

A Scalable Command and Control System for Human-Machine Work Systems

Aaron C. Morris, Charles K. Smart, and Scott M. Thayer

Robotics Institute
5000 Forbes Avenue
Carnegie Mellon University
Pittsburgh, PA 15213
email: acmorr@ri.cmu.edu, cks@andrew.cmu.edu, sthayer@ri.cmu.edu

Abstract. The emergence of complex work systems has yielded new challenges for efficient and reliable collaboration between humans and machines. Robots are now working autonomously beside human counterparts to accomplish critical tasks; however fully autonomous robot action is still considered unreliable. This paper examines an approach to increasing the robustness, reliability, and efficiency of human-machine work systems by dynamically establishing dynamic control relationships between humans and robots as well as altering the effective autonomy manifested by each robot. The process involves **workload estimation** to determine the parameters of the system, **workload optimization** to analyze and modify system parameters, and **workload mitigation** to enact these modifications in a non-intrusive manner. Furthermore, heuristic approaches to approximating an optimal system configuration for real-time environments are also addressed and simulated.

Key words: human-machine work systems; workload estimation, optimization, and mitigation

1 Introduction

Unstructured and hostile environments impose risk to exposed humans and present ideal domains for robotic applications; however, the conditions of these environments present significant challenges (i.e. complex obstacles, lack of sufficient prior knowledge, etc.) for autonomous robot operation and can impede a robot from achieving a desired goal. The emerging use of multiple robots has compounded these problems since the success of one robot may depend upon the performance of other robots within the collective.

To mitigate these challenges, an approach that integrates human cognition into the robot control structure is proposed. As first discussed in [1], researchers from a spectrum of fields have developed technologies aligned with this theme. Some recent examples of this research include human-computer interfaces [2], distributed software agents [3], human factors modeling [4], and adjustable

autonomy [5]. Even components of human thought have been modeled [6] [7] in order to create a more seamless connection between human and computer.

This paper discusses the enabling merger of computational cognitive estimation with robot control theory to produce a system enabling a small group of humans to manage a larger group of semi-autonomous robots. The bulk of the work outlined herein examines the mathematical model used to create this linkage between human cognition and robot control. This work is presented as a three-part process (**workload estimation**, **workload optimization**, and **workload mitigation**) detailing the system representation. Finally, an analysis of simulations is presented that portends improvements in human-machine control.

2 The Human-Machine Work System

A human-machine work system is a collaboration of multiple heterogeneous “natural and artificial cognitive systems” [2] engaged in the execution of shared tasks. The corresponding notation for representing this system is $\mathbf{H} = \{h_1, h_2, \dots, h_M\}$ for M humans, $\mathbf{R} = \{r_1, r_2, \dots, r_N\}$ for N robots, and $\mathbf{T} = \{t_1, t_2, \dots, t_P\}$ for P tasks.

Robots are also decomposed into two sub-layers identified as subsystems and functions. A subsystem describes a particular interface, behavior, or capability of a robot and is denoted as $r = \{s_1, \dots, s_i, \dots, s_K\} \Rightarrow r \subseteq \mathbf{S}$ where s_i is a subsystem (i.e. *navigation, target recognition, manipulation, etc*) within the set of all subsystems \mathbf{S} . A function represents a particular action composing the robotic subsystem and is denoted as $s_i = \{f_1, \dots, f_j, \dots, f_j\} \Rightarrow s_i \subseteq \mathbf{F}$ where f_j is a particular function (i.e. *sensor feedback, path planning, and steering*) within the set of all functions \mathbf{F} . Furthermore, \mathbf{T} , \mathbf{H} , \mathbf{R} , \mathbf{S} , and \mathbf{F} create a system that operates by estimating workload parameters, optimizing these parameters through analysis of system configuration, and enacting an optimized configuration through alteration of robot autonomy and human-function pairing (Figure 1).

