RESEARCH & TECHNOLOGY DEVELOPMENT CENTER

Consensus Variable Approach to Decentralized Adaptive Scheduling

Kevin L. Moore and Dennis Lucarelli

Systems and Information Science Group
Research and Technology Development Center
Johns Hopkins University Applied Physics Laboratory

presented at

5th International Conference on Cooperative Control and Optimization

University of Florida, Gainesville, Florida
20 January 2005

Outline

- **Motivating Problem: Adaptive Decentralized Scheduling**
- Consensus Variables
- Forced and Constrained Consensus Variables
- Example: Strike Mission
- Concluding Comments

Adaptive Decentralized Scheduling

- This work motivated by DARPA Coordination Decision Support Assistants (COORDINATORs) Program (BAA # 04-29)
- Interested in applications where there are
  - Multiple agents
  - Spatially distributed
  - Interacting through a sequence of inter-dependent tasks
  - That must be executed according to a prescribed schedule
  - With a prescribed allocation of tasks to resources
- Can typically solve such problems “up-front,” using some type of planning and scheduling algorithm
- However, when change occurs that upsets these plans during execution, mission plans must be adapted

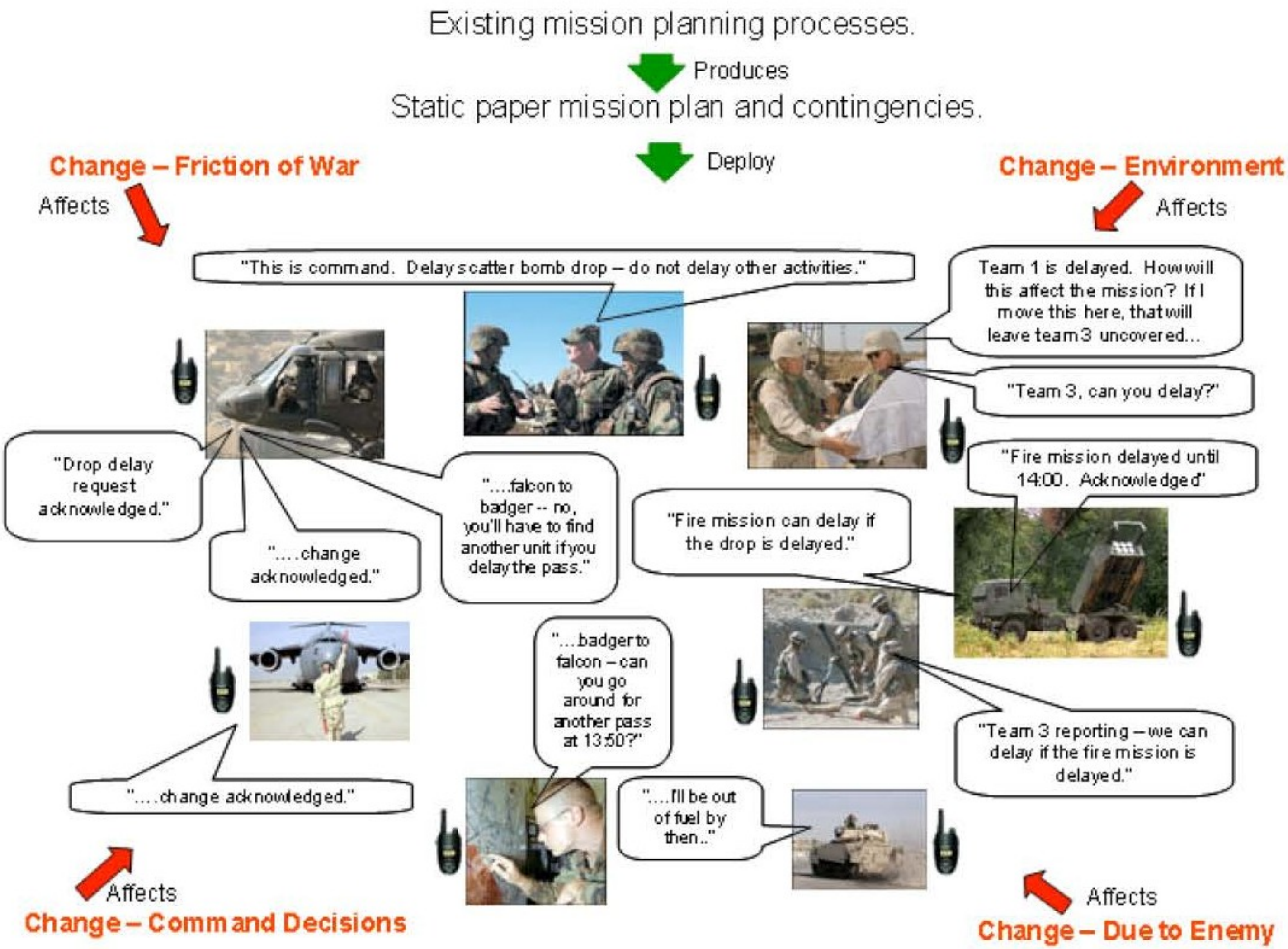


Figure 1 - Today Coordination Is Manual And Distracts Human Units

(From the DARPA Coordination Decision Support Assistants (COORDINATORs) Program (BAA # 04-29) Proposer Information Pamphlet

Adaptive Decentralized Scheduling

- Mission schedule adaptation
  - In many cases the luxury to re-plan is not available
  - In the "heat of battle" new schedule and contingencies must often be determined "on-the-spot"
  - Via team-to-team communications
  - Usually without the benefit of advanced planning tools and global domain knowledge
- The result is that coordination efforts can distract team members from the task at hand and that mission success can be compromised
- Goal: develop
  - Distributed computational system
  - Adapt existing mission plans online, in real time
  - Making changes to task timings and allocations and
  - Selecting from pre-planned contingencies

