Towards Precise Measures of Individual Performance in Complex Tasks

Roussel Rahman (roussel.rahman@gmail.com) &
Wayne D. Gray (grayw@rpi.edu)
Cognitive Science Department, Rensselaer Polytechnic Institute
Troy, NY 12180 USA

Abstract

Simple laboratory tasks typically allow one or a few methods of task performance. In contrast, moderately complex tasks, such as video games, provide many methods of task performance which, in essence, provide many ways of completing the task without necessarily completing all possible components. Although performance on complex tasks improves with practice, the improvements do not represent the simple effects of power-law learning but, rather, they tend to reflect the discovery and practice of a diverse set of methods. Understanding what we see during complex task learning, requires us to evaluate individual performance against benchmarks of optimality. In this report, we use the game of Space Fortress (SF) as a complex experimental paradigm in which we demonstrate two alternative measures that reveal scopes of individual differences in the discovery and implementation of an optimal method that would be missed by traditional measures of the game.

Keywords: Complex Task Learning; Individual Learning; Plateaus; Dips; Leaps; PDL; SpotLight

Introduction

General laws which explain human learning as a function of practice (e.g., the power law or the exponential law) implicitly assume that practice alone is sufficient to reach the asymptote of performance. Although such assumptions may be reasonable for simple tasks that afford few alternative methods, they do not hold for more complex tasks, such as video games, which afford many alternative methods. A growing body of work (e.g., Siegler, 1987; Rickard, 1997; Delaney, Reder, Staszewski, & Ritter, 1998; Towne, Boot, & Ericsson, 2016; Thompson, McColeman, Blair, & Henrey, 2019; Rahman & Gray, 2020; van der Mijn & van Rijn, 2021) shows that individuals demonstrate both inter- and intra-individual differences of task execution methods during learning and also that the practice benefits are largely localized to the specific methods practiced. Indeed, even for seemingly simple video games (e.g., Pacman or Tetris), it may be difficult to identify the optimal method from amongst its numerous alternative possibilities.

The difficulty in finding the best or even an appropriate method can be observed in many real-world tasks; for example, finding the fastest route in traffic, finding a sure-win formula for Chess or Football, solving mathematical problems, even choosing the tasks to learn in a lifetime. How do humans search for and find the optimal method(s) in such tasks? To reach the asymptote, optimal methods must be discovered or invented. Therefore, theories of complex task learning must include an account of how the individuals' task execution methods evolve with learning to reach the optimal one(s) at the asymptote of performance.

Until now, we portrayed the complexity of complex tasks from a performer's perspective. But similar difficulties also persist for the researchers of complex skill learning in deciding where to look for measurable changes and which measures to use (Gray & Lindstedt, 2017). Looking at the wrong or imprecise measures can easily lead to false negatives of learning or training benefits, as underlying improvements may remain undiscovered (Gray, 2017). Moreover, if the asymptote(s) of performance and the corresponding optimal method(s) are both unknown, it is difficult to ensure that increments in performance measures are indeed steps towards the asymptote. The reason is that individuals may be using suboptimal methods that would lead to plateaus instead of the asymptote (Gray, 2017; Rahman & Gray, 2020).

An approach that has been useful in evaluating complex task performance is comparing performance against benchmarks of optimality. For example, Anderson, Kleinberg, and Mullainathan (2017) recently investigated the predictors of blunders in chess endgames, by comparing each move against known optimal moves. Relevantly, they found that the players are more likely to err in positions with fewer optimal or near-optimal moves within very large pools of possible moves. This relationship was consistent across all skill levels, even for the best human players with ELO ratings above 2300. In cases where optimal performance is not known, expert performance may serve as a substitute. For example, van Meeuwen et al. (2014) compared performance of novice air-traffic controllers against experts' performance to investigate how effective strategies are formed in solving complex visual problems (e.g., finding the optimal landing order for incoming planes).

In this work, we explore the benefits of evaluating individual performance against benchmarks of optimality in a historic experimental paradigm – the complex game of SF (Mané & Donchin, 1989). Since its development, SF has been used in many studies of complex skill learning to enrich our understanding of human learning process. However, several studies observed that two very important measures of SF – Velocity and Control – that represent the most fundamental skill needed in the game (flying in the game universe), are prone to ceiling effects; consequently, the measures asymptote before humans do (Boot et al., 2010; Destefano, 2010;

Gray, 2017). Here, we use two alternative measures – (1) angular velocity of player ship and (2) approximations of Pi (π) from ship paths – both tailored to capture progress towards optimal flight strategy of moving in slow, small circles around the enemy (i.e., the Fortress). These measures depict a much clearer picture of individuals' route to optimality and reveal scopes of changes in individuals' performance that would be missed by the traditional measures of SF.

