Networked and Distributed Control: Assignment 4

Niels Stienen, 5595738

I. Problem 1

For this problem the goal is to solve a multi-aircraft coordination problem, where a linear time-invariant dynamical system can describe each aircraft i

$$x_i(t+1) = Ax_i(t) + Bu_i(t), \quad x_i(0) = x_{i,0}, \\ \text{for} \quad i = 1, ..., 4, t = 0, ..., T_{final} - 1$$

The objective is to find a decomposition-based algorithm for coordinating towards a common target state at T_{final} to ensure that

$$x_i(T_{final}) = x_f, \quad \forall i$$

while satisfying an element-wise limitation on the control inputs

$$|u_i(t)| \leq \frac{u_{max}}{T_{final}}, \quad \forall i, t$$

The objective function to be minimized is the function

$$\sum_i \sum_t x_i(t)^\top x_i(t) + u_i(t)^\top u_i(t)$$

> Derive a solution based on dual decomposition using the projected subgradient method. Show a plot that demonstrates the convergence of the optimization process. Show a plot of the resulting aircraft state trajectories after convergence, and compare with the centralized optimal solution.

Before applying the dual decomposition the full optimization problem is stated as below with T_{final} denoted by T_f for brevity.

$$\begin{aligned} \min_{u_i, x_f} \quad & \sum_{i=1}^{4} \sum_{t=0}^{T_f-1} x_i(t)^\top x_i(t) + u_i(t)^\top u_i(t) + x_i(T_f)^\top x_i(T_f) \\ \text{s.t.} \quad & x_i(t+1) = A_i x_i(t) + B_i u_i(t), \quad \forall i, t, \\ & x_i(t) \in \mathbb{R}^4, \quad \forall i, t, \\ & u_i(t) \in \left\{ \mathbb{R}^2 \,\middle|\, |u_i(t)| \leq \frac{u_{max}}{T_f} \right\}, \quad \forall i, t, \\ & x_i(T_f) = x_f, \quad \forall i, \\ & x_f \in \mathbb{R}^4 \end{aligned} \tag{1}$$

The state update constraint can be removed by using a prediction matrix for the future states of x_i dependent on only the initial state $x_i(0)$ and control inputs u:

$$\mathbf{x}_i = T_i x_{i,(k)} + S_i \mathbf{u}_i, \quad \forall i, t \tag{2}$$

$$\mathbf{x}_i = \begin{bmatrix} x_{i,(k+1)} \\ x_{i,(k+2)} \\ \vdots \\ x_{i,(k+N)} \end{bmatrix}, \quad \mathbf{u}_i = \begin{bmatrix} u_{i,(k)} \\ u_{i,(k+1)} \\ \vdots \\ u_{i,(k+N-1)} \end{bmatrix} \tag{3}$$

$$T_i = \begin{bmatrix} A_i \\ A_i^2 \\ \vdots \\ A_i^N \end{bmatrix}, \quad S_i = \begin{bmatrix} B_i & 0 & \dots & 0 \\ A_i B_i & B_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_i^{N-1} B_i & A_i^{N-2} B_i & \dots & B_i \end{bmatrix} \tag{4}$$

Using these matrices the cost function is changed to

$$f_i(x_i(0), \mathbf{u}_i) = (T_i x_i(0) + S_i \mathbf{u}_i)^\top (T_i x_i(0) + S_i \mathbf{u}_i) + \mathbf{u}_i^\top \mathbf{u}_i \tag{5}$$

To use `quadprog()` for implementation in Matlab the cost function is expanded to the form $u^* = \min_u \frac{1}{2} u^\top H u + h^\top u$.

$$\begin{aligned} f_i(x_i(0), \mathbf{u}_i) &= (T_i x_i(0) + S_i \mathbf{u}_i)^\top (T_i x_i(0) + S_i \mathbf{u}_i) + \mathbf{u}_i^\top \mathbf{u}_i \\ &= x_i(0)^\top T_i^\top T_i x_i(0) + x_i(0)^\top T_i^\top S_i \mathbf{u}_i + \mathbf{u}_i^\top S_i^\top T_i x_i(0) \\ &\quad + \mathbf{u}_i^\top S_i^\top S_i \mathbf{u}_i + \mathbf{u}_i^\top \mathbf{u}_i \\ &= \mathbf{u}_i^\top \underbrace{(S_i^\top S_i + I)}_{H} \mathbf{u}_i + 2 \underbrace{x_i(0)^\top T_i^\top S_i}_{h} \mathbf{u}_i + \underbrace{x_i(0)^\top T_i^\top T_i x_i(0)}_{Constant} \end{aligned}$$

