HYBRID PSEUDO-RANDOM (HYPR) DYNAMIC SCENARIO GENERATION CAPABILITY

Ken Saari
Dr. Dennis Bushmitch, Eugene Ivanyutin
The MITRE Corporation, Eatontown, NJ

ABSTRACT

This paper discusses a new capability, the Hybrid Pseudo-Random (HyPR) Dynamic Force Generator. This capability (implemented in MATLAB) takes as input the geographic positions of a relatively small key subset of actual ground forces. It quickly generates realistic, dynamic positions for all remaining specific platforms in a military force structure. This capability was initially designed to generate scenario-linked laydowns for an entire ~3000 platform U.S. Army Future Combat System (FCS) Brigade Combat Team (BCT) with varying configurations. Each laydown was required to be correlated to a Horseblanket/Unit Requirements Sheet (URS) and established Operational and Organizational guidelines (O&Os) for FCS and its spinouts (e.g. Infantry BCT (IBCT)). Results were initially applied to a wireless connectivity analysis tool and an OPNET-based Mobile Ad-Hoc Network (MANET) system performance model of Joint Tactical Radio System (JTRS) and other radio systems/protocols. HyPR provided all platform locations on-the-move during specific battle scenario segments. HyPR provides a speedy and robust new capability to develop high fidelity scenarios, including movements with greatly scaled up forces.

I. BACKGROUND

Traditionally, it has been a difficult problem to develop operationally relevant, large-scale (e.g., Brigade size) scheme and maneuver force laydowns with future technologies under Operational and Organizational (O&O) constraints, especially in a timely manner.

The Hybrid Pseudo-Random (HyPR) capability was developed due to the lack of availability of such scenarios with complete sets of platform positions for in-depth analyses. Thus, a methodology was needed to quickly and realistically generate missing platform positions. Given that the forces employ clusters of soldier and vehicle platforms grouped together with their command vehicle platform, pseudo-random position generation via Gaussian distributions can be used to approximate such clusters. The force structures employ a hierarchy of platform clusters, and in a Brigade Combat Team (BCT), the Brigade Commander and directly subordinate platforms (including Battalion Commanders) constitute the highest level cluster. The next echelon level down the hierarchy begins at the Battalion Commanders, and includes their subordinate Company Commanders, who each have their own subordinate clusters. In this manner, platform clusters are assembled in the hierarchy down to the Rifleman. Using this methodology, small sets of key battle scenario platform locations may now be speedily scaled up to Brigade-sized (or larger) dynamic scenarios.

II. INTRODUCTION

The approach to achieve the objective was the creation of a set of two modules in MATLAB shown in Figure 1: the HyPR Force Generator and Motion Generator. The Force Generator inputs a subset of platform positions from a military force structure (e.g., BCT) and completes the force with pseudo-randomly generated positions for the remaining platforms. Thus, via the input of the actual positions and parameterization for the generation of the remaining platforms, both the resulting laydown scale and fidelity is completely user configurable.