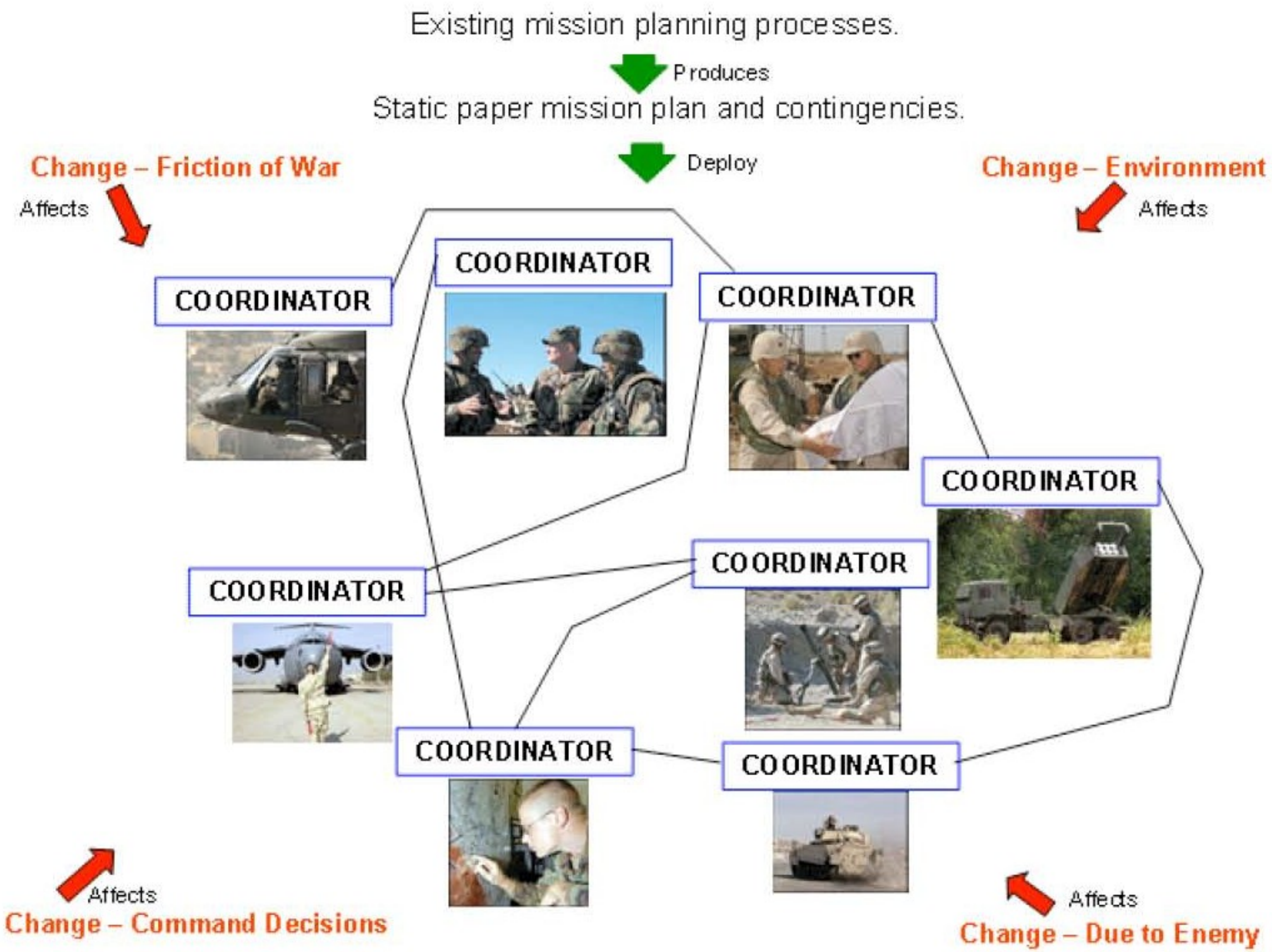


Figure 2 - With COORDINATORs Humans Can Focus On The Big Picture

(From the DARPA Coordination Decision Support Assistants (COORDINATORs) Program (BAA # 04-29) Proposer Information Pamphlet

Outline

- Motivating Problem: Adaptive Decentralized Scheduling
- **Consensus Variables**
- Forced and Constrained Consensus Variables
- Example: Strike Mission
- Concluding Comments

Consensus Variable Perspective

- Assertion:
  - Multi-agent coordination requires that *some* information must be shared
- The idea:
  - Identify the essential information, call it the *coordination or consensus variable*.
  - Encode this variable in a distributed dynamical system and come to consensus about its value
- Examples:
  - Heading angles
  - Phase of a periodic signal
  - Mission timings
- In the following we build on work by Beard, *et al*. to use consensus variables to solve the adaptive decentralized scheduling problem

Consensus Variables

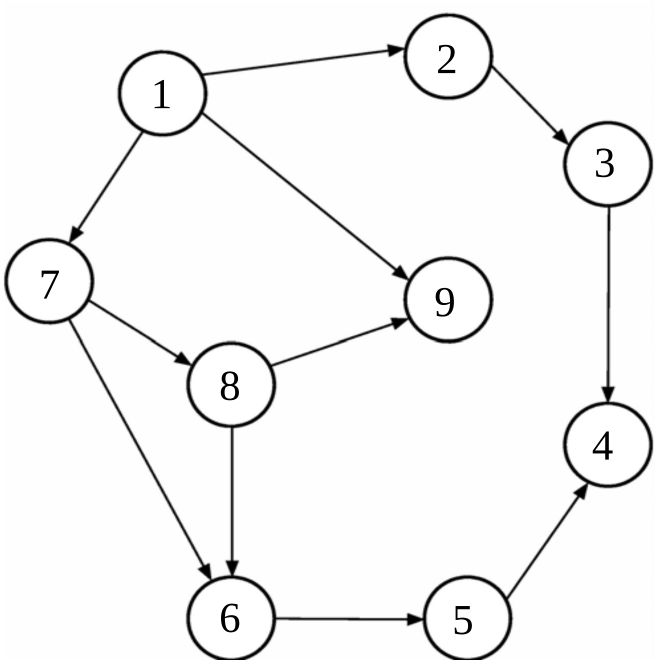
- Suppose we have N agents with a shared *global* consensus variable ξ
- Each agent has a *local* value of the variable given as ξ_i
- Each agent updates their local value based on the values of the agents that they can communicate with

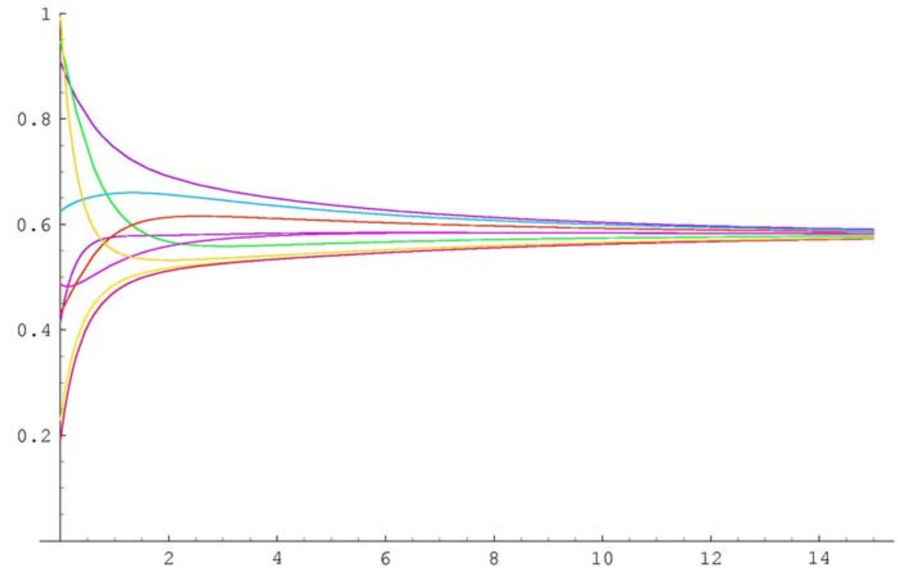
$$\dot{\xi}_i(t) = -\sum_{j=1}^{N} k_{ij}(t)G_{ij}(t)(\xi_i(t) - \xi_j(t))$$

where k_{ij} are gains and G_{ij} defines the communication topology graph of the system of agents

- Key result from literature: If the graph has a spanning tree then for all i $\xi_i \longrightarrow \xi^*$

Example: Single Consensus Variable





$$
\begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \\ \dot{\xi}_5 \\ \dot{\xi}_6 \\ \dot{\xi}_7 \\ \dot{\xi}_8 \\ \dot{\xi}_9 \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{21} & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{32} & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{43} & -k_{43}-k_{45} & k_{54} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_{56} & k_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_{67}-k_{68} & k_{67} & k_{68} & 0 \\ k_{71} & 0 & 0 & 0 & 0 & 0 & -k_{71} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{87} & -k_{87} & 0 \\ k_{91} & 0 & 0 & 0 & 0 & 0 & 0 & k_{98} & -k_{91}-k_{98} \end{bmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \\ \xi_9 \end{pmatrix}
$$

Outline

- Motivating Problem: Adaptive Decentralized Scheduling
- Consensus Variables
- **Forced and Constrained Consensus Variables**
  - From "Forced and Constrained Consensus Among Cooperating Agents," K.L. Moore and D. Lucarelli, to appear in *Proceedings of 2005 IEEE International Conference on Networking, Sensing, and Control*, Tuscon, AZ, March 2005
- Example: Strike Mission
- Conclusion

Extension 1 - Forced Consensus

- **Forced Consensus**
  - Sometimes we may like to force all the nodes to follow a hard constraint
  - This can be done by injecting an input into a node as follows

$$\dot{\xi}_i(t) = -\sum_{j=1}^{N} k_{ij}(t) G_{ij}(t)(\xi_i(t) - \xi_j(t)) + b_i u_i$$

  - Then we use a feedback controller as given in the following

- **Theorem** Let A be a set of agents with $b_k = 1$, $b_i = 0, \forall i \neq k$, and