The final equality constraints can be written in terms of the initial state $x_i(0)$ and control input vector $\mathbf{u}_i$ as well:

$$x_i(T_f) = \overbrace{A_i^{T_f} x_i(0)}^{b_i^{eq}} + \overbrace{\begin{bmatrix} A_i^{T_f-1} B_i & \dots & A_i B_i & B_i \end{bmatrix}}^{A_i^{eq}} \mathbf{u}_i \tag{6}$$

$$x_i(T_f) = A_i^{eq} \mathbf{u}_i + b_i^{eq} \tag{7}$$

Lastly, the inequality constraint for the control input can be rewritten to a matrix constraint such that the absolute is no longer needed which will make coding easier. Defining $\mathbb{1} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^\top$ the inequality constraints become

$$|\mathbf{u}_i(t)| \leq \mathbb{1} \frac{u_{max}}{T_f} \Leftrightarrow \overbrace{\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}}^{A^u} \mathbf{u}_i \leq \overbrace{\begin{bmatrix} \mathbb{1} \frac{u_{max}}{T_f} \\ \mathbb{1} \frac{u_{max}}{T_f} \end{bmatrix}}^{b^u} \tag{8}$$

The full optimization problem now is:

$$\begin{aligned} \underset{u_i, x_f}{\text{minimize}} \quad & \sum_{i=1}^{4} f_i(x_i(0), \mathbf{u}_i) \\ \text{subject to} \quad & x_i(0) \in \mathbb{R}^4, \quad \forall i, \\ & \mathbf{u}_{i,j} \in \mathbb{R}^2 \quad \forall i, j, \\ & A^u \mathbf{u}_i \leq b^u \quad \forall i, \\ & A_i^{eq} \mathbf{u}_i + b_i^{eq} = x_f, \quad \forall i, \\ & x_f \in \mathbb{R}^4 \end{aligned} \tag{9}$$

The coupling constraint for the final state is dependent on an additional decision variable. This variable can be eliminated by rewriting the constraint into a different set of coupling constraints. The notation $x_i(T_f)$ will still be used here as it keeps the result clearer.

$$x_1(T_f) = x_2(T_f) \tag{10}$$
$$x_1(T_f) = x_3(T_f) \tag{11}$$
$$x_1(T_f) = x_4(T_f) \tag{12}$$

The Lagrangian of the problem then becomes

$$\mathcal{L}(\mathbf{u}, \lambda) = \sum_{i=1}^{4} f_i(\mathbf{u}_i) + \lambda_{1,i}^\top (x_1(T_f) - x_i(T_f)) \tag{13}$$

The resulting higher level problem is maximising the dual problem

$$\begin{aligned} &\underset{\lambda}{\text{maximize}} && \inf_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \lambda) \\ &\text{subject to} && \lambda \geq 0 \end{aligned} \tag{14}$$

Making use of the projected subgradient method the subgradient is updated

$$\begin{bmatrix} \lambda_{1,2}^{k+1} \\ \lambda_{1,3}^{k+1} \\ \lambda_{1,4}^{k+1} \end{bmatrix} = \mathcal{P}_{\mathbb{R}^4}\left(\begin{bmatrix} \lambda_{1,2}^{k} \\ \lambda_{1,3}^{k} \\ \lambda_{1,4}^{k} \end{bmatrix} + \alpha \begin{bmatrix} x_1(T_f) - x_2(T_f) \\ x_1(T_f) - x_3(T_f) \\ x_1(T_f) - x_4(T_f) \end{bmatrix} \right) \tag{15}$$

The objective function matrices for the lower-level problems are:

$$H = \left(S_i^\top S_i + I\right) \tag{16}$$
$$h_1 = x_1(0)^\top T_1^\top S_1 + \frac{1}{2}(\lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4})^\top x_1(T_f) \tag{17}$$
$$h_i = x_i(0)^\top T_i^\top S_i - \frac{1}{2}\lambda_{1,i}^\top x_i(T_f), \quad \text{for } i \neq 1 \tag{18}$$
$$C = x_i(0)^\top T_i^\top T_i x_i(0) \tag{19}$$

Using the cost function and constraints the dual decomposition problem is solved using `quadprog()` which resulted in the optimal control input such that the states of all aircraft are converged to within an error of 10^{-10}, see Figure 1. The error is defined as the two-norm of the difference with the central solution divided by the two-norm of the central solution:

$$\frac{\left\| \mathbf{x}_{i,f} - \mathbf{x}_f^* \right\|_2}{\left\| \mathbf{x}_f^* \right\|_2} \tag{20}$$

The state trajectories of each aircraft converge to the centralized optimal solution as shown in Figure 3. In addition, the XY plane has been plotted showing the path the aircraft traveled and eventually converged in the final state as found with the centralized solution, as shown in Figure 2.

Investigate the effect of step size (for a constant step) and step size update sequence (for variable step) on the convergence of the standard subgradient method. Show the results using a logarithmic error sequence plot.