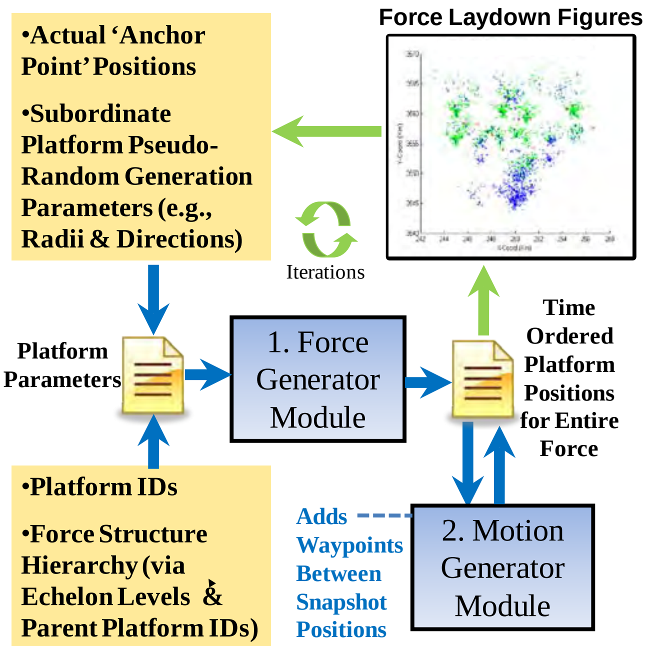

Figure 1 –Methodology and Data Flow

As shown in Figure 1, the Force Generator first inputs a Platform Parameters file, which defines each platform and includes hierarchical force structure definitions. As illustrated in the figure via the "Iterations" loop, actual scenario-linked positions (termed ‘anchor points’ in this paper) and associated parameters can be added to the Platform Parameters file incrementally during force generation. Anchor points are key platform positions across the upper echelons in the force structure hierarchy that typically can be obtained from battle scenarios. The developing force structure can be visualized via force laydown displays as the anchor point data is input until realistic snapshots are achieved. Second, the Motion Generator inputs the static snapshot positions created by the Force Generator and adds movement in the form of waypoint positions over time for each platform in the force.

The HyPR capability was created as baseline software for the U.S. Army Future Combat System (FCS) Brigade Combat Team (BCT) Network Systems Integration (NSI) Network Analysis and Integration Laboratory (NAIL) at Fort Monmouth, NJ. It was initially used to generate an Infantry BCT (IBCT) scenario. HyPR has also been shared both within and outside of the NAIL for other purposes (e.g., it has been shared within MITRE and with the Army Communications-Electronics Research, Development, and Engineering Center (CERDEC), and it has been delivered to US Army Training and Doctrine Command (TRADOC) at Fort Leavenworth).

This paper is organized as follows. The Force Laydown Generator, which generates initial static force laydown scenario snapshots, is described in Sections III and IV. The Motion Generator, which adds movement to the scenario snapshots, is described in Sections V-VII. Section VIII summarizes the implementation and presents some initial dynamic scenario results. Section IX presents conclusions.

III. HYPR FORCE LAYDOWN GENERATOR

Prior to motion generation, the HyPR Force Generator Module combines actual positions with pseudo-randomly generated platform positions to generate laydowns for static scenario snapshots. A platform is defined as a vehicle or a soldier, and a vehicle can be a ground vehicle or an Unmanned Aerial Vehicle (UAV). The actual platform positions serve as ‘anchor points’ for the pseudo-randomly generated platform positions. The anchor point positions can be tied to specific battle phase snapshots from official scenarios. For example, while only a small subset of key anchor points were available in the scenario snapshots used for the IBCT analysis recently completed by the FCS NAIL, HyPR ensured that all platforms from the associated U.S. Army Horseblanket were present in the resulting laydowns.

The pseudo-randomly generated positions are generated under strict constraints in order to achieve representative force laydowns. These constraints are related to the Organizational and Operational (O&O) and other guidelines that are applicable to a particular military force structure, which are explained in Section IV.

The methodology illustrated in Figure 1 to generate a scenario laydown using HyPR is as follows. A Platform Parameters input file is created containing a record for each platform, in which the force structure hierarchy is defined numerically via echelon levels and parent platform associations. Among other parameters, each platform is assigned an Echelon Level, Platform ID and Parent Platform ID. The highest level platform (e.g., Brigade Commander) is assigned Echelon Level 1 and no parent platform. The levels increase down the hierarchy, as child platforms, which can be a mix of anchor point and non-actual positions, are assigned to the parent platforms. Child platforms are directly subordinate to their parents in the force structure (e.g. a Company Commander will have a Brigade Commander as a parent platform). Next, actual scenario-linked ground platform positions (i.e., the anchor points) and associated parameters are added to the file. These parameters control the pseudo-random generation of the non-anchor point child positions down the echelon hierarchy.

As illustrated in Figure 1 with the “Iterations” loop, the addition of anchor points and associated parameters can be performed incrementally until the desired force laydown realism is achieved. The initial result is a static laydown snapshot in which all remaining non-anchor point ground platforms are deployed around the anchor point positions, typically within decreasing deployment radii from parent platforms down the force structure hierarchy. Parameters at each platform for subordinate (child platform) position generation via Gaussian distributions include maximum radius and direction. Given that Brigade and above asset UAV (e.g., Class IV UAV) positions have less dependency on the anchor point positions; they are typically specified independently of the Platform Parameters file via an additional UAV Position input file.

The process of generating laydown snapshots described above can be repeated until a set of time-ordered scenario snapshots is achieved that span a desired scenario window length (e.g., one hour vignette). Next, the Motion Generator can be used to generate waypoint positions between successive snapshots, thus creating continuous force movement within the scenario time window.