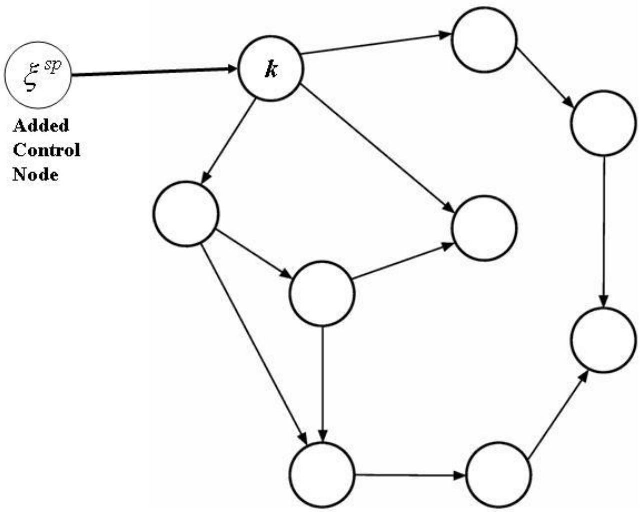
$$u_k(t) = k_p(\xi^{sp} - \xi_k)$$

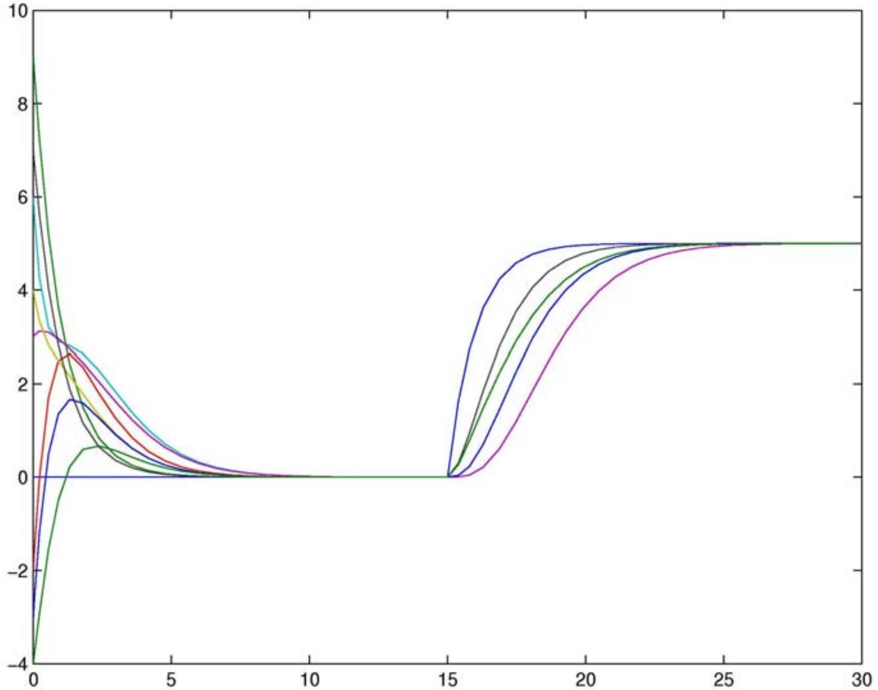
where ξ^{sp} is a constant setpoint and $k_p > 0$ is a constant gain. Then the consensus strategy achieves global asymptotic consensus for A, with

$$\lim_{t \to \infty} \xi_i(t) = \xi^{sp} \quad \forall i$$

if and only if node k is a spanning node for the communication graph G.

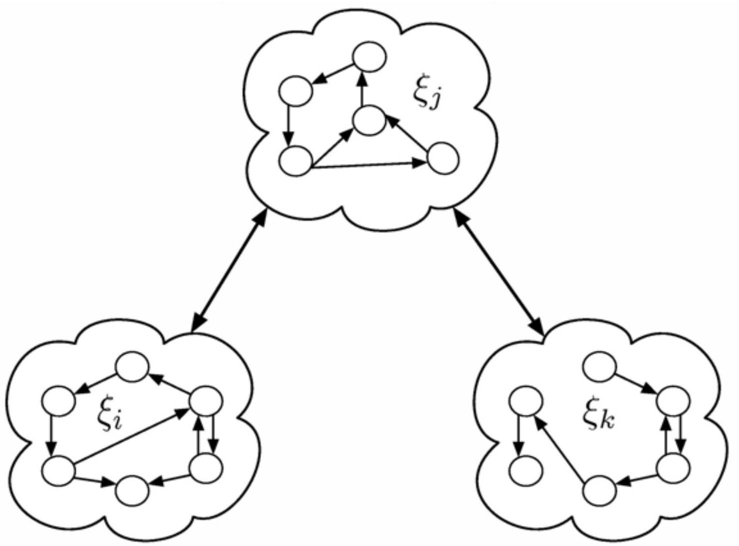
Example – Forced Consensus





Extension 2 – Multiple, Constrained Consensus

- Often we will have multiple consensus variables in a given problem



- It can be useful to enforce constraints between these variables, specifically, to have $\xi_i = \xi_j + \Delta_{ij}$
- Again we can give a feedback control strategy to achieve this type of constrained consensus between groups of agents

Theorem Let A^a and A^b be two set of agents, each negotiating locally about consensus variables ξ^a and ξ^b, respectively, and each with communication graphs G^a and G^b defined by communication topologies G_{ij}^a and G_{ij}^b, respectively. Suppose

1. Each agent set updates the local values of their consensus variable by

$$\begin{aligned}\dot{\xi}_i^a(t) &= -\sum_{j=1}^{n^a} k_{ij}^a(t)G_{ij}^a(t)(\xi_i^a(t)-\xi_j^a(t))+b_i^a u_i^a \\ \dot{\xi}_i^b(t) &= -\sum_{j=1}^{n^b} k_{ij}^b(t)G_{ij}^b(t)(\xi_i^b(t)-\xi_j^b(t))+b_i^b u_i^b\end{aligned}$$

   where $b_{k^a}^a = 1, b_i^a = 0, \forall i \neq k^a$ and $b_{k^b}^b = 1, b_i^b = 0, \forall i \neq k^b$

2. The two agent sets communicate to each other via the nodes k^a and k^b using the following agent-to-agent consensus update law:

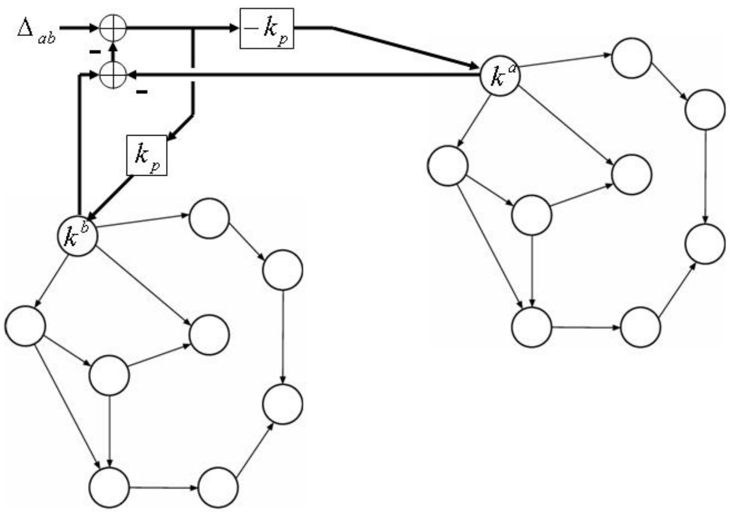
$$\begin{aligned}u_{k^a}^a &= -(\Delta_{ab}-(\xi_{k^b}^b-\xi_{k^a}^a)) \\ u_{k^b}^b &= \Delta_{ab}-(\xi_{k^b}^b-\xi_{k^a}^a)\end{aligned}$$

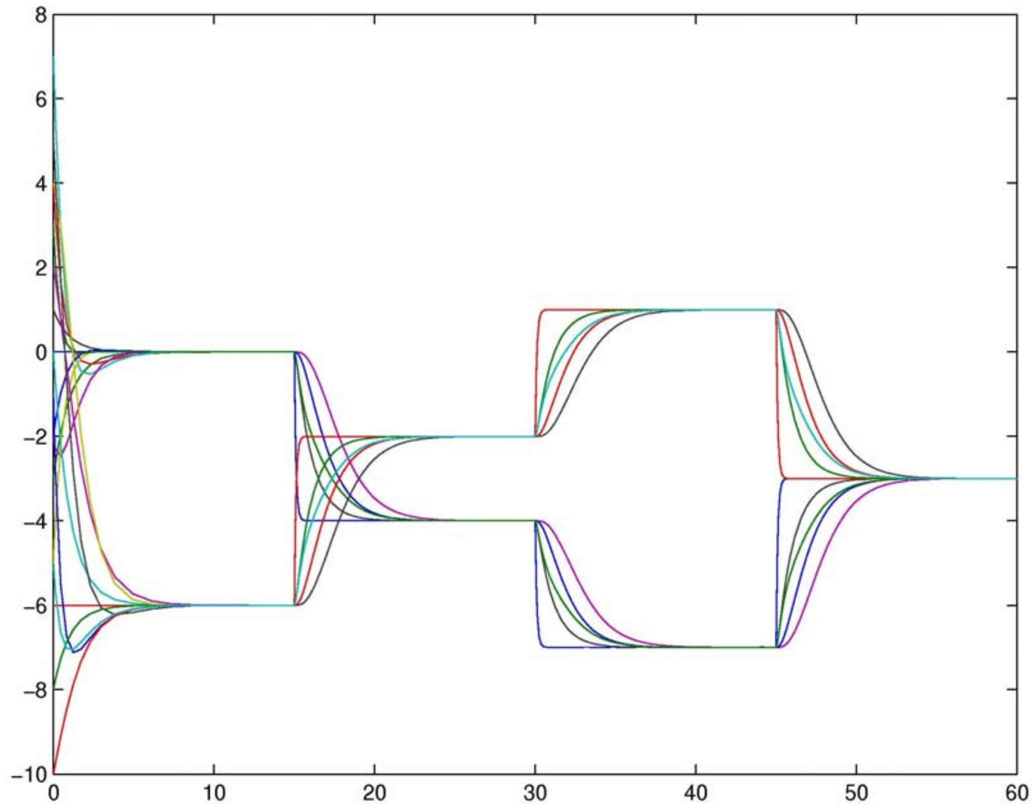
Then the consensus strategy achieves global asymptotic global consensus for each set A^a and A^b, with

$$\begin{aligned}\xi_i^a &\rightarrow \xi^{a^*} \\ \xi_i^b &\rightarrow \xi^{b^*} \\ \xi^{b^*} &= \xi^{a^*}+\Delta_{ab}\end{aligned}$$

if and only if nodes k^a and k^b are spanning nodes for the graphs G^a and G^b, respectively.