The Game of Space Fortress

The game of SF was developed by Mané and Donchin (1989) as a common complex task for different research groups to study complex skill acquisition. The goal was to create an experimental task representative of real-world complex tasks incorporating dimensions of complexity based on existing research. Complexity of SF stems from both the multiplicity of tasks to be performed and the specificity of the ways they need to performed. In each game of SF, the player flies a ship (yellow plane in Figure 1) equipped with a limited number of missiles to engage in a five-minute battle against the Fortress (located at the center of the screen). The Fortress needs to be destroyed in two steps: (1) make it vulnerable by 10 or more hits (at intervals $>$ 400ms), then (2) a double shot with an interval within 250-400ms to destroy it – any deviation results in instant recovery of the Fortress. The Fortress fights back by shooting shells at the ship; in addition, its minions (the mines, Figure 1) spawn periodically at random locations to chase the ship. There are two types of mines, each of which requires identification by letter-codes shown at start screen and specific handling. The player must also protect the ship from getting hit by enemies, as four hits would result in ship destruction. Both the ship and the Fortress respawn upon destruction and the battle resumes. Finally, the player needs to manage the ship's arsenal. Each game starts with a full arsenal and the player receives several bonus opportunities to replenish the arsenal. The player can still shoot missiles with depleted arsenal, but sacrificing points per missile.

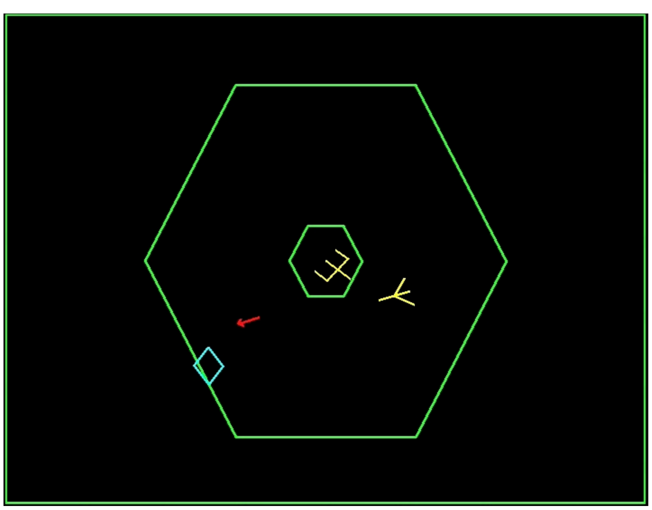

Figure 1: Screenshot of Pygame Space Fortress 4 (Destefano, 2010). The Fortress is at the center; the player's ship (yellow) have recently fired a missile (red) at a mine (blue diamond).

Game-generated Scores as Performance Measures

The objective of the game is to maximize the Total score, which is the sum of four subscores – Points, Speed, Control and Velocity – capturing performance in different subtasks. The Points score serves as a measure of several skills together; such as, skills in fighting the Fortress and the mines, defending own ship, managing resources. The Speed score rewards speed of killing mines and penalizes if mines escape.

The rest of the two scores are both measures of ship maneuvering skills. The Control score measures the performers' control over OS' spatial location; the player is rewarded at a higher rate for staying within the large hexagon than outside (Figure 1). The Velocity score measures the performers' control over OS' velocity; the player is rewarded for flying the ship within a speed limit and penalized for any violations. As mentioned previously, these two measures are prone to ceiling effects and do not consistently reflect improvements in associated skills. In the next section, we briefly review findings of optimal/expert flight behavior, before discussing alternative measures.

