correlation of filtered and unfiltered assignments

algorithm that reduces the effect of the high frequency noise on the planner. We demonstrated the effectiveness of this scheme in simulation. Simulation results showed good signal-tracking and noise-rejection properties. Future work will investigate the role of robust approaches to the task assignment and their relation to the recently developed noise rejection algorithms.

Acknowledgment

Research funded in part by AFOSR grant #F49620-01-1-00453.

References

[1]Office of the Secretary of Defense, "Unmanned Aerial Vehicles Roadmap," Technical Report, December 2002.

[2]P. R. Chandler, S. Rasmussen, "UAV Cooperative Path-Planning," Proc. of the *AIAA GNC,* Aug. 14–17, 2000. AIAA-2000-4370.

[3]J. Tierno and A. Khalak, "Frequency Domain Control Synthesis For Time-Critical Planning," Proceedings of the *IEE European Control Conference,* Sept. 2003.

[4]J. W. Curtis, and R. Murphey, "Simultaneous Area Search and Task Assignment for a Team of Cooperative Agents," AIAA *GNC*, August 2003.

[5]J. M. Mulvey, R. J. Vanderbei, S. A. Zenios, "Robust optimization of large-scale systems," *Oper. Res.* Vol. 43 (1995) pp. 264–281.

[6]M. Sim and D. Bertsimas, *The Price of Robustness,* MIT Sloan Internal Tech. Report, 2002.

[7]L. F. Bertuccelli, *Robust Planning for Heterogeneous UAVs in Uncertain Environments,* SM Thesis, MIT Department of Aeronautics and Astronautics, May 2004.

[8]J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, "Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environment," IEEE *CDC,* Dec. 2002.

[9]M. Alighanbari, Y. Kuwata, and J. P. How, "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering," IEEE *ACC,* June 2003.

[10]R. Kastner, C. Hsieh, M. Potkonjak, and M. Sarrafzadeh, "On the Sensitivity of Incremental Algorithms for Combinatorial Auctions," WECWIS 2002: Newport Beach, California, USA.

[11]Personal communication with Jorge Tierno, Sept. 2003.

[12]Percival and Walden, Spectral Analysis for Physical Applications, Cambridge University Press, 1993.

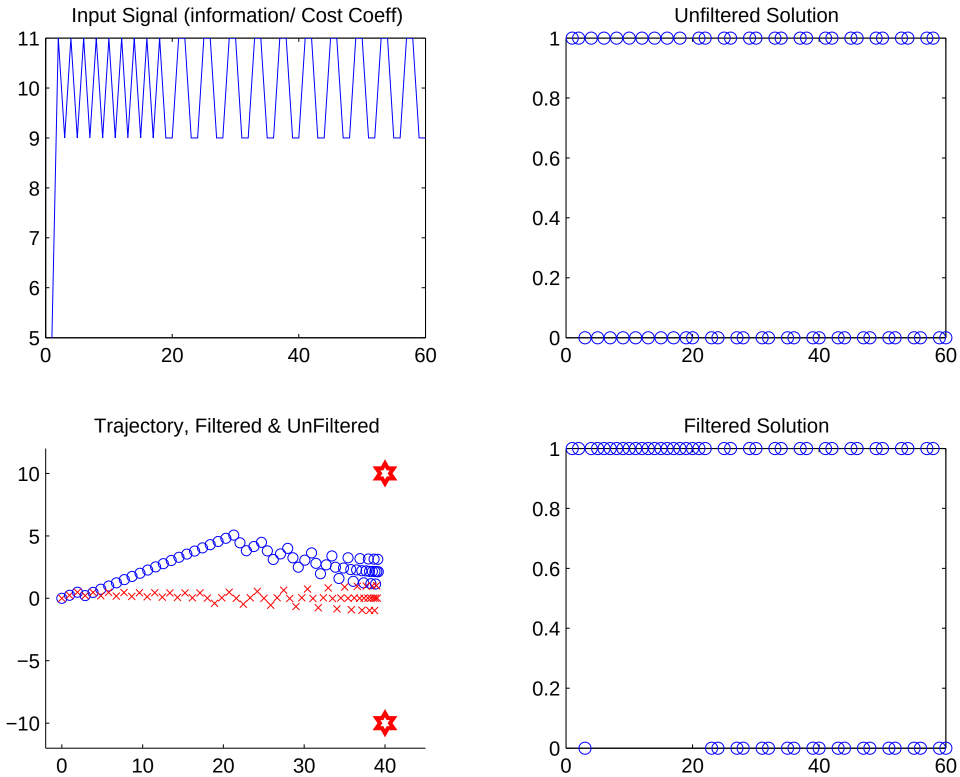

Fig. 10: The result of 1-UAV 2-Targets simulation for filtering with cut-off frequency of 1.

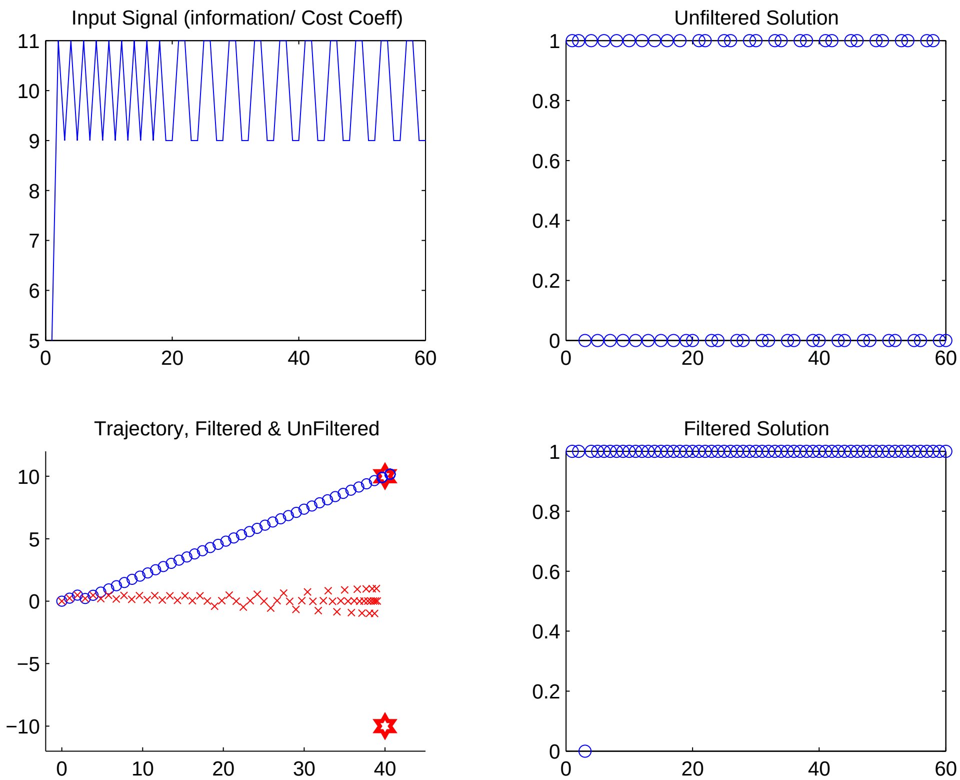

Fig. 11: The result of 1-UAV 2-Targets simulation for filtering with cut-off frequency of 0.5.