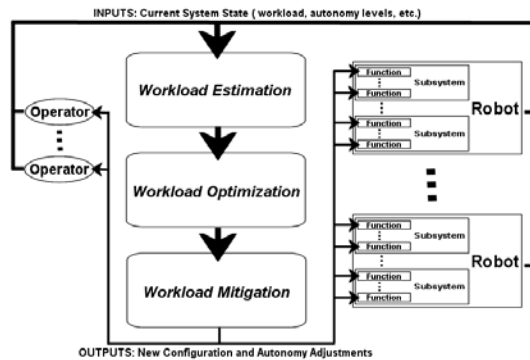


Fig. 1. Architecture of the human-machine work system.

From these baseline definitions, human-machine work systems can be evolved into complex structures that consist of multiple tasks distributed over multiple robots and managed by multiple operators. To formalize and constrain system organization, the following axioms will be used.

- (1) A function can be controlled by only one operator**
- (2) An operator can control multiple functions**
- (3) Tasks can occupy multiple subsystems**
- (4) A single subsystem can concurrently execute multiple tasks**

3 Workload Estimation

The human's role as an administrator requires the injection of control signals at the robot functional level. Human cognition, however, is an exhaustible resource that restricts the number of functions they can manage. Additional activities beyond a given threshold will sacrifice the performance of some (or all) concurrent activities [4]. To quantitatively monitor this threshold, the ACT-R (Adaptive Character of Thought – Rational) cognitive architecture [6] is utilized. According to ACT-R, human cognition can be separated into six distinct, quantifiable areas consisting of working memory, long-term memory, vision, speech, motor, and audition. With this representation, human cognitive capacity can be modeled in a vector form, $\vec{\Lambda} = [\lambda_{vm}, \lambda_{lm}, \lambda_v, \lambda_s, \lambda_m, \lambda_a]^T$, to quantify the capacity within each area of the human brain.

In a similar manner, a formal means of representing the induced cognitive load of human interaction with functional components is derived. The cognitive loading resulting from human interaction with robotic functions is defined as $\vec{\Gamma} = [\varphi_{vm}, \varphi_{lm}, \varphi_v, \varphi_s, \varphi_m, \varphi_a]^T$ corresponding to the cognitive load induced upon a particular region of the human brain as measured by ACT-R.

It is important to note that the level of autonomy governing the behavior of a robot function will have a profound impact upon the induced cognitive load [7]. For example, robot functions operating under full autonomy may require little or no attention from the operator and consequentially, the induced cognitive load will be relatively small. The approach to modeling this effect is discussed in the following section.

4 Adjustable Autonomy and Workload Mitigation

To enable environmental adaptation, this system requires that robot autonomy dynamically vary as the robot encounters uncertainties. Adjustable autonomy ([8] and [5]) allows robot functions to be controlled at various levels of human

interaction so that the shared workload of humans and robots may be mitigated to the constraints of the work system. For example, a robot may require only minimal supervision during a navigation task until complex terrain obstructs its path. A lower level of autonomy may then be invoked so that human control assists the robot in avoiding obstacles. Adapted from [3], we define four general autonomy levels for robot functions:

1. **Fully autonomous:** the highest level of autonomy where control is determined by intrinsic functional capabilities. Operators may periodically monitor and asynchronously alter the progress of a unit, but the robot is otherwise independent.
2. **Semi-Autonomous:** the autonomy level that maintains the independence of robot functions with minimal intervention from the human operator. Interaction between operator and function is infrequent, yet sometimes critical, causing small or moderate cognitive loadings.
3. **Indirect Manual Control:** the level of autonomy that requires human intervention for discrete yet frequent time intervals. The cognitive demand resulting from function control can vary from moderate to relatively high.
4. **Direct Manual Control:** the lowest level of autonomy that requires continuous and direct functional control. The cognitive demand under this level of autonomy is consistently high (and perhaps maximal).