Example – Multiple, Constrained Consensus





JOHNS HOPKINS
UNIVERSITY
Applied Physics Laboratory

Outline

- Motivating Problem: Adaptive Decentralized Scheduling
- Consensus Variables
- Forced and Constrained Consensus Variables
- **Example: Strike Mission**
- Concluding Comments

Example: Strike Mission

- Three teams (each team or unit is considered an agent)
  1. Air drop team MH-J = Unit 1
  2. Special Forces team SF = Unit 2
  3. Seal Team and their boat MK-V = Unit 3
- Each team *i* has a series of ordered tasks *j*, denoted *Tij*
- The tasks of some teams are pre-requisite for the tasks of some other teams
- For some tasks there are different contingencies for carrying out the task
- Different contingencies have different costs
  - In our example contingencies are parameterized by time-to-complete
- Goal is to develop a decentralized coordination algorithm to adapt required start and end times for specific tasks based on changes in
  - Required mission end time
  - Changes in individual task execution times (e.g., disturbances)

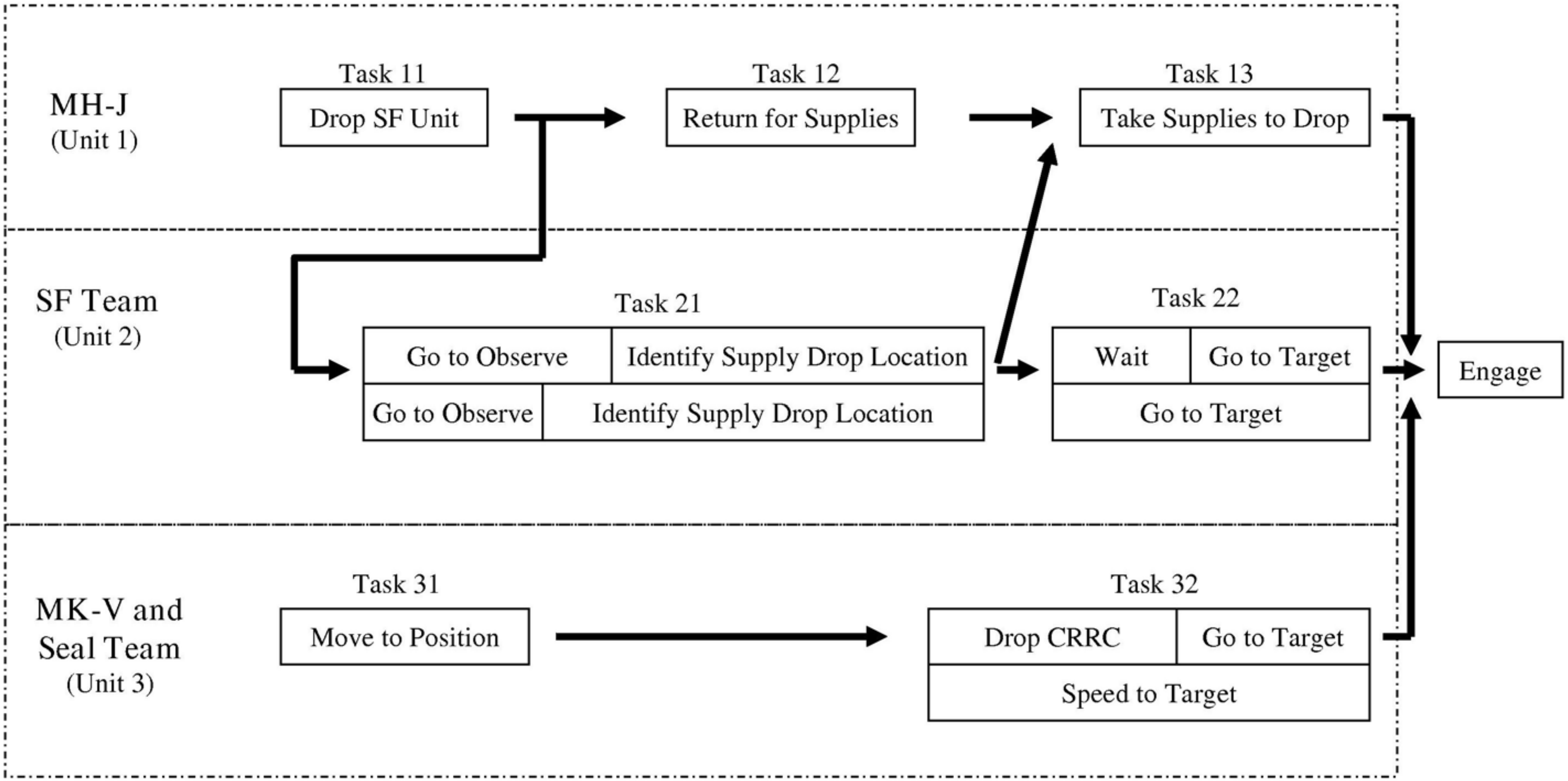
Example: Strike Mission

Scenario:

- Air Drop team deploys SF team and returns to pick up supplies
- Simultaneously Seal Team moves to beach landing
- SF Team moves to observation position to identify drop location
- SF Team relays drop location to Air Drop team and then moves to drop location
- When supplies are dropped and SF and Seal Team are in place, then all teams execute

