Computer War

Boris V. Raushenbakh

Professor of Theoretical Mechanics and Control, Moscow Physical-Technical Institute; Member, Committee of Soviet Scientists for Peace against the Nuclear Threat. Dr. Raushenbakh is a member of the Academy of Sciences in the USSR and the International Academy of Astronautics. His work has been awarded the Lenin Prize.

The need for a new way of thinking in our nuclear age has been lately affirmed by many. Man has become all too powerful; so great is his might that he can annihilate all life on the Earth, his own kind included, a situation that was deemed unthinkable early in this century. Under these new conditions, man cannot afford to think and act solely in terms of the welfare of his own kin, his own community, or his own country. Nowadays man must also heed the global consequences of his actions.

The global nature of human activities is discernable in various ways and makes itself felt in the exhaustion of our resources, in ecological problems and, unquestionably, in the arms race.

When humanity meets the challenge of ecological problems, “to err is human” may be an acceptable rationale. The ecological processes are sufficiently slow that they can be observed and studied with a certain measure of detachment so that necessary adjustments can be made to change the human course. If these adjustments prove ineffective, new measures can be taken and the problem would ultimately be solved by trial and error.

In the event of a nuclear war, errors would produce entirely different consequences. There would be no time to correct mistakes. The first mistake is likely to become the last. The time of a ballistic missile flight is measured in minutes and hence the duration of a nuclear conflict is very short.

Computers in War

Due to the short duration of battle operations and the extreme complexity of military equipment and its control, computers have become an indispensable element of weapons systems. As a rule they are man's helpers and are capable of helping to control sophisticated weaponry. However, with more complex equipment and shorter duration battles, humans could be forced out of the decision loop and crucial decisions left to computers.

> *"In the event of a nuclear war … the first mistake is likely to become the last."*

The possibility of triggering this kind of "computer war" is a reality if launch on warning strategies are adopted or if current plans to militarize space are carried out. Total computerization of any battle system is fraught with grave danger. This paper will use space-based weapons for illustrative purposes. This example helps concretize the analysis and is extremely relevant to current defense planning. Some space-based weapons would require complete computerization due to their virtually instantaneous propagation of destructive energy, literally a fraction of a second.

The conventional three-component formula of military action control consisting of a warning as to the emerging situation, waiting for a decision to be made by the authority, and execution of the order issued, is rendered invalid by this kind of "computer war." No person can appraise the situation and make a correct decision in a matter of seconds, nor is it feasible to wait until the decision is made by the political or military authority. The decision, therefore, will be made by suitably programmed computers.

People thus become hostages of computers. In terms of potential nuclear war, the very existence of mankind is becoming dependent on hardware and software. In a situation like this, the discussion of the possibility of an accidental triggering of an attack by one of the sides, an attack by mistake or chance coincidence, ceases to be academic.

The Effect of Secrecy

The issue of accidental and unprovoked triggering of a nuclear conflict is now being increasingly perceived as one of the most grave dangers threatening humanity. Such a course of events may be set off by various causes. Here we shall confine ourselves to the discussion of those causes that involve computers.

Two such causes that are normally mentioned meet the eye. The first cause consists in a hardware failure. Malfunctioning of some element of the system may not only cripple its effectiveness but also result in triggering an unprovoked attack. The second cause lies in errors that may creep into software.

"The decision … will be made by suitably programmed computers. People thus become hostages of computers."

There also exists a third source of danger that has so far been largely disregarded. It involves neither malfunctioning hardware, nor errors in software and is, therefore, unremovable. This cause is associated with a lack of concordance in the software of two counteropposed systems. Computers in these systems will be fed data representing a model of the potential enemy's system attributes, rather than factual data on these attributes. Because each side maintains secrecy concerning its design, aspects of the model may be imprecise and, occasionally, downright false. It is this substitution of an unavoidably imprecise model for the actual properties of the potential enemy's system that we call a lack of concordance in the software of the two systems.

To simplify the ensuing examination of this problem, let us make the improbable assumption that the software of the two counteropposed systems is error-free and that the hardware is fail-safe. The only errors that will be allowed for are the errors in planning, i.e. mistakes stemming from insufficient information on the opposing system.

Stability

Space-based multifunctional systems will make up a certain strike capability complex. To retain effectiveness against surprise attack, a fair share of resources will be spent on prompt detection of ballistic missile launchings as well as on detection of preparations for launching, preparations for activating space-based weaponry, and other support operations.

Let us now proceed from two all but obvious assumptions: that detection of operations immediately preceding the use of space-based weaponry is feasible, and that both sides refrain from plunging into a nuclear conflict on early detection of signs that may be interpreted as preparations to attack.