Review of Optimal (Flight) Strategies in SF

As mentioned earlier, SF was developed as a common paradigm to compare different training regimens (Mané & Donchin, 1989). The original game included only the Points score. The other three scores were added by Gopher, Weil, and Siegel (1989) for their *Emphasis Change* study. In their experiment, players practiced in the whole task, but were instructed to prioritize different parts at different points during practice. In contrast, Frederiksen and White (1989) adopted a part-task training approach by discretizing the gameplay into sub-tasks and trained the players by building up from small to more integrated subtasks. The purpose was to develop a better understanding of the dynamics of the SF universe. The researchers first identified the gameplay variables that affect task execution methods and the high-level goals of the game, after verbal protocol analysis of expert players' methods; then, decided on an optimal method as the foundation of a hierarchical training regimen.

Frederiksen and White observed that the optimal methods in low-level subtasks of Space Fortress are regulated by three high-level goals: (1) Hit Fortress without getting hit (2) Handle mines with the least possible disruption to the first goal (3) Allocate resources to maximize the Points score. For the two first goals, they suggested that players should fly around the Fortress in circles constructed by a series of pre-planned, linear trajectories at low speeds, and when a mine appears, players should wait until mines move to locations which require minimum deviation from the circles.

Recent works confirm that the flight paths of expert players indeed converge to circles around the Fortress (Destefano, 2010; Towne et al., 2016; Rahman & Gray, 2020). The need to construct circles with small lines stems from the constraints imposed in SF's input system: a player can either move the ship along straight lines (using *Thrust* key) or rotate (using *Rotate* key) to change ship direction, but cannot simultaneously use both to move at angles. To obtain a circular path (and to attack the enemies), a player needs to periodically repeat a sequence of keypresses throughout the game: *Thrust-Rotate -(Shoot)*. This way, movement constraints force players to construct the circles with numerous straight lines. Similar constraints also exist for joystick-based input systems. The need to precisely synchronize actions indicates that even if a high-level description of the optimal method is known, it cannot be implemented without understanding the mechanics of the SF universe and mastering the low-level action components. Confirming this view, Rahman and Gray (2020) found that players demonstrate both inter- and intra-individual differences of flight paths during learning, even when explicitly instructed on the optimal choice of slow circles.

Recent works also fine-tune Frederiksen and White's suggestions for optimal flight control. For example, Destefano (2010) observed that although low velocities are optimal for the Velocity score, players would benefit in defending against the Fortress by flying the ship faster. The reason is: the Fortress only fires at the ship if it can locate the ship for more than 1 second, in one of 36 equally divided segments around the Fortress (Destefano, 2010, pp. 34, Figure 14). Hence, by moving at an angular velocity faster than 360/36 = 10 degrees/second, a player can prevent the fortress from shooting at the ship. Therefore, the upper limit of velocity is the speed limit for Velocity score, whereas the lower limit of velocity is determined by the minimum angular velocity. In the next section, we knit these pieces of information together for a more precise description of optimal flight control in SF.

Methodology

Dataset used

We use the dataset from Destefano (2010). This dataset is publicly available (`osf.io/v5mzx/`) and has been used in several previous studies (Destefano & Gray, 2016; Gray, 2017; Rahman & Gray, 2020). We chose this dataset as it contains millisecond-level performance records of nine individuals over 31 hours of gameplay. Each individual played 8 games in each 1-hr session (one session per day), resulting in total 248 games per player. As the players needed time to familiarize themselves with the complex rules of SF, we exclude the data from the first day. Therefore, the final dataset contains 240 games for each player.

To provide a glimpse of the richness of information, the dataset contains about 40 game-aggregated measures to capture performance in different subtasks (e.g., number of Fortress kills, ship deaths, missiles bought). More importantly for our work, 9000 datapoints were collected at 30 Hz frequency from each 5-minute game, documenting each keypress by the player, each tiny movement by the Fortress or the mines and many more. The detailed records at the lowest-level performance mean that a researcher can develop performance measures tailored to answer specific research questions, at any level of the complex task.

Description of Optimal Flight Performance

To maximize the Control score, a player must fly the ship in the area between the large and small hexagons; using specifications of the hexagons, this statement can be written as:

$$50 < \textbf{Ship distance from Fortress} \text{ (in pixels)} < 182$$

To maximize the Velocity score, a player only needs to maintain *linear velocities* below a specified limit. But moving at a fast enough *angular velocity* in circles would prevent the Fortress from firing at the ship. Therefore:

$$\text{limit from ang. vel.} < \textbf{Linear velocity} \text{ (pixels/sec)} < 120$$

$$10 < \textbf{Angular velocity} \text{ (degrees/sec)} < \text{limit from lin. vel.}$$

Measures of Flight Control

To demonstrate the reasons for ceiling effects of Control and Velocity scores, we examine their main constituents – respectively, distance from the Fortress and ship velocity. As examples of alternative measures, we demonstrate two measures: (1) angular velocity of the OS and (2) approximations of π using Archimedes' method.