To investigate the effect that the step size and step size update sequence have on the convergence several different update sequences are chosen. For all sequences, various values of α are tested and a comparison between all sequences is shown in Figure 8.

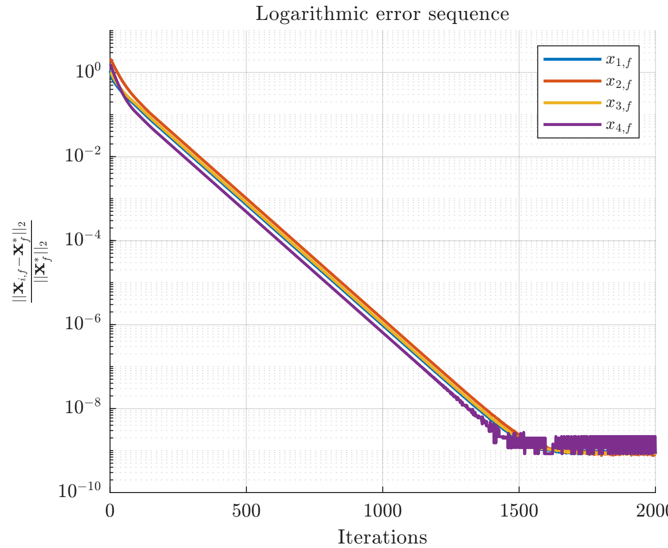

Fig. 1: Logarithmic error sequence of each aircraft compared to the optimal centralized solution for $\alpha = 0.5$

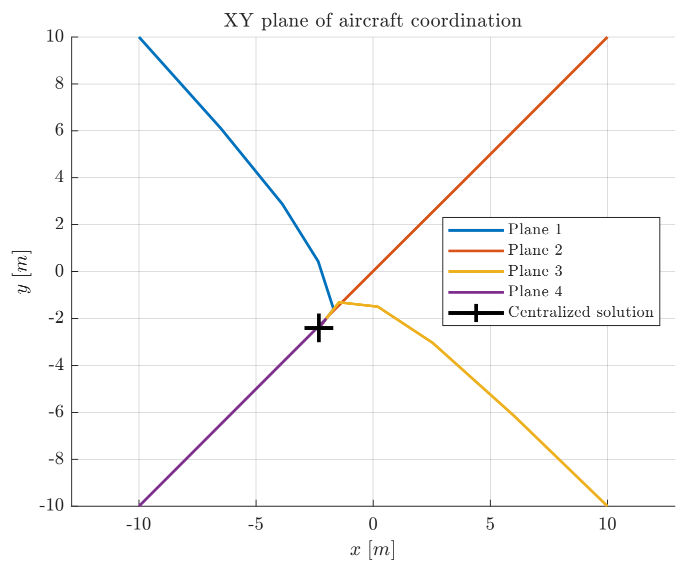

Fig. 2: XY plane showing the aircraft converging to a location southwest of the average initial position

The update sequences used are

$$\alpha_k = \alpha_0 \tag{21}$$
$$\alpha_k = 0.99^k \alpha_0 \tag{22}$$
$$\alpha_k = \frac{\alpha_0}{k} \tag{23}$$
$$\alpha_k = 2^{-length(l_0)}\alpha_0 \tag{24}$$

where the power of the last update sequence is defined as the number of leading zeros in the norm of the maximum error at a given iteration. The result is an update sequence that only changes step size when the maximum error has decreased an order of magnitude.

For the constant step size increasing the value of α causes

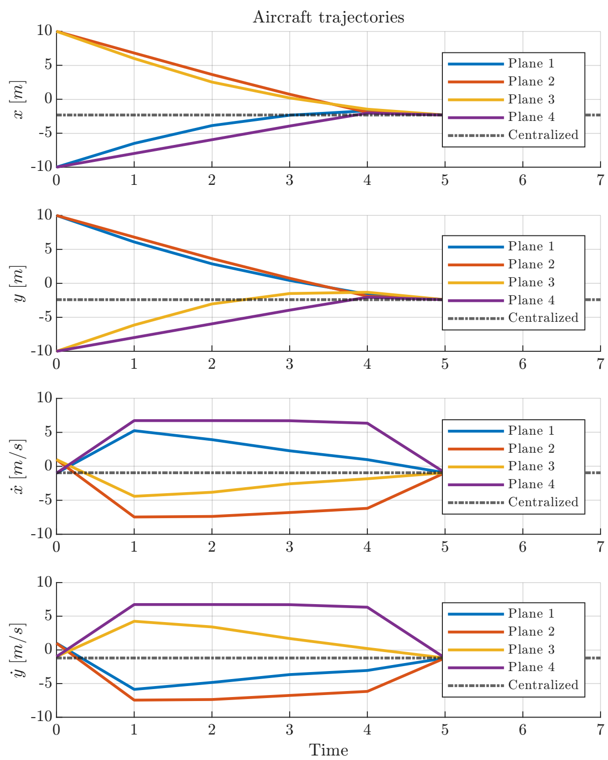

Fig. 3: State trajectories for each aircraft showing convergence to the centralized optimal solution at T_{final}.

the error to converge faster, as seen in Figure 4, up to an upper bound around $\alpha_k = 1.0$. If the value of α is chosen to be greater the error will increase after a few iterations.