IV. STATIC LAYDOWN RESULTS

Figure 2 illustrates generic guidelines for an IBCT obtained via O&O guidelines and Subject Matter Expert (SME) input. The platforms in the initial generic IBCT laydown snapshot generated with HyPR (shown in Figure 3) were deployed approximately within the areas illustrated in Figure 2. The overall footprint was derived from O&O guidelines and the sizes and relative positions of the areas in the figure were primarily derived from SME input. Anchor points (shown in Figure 4) and parameters were added to the Platform Parameter file within iterations until the laydown conformed to the guidelines. When using an actual scenario, the relative locations of the Reconnaissance Squadron and Infantry Battalion platforms can differ and change over time, as dictated by the anchor points.

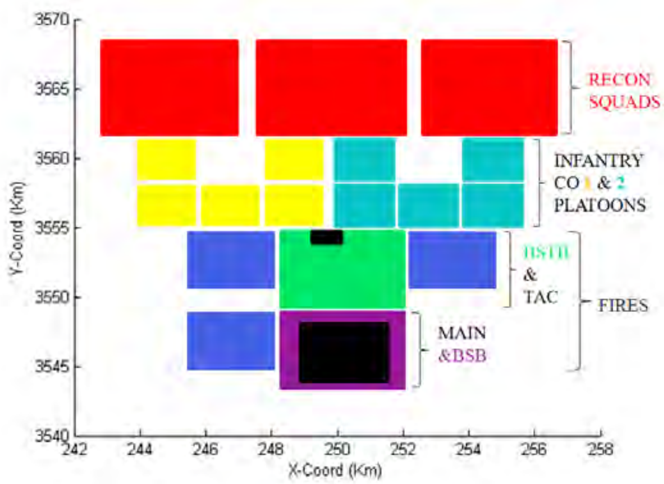

Figure 2 –Generic IBCT Guidelines

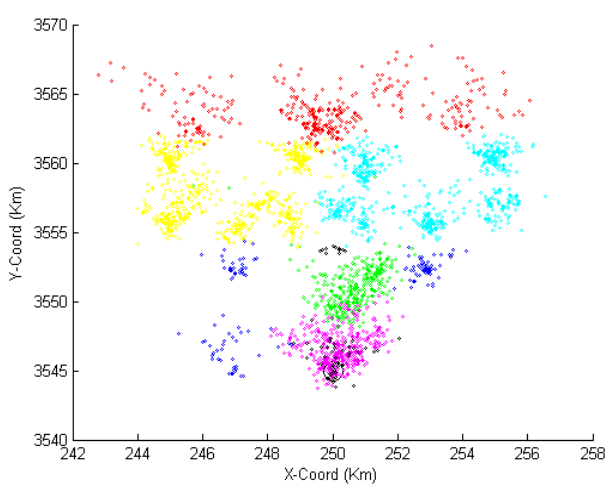

Figure 3 –Laydown Color-Coded by Battalion

HyPR can generate up to six laydown snapshot pop-up figures, each of which display the platform positions and highlight different aspects of the force structure:

1. Color-coded by Battalion (as in Figure 3).
2. Color-coded by Echelon Level.
3. Soldier versus ground vehicles plus UAVs.
4. Vehicles and URS-associated soldiers.
5. Highlighted Anchor Points (as in Figure 4).
6. Color-coded by highest ranked radio type.

Figure 4 illustrates the anchor point positions used to generate the snapshot shown in Figure 3. Anchor point platforms are color coded by echelon level.