We chose angular velocity because, in circular motions, the radius of the circle (i.e., the distance from Fortress) and the linear velocities on the circles follow this mechanistic relation: $linear\ velocity = angular\ velocity * radius$ (Beer et al., 1972). Previously, it has been noted that the Control and Velocity scores are correlated (Boot et al., 2010; Gray, 2017); the aforementioned relation indicates that this correlation would exist only when the optimal method of moving in circles is adopted.

Finally, as a measure of goodness of the circles, we use approximations of π from the individuals' flight paths using Archimedes' method. The method of constructing circles with many lines, closely resembles Archimedes' method of calculating π by approximating circumferences of circles from perimeters of polygons. In both cases, with increasing number of sides, the polygons converge to the circles, resulting in better approximations of π.

Results and Discussions

Discrepancies between the Stories Revealed by the Control Score and its Constituents

As mentioned earlier, several previous works noted that the Control and the Velocity scores asymptote before humans reach the limit of performance and thereby hide underlying improvements in low-level constituents of the scores. In Figure 2, we show the game-averages of the Control score's main low-level constituent – ship distance from the Fortress – which confirms that the players improved in flying close to the Fortress deep into practice. However, the Control score would stop showing these improvements beyond the red-dashed line. The reason is, to max out the score, a player only

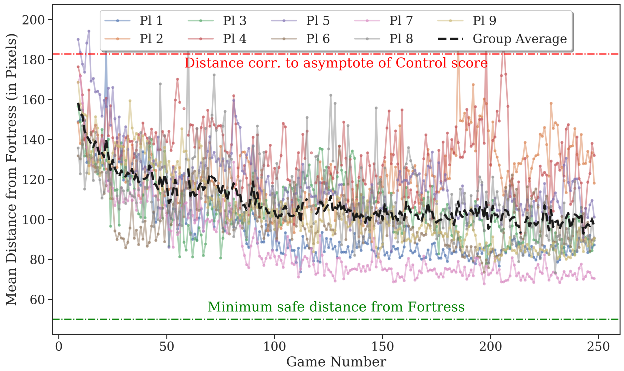

Figure 2: Mean distance for all players, almost none of whom seem to have reached the limit of performance even at the end of practice.

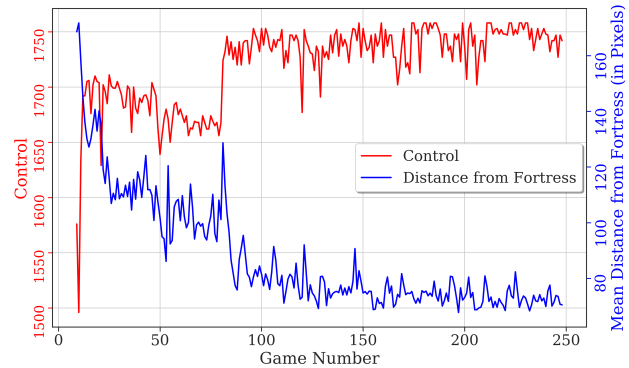

Figure 3: Control score vs its main constituent (mean distance from Fortress) for Player 7.

needs to stay within the large hexagon which corresponds to the ship distance denoted by the horizontal red-dashed line. However, as can be observed, players continue to improve beyond this threshold and gradually approach the minimum safe distance from the Fortress (green-dashed line).

The asymptote of the Control score is clearly false, but what about the score's ability to depict changes in individuals' performance with learning? Figure 3 demonstrates the Control score (red line) along with the game averages of the distance from Fortress (blue line), for our best player (Player 7) alone. Previously, it has been demonstrated that Player 7 went through a period (games 50-80) of extensive explorations of optimal flight paths before permanently adopting the optimal flight paths of circles around the Fortress (Rahman & Gray, 2020, samples of within-game trajectories in Figure 3, pp. 981). But the player's Control score in Figure 3 shows hardly any signs of these explorations. Rather, the player seemed to have suddenly leapt from a plateau to the asymptote. The mean distance (blue line) shows the improvement to be much more gradual than the Control score does, with no obvious plateau in the preceding period.