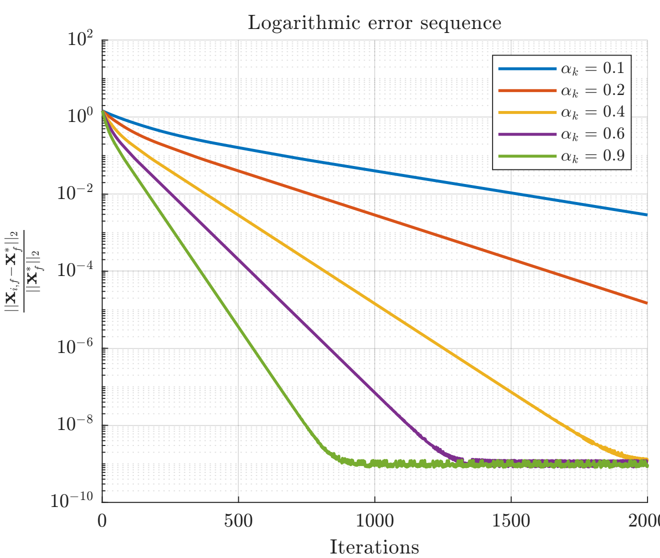

Fig. 4: Comparison of iterations needed for convergence for various constant values of α

The update sequence $\alpha_k = 0.99^k \alpha_0$ performs similarly to the constant step size needing around $1-2\%$ more iterations before convergence for the same α_0. The various values tested are shown in Figure 5.

The final two update sequences both show that they are decreasing too fast for convergence close to the error that the constant step size achieves. Figure 6 shows the third update sequence, and Figure 7 shows the final update sequence where the change in step size is noticeable through the sharp switch in gradient of the logarithmic error sequence.

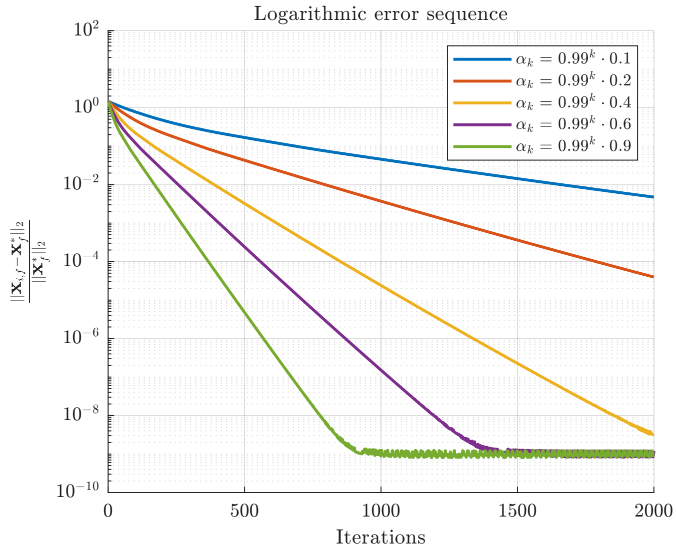

Fig. 5: Comparison of iterations needed for convergence for various values of $\alpha_k = 0.99^k \alpha_0$

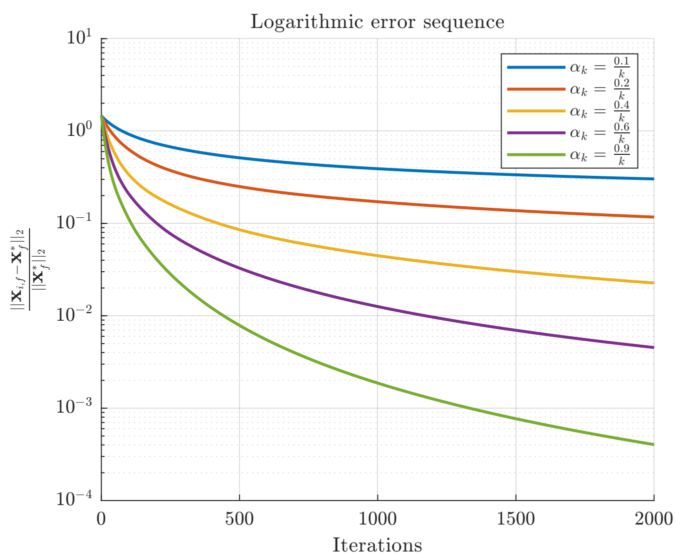

Fig. 6: Comparison of iterations needed for convergence for various values of $\alpha_k = \frac{\alpha_0}{k}$

Comparing the types of update sequences considered the best methods ended up being the Nesterov accelerated gradient which will be covered hereafter and the constant step size. The leading zeros approach outperformed the other update sequences for the first hundred iterations but was quickly surpassed by the other update sequences once the error had dropped to below 10^{-2}.

