Suppression Model

This document will discuss the changes to the direct fire suppression model introduced in Australia in the mid 1990s. The primary motivation for many of these changes was to support an infantry firepower study[1]. The fundamental problem with the Janus model was that the “theoretical” lethality of infantry small arms was too high, both in terms of rounds fired per enemy casualty and casualties inflicted over time. The Aide Memoire suggested that a platoon assault should take approximately 30 minutes, but given movement rates and firepower in Janus this scenario would be over in a few minutes. Suppression was one of the issues explored to improve the situation. The changes that were made were very simple from a coding point of view, but introduced a wide array of variation in the effect of suppression.

Janus(T)

The original Janus(T) developed by the US Army included a simple suppression model. Suppression could be caused by both direct and indirect fire. In the case of direct fire[2], all direct fire weapons had the potential to cause suppression against all ground platforms. Air targets could not be suppressed.

When an entity engaged a target with direct fire, Janus(T) calculated a single shot kill probability (SSKP) by multiplying the probability of hit (PH) and the probability of kill (PK). If this SSKP[3] was below a hard coded threshold of 0.05, no shot was fired. Otherwise, the shot was fired. If the shot resulted in a kill, the target was automatically suppressed[4]. If the shot was not a kill, then an additional random draw was made against a probability of suppression (PS) to determine if the target became suppressed.

This algorithm is described below.

Janus(T) Direct Fire Suppression Algorithm

```
calculate SSKP
if SSKP < 0.05 then
       do not shoot
else
       if random_draw1 < SSKP then
              target killed (and suppressed)
       else
              if random_draw2 < PS then
                     target suppressed
```

Importantly, the PS was a single value, defined when the scenario was created, that applied to all weapons and all target types. That is, assuming the shot did not kill the target, the probability that a rock thrown at a tank[5] had the same probability of causing suppression as a 120mm round fired at the same tank.

In Janus(T) suppression was measured as time. If an entity was suppressed it was suppressed for a fixed amount of time depending on whether it was a foot mover or other ground mover. Once again, this time value was independent of the weapon or the target type. A round fired at an armoured vehicle would suppress it for exactly the same amount of time as the same round fired and an unarmoured vehicle.

For example: Assume the global PS value is 0.5. If a rifleman fired at another rifleman with a PH of 0.4 and a PK of 0.8, the resulting SSKP was 0.32, which is high enough for the shot to be fired. If the shot

---

[1] The author has no references to any reports that were written on this study. It was conducted by an officer from the School of Infantry. The date is uncertain, but it was between 1993 and 1995 (inclusive).

[2] Indirect fire suppression may be described in a future document.

[3] Both the PH and PK were calculated based on weapon type, target type, range and target posture.

[4] This only mattered if the target entity was an aggregate. If it was a single entity platform, then the target was simply destroyed.

[5] This is an extreme example used to illustrate the situation. In reality, it is unlikely the “rock” would have met the initial SSKP threshold to take the shot in the first place, but the point is that the neither nature of the weapon or the target were considered when calculating the probability of causing suppression.

did not kill the target a second random draw is made and compared with the PS of 0.5. If this is less than the PS, the target becomes suppressed.

Suppression was a binary state. Either an entity was suppressed or not. Once suppressed, an entity could not move or shoot until the required time elapsed. This applied to every type of ground based system, so even a tank that was suppressed stopped moving for a short period of time.

Janus(T) Suppression Behaviour

```
if entity due to move then
       if entity suppression time > clock then
              do not move
if entity due to shoot then
       if entity suppression time > clock then
              do not shoot
```

If a target was suppressed again before the initial suppression time elapsed, the time at which the target was clear of suppression was simply increased to the current game time plus the global suppression time.

For example: Assume the global suppression time is 30 seconds. If, at time 1:00 a target was suppressed, it would remain suppressed until time 1:30. However, if it was suppressed again at time 1:15, it would then remain suppressed until time 1:45.

Janus(AS)

The first change that was made was to separate the PH and PK out of the SSKP calculation and make individual random draws. This was done for other reasons but was important for the infantry firepower study. Data collected from historical records[6] showed that the actual SSKP from rifle fire was quite low if measured simply by rounds fired against casualties. A simple calculation based on rules of thumb in the Army Aide Memoire suggested that the SSKP would be approximately 0.002[7]. If that number was used in the database, no rifleman would fire due to the SSKP threshold constraint.

Therefore, the threshold was instead applied to the PH value alone, meaning that as long as the PH was above 0.05[8], the PK could be quite low to give the required very low SSKP. This meant that instead of the two possible outcomes from the simple SSKP test (kill or miss), there were now three outcomes: no hit, hit but no kill, and hit with kill. We then decided that the “hit but no kill” state would equate to a suppression result, doing away with the global suppression ratio that had been applied to misses and replacing it with a value dependent on the weapon and target pair.

The new algorithm is described by the following pseudo code.