Although we discuss only the Control score here, the same discrepancies were also observed for the Velocity score and its only low-level constituent, ship velocity. To find the reasons behind these discrepancies, we next look at how these scores are constructed.

Discontinuous Reward Functions ⇒ Disproportionate Rewards with Performance

The step functions for rewards in Equations 1 and 2 explain why we see a stepwise progression of the scores despite continuous improvement of players in associated low-level performance, as step functions convert continuous input to stepwise, discontinuous outputs. To elaborate, these two scores are largely insensitive to any intermediate improvements in flying skills apart from right at the transition point of the function. For example, Velocity score rewards would be the same for flying OS at 10x, 2x and 1.01x of the speed limit, but different at 0.99x. The situation is analogous to having a digital watch showing only the hours of time; as the changes of minutes or seconds would not be observable, progression of time would seem to follow a step function to uninformed eyes.

$$\text{Control score reward} = \begin{cases} +6 \text{ per second;} & \text{Inside large hexagon} \\ +3 \text{ per second;} & \text{Out of large hexagon} \end{cases} \tag{1}$$

$$\text{Velocity score reward} = \begin{cases} +7 \text{ per second;} & \text{within speed limit} \\ -7 \text{ per second;} & \text{above speed limit} \end{cases} \tag{2}$$

In summary, the disproportionate relation between reward and performance level leads to (i) the asymptotes of the game-generated scores not being equal to the asymptotes of players skills, and may lead to (ii) false plateaus in individual performance hiding underlying improvement and (iii) false leaps by rewarding long-term improvement in one burst.

Exploiting Knowledge of Optimal Methods to Capture Progress towards Optimality

Spurious plateaus and asymptotes are likely to lead researchers to false conclusions about learning patterns and training effects, especially when the performance records are not as detailed as ours and are studied only at a high level. Therefore, the step-wise reward functions of the Control and the Velocity scores should be replaced with more continuous functions to develop measures sensitive to fine-grained changes in performance. One option is to adopt a reductionist view and deconstruct elements of performance to investigate progress towards optimality in each element. However, as all subtasks within a whole complex task are not independent, the true asymptote of performance in the whole task would inevitably be lower than the one estimated from parts. A more practical approach is to use the knowledge of optimal methods from previous works to identify or develop measures that would unambiguously reflect progress towards optimality.

For example, as for flight paths in SF, we know that the optimal method is to move in circles around the Fortress. To investigate progress towards this optimal method, we may

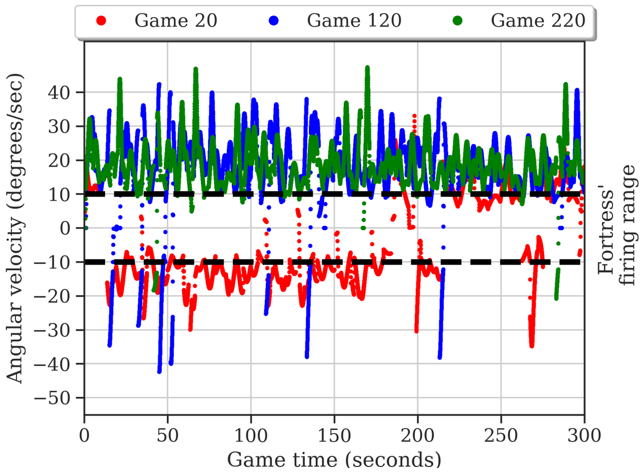

Figure 4: Instantaneous angular velocity (positive in the clockwise direction) in three sample games for Player 7

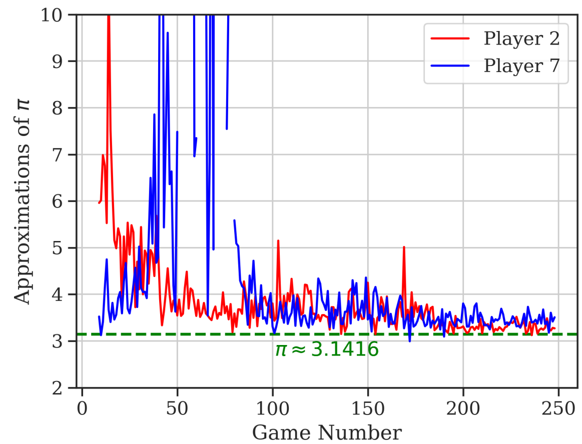

Figure 5: Approximations of π using Archimedes' method, for our best player (Player 7) and the worst (Player 2).

simply investigate how these circles improve with learning. The circles are implemented by controlling the ship's angular velocity, which we use as our first measure. Next, to investigate the goodness of end results (i.e., the circles), we use approximations of π from ship's paths.