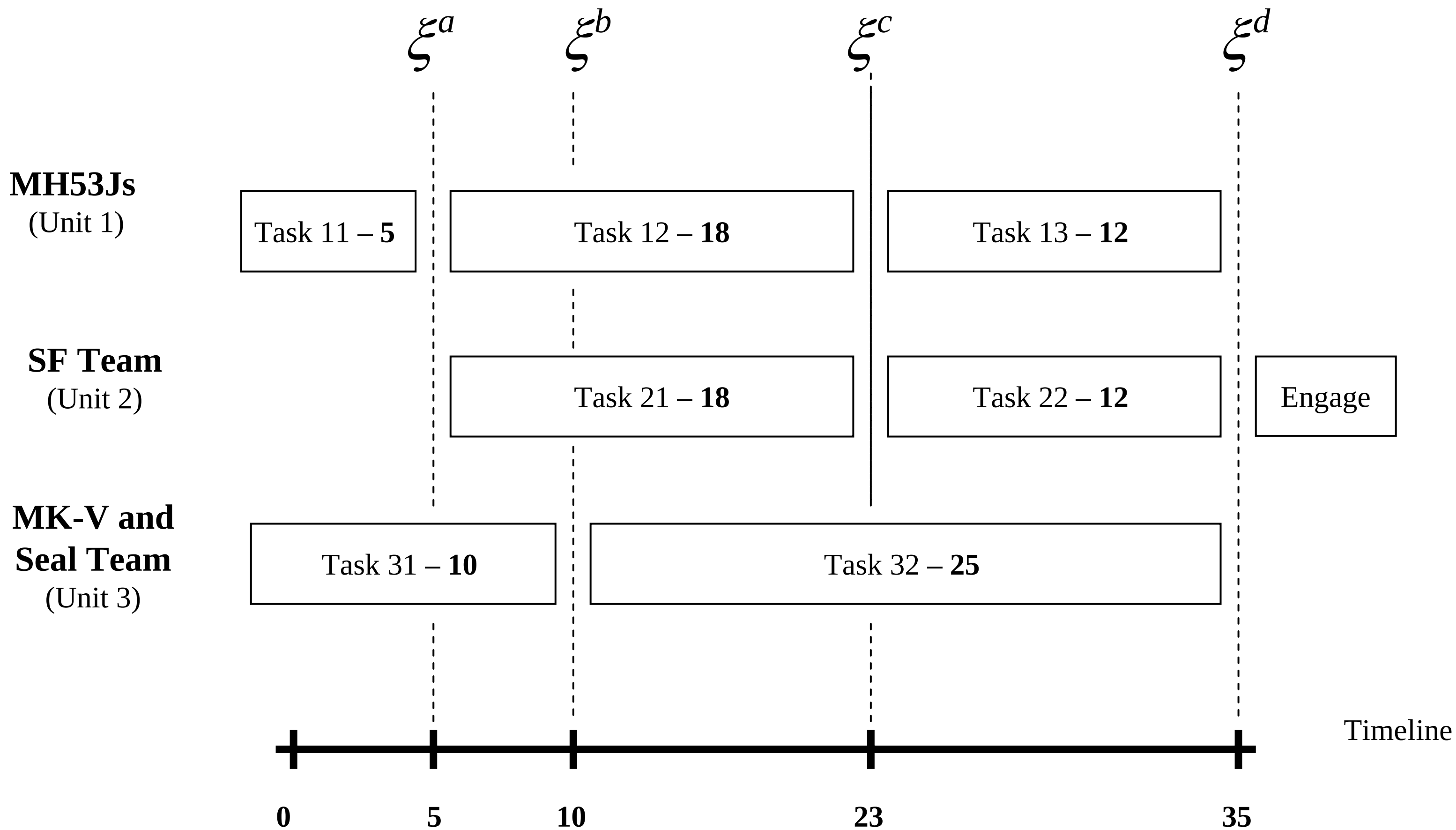
Strike Mission Task Dependencies

- **Synchronized Strike Mission:**



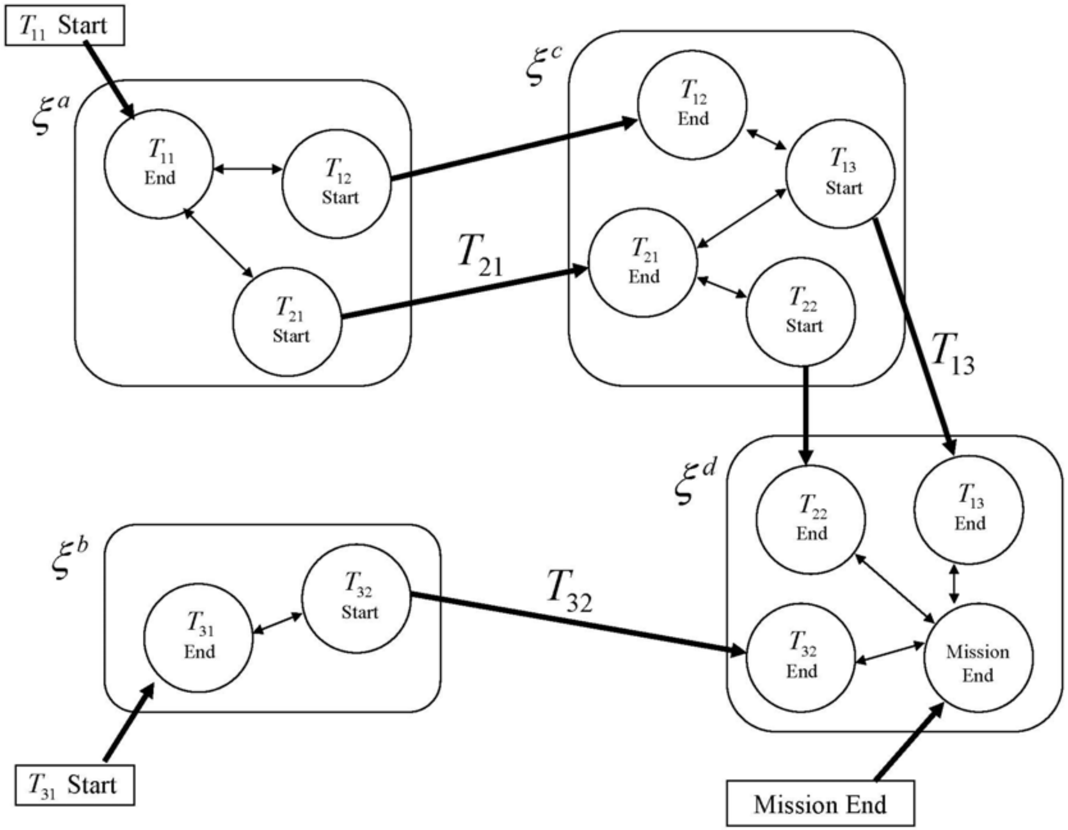
Consensus Variable Definitions

- **Key concept**: consensus variables are chosen to be task intersection times (nominal mission durations and consensus times are shown):



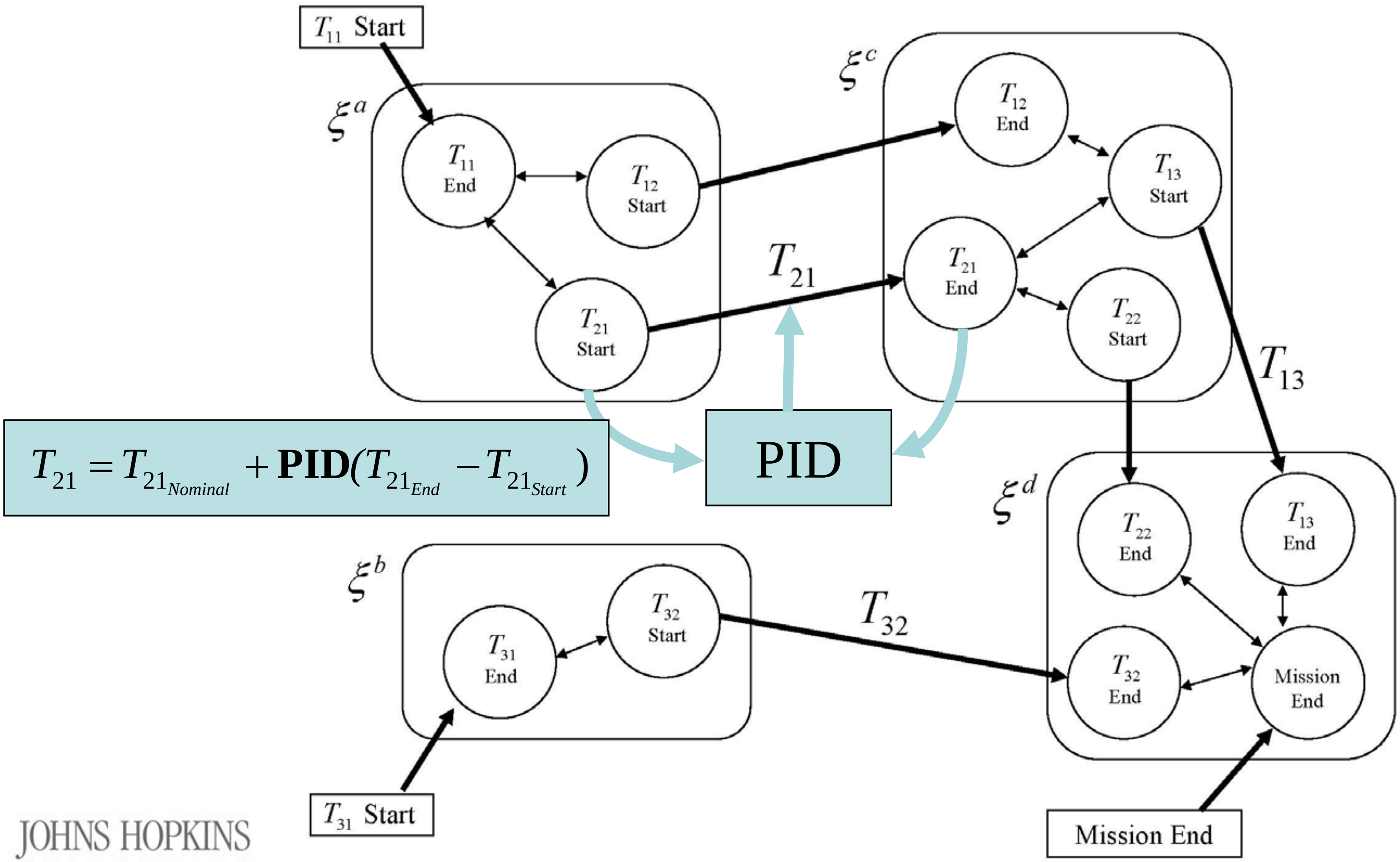
Agent Topology for Example

- Use forced offset to define start time and engagement setpoints and use prescribed task durations to constrain the offset between consensus variables:



Agent-Level Contingency Selection

- One additional feature – adjustment of task times:



Example Adaptive Behavior

- To describe the global system behavior, define:

$$\xi_1^a = T_{11_{End}}$$

$$\xi_2^a = T_{12_{Start}}$$

$$\xi_3^a = T_{21_{Start}}$$

$$\xi_1^b = T_{31_{End}}$$

$$\xi_2^b = T_{22_{Start}}$$

$$\xi_1^c = T_{12_{End}}$$

$$\xi_2^c = T_{21_{Start}}$$

$$\xi_3^c = T_{13_{Start}}$$

$$\xi_4^c = T_{22_{Start}}$$

$$\xi_1^d = T_{32_{End}}$$

$$\xi_2^d = T_{22_{end}}$$

$$\xi_3^d = T_{13_{End}}$$

$$\xi_4^d = MissionEnd$$

Global System Model

- The resulting overall system equations have incorporated:
  - Initial condition offsets
  - Task-length constraints between consensus variables
  - Task-length adjustment to respond to changes

$$
\begin{aligned}
\dot{\xi}_1^a &= -k_{12}^a(\xi_1^a - \xi_2^a) - k_{13}^a(\xi_1^a - \xi_3^a) + (T_{11} - \xi_1^a) \\
\dot{\xi}_2^a &= -k_{21}^a(\xi_2^a - \xi_1^a) \\
\dot{\xi}_3^a &= -k_{31}^a(\xi_3^a - \xi_1^a) - k_{32}^{ac}(T_{21} + \xi_3^a - \xi_2^c) \\
\dot{\xi}_1^b &= -k_{12}^b(\xi_1^b - \xi_2^b) + (T_{31} - \xi_1^b) \\
\dot{\xi}_2^b &= -k_{21}^b(\xi_2^b - \xi_1^b) - k_{21}^{bd}(T_{32} + \xi_2^b - \xi_1^d) \\
\dot{\xi}_1^c &= -k_{13}^c(\xi_1^c - \xi_3^c) \\
\dot{\xi}_2^c &= -k_{23}^c(\xi_2^c - \xi_3^c) - k_{24}^c(\xi_2^c - \xi_4^c) + k_{23}^{ac}(T_{21} + \xi_3^a - \xi_2^c) \\
\dot{\xi}_3^c &= -k_{31}^c(\xi_3^c - \xi_1^c) - k_{32}^c(\xi_3^c - \xi_2^c) - k_{33}^{cd}(T_{13} + \xi_3^c - \xi_3^d) \\
\dot{\xi}_4^c &= -k_{42}^c(\xi_4^c - \xi_2^c) \\
\dot{\xi}_1^d &= -k_{14}^d(\xi_1^d - \xi_4^d) + k_{12}^{bd}(T_{32} + \xi_2^b - \xi_1^d) \\
\dot{\xi}_2^d &= -k_{24}^d(\xi_2^d - \xi_4^d) \\
\dot{\xi}_3^d &= -k_{34}^d(\xi_3^d - \xi_4^d) + k_{33}^{cd}(T_{13} + \xi_3^c - \xi_3^d) \\
\dot{\xi}_4^d &= -k_{41}^d(\xi_4^d - \xi_1^d) - k_{42}^d(\xi_4^d - \xi_2^d) - k_{43}^d(\xi_4^d - \xi_3^d) + \mathrm{PID}(\mathrm{SP} - \xi_4^d) \\
T_{13} &= T_{13_{\mathrm{nominal}}} + \mathrm{PID}(\xi_4^d - \xi_3^d) \\
T_{21} &= T_{21_{\mathrm{nominal}}} + \mathrm{PID}(\xi_3^c - \xi_2^c) \\
T_{32} &= T_{32_{\mathrm{nominal}}} + \mathrm{PID}(\xi_4^d - \xi_1^d)
\end{aligned}
$$

Example Adaptive Behavior

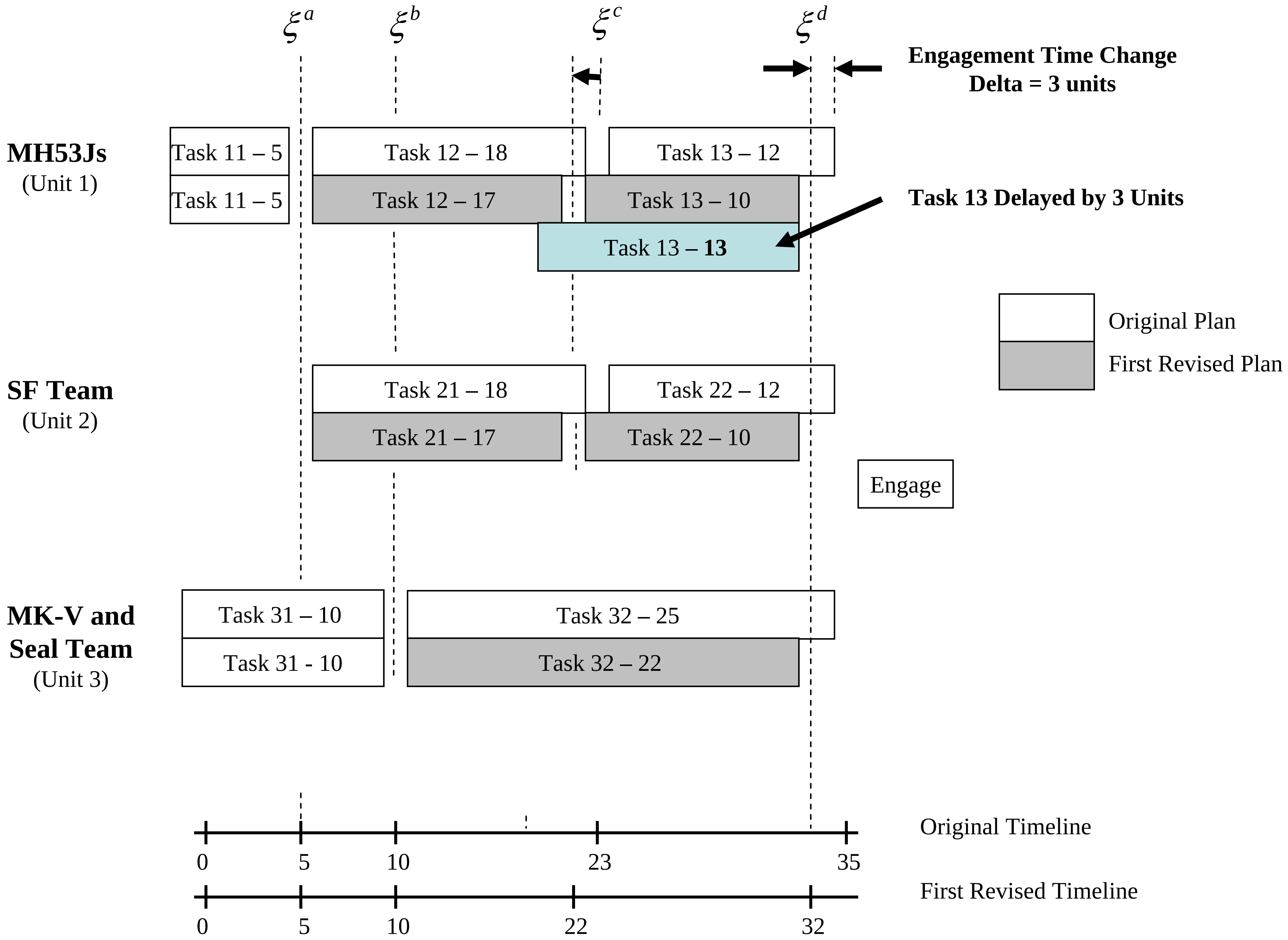
- Consider the resulting consensus variable values as the system adapts to two events:

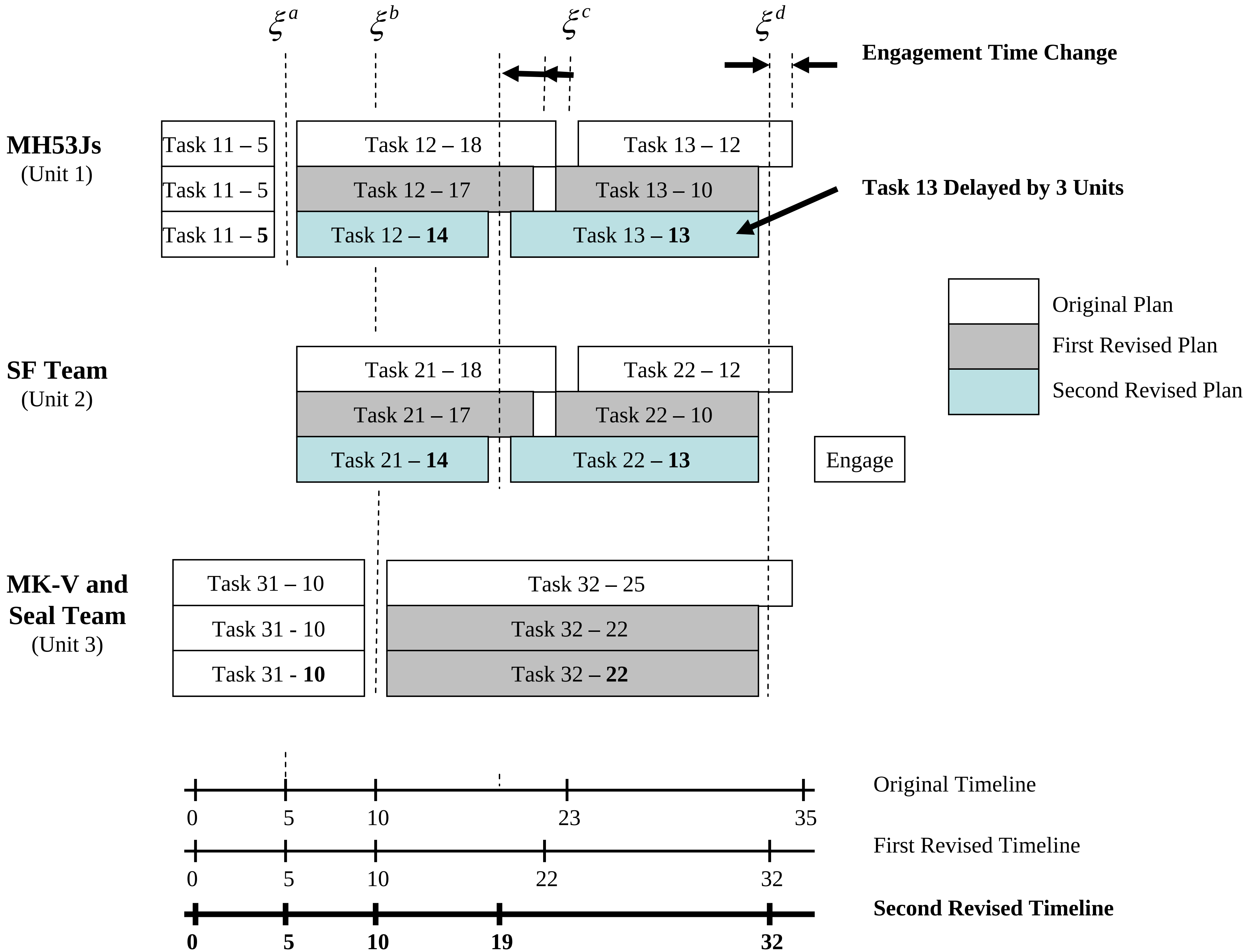
1. Change in Engagement Deadline

   Represented by a change in the setpoint for ξ^d

2. Change in Task time for Task T_{13}

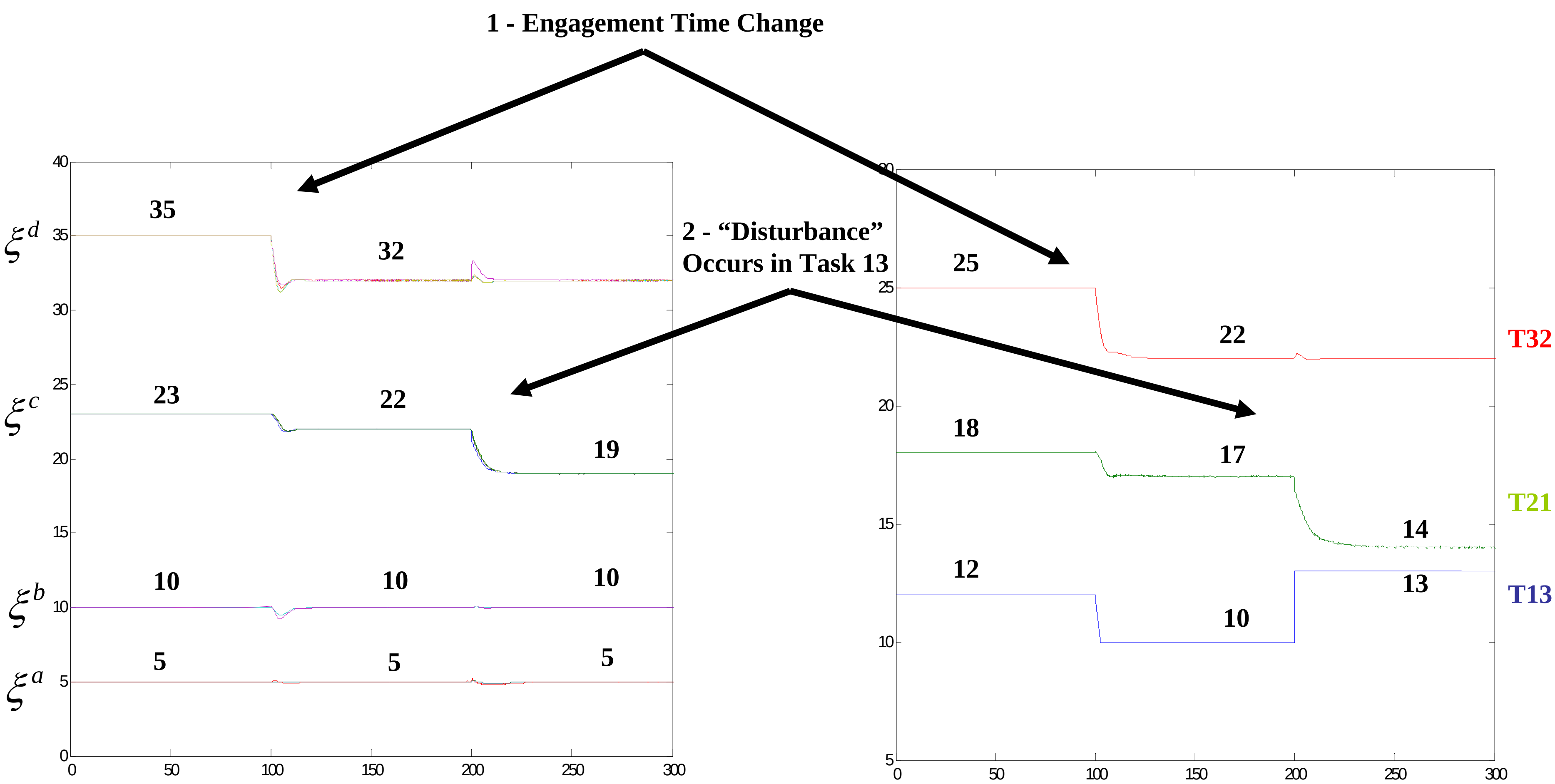
   Represented by a change in $T_{13_{Nominal}}$


ξ^a
ξ^b
ξ^c
ξ^d
Engagement Time Change
Delta = 3 units
MH53Js
(Unit 1)
Task 11 – 5
Task 12 – 18
Task 13 – 12
Task 11 – 5
Task 12 – 17
Task 13 – 10
Task 13 Delayed by 3 Units
Task 13 – 13
Original Plan
First Revised Plan
SF Team
(Unit 2)
Task 21 – 18
Task 22 – 12
Task 21 – 17
Task 22 – 10
Engage
MK-V and
Seal Team
(Unit 3)
Task 31 – 10
Task 32 – 25
Task 31 - 10
Task 32 – 22
Original Timeline
0
5
10
23
35
First Revised Timeline
0
5
10
22
32