Janus(AS) Direct Fire Suppression Algorithm

```
determine PH
if PH < 0.05 then
       do not shoot
else
       if random_draw1 > PH then
              miss
       else
              target suppressed
              determine PK
              if random_draw2 < PK then
                     target killed
```

---

[6] One of the DSTO scientists working in the team had been researching data from Vietnam era contact reports that showed thousands of rounds had to be expended for each enemy casualty inflicted.

[7] Logistics data from the Aide Memoire suggested that a platoon assault against an enemy section would consume at least half of the ammunition from the two assaulting sections or approximately 100 rounds per man. Assuming that after the enemy received 5 casualties they either withdrew or surrendered this results in a SSKP of 5/ 2000 or 0.0025. Even if it is assumed that multiple shots hit each casualty, the SSKP is still well below the 0.05 threshold value.

[8] This default PH threshold was later changed so that the user could alter it during play on an entity by entity basis as a means of imposing fire control.

The second change related to the behaviour of an entity that was suppressed. Instead of the simple "no move or shoot" rule, three new fields were added to the database for each platform type: a factor (between 0.0 and 1.0) for movement, firepower and detection. When a suppressed entity attempted to move, its maximum speed was multiplied by its movement factor, when it attempted to shoot, its PH value was modified by its firepower value, and when it attempted to detect a new target, its probability of detection was multiplied by its detection factor. Using this scheme, the original behaviour could still be modelled by setting the movement and firepower factors to zero.

Janus(AS) Suppression Behaviour

```
If entity due to move then
       if entity is suppressed then
              maximum speed = maximum speed * speed modifier
if entity due to shoot then
       else if entity is suppresed then
              PH = PH * PH modifier
if entity due to detect then
       if entity is suppressed then
              PD = PD * detection modifier
```

This change meant that instead of forcing a suppressed armoured vehicle to stop moving and shooting, it could be degraded by whatever value was desired (including no degradation at all). For example, it was possible to model a tank "closing down" by just reducing its speed slightly and degrading its ability to detect and shoot to represent the reduced visibility due to reliance on periscopes and other limited sensors.[9]

The study scenario pitted a blue platoon assaulting a red section in a hasty defensive position. To setup the data, both sides were equipped with the same weapons meaning that at a given range and target posture they had the same PH and PK values. Where the target was in "defilade"[10] the PH was set lower than the same target at the same range "exposed". Thus, at a given range it was easier for the defenders to "hit" the attackers. However, the weight of fire from the larger number of attackers was enough to gradually suppress the defenders forcing them to then fire with the PH value modified by their suppression factor and allowing the attackers to recover from their own suppression and resume moving forward. In combination with these PH settings, the PK was set to a very low number against exposed targets and zero against defilade targets until the range reduced to a very close value.

These conditions created a balancing effect where the defenders initially suppressed most of the moving attackers, who immediately stopped and went prone allowing them to gradually recover from their suppression while simultaneously building up suppression on the defenders. As the attackers slowly inched forward, the firepower balance gradually tipped in their favour allowing them to maintain constant suppression on the defenders and gain greater freedom of movement until they could close to within lethal range and defeat the enemy. Once this situation could be modelled it was then possible to conduct the study which focused on changing the number of men and varying the number of machine guns in the section.

The input data were never "validated", but the model proved to have more credibility with the user than the previous simpler model.

A subsequent additional change was to replace the binary suppression state with a suppression value. Each weapon versus target pair was assigned a suppression amount and when suppressed, the target's current suppression amount was increased according to which weapon fired at them. Over time, this suppression amount decayed until it returned to zero. The rate of recovery was globally set by redefining the old suppression time field to mean the time it takes to recover from 1.0 units of suppression.

When testing to see if an entity was suppressed, the current suppression amount was examined. If it was greater than 1.0, the entity behaved as if it was suppressed as described above. If its current value was less then 1.0, then a random number was drawn and if that was lower than the current suppression

[9] This approach could have been extended by applying these factors to specific sensors or weapons or also reducing rate of fire or altering other characteristics. These options were discussed but not pursued at the time in the interest of simplicity.

[10] "Defilade" was the term used to reflect entities that were partially concealed or prone.

amount then it behaved as if it was suppressed for that tie tick, but if not then it behaved as if not suppressed.

For example, a shot from a rifle at a soldier might add 0.1 to the suppression amount on the target, while a shot from a sniper rifle might add 0.5 and thus cause more suppression.

This additional level of complexity achieved two effects: it allowed different weapons to generate different suppression amounts; and it modelled a gradual increase and decrease in the effect of suppression.

A final change that was coded, but never actually tested or activated was to vary the rate at which an entity recovered from suppression. The default was that a fixed amount of suppression was removed every two seconds of game time. However, this value could be varied on an entity by entity basis (not platform type by platform type) to represent some entities able to recover from suppression faster or slower: for example to represent the greater resilience of special forces or the fragility of civilians or untrained troops.