Angular Velocity: The Velocity score aims to capture players' control over the ship based on its linear velocity (i.e., the rate of change of linear position) of OS; but for circular motions, angular velocity (i.e., the rate of change of angular position) is a more appropriate determinant of control. To create a perfect circle, the player needs to maintain a constant angular velocity throughout. This requirement, due to the input constraints of SF, needs to be approximated by consistently oscillating about steady reference values. In addition, the Fortress can be prevented from shooting by maintaining an angular velocity greater than 10 degrees/second within the circles. Therefore, instantaneous angular velocity provides an excellent measure to investigate within-game flight control.

Figure 4 shows the instantaneous angular velocities in three example games (at 100-game intervals) played by Player 7. As can be seen, from as early as the 20^{th} game (shown in red), the player demonstrates remarkable consistency in maintaining a wave-like angular velocity about steady reference values. In the 120^{th} game (blue), the player shows marked improvements in controlling the angular velocity and in staying away from the Fortress' firing range (marked in figure 4), and then shows comparatively smaller improvements in the 220^{th} game (green). As mentioned earlier, Player 7 extensively explored and practiced different flight paths within games 50-80 before permanently adopting the circles, providing a specific explanation for the diminished returns from practice. Finally, based on the patterns observed in angular velocity, we can safely conclude that the player indeed progressed towards optimal flight performance with practice.

Approximations of π using Archimedes' Method: Archimedes had observed that, with increasing number of sides (n), regular n-sided polygons become increasingly better approximations of circles. This simple observation led him to develop one of the earliest methods to calculate π as the ratio between the perimeter of the n-polygon and its largest diagonal. Although the flight paths taken by our players are not regular polygons, this ratio can still be used to approximate π for each full circle around the Fortress.

Figure 5 demonstrates the game averages of these π values for two players: Players 7 and 2, respectively the best and the worst performing players according to the Total scores achieved in the last 50 games. As mentioned earlier, Player 7 experimented with different flight paths (e.g., moving along lines or half-circles) within games 50-80. The impact of these experimentations are clearly observable, as either π could not be calculated (13 games) or were very inaccurate during this period. To facilitate comparison between the players, we limit the y-axis to show only values below 10, which occludes 11 games for Player 7 and one for Player 2.

Although these players demonstrate opposing trends early in practice, both players can be observed to be approaching the asymptote (i.e, the true value of π), yielding increasingly better approximations of π with more practice. To illustrate the level of accuracy reached at the end of practice, average π in the last 50 games is 3.3 (SD = 0.09) for Player 2 and 3.4 (SD = 0.14) for Player 7.

To note, the approximations of π from each circle around the Fortress can also be used as a within-game measure of performance in maintaining the circles. We skip this demonstration due to space constraints, but the within-game approximations fluctuate a lot more than the game-averages do, indicating substantial detours from the circles. Therefore, even though the game-averages suggest that the players are approaching the asymptote of performance, ample room for improvement may still remain.

Conclusions

In this work, we highlight the need to evaluate individual performance in complex tasks against benchmarks of optimality. Individuals demonstrate ample differences of task execution methods in complex tasks, therefore, looking in the wrong scopes of improvement may lead to false negatives regarding individuals' training or practice benefits. In such cases, measures tailored to capture performance within scopes of optimality, provides a commonground to compare different individuals and search for general patterns underneath the individual differences.

For our demonstrations, we use the complex game of SF and investigate individuals' acquisition of one fundamental skill – flying the ship – using two measures tailored to capture progress towards the optimal flight strategy. We chose angular velocity as our first measure, as the optimal path of circles needs to be implemented by controlling the angular velocity. Second, we use approximations of π as a measure of goodness of the circles created.