Implement an accelerated version (e.g. Nesterov method) of the subgradient updates. Show the results using a logarithmic error sequence plot.

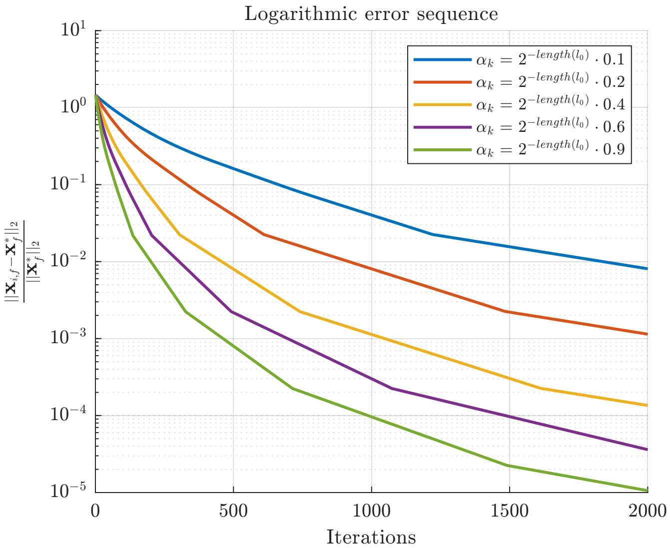

Fig. 7: Comparison of iterations needed for convergence for various values of $\alpha_k = 2^{-length(l_0)}\alpha_0$, where l_0 are the leading zeroes of the maximum error Δx_f.

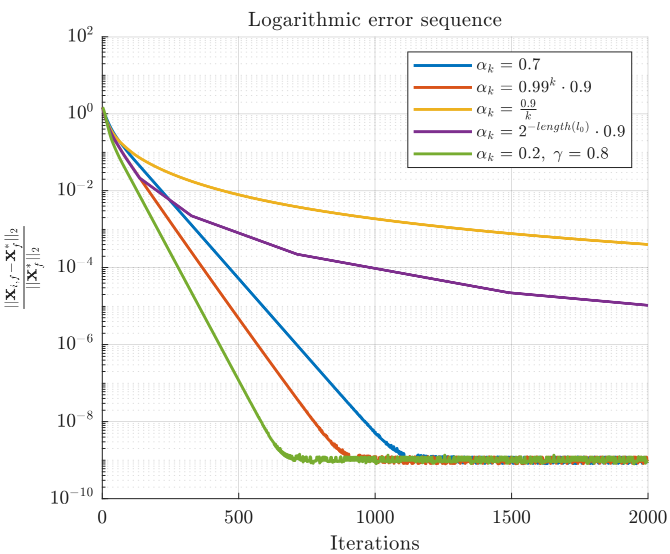

Fig. 8: Comparison of iterations needed for convergence for all previously described stepsize methods.

For implementing the Nesterov accelerated gradient the step size is set to constant again. The Nesterov accelerated gradient is implemented by augmenting Equation 15 as

$$\psi_{k+1} = \gamma\psi_k + \alpha \begin{bmatrix} x_1(T_f) - x_2(T_f) \\ x_1(T_f) - x_3(T_f) \\ x_1(T_f) - x_4(T_f) \end{bmatrix} \tag{25}$$

$$\lambda_{k+1} = \mathcal{P}_{\mathbb{R}^4}(\lambda_k + \psi_{k+1}) \tag{26}$$

with

$$\psi_k = \begin{bmatrix} \psi_{1,2}^k \\ \psi_{1,3}^k \\ \psi_{1,4}^k \end{bmatrix}, \qquad \lambda_k = \begin{bmatrix} \lambda_{1,2}^k \\ \lambda_{1,3}^k \\ \lambda_{1,4}^k \end{bmatrix} \tag{27}$$

Here the constant γ can be seen as a momentum term where the subgradient update direction is adjusted instead of changed at each step. Due to the previous direction still being considered with a factor γ. Different combinations of α and γ were considered with the best value of γ found to be between 0.8 and 0.9. As either α or γ increased close to one the error sequence started to have oscillatory behaviour caused by the subgradient update overshooting the optimal solution. Both values need to be considered together when tuning these parameters as Figure 9 shows that for small values of α, $\gamma = 0.9$ performed significantly better whereas for large values of α the opposite was seen as performance decreased for $\gamma > 0.8$.