Any of these four levels of autonomy may be invoked during functional execution of a task. Both robot functions and human operators have the capacity to request an autonomy adjustment. Adjustment can impact the cognitive load applied to the human operator: either amplifying or attenuating the load depending upon the direction of transition. To represent adjustable autonomy $\bar{\Gamma}$ must be to vary as the level of autonomy varies. By this definition, an autonomy alteration can be represented as

$$\bar{\Gamma} \xrightarrow{\text{autonomy adjustment}} \bar{\Gamma}' \text{ or } T(\bar{\Gamma}) \xrightarrow{\text{autonomy adjustment}} \bar{\Gamma}' \text{ where } T \text{ is a linear transform.}$$

To compute T , the four autonomy levels are assumed to be qualitatively consistent across the entire set of functions. This assumption is reasonable since the same joystick and monitor set-up used to control a *steering* function for the *navigation* subsystem will also control a *guidance* function for the *manipulation* subsystem. Thus, for a given function under a base autonomy level, the resulting cognitive load under an alternative autonomy level can be modeled as a scaled multiple of the base level or $\varphi'_i \propto \varphi \Rightarrow \varphi'_i = \delta_{ik} \varphi_i$ where each $i \in \{wm, ltm, \dots, a\}$ is a cognitive component at a default autonomy level and δ_{ik} is the constant of proportionality for the desired level of autonomy $k \in \{level1, \dots, level4\}$. A diagonal scaling matrix Δ_k is used to map functional cognition vectors at the default autonomy level to a new level k so that $\Delta_k \bar{\Gamma} = \bar{\Gamma}'$

Furthermore, by selecting Δ_k to be isomorphic, a transform from level k with Δ_k to level k' with $\Delta_{k'}$ can be expressed as $\Delta_k \cdot \Delta_k^{-1}$. Hence, the transform from any autonomy level to another is determined by multiplying the cognitive loading vector by $\Delta_k \cdot \Delta_k^{-1}$. The existence of an inverse matrix Δ_k^{-1} to perform level jumping is a key benefit for modeling autonomy adjustment in the form of a linear transformation. In addition, the use of linear transformations (based upon the properties of these transformations to act on the basis of a vector space [10]) also minimizes:

1. Prior knowledge of the system
2. The data representation of the system
3. The computational time required obtaining a new level of autonomy.

In fact, the construction of each Δ_k would require prior knowledge of cognitive loading values for only one function across all autonomy levels. The subsequent scaling components can then be acquired by taking ratios of components from these known cognitive loading vectors.

5 Workload Representation

Given $\bar{\Lambda}$ for each operator within the human population and the $\bar{\Gamma}$ for each robotic function within the set \mathbf{F} , the relationship binding humans to robot functions can be stated as: for each $h_i \in \mathbf{H} \rightarrow h_i \Leftrightarrow \bar{\Lambda}_i$ where $i = 1$ to M , a matrix of the form $[\bar{\Lambda}_1^T, \bar{\Lambda}_2^T, \dots, \bar{\Lambda}_M^T] = \mathbf{H}_C$ is created. This $6 \times M$ matrix represents the entire human cognitive capacity for a human-machine work system. In addition, the previous representation can be extended to functions, subsystems, and robots by stating for each $f_i \in \mathbf{F} \rightarrow f_i \Leftrightarrow \bar{\Gamma}_i$ where $i = 1..J$, a matrix of the form $[\bar{\Gamma}_1^T, \bar{\Gamma}_2^T, \dots, \bar{\Gamma}_J^T] = \mathbf{F}_C$ is created. This $6 \times J$ matrix represents the functional cognitive loading for the all system functions. Additionally, using the definition for robots, subsystems, and functions, the following is derived:

$$r = \{ \{ f_{11}, \dots, f_{1j}, \dots, f_{1Y} \}, \dots, \{ f_{i1}, \dots, f_{ij}, \dots, f_{iY} \}, \dots, \{ f_{X1}, \dots, f_{Xj}, \dots, f_{XY} \} \} \cdot$$

$$r = \{ \{ \bar{\Gamma}_{11}^T, \dots, \bar{\Gamma}_{1j}^T, \dots, \bar{\Gamma}_{1Y}^T \}, \dots, \{ \bar{\Gamma}_{i1}^T, \dots, \bar{\Gamma}_{ij}^T, \dots, \bar{\Gamma}_{iY}^T \}, \dots, \{ \bar{\Gamma}_{X1}^T, \dots, \bar{\Gamma}_{Xj}^T, \dots, \bar{\Gamma}_{XY}^T \} \}$$