These measures – directed to capture performance within scopes of optimality – are able to reveal scopes of consistency and changes in individuals' performance that would be missed by undirected measures. For example, our results indicate that the individuals did realize that the circular paths need to be achieved by maintaining a consistent (optimally, constant) angular velocity and improved in doing so with practice. Excellent approximations of π towards the end of practice show that these players attained near-asymptotic skill levels in executing the optimal flight strategy. Importantly, as the asymptotes are known for both measures (i.e., constant angular velocity and the true value of π), improvements in these measures can be unambiguously interpreted as progress towards optimal performance. The known asymptotes also allow us to reliably investigate within-game performance of individuals with the same measures and identify both the scopes of current expertise and for further improvements.

Although our demonstrations are in one game only, the game of SF represents real-world complex tasks that present performers with the general difficulty to identify optimal methods among many alternatives. Evaluating performance against benchmarks of optimality would help us find general explanations for how individuals' different routes converge towards the same optimal methods and when do they diverge towards plateaus of stable, suboptimal performance. This way, by helping to uncover the evolution of individuals' task execution methods, precise measurement and evaluation of individuals' performance can help us progress towards the general laws of individual learning.

Acknowledgments

Correspondence should be sent to Roussel Rahman, Cognitive Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180. Email: roussel.rahman@gmail.com . The work was supported, in part, by grant N00014-16-1-2796 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer.

References

Anderson, A., Kleinberg, J., & Mullainathan, S. (2017). Assessing human error against a benchmark of perfection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *11*(4), 1–25.

Beer, F. P., Johnston Jr, E. R., Mazurek, D. F., Cornwell, P. J., Eisenberg, E. R., & Sanghi, S. (1972). *Vector mechanics for engineers* (Vol. 1). Tata McGraw-Hill Education.

Boot, W. R., Basak, C., Erickson, K. I., Neider, M., Simons, D. J., Fabiani, M., ... others (2010). Transfer of skill engendered by complex task training under conditions of variable priority. *Acta Psychologica*, *135*(3), 349–357.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science*, *9*(1), 1–7.

Destefano, M. (2010). *The mechanics of multitasking: The choreography of perception, action, and cognition over 7.05 orders of magnitude*. Unpublished doctoral dissertation, Rensselaer Polytechnic Institute.

Destefano, M., & Gray, W. D. (2016). Where should researchers look for strategy discoveries during the acquisition of complex task performance? The case of Space Fortress. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society* (pp. 668–673).

Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychologica*, *71*(1-3), 89–146.

Gopher, D., Weil, M., & Siegel, D. (1989). Practice under changing priorities: An approach to the training of complex skills. *Acta Psychologica*, *71*(1-3), 147–177.

Gray, W. D. (2017). Plateaus and asymptotes: spurious and real limits in human performance. *Current Directions in Psychological Science*, *26*(1), 59–67.

Gray, W. D., & Lindstedt, J. K. (2017). Plateaus, dips, and leaps: Where to look for inventions and discoveries during skilled performance. *Cognitive Science*, *41*(7), 1838–1870.

Mané, A., & Donchin, E. (1989). The space fortress game. *Acta Psychologica*, *71*(1-3), 17–22.

Rahman, R., & Gray, W. D. (2020). Spotlight on dynamics of individual learning. *Topics in Cognitive Science*, *12*(3), 975–991.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, *126*(3), 288.

Siegler, R. S. (1987). The perils of averaging data over strategies: An example from children's addition. *Journal of Experimental Psychology: General*, *116*(3), 250–264.

Thompson, J. J., McColeman, C. M., Blair, M. R., & Henrey, A. J. (2019). Classic motor chunking theory fails to account for behavioural diversity and speed in a complex naturalistic task. *PLOS ONE*, *14*(6), 1-24.

Towne, T. J., Boot, W. R., & Ericsson, K. A. (2016). Understanding the structure of skill through a detailed analysis of individuals' performance on the space fortress game. *Acta Psychologica*, *169*, 27–37.

van der Mijn, R., & van Rijn, H. (2021). Attention does not affect the speed of subjective time, but whether temporal information guides performance: A large-scale study of intrinsically motivated timers in a real-time strategy game. *Cognitive Science*, *45*(3), e12939.

van Meeuwen, L. W., Jarodzka, H., Brand-Gruwel, S., Kirschner, P. A., de Bock, J. J., & van Merriënboer, J. J. (2014). Identification of effective visual problem solving strategies in a complex visual domain. *Learning and Instruction*, *32*, 10–21.