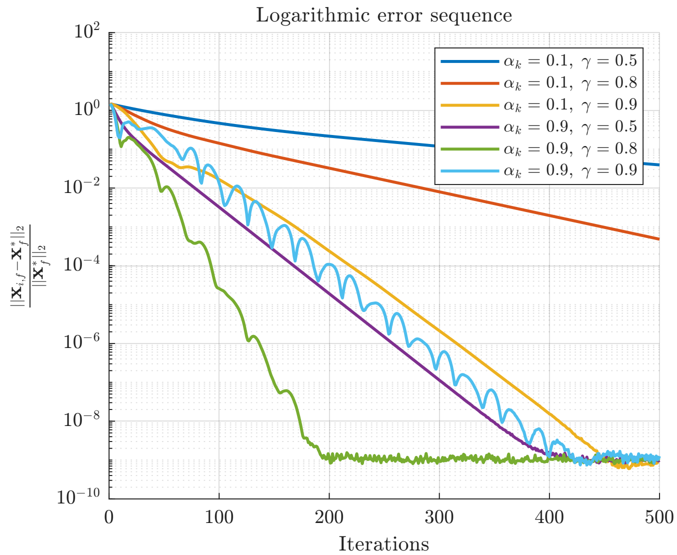

Fig. 9: Comparison of iterations needed for convergence for various values of α_k and γ using Nesterov's accelerated gradient.

Implement a combined consensus/incremental subgradient approach and investigate the effect of the number of consensus steps in each subgradient iteration on the convergence rate. Using the following consensus matrix between the agents:

$$W = \begin{bmatrix} 0.75 & 0.25 & 0 & 0 \\ 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0.25 & 0.5 & 0.25 \\ 0 & 0 & 0.25 & 0.75 \end{bmatrix}$$

For the consensus combined with an incremental subgradient approach there no longer is a master problem. Instead, a consensus matrix (W) is used to reach a consensus on the terminal state. For this approach, the original minimization problem is changed with each subproblem having its own local subgradient λ. The new objective function matrix is

$$h_i = x_i(0)^\top T_i^\top S_i + \frac{1}{2}\lambda_i^\top A_i^{eq} x_i(T_f), \quad \forall i \tag{28}$$

These local subgradients are then updated using ϕ consensus iterations giving the subgradient update as

$$\lambda_{k+1} = \lambda_k + \alpha\big(\mathbf{x}_{f,k} - W^\phi \mathbf{x}_{f,k}\big) \tag{29}$$

where

$$\mathbf{x}_{f,k} = \begin{bmatrix} x_f^1 \\ x_f^2 \\ x_f^3 \\ x_f^4 \end{bmatrix} \tag{30}$$

and W is extended through `W=kron(W, eye(4))` such that consensus is reached on each state of the aircraft.

Comparing different combinations of α and consensus iterations shows that as the number of consensus iterations increases the performance gain decreases asymptotically with no visible improvement past 50 consensus iterations.

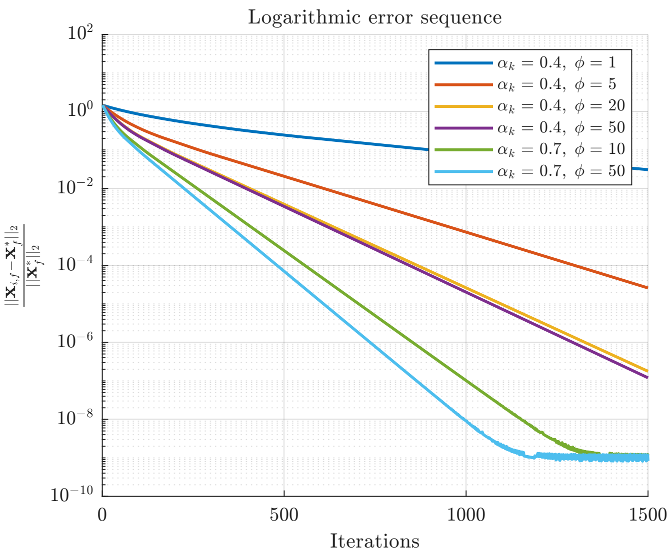

Fig. 10: Comparison of iterations needed for convergence for various combinations of α and number of consensus iterations per subgradient update.

II. Problem 2

Consider the multi-aircraft coordination formulation in Problem 1 and implement a consensus ADMM approach to solve this problem. Show a plot that demonstrates the convergence of the optimization process. Show a plot of the resulting aircraft state trajectories after convergence, and compare with the centralized optimal solution.