ξ^a
ξ^b
ξ^c
ξ^d
Engagement Time Change
MH53Js
(Unit 1)
Task 11 – 5
Task 12 – 18
Task 13 – 12
Task 11 – 5
Task 12 – 17
Task 13 – 10
Task 11 – **5**
Task 12 – **14**
Task 13 – **13**
Task 13 Delayed by 3 Units
Original Plan
First Revised Plan
Second Revised Plan
SF Team
(Unit 2)
Task 21 – 18
Task 22 – 12
Task 21 – 17
Task 22 – 10
Task 21 – **14**
Task 22 – **13**
Engage
MK-V and
Seal Team
(Unit 3)
Task 31 – 10
Task 32 – 25
Task 31 - 10
Task 32 – 22
Task 31 - **10**
Task 32 – **22**
0 5 10 23 35
Original Timeline
0 5 10 22 32
First Revised Timeline
0 5 10 19 32
Second Revised Timeline

Adapting to Two Events



(a) Consensus Variable Values

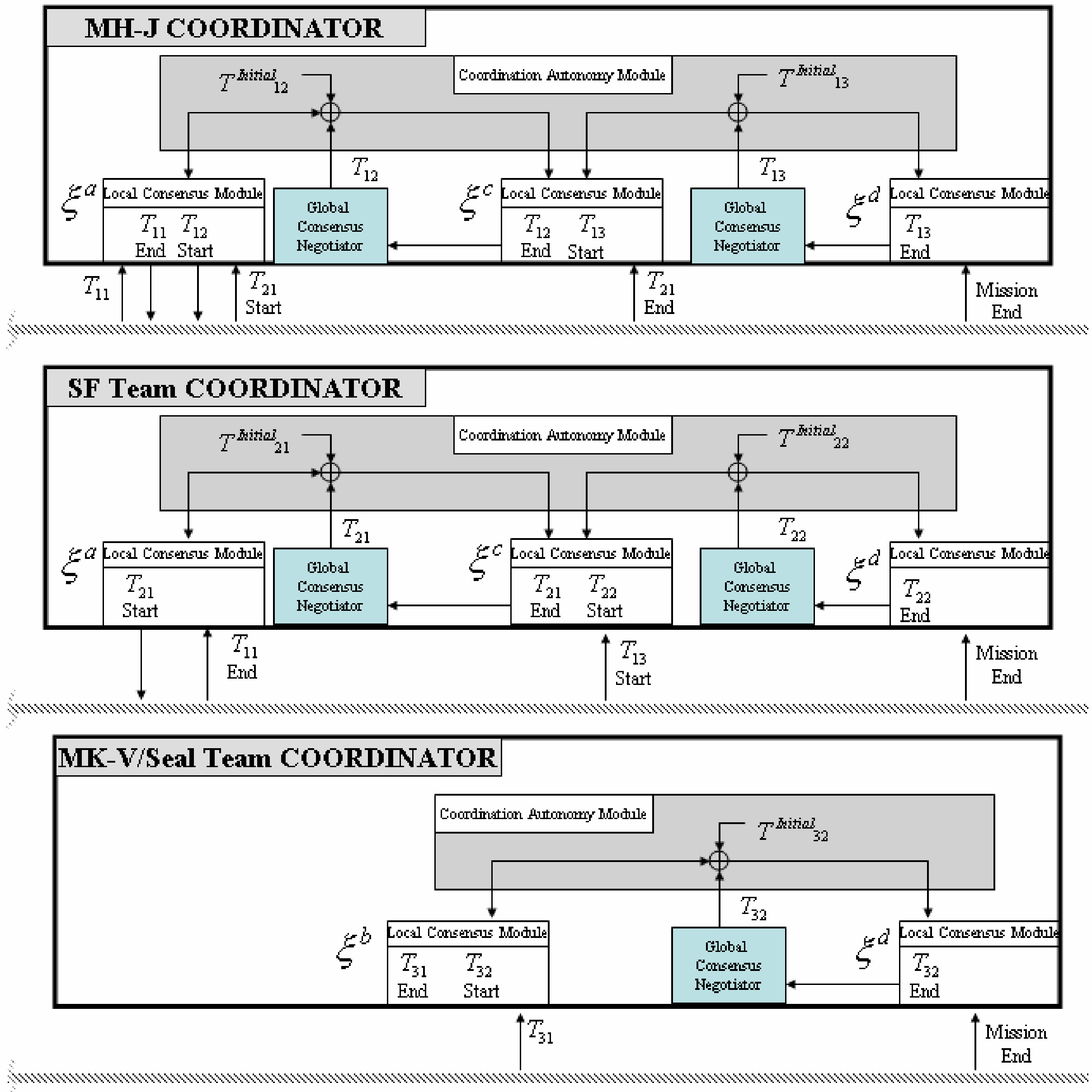
(b) Key Task Timings

Where the Variables Live

- Values of the various consensus variables actually evolve in different places:
  - Unit 1
  - Unit 2:
  - Unit 3:
  - Central Command:
- We also think about computations as being
  - Global
  - Local

$$
\begin{aligned}
\dot{\xi}_1^a &= -k_{12}^a(\xi_1^a - \xi_2^a) - k_{13}^a(\xi_1^a - \xi_3^a) + (T_{11} - \xi_1^a) \\
\dot{\xi}_2^a &= -k_{21}^a(\xi_2^a - \xi_1^a) \\
\dot{\xi}_3^a &= -k_{31}^a(\xi_3^a - \xi_1^a) - k_{32}^{ac}(T_{21} + \xi_3^a - \xi_2^c) \\
\dot{\xi}_1^b &= -k_{12}^b(\xi_1^b - \xi_2^b) + (T_{31} - \xi_1^b) \\
\dot{\xi}_2^b &= -k_{21}^b(\xi_2^b - \xi_1^b) - k_{21}^{bd}(T_{32} + \xi_2^b - \xi_1^d) \\
\dot{\xi}_1^c &= -k_{13}^c(\xi_1^c - \xi_3^c) \\
\dot{\xi}_2^c &= -k_{23}^c(\xi_2^c - \xi_3^c) - k_{24}^c(\xi_2^c - \xi_4^c) + k_{23}^{ac}(T_{21} + \xi_3^a - \xi_2^c) \\
\dot{\xi}_3^c &= -k_{31}^c(\xi_3^c - \xi_1^c) - k_{32}^c(\xi_3^c - \xi_2^c) - k_{33}^{cd}(T_{13} + \xi_3^c - \xi_3^d) \\
\dot{\xi}_4^c &= -k_{42}^c(\xi_4^c - \xi_2^c) \\
\dot{\xi}_1^d &= -k_{14}^d(\xi_1^d - \xi_4^d) + k_{12}^{bd}(T_{32} + \xi_2^b - \xi_1^d) \\
\dot{\xi}_2^d &= -k_{24}^d(\xi_2^d - \xi_4^d) \\
\dot{\xi}_3^d &= -k_{34}^d(\xi_3^d - \xi_4^d) + k_{33}^{cd}(T_{13} + \xi_3^c - \xi_3^d) \\
\dot{\xi}_4^d &= -k_{41}^d(\xi_4^d - \xi_1^d) - k_{42}^d(\xi_4^d - \xi_2^d) - k_{43}^d(\xi_4^d - \xi_3^d) + \mathrm{PID}(\mathrm{SP} - \xi_4^d) \\
T_{13} &= T_{13_{\text{nominal}}} + \mathrm{PID}(\xi_4^d - \xi_3^d) \\
T_{21} &= T_{21_{\text{nominal}}} + \mathrm{PID}(\xi_3^c - \xi_2^c) \\
T_{32} &= T_{32_{\text{nominal}}} + \mathrm{PID}(\xi_4^d - \xi_1^d)
\end{aligned}
$$

Agent Architecture



Concluding Comments - 1

- We have presented a consensus variable approach to adaptive decentralized scheduling
  - Introduced the ideas of forced and constrained consensus
  - Applied these ideas by defining task start and stop times in a mission to be the consensus variables to be negotiated by cooperating teams
  - Showed an architecture for implementing the ideas
- Our approach is differentiated from classical approaches to schedule adaptation:
  - It is provable and, we believe, scalable
  - Global communication is not required
  - We do not do re-planning

Concluding Comments - 2

- Future work aims to extend these ideas in several ways
  - Uncertainties in constraints and communications can be handled explicitly and algorithmically using a Kalman filtering approach
  - We are exploring the effect of structural changes, such as node loss, and how to handle them using re-configurable control ideas
  - We are applying the approach to handle other variables, such as resources, and to explicitly handle the trade off between local and global cost functions during consensus negotiations
  - We are considering how to include probabilistic considerations, making it possible to place confidence intervals on contingency options