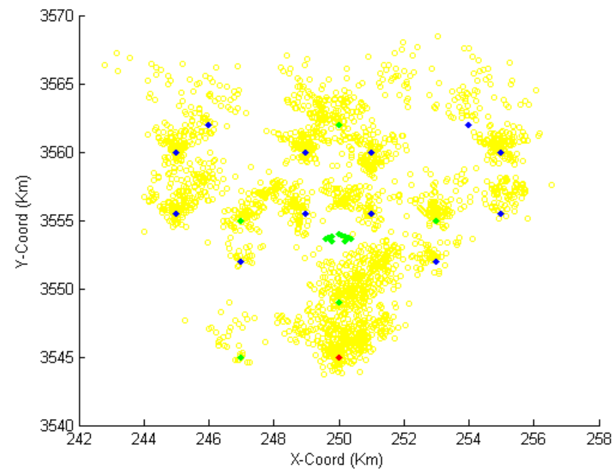

Figure 4 –Laydown Highlighting Anchor Points

Automated O&O checks can be employed to ensure that guidelines are not violated for any given force, such as maximum distances to anchor point child platforms at each echelon level. Additionally, Figure 5 illustrates two types of calculations performed for two sets of ground platforms: platforms excluding Reconnaissance platforms (which have the greatest modular mobility) and the full set of ground platforms. Both the maximum distances (D1 and D2) from outlying platforms to the headquarters and the overall areas encompassing the platforms (the black and red dashed boxes) are checked.

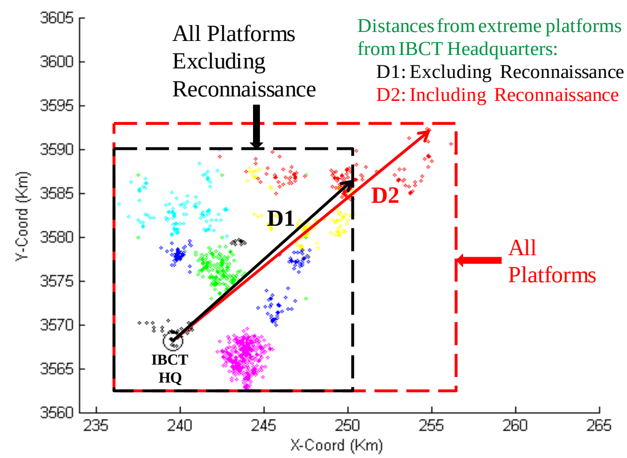

Figure 5 –Automated O&O Checking

V. HYPR MOTION GENERATOR

The objective for the HyPR Motion Generator is to add time ordered waypoints to sets of Platform Position files created by the HyPR Force Generator. It inputs successive pairs of scenario snapshots (captured in the Platform Position files), and adds waypoints between them to generate platform motion for M&S. Dynamic Anchor Point positions tied to a scenario can be obtained from various Army Force deployment/maneuver scenario sources or by other means. The result is sets of continuous positions over time for all platforms between successive static snapshot positions. The number of waypoints between snapshots is user-configurable, and is correlated to the desired time increment (i.e., increasing the time increment between the sets of waypoints decreases the number of sets of waypoints between any given pair of snapshots). Configurable degrees of randomness can be added to waypoints to obviate unrealistic lock-step movements, in which spatial relationships between platforms remain unchanged.

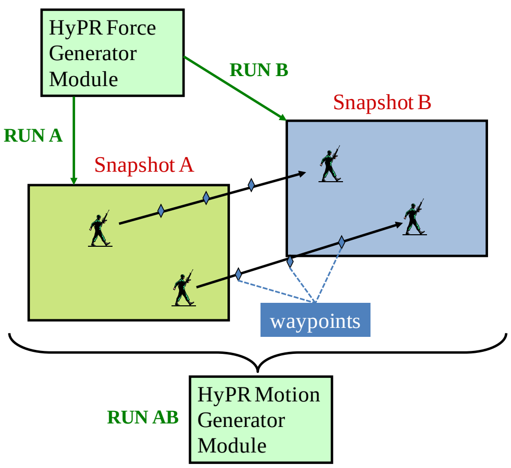

Figure 6 –Creating Waypoints from Snapshots

Following snapshot generation, the Motion Generator is run for each successive snapshot pair to generate the waypoints, which are illustrated in simplified form in Figure 6 as diamonds along the paths of soldiers between their Snapshot A and B positions. As shown, one Motion Generator run (designated AB) follows two Force Generator runs (designated A and B). Thus, continuous motion in the form of waypoint positions guided by successive snapshot anchor point positions is facilitated. Potential obstacles such as mountains and bodies of water can be avoided by careful and adequate snapshot specifications. Greater realism can be achieved by increasing the number of snapshots guiding motion within any given time period. For example, if an obstacle exists between a set of two snapshots that is to be avoided, additional intermediate snapshot(s) can be created to guide the motion of the force around the obstacle. It is also possible to guide anchor point platforms along roadways via successive positions along the path in multiple snapshots.