The augmented Lagrangian for ADMM consensus is given as

$$L_\rho(u, x_f, \lambda) = \sum_{i=1}^{N} \left(f_i(u_i) + \lambda_i^\top (x_i(T_f) - x_f) + \frac{\rho}{2} \|x_i(T_f) - x_f\|_2^2 \right) \tag{31}$$

with the update rules for x_i, x_f and λ_i as

$$x_i^{k+1} = \arg\min_{u_i} \Big(f_i(u_i) + \lambda_i^\top (x_i(T_f) - x_f) + \frac{\rho}{2} \|x_i(T_f) - x_f\|_2^2 \Big) \tag{32}$$

$$x_f^{k+1} = \frac{1}{N} \sum_{i=1}^{N} \left(x_i^{k+1} + \frac{1}{\rho} \lambda_i^k \right) \tag{33}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho \left(x_i^{k+1} - x_f^{k+1} \right) \tag{34}$$

The objective functions for use in quadratic programming then are

$$H = \left(S_i^\top S_i + I \right) + \rho \left(A_i^{eq\top} A_i^{eq} \right) \tag{35}$$

$$h_i = x_i(0)^\top T_i^\top S_i + \left(\frac{1}{2} \lambda_i^\top + \rho (b_i^{eq} - \mathbf{x}_f) \right) A_i^{eq} \tag{36}$$

Figure 11 shows that the consensus ADMM method converges in approximately one-tenth of the iterations that the dual decomposition problem needed. The state trajectories and XY plane are the same which is to be expected, Figure 12 and Figure 13 respectively.

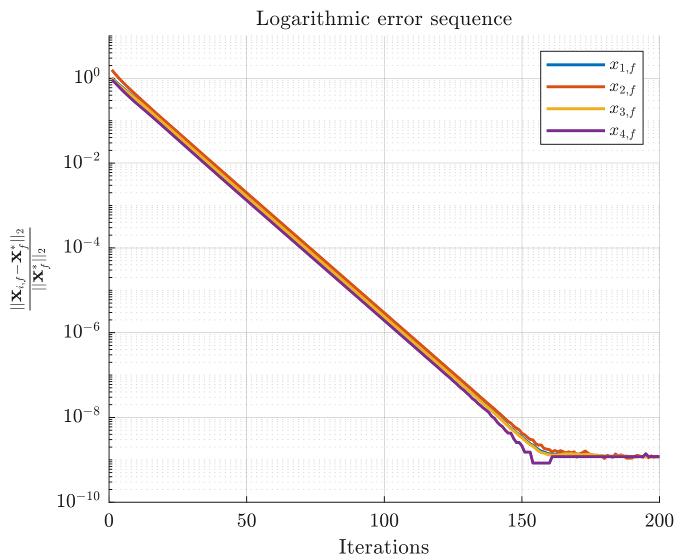

Fig. 11: Logarithmic error sequence of each aircraft compared to the optimal centralized solution using ADMM consensus with $\rho = 5$.

Investigate the effect of the parameter ρ on the convergence rate. Show the results using a logarithmic error sequence plot.

The effect of ρ was first investigated by trying different values. This caused some odd behaviour in iterations needed for convergence that was further investigated. Figure 14 shows the number of iterations needed to converge to an error less or equal to 10^{-5} for varying values of ρ. Two local minima are present with an increasing number of iterations needed for the area in between. By using these local minima and local maximum as some of the parameters of ρ the comparison seen

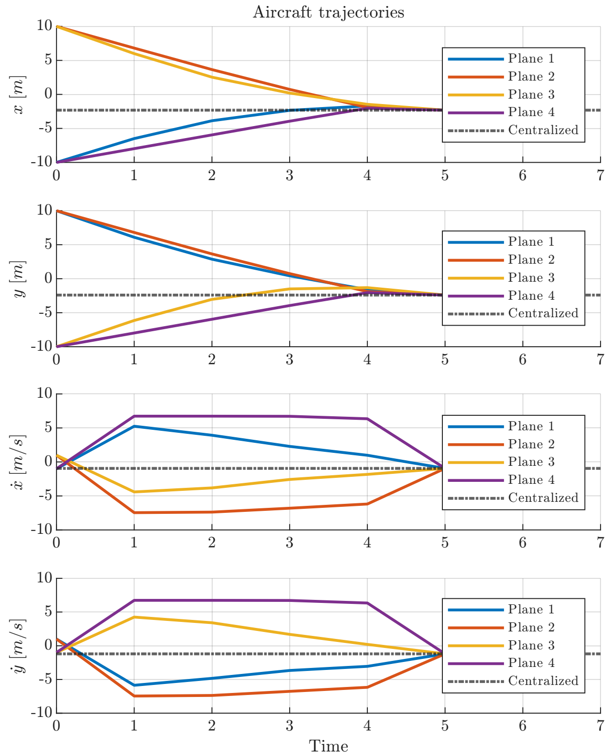

Fig. 12: State trajectories for each aircraft showing convergence to the centralized optimal solution at T_{final}.