Each robot r in the population of N robots is composed in this manner. No explicit association among functions, subsystems, and robots is enforced since axioms **(1)** and **(2)** state the existence of a direct relationship between humans and functions; however, system designers are free to enforce any required constraints. In this

manner, an operator constrained to control a particular robot can equivalently be constrained to control only those functions associated with that robot. This $6 \times (J \cdot K \cdot N)$ matrix (denoted \mathbf{R}_c) can be used to calculate the cognitive loading of a work system (denoted \mathbf{H}_L) with the equation $\mathbf{R}_c \mathbf{M} = \mathbf{H}_L$. \mathbf{M} is a $(J \cdot K \cdot N) \times M$ Boolean configuration matrix that has the effect of summing all $\vec{\Gamma}$ s in \mathbf{R}_c that are controlled by a particular operator.

5.1 Workload Optimization

The next objective of human-machine work systems is to maximize the number of functions under human supervision while concurrently minimizing the cognitive load induced on each operator. These constraints act in opposition to each other: maximization seeks to consume human cognition while minimization tends to withhold it. To resolve the conflicting constraints, an optimization procedure must create a reasonable balance based upon the current state of the system.

Minimization: Recall that every column in \mathbf{H}_C is a cognition capacity vector $\vec{\Lambda}$ and every column in \mathbf{H}_L is cognitive loading vector defined as

$$\vec{F}_j = \sum_{k=1}^{J \cdot K \cdot N} \vec{\Gamma}_{kj} m_{kj}$$

This definition implies that the j^{th} operator having a cognitive capacity of $\vec{\Lambda}_j$ must command a set of functions inducing a cognitive load of \vec{F}_j . Thus, the minimizing constraint requires $\lambda_{ij} \geq \phi_{ij}$ for each element λ_{ij} in $\vec{\Lambda}_j$ and ϕ_{ij} in \vec{F}_j for all $i \in \{wm, ltm, \dots, a\}$ ensuring that operators do not become cognitively overloaded.

Maximization: Let Z be number of all-zero rows in the matrix \mathbf{M} (i.e. functions assigned to no operator). Recall M is the number of columns in \mathbf{M} and $(J \cdot K \cdot N)$ is the number of rows in \mathbf{M} . The maximizing constraint desires that the number of managed functions be greater than the number operators in the system or $(J \cdot K \cdot N) - Z > M$ ensuring the maximum number of functions is controlled.

Ideally, all functions should be operator supervised; however, complete functional coverage for a relatively large system may not be achievable, especially during periods of heavy system activity. This complication creates a trade-off between the number of functions obtaining operator attention and the number of functions each operator can adequately manage. Therefore, a cost metric must be established in order to determine the optimal functional assignment given the cognitive capacity of each operator and the optimal level of autonomy governing each function.

5.2 Cost Metric

The cost metric encodes a number of systemic parameters that reflect the situational significance of a function. These parameters assist in determining an optimal configuration of humans, functions, and autonomy levels.

- **Functional Priority:** encodes the relative priority of a function. It provides precedence to critical functions when cognitive loading becomes excessive.
- **Initial Configuration:** is the initial configuration matrix \mathbf{M}_0 allowing the system to preserve the default grouping structure whenever possible.
- **Pairing Authorization:** encodes the permission granting or forbidding the existence of a pairing so that certain robot functions (i.e. those functions controlling weaponry or sensitive sensory equipment) are available to only authorized operators.
- **Request Wait Time:** represents the elapsed time of a function's or operator's request to adjust autonomy in an attempt to prevent starvation: the act denying functions from receiving operator attention.
- **Control Inertia:** records the time elapsed between a linked operator and function (provided the function is not fully autonomous) to prevent unnecessary context switching between operator-to-function assignments

By quantifying these parameters, the cost function can be tailored to suit the requirements of any work system and allow efficiency to be determined by comparing relative system cost. For example, the minimum system cost can be cast as the goal for searching the configuration space. The resulting configuration defined at the goal will be the system's \mathbf{M} .