Thus, every leg of the motion (i.e., the set of waypoints between a pair of snapshots) is defined by a 'Snapshot A' (starting positions) and a 'Snapshot B' (ending positions). Following the generation of all desired legs of motion, a continuous set of waypoint positions defined for the entire force over the desired scenario window timeframe exists. Subsequently, these positions can be used for various applications (e.g., radio network connectivity analysis and discreet event simulations).

VI. MOTION GENERATION ALGORITHM

Refer to Figure 7 for an illustration of successive motion segments. The first leg/segment of platform motion uses all of the first set of 'Snapshot A' positions (both actual anchor points and non-actual/pseudo-randomly-generated) as starting positions for all platforms. Ending positions of motion legs next become starting positions of subsequent legs to ensure continuity through multiple legs of motion. Thus, for all legs following the initial leg, the anchor point positions taken from the snapshots guide the motion, while the non-actual positions are functions of the motion. In effect, the snapshots are modified in that they retain the identical anchor point positions, but may reposition the remaining platforms.

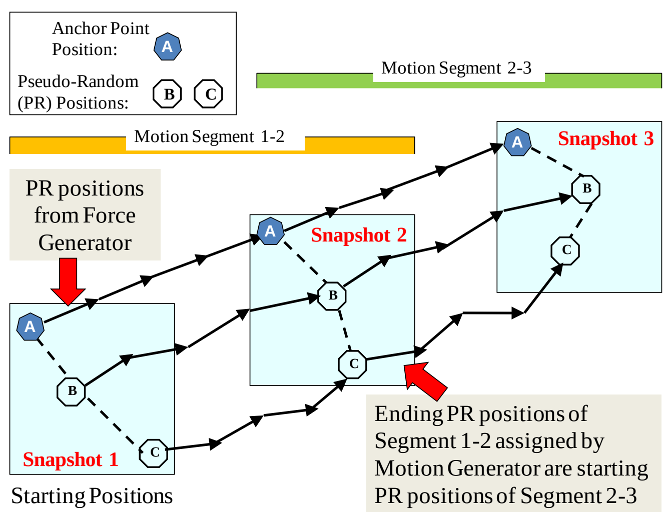

Figure 7 –Successive Motion Segments

Figure 8 illustrates the manner in which random offsets are optionally generated at waypoints for non-actual platforms. Anchor point positions are assigned waypoints in a straight line between their positions in the pair of snapshots guiding the motion. A is an anchor point position and B is a subordinate non-actual platform position, both of which are moving from Snapshot 1 to Snapshot 2 locations. A single set of waypoint positions are illustrated as A' and B'. Initially, the waypoint position resulting from move M for a non-actual platform B is calculated, which retains the identical spatial relationship (Delta X and Y) with respect to anchor point A platform's position at the previous set of waypoints. Then, a random offset R is added to the position within a set of bounds, resulting in a Delta X' and Y'. This process proceeds down the force structure hierarchy, and there may be several levels of non-actual platforms subordinate to an actual position. Thus, random deltas can become cumulative for platforms at the lower end of the hierarchy.

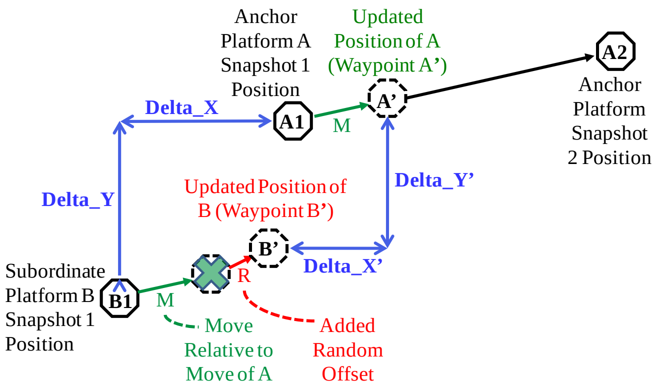

Figure 8 – Random Offsets Added to Waypoints

The radius to the parent platform, which bounds random offset generation, is a parameter associated with both the platform's echelon level and the relative velocity of the associated actual position. The radii are scaled such that the nominal value specified at an echelon level is applied only to platforms traveling at the greatest velocity in a particular leg of motion. The remaining platforms utilize radii scaled down to the ratio of their own velocities with respect to the maximum velocity. Thus, slower moving platforms have proportionately smaller random movement components. The end result is smooth and consistent motion throughout the force structure.