If system A has detected the preliminary operations of system B, it must proceed with similar preparations, but refrain from immediate attack since the actions of system B may have been misinterpreted. System A will be provoked to attack only after detection of a sufficiently large number of danger signs. Even then, there may be an alternative to a nuclear attack among the system's capabilities. Thus there is a certain gradualness of countermeasures aimed at ruling out the possibility of triggering a nuclear conflict by accident.

To make these rather general observations more graphic, the following pattern of system A's response to system B's behavior may be suggested. Suppose system A's designers constructed a sequence of actions based on the assumption that simultaneous appearance of six danger signals is critical. Then the actions of system A may be represented in the following manner:

Observation	*Response*
The first emergence of one sign	Enhance observation such as activation of supplementary tracking systems
Simultaneous emergence of two signs	Relatively time-consuming support operations to put the system into the ready-for-action mode
Simultaneous emergence of three signs	Intermediate readiness
Simultaneous emergence of four signs	Full readiness
Simultaneous emergence of five signs	Nonnuclear military action (e.g. destruction of some satellites in system B)
Simultaneous emergence of six signs	Nuclear war

The above scheme is but an illustration. The critical number of danger signals may be different depending on the nature of the signs. Also, countermeasure patterns are far from being this elementary. Yet whatever the actual programs are, they will always proceed from the need to gradually step up the response so as to make it adequate to the potential threat. It is equally obvious that disappearance of danger signals (reduction in their number) will entail the corresponding annulment of countermeasures. Given that the system B behavior was simply misinterpreted, the disappearance of danger signals or their modified interpretation (e.g. they might be generated by some rare natural phenomena) will bring system A back into the initial state.

The described sequence of actions, their gradualness, and reversibility make system A stable. That is, slight perturbations (a small number of danger signals) cause the system to act "proportionately," adding or subtracting countermeasures according to the above plan. This mechanism appears to have a safety valve to prevent explosive development of the process culminating in a nuclear conflict on marginal grounds. This "proportionality" seems to rule out accidental triggering of a nuclear conflict.

System B is most likely to be designed along the same lines and will also be stable and have the same built-in “proportionality” discussed above. The stability of systems A and B taken separately, however, does not imply the stability of the large system A+B.

Instability

Examination of the large system A+B, i.e. of the interacting systems A and B, shows that conventional techniques used in designing, modeling, processing, and testing of large systems to ensure stability of their concurrent operation cannot be fully implemented. Since systems A and B belong to adversaries, the design and debugging of each system will go on independently and under tight security cover. Their “marriage” will take place only when they are deployed and put on round-the-clock duty. It is in the first conflict situation that they will start functioning together, and a military action may be their first test.

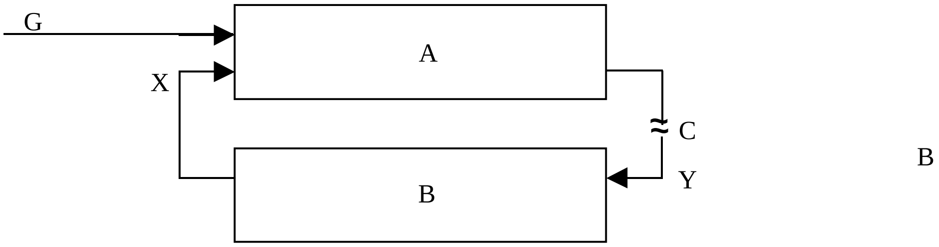
Modern control theory holds that integration of two stable systems into one large system frequently entails instability of the latter. (1) The point is illustrated in Figure 1.

Rectangles A and B designate the corresponding systems, while lines X and Y which end in arrows stand for information flows which systems A and B exchange after they have been integrated into the large system A+B. These “information flows” represent danger signs detected by one system in the other.

Let us examine the problem by turning to the “open loop” system, obtained by assuming that at point C the communication between A and B is broken and the transfer of the information Y into system B does not take place. That is system B is not capable of detecting the processes going on in system A.

Suppose also that under peacetime operation of system B, some processes occur in this system that are registered by system A. (The figure shows it as the information X input.) Let us assume that these processes are not associated with preparations for attack, but stem from some minor

Figure 1. Computer War



malfunction, testing, or other similar procedures. Let us now assume that system A perceives these operations as signals indicating preparation of system B to attack. (Since system A does not have complete information on system B, mutual suspicion is not only understandable, but also warranted.) Having received the information X, system A will switch to the operational mode appropriate to that information. If after some time interval the malfunctioning in system B is corrected or testing is completed, information X ceases to carry the danger signs and system A returns to the observational mode. Here the occasional emergence of danger signs produces no tragic consequences.