5.3 The Workload Optimization Algorithm

The Transportation Algorithm is a well-documented problem of optimization that parallels the intentions of workload optimization. In summary, the Transportation Algorithm involves the determination of an optimal shipping network for groups of suppliers and consumers. Similarly, workload optimization involves the determination of an optimal configuration of operators (suppliers of cognitive capacity) and functions (consumers of cognitive capacity). Despite the similarities, the baseline Transportation Algorithm cannot be directly applied to workload optimization due to the following complications:

1. The Transportation Algorithm optimizes scalar quantities whereas scalable command and control relies upon vector quantities
2. The Transportation Algorithm produces a shipment matrix consisting of real-number values with multiple suppliers connected to a single consumer whereas scalable command and control produces a configuration matrix of Boolean values with a single operator per function

Complication (1). This situation requires modification to the vector components. As such, the cognitive capacity and loading vectors are collapsed into scalar components by selecting the smallest component from each $\bar{\Lambda}$ for operator representation and the largest component from each $\bar{\Gamma}$ for function representation. This reduction of dimensionality does affect the optimality of the configuration; however, the approximation drastically reduces the configuration space to improve the speed of computation.

Complication (2). This situation requires manipulation of continuous flow into discrete containers. To approximate an optimal configuration, the maximal flow component from each column of the configuration matrix is selected as the human-function match. Occasionally, selecting the maximal component can lead to cognitive overloading; however, if this case occurs, the work system varies the level of autonomy for any overloading functions and reprocesses the configuration.

Finally, these solutions are only reasonable when the cognitive loading vectors are small relative to cognitive capacity vectors. When these vectors are on the same order of magnitude, the accuracy of approximation algorithm will degrade resulting in a systemic tendency to unnecessarily increase levels of functional autonomy.

6 Simulation Results and Conclusions

To demonstrate the capabilities of scalable command and control in human-machine work systems, a simulation of the system described in sections 3, 4, and 5 was implemented. The cost function used during the simulation was of the form $C = f(\mathbf{M}_0, i, j) \cdot e^{g(\mathbf{p}, \mathbf{q}, \mathbf{Z}_{ij})} \cdot (1 + \alpha \mathbf{A}_{ij})$ where C is cost; \mathbf{M}_0 is the preferred configuration; \mathbf{p} is functional priority; \mathbf{q} is request wait time; \mathbf{Z} is task inertia; and \mathbf{A} is the authority. The function $f(*)$ returns a constant that reflects the grouping assignment while $g(*)$ returns a composition of the priority, request time, and control inertia. This composition is placed into an exponential operator to give numerical importance to the mentioned parameters. Finally, α is selected to be significantly large such that when operator-robot pairings are prohibited, C_{ij} becomes extremely costly thereby effectively prohibiting the potential pairing.

To obtain the results in Table 1, 10 trials were simulated at each human-robot configuration. Each simulation lasted 100 iterations; each robot had between one and two subsystems (randomly selected); and each subsystem was allowed to have between one and three functions (randomly selected). To serve as a comparison, a second system was created using a fixed-assignment configuration where operators were only permitted to control pre-assigned functions up to their cognitive capacity. Autonomy alterations were requested on a Gaussian

distribution (Figure 3). Finally, 10% noise was added to simulate random disturbance in workload estimation and communication errors. The results include the average number of functions (AF), the unassigned function per iteration (UF), the unassigned functions with high priority per iteration (UFHP), total autonomy switches (TAS), and total context switches (TCS).

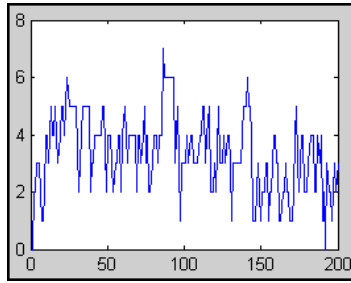


Fig. 3. Autonomy-transition intensity characteristic of our simulation runs.