A description of the algorithm that updates subordinate platforms at each set of waypoints at a particular time is as follows. Starting at the highest echelon level platform, all child and descendent platforms are traversed down the force structure hierarchy. For each anchor point platform encountered, the waypoint positions are obtained deterministically, resulting in equal spacing within a straight line between the pair of actual positions in the two snapshots encompassing the motion. However, when non-actual platforms are encountered, waypoint positions are calculated similarly, but random offset values can be added to them, thus preventing platform motion in straight lines. The random offset for a particular platform is generated within a finite loop until if and when 'maximum delta' constraints are achieved. If these constraints cannot be achieved within maximum iterations, execution is halted and the user is warned to correct the associated parameters.

Constraints on this random offset generation were created to control the nature of the movement. Specified at each echelon level, and illustrated in Figure 9, they include the following:

- Nominal Radii of Random Offset Areas for subordinate (child) platforms.
- Maximum Deltas for relative positions of a platform with respect to its anchor point platform's original position and parent platform's current position.

Working together, the constraints imposed on random offset generation will control how structured or unstructured a formation will remain at successive sets of waypoints. They will also ensure that the motion at successive sets of waypoints is smooth and free from wild jumps in the relative distances between platforms in the force. Constraining relative distances of platforms to their parent platforms within ranges of their immediately prior values will minimize such jumps. Constraining distances to anchor point platforms within ranges of their original relative distances will keep formations intact.

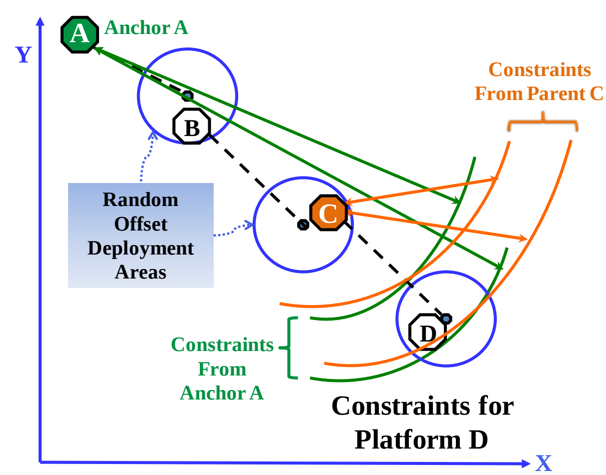

Figure 9 –Constraints on Random Offsets

Refer to Figure 9 for an illustration of an example of the constraints imposed upon random offset generation for a platform (labeled 'D'). In this example, Platform A is an anchor point platform that has just moved to the next waypoint in its trajectory, and Platforms B, C and D are non-actual platforms at successive levels down the echelon hierarchy from Platform A. The non-actual platforms will move relative to Platform A with the added random offsets. The circles represent the random offset deployment areas (constrained by radii from the immediately previous locations relative to the parent platforms). The arcs show the min/max radii constraints for Platform D; the green arcs apply to anchor Platform A and the orange arcs apply to the parent Platform C.

VII. MOUNTED & DISMOUNTED MODES

HyPR can be run in either Mounted or Dismounted mode. In Mounted mode, the soldier platforms are not explicitly deployed in the laydown. In this mode, it is assumed that they are either riding in the ground vehicle platforms, or will arrive at a later time via non-organic means (such as air lifting). Mounted mode is typically appropriate for early stages of the battle, when the force is in transit to the objective area. When the scenario later calls for Dismounted Soldiers on the ground, the tool is then run in Dismounted mode, which explicitly deploys a fraction or all of the soldier platforms.

The modeling of dismounted soldiers and vehicles in motion at the same time presents challenges to the HyPR methodology and requires assumptions. For example, it is an obvious fact that vehicles can achieve velocities well in excess of dismounted human velocities. Given that vehicles are moving at such velocities, it may or may not be assumed that the dismounted soldiers will follow the vehicles to the same general destination area. If this assumption can be made, then the HyPR Motion Generator can be run with a special option in Dismounted mode, which employs a "Lagging Soldier" algorithm. Figure 10 illustrates the optional Lagging Soldier algorithm behavior, in which A is an anchor point position, V is a non-actual vehicle position and the soldier icon is a non-actual soldier position.