"… integration of two stable systems into one large system frequently entails instability ..."

The above process indicating the "stability" of system A proved possible because information Y on the action of system A was not transferred to system B. Let us now consider the behavior of the two systems when the feedback is closed at point C and they turn into a single system A+B.

Suppose both systems are "stable" in the described sense, and function perfectly. System A receives signal G which is not relevant to the operation of system B but stems from an unusual phenomenon, some chance occurrence in space, or anything that may be interpreted as a danger signal by system A. Let us assume that the signal is not too alarming and that system A will only go through the initial stages of response. As soon as information Y on these measures is received by system B it will reciprocate and the system A input will deal with two simultaneous signals G and X, the latter being due to system B's actions.

The appearance of two danger signals instead of one will cause system A to take another step toward attack, which will immediately change the information Y input into system B, causing further alarm. The appropriate response in system B will change the content of information X, which will become progressively more alarming, producing further action by system A. Eventually this will trigger off an explosive process of measures and countermeasures, leading to a nuclear conflict.

The above example is instructive in that it indicates the possibility of an entirely "unprovoked" nuclear attack triggered by the interaction of two perfectly operating computer-based systems, each of which taken separately is "stable."

Control theory readily explains the instability of the system A+B as generated by positive feedback. (System A responds to system B's actions in such a way that the latter is further activated.) (1)

All-or-Nothing Control

It may appear that designing either system to disregard minor danger signs would solve the problem and remove the instability of system A+B. The limiting case would be to block the response of one system to perceived preparations of the other, but have a hair-trigger that would execute a nuclear attack as soon as an attack was launched by the other side.

The problem arising in the development of such a system with an all-or-nothing response (termed "bang-bang control") is pinpointing the threshold value of the danger signs which would imply imminence of an attack and would give sufficient grounds for a preemptive response. Analysis shows that there is more to this issue than meets the eye.

"Where man might stop, the computer goes on, for computers know no moral code."

Suppose system A's designers know that a certain attack scenario may be reliably identified on the emergence of five known danger signs. Considering the lack of complete concordance of their software with the factual properties of their adversary's system, it is touch-and-go when three or four signs out of five are registered. Should they attack or shouldn't they? They should attack if the lack of some signs is a "feint" or a result of differences between the factual properties of system B and the model of these properties stored in system A's computer. The attack should by no means be launched if this lack of signs signifies any occurrence other than the attack, or system A's response would trigger a war by mistake. To compound the problem, this decision should be built into the software long (probably years) before the decision must be made by system A's computer. This uncertainty can bring about a fatal lack of the retaliative response or an even more fatal overreaction, an accidental nuclear war.

The task that software designers on both sides have to face is compounded by the possibility of false target launchings. The side that launches false targets tries to overstate the corresponding signs rather than to disguise them, in an attempt to undermine the deterrent potential of the other side. Software designers will thus have to seek additional signs which would help differentiate between false and real targets. Uncertainty in the interpretation of the incoming information will be greatly magnified, further "destabilizing" the software, with a consequent increase in the probability of an inadequate or dangerous response, including a nuclear attack.

Software is likely to include parts based on the proportionate response concept and parts of the bang-bang design or other modes of response known in control theory. Analysis shows that this in no way invalidates the conclusions. When the mutually uncoordinated systems A and B are integrated into combined system A+B, they will obey the laws that are pertinent to system A+B and are unknown to designers of respective software.

The above example shows that the cardinal properties of a large system and of its components may be qualitatively different. (Systems A and B are separately stable, while system A+B is unstable.) It follows that neither party can guarantee a “reasonable” behavior of system A+B. It should be emphasized that political and military authorities will have no time to interfere with the instantaneous hostilities triggered by mistake.

Conclusion

Humanity thus entrusts its fate to computer systems that, even if they function perfectly (no malfunctioning occurs, there are no errors in software design and execution), follow logic known to no man. Given some entirely unknown circumstances, this logic can lead to war and hence to death of humankind. Where man might stop, the computer goes on, for computers know no moral code.

To avert such a course of events, a new way of thinking is required today. The viewpoint of the separate systems A or B must give way to the viewpoint of the large system A+B, that is the viewpoint of the entire human family. And from this vantage point, any military strategy which would force the use of computers to override human reaction time is seen as an irresponsibly dangerous act.

References

1. Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini, *Feedback Control of Dynamic Systems* (Reading, Massachusetts: Addison-Wesley, 1986).