Table 1. Simulation Results

H:R*	AF	Method	UF	UFHP	TAS	TCS
3:1	2.5	Optimized	0.00	0.00	54.00	0
		Static	0.00	0.00	54.00	NA
3:3	6.9	Optimized	0.15	0.05	145.10	10.8
		Static	0.34	0.11	145.10	NA
3:9	20.8	Optimized	9.90	0.95	240.20	4.8
		Static	12.46	2.90	240.20	NA
8:2	47.0	Optimized	23.66	1.06	232.00	54
		Static	27.06	4.72	232.00	NA

*H:R is the number of humans to robots used for a given run.

The results reflect the intentions of scalable command and control. Both the fixed and optimized procedures produced identical results when ample cognitive capacity was available to meet all cognitive loading conditions (resulting in no unnecessary assignment switches). When system complexity was increased, the system still performed quite well by consistently maintaining a lower number of unmanaged functions than static control. The last set of trials (those using 8 humans and 18 robots) demonstrates the true strength of the optimization procedure. The average number of unassigned functions per iteration is nearly half of what the fixed procedure generated. Furthermore, of those unassigned functions, an average of 1.06 of the top priority functions were left unassigned while the fixed method missed an average of 4.72. Clearly, if these top-priority functions were absolutely critical to the successful completion of a task, then the optimizing procedure would be the preferred implementation.

Scalable Command and Control demonstrates potential as a multi-human, multi-robot control mechanism. The simulations estimate that optimization consistently out-performs static control structures as system complexity increases. This fact is most apparent when the static control paradigm left 300% to 470% more top-

priority functions unassigned. The use of optimization also maintains a ratio of approximately 3.33 functions per operator for complex systems (i.e. the simulations with more than one function per operator) whereas static assignment produced only 2.7. This 21% difference, when accumulated across the operator population, results in a marked loss of efficiency. The time complexity of the approximation algorithm can be estimated as Kn^α where K is the number of autonomy levels and n^α represents polynomial time complexity of the standard Transportation Algorithm. By pruning the shipment matrix, the problem becomes analogous to the “bin packing” problem understood to be NP-hard [12]. The polynomial time approximation used herein is more amenable to the real-time requirements of unstructured operations, at the expense of optimality.

Reference

1. N. Jordan. (1963) Allocation of functions between men and machines in automated systems. *Journal of applied psychology*, Vol. 47, No. 3.
2. E. Hutchins. (1995). *Cognition in the wild*. Cambridge, MA
3. T. Fong, C. Thorpe, C. Baur. (2001) Collaboration, Dialogue, and Human-Robot Interaction. 10th International Symposium of Robotics Research, November 2001, Lorne, Victoria, Australia.
4. J. Ferber. (1999) *Multi-Agent Systems*. Addison-Wesley, London.
5. C. Lebiere, J. R. Anderson, D. Bothell. (2001) Multi-tasking and cognitive workload in an ACT-R model of a simplified air traffic control task. *Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation*. Norfolk, Va.
6. D. Kortenkamp, Keirn-Schreckenghost, R. Peter Bonasso. (1991). Adjustable Control Autonomy for Manned Space Flight Systems. *Proc. of IEEE*.
7. J.R. Anderson, C. Lebiere. (1998) *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.
8. C. Lebiere, J. R. Anderson, D. Bothell. (2001) Multi-tasking and cognitive workload in an ACT-R model of a simplified air traffic control task. In *Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation*. Norfolk, Va.
9. K. S. Barber, C. E. Martin. (1999) *Specification, Measurement, and Adjustment of Agent Autonomy: Theory and Implementation*. Autonomous Agents and Multi-Agent Systems.
10. R. Larson, (2000) *Elementary Linear Algebra*. 4th Ed. Houghton Mifflin Company. Boston, MA.
11. M H. Sohn, S. Ursu, J. R. Anderson, V. A. Stenger, C. S. Carter. (2000) The role of prefrontal cortex and posterior parietal cortex in task switching. *Proceedings of the National Academy of Sciences*, vol. 97, no. 24, pp. 13448-13453.
12. G. Strang. (1986) *Introduction to Applied Mathematics*. Wellesley-Cambridge Press. Wellesley, MA.