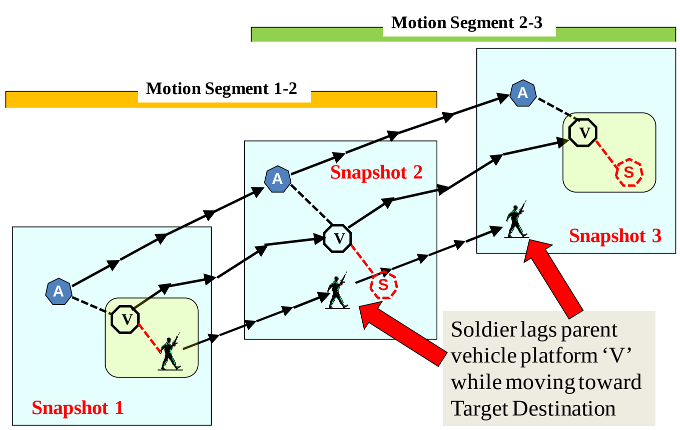

Figure 10 –Lagging Soldier Motion Segments

The algorithm allows for Dismounted Soldier platforms to lag their parent vehicle platforms when the vehicles are moving at faster-than-human velocities. Soldiers will move at human velocities toward their parent vehicles positions in the next snapshot, while attempting to end up with the same relative position to the vehicle that was established in the first snapshot. If and when in a subsequent segment of motion between snapshots the vehicle slows down sufficiently, stops or reverses direction such that the soldier catches up, the soldier will remain with the vehicle as before as long as possible. Currently, when this option has not been invoked in Dismounted Mode, warnings will be generated if faster-than-human velocities are achieved by the soldiers on the ground. However, this algorithm may undergo further refinement in the future, as dictated by scenario requirements.

VIII. IMPLEMENATION AND INITIAL DYNAMIC RESULTS

HyPR was implemented entirely in MATLAB. Various sets of default parameters can be set via a pair of master M-files, which invoke unique General User Interfaces (GUIs) for the HyPR Force Generator and Motion Generator; they are displayed in Figures 11 and 12, respectively. The GUIs control operation of the software and display critical parameters, options and input/output files that define each run, while facilitating user-input changes to them. The GUIs also control and display the force and motion generation results graphically via embedded figures and multiple pop-up figure options. Given the significant variances in configurations that each software module can employ between executions, the GUIs provide a concise visual means to confirm that the critical values are set properly.

For example, the Force Generator GUI permits the read back of previously generated ground platform positions, while facilitating the redeployment of UAVs and/or percent of Dismounted Soldiers. Additionally, the Motion Generator GUI has input parameters that control and bound the random variations that can be introduced for the non-actual waypoints are specified for each hierarchical echelon level. After the Motion Generator tool has been run for all desired legs of the motion, a movie button exists to generate a motion animation.

A set of snapshots were created and employed in the generation of a demonstration animation to illustrate HyPR's ability to put a BCT-sized laydown in motion. Two such snapshots (minus terrain map backgrounds for clarity) are shown in Figures 13 and 14.