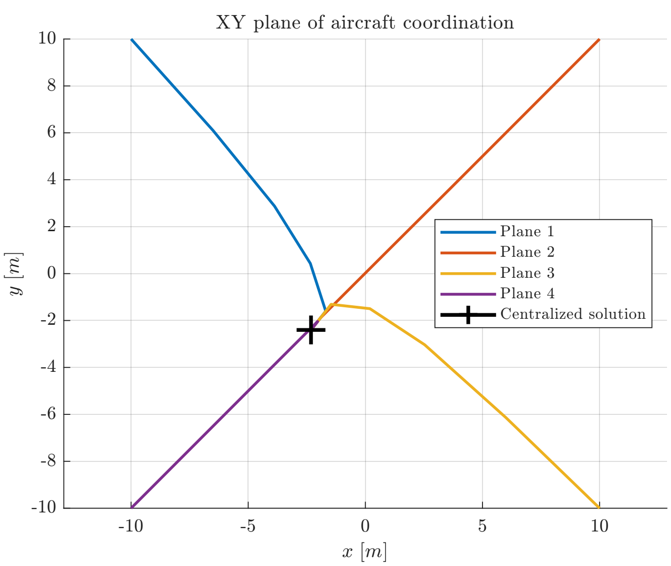

Fig. 13: XY plane showing the aircraft converging to a location southwest of the average initial position.

in Figure 15 was made. The performance of consensus ADMM can be split into three regions with the edges defined as the local minima.

$$\rho < 6.5 \tag{37}$$

$$6.5 < \rho < 10 \tag{38}$$

$$\rho > 10 \tag{39}$$

As ρ becomes smaller the iterations needed for convergence increase, for $\rho > 10$ the iterations needed for convergence also start increasing again. However, now the error after convergence also increases with increasing ρ. For values of ρ in the middle region the convergence is fastest but there is some oscillatory behaviour that also explains why there were multiple minima. Therefore $\rho = 5$ was chosen for the final state trajectories.

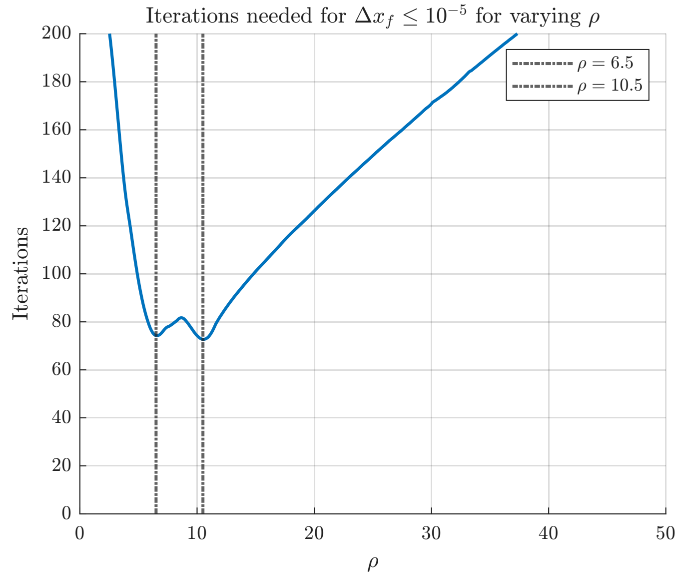

Fig. 14: Number of iterations needed for convergence within a set threshold of $\Delta x_f \leq 10^{-5}$. The two local minima of are shown with dashed lines.

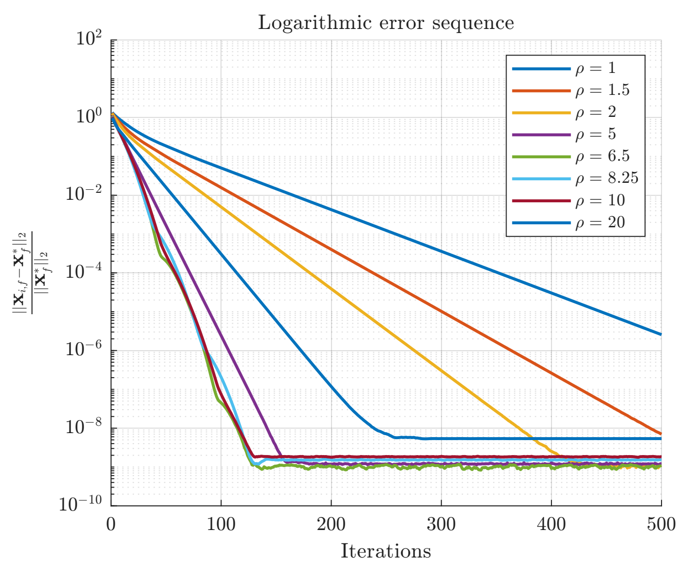

Fig. 15: Comparison of iterations needed for convergence for various values of ρ using ADMM consensus.