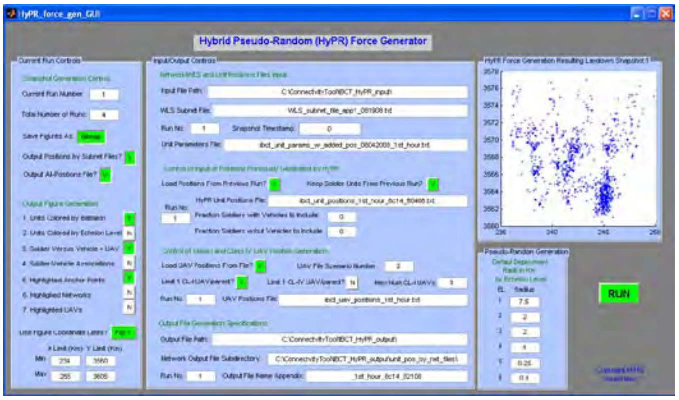

Figure 11 –HyPR Force Generator GUI

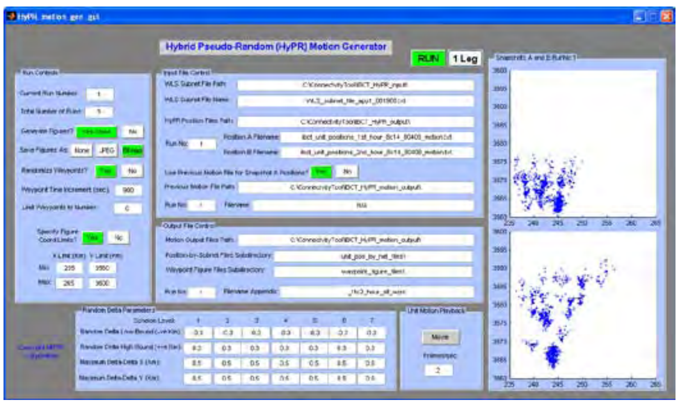

Figure 12 –HyPR Motion Generator GUI

The ground platforms employed are color-coded at the Battalion level and the black circles represent UAVs. The snapshots employed resulted in multiple successive legs of motion waypoints. The first snapshot was previously shown in Figure 3. An intermediate snapshot is shown in Figure 13 and the final snapshot is shown in Figure 14. The demonstration illustrates continuous motion between the snapshots, in which the IBCT platforms are initially grouped close together, and then spread out toward an objective area (which is located near the upper right quadrant of the figures). The red Reconnaissance Squadron platforms initially advance toward the objective (Figure 13), and subsequently retreat as the (yellow and light blue) Infantry Battalion platforms advance (Figure 14).

The first significant analysis task utilizing HyPR was successfully completed in March 2009, when final analysis results were presented and delivered to the customer. It was utilized to generate the on-the-move positions for ~3000 platforms that drove the radio communications analysis and OPNET System Performance Models of IBCT MANETs, which included detailed models of JTRS and other radio protocols.

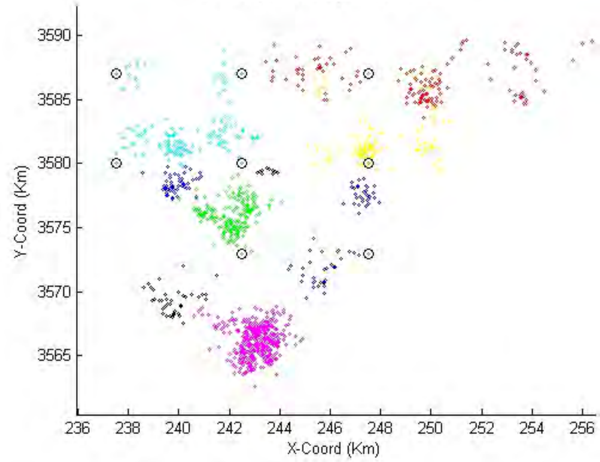

Figure 13 –Intermediate Animation Snapshot

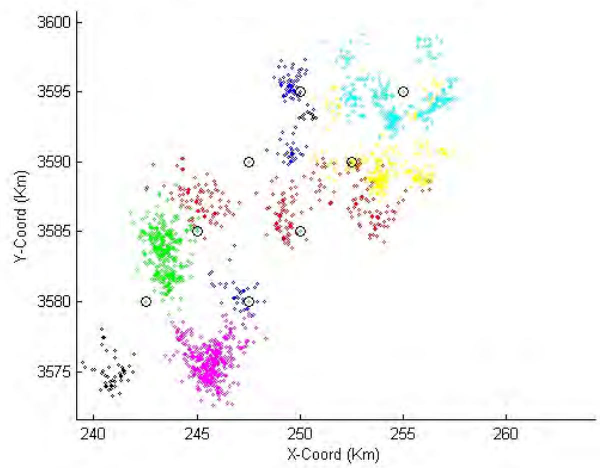

Figure 14 –Final Animation Snapshot

IX. CONCLUSIONS

The capabilities of HyPR have been proven via its successful use within the FCS Program, MITRE and recent formal delivery to TRADOC. Both the scale and fidelity of the resulting scheme and maneuver force laydowns are completely user-configurable. Even with just a small subset of key actual positions from a scenario, it can quickly generate representative positions for all remaining specific platforms in a military force structure. It has been used to generate scenario-linked laydowns and facilitate associated path loss value calculations between specific platforms for an entire ~3000 platform IBCT while on-the-move. This capability is not limited to work with BCT force structures, and additional information regarding potential usage of HyPR is available from the FCS NAIL upon request. HyPR continues to support various analyses both within and outside of the FCS NAIL.