When should the chicken cross the road?: Game theory for autonomous vehicle - human interactions

C.W. Fox[a,b,c], F. Camara[c,b], G. Markkula[c], R.A. Romano[c], R. Madigan[c] and N. Merat[c]

February 6, 2018

Abstract

Autonomous vehicle control is well understood for localization, mapping and planning in un-reactive environments, but the human factors of complex interactions with other road users are not yet developed. This position paper presents an initial model for negotiation between an autonomous vehicle and another vehicle at an unsigned intersections or (equivalently) with a pedestrian at an unsigned road-crossing (jaywalking), using discrete sequential game theory. The model is intended as a basic framework for more realistic and data-driven future extensions. The model shows that when only vehicle position is used to signal intent, the optimal behaviors for both agents must include a non-zero probability of allowing a collision to occur. This suggests extensions to reduce this probability in future, such as other forms of signaling and control. Unlike most Game Theory applications in Economics, active vehicle control requires real-time selection from multiple equilibria with no history, and we present and argue for a novel solution concept, *meta-strategy convergence*, suited to this task. [1]

1 Introduction

Automated vehicle (AV) localization, mapping, and planning have recently become practically feasible due to price falls in computer processing power. The problem of simultaneous localization and mapping (SLAM) is well understood [16], and despite its exact solution being NP-hard [15], good approximations exist such as particle filtering, which make use of large compute power to draw samples near solutions. Route planning in non-interactive environments also has well known tractable solutions such as the A-star algorithm. Given a route, localizing and control to follow that route then becomes a similar task to that performed by the 1959 General Motors Firebird-III self-driving car [1], which used electromagnetic sensing to follow a wire built into the road. Such path following, using wires or SLAM, can then be augmented with simple safety logic to stop the vehicle if any obstacle is in its way, as detected by any range sensor. Free and open source systems for this level of 'self-driving' are now widely available [6].

In contrast, problems that these vehicles will face around interacting with other road users are much harder both to formulate and solve. Autonomous vehicles do not just have to deal with inanimate objects, sensors, and maps. They have to deal with other agents, currently human drivers and pedestrians and eventually other autonomous vehicles, all of which may be at least as "intelligent" and "rational" as they are, and in competition with them for space, time and priority on the road. Recent studies have shown [9] that in trials of autonomous minibuses, pedestrians knowingly obstruct autonomous vehicles around once every three hours – enough to occur once every day on a long commute. Once human road users know that AV safety systems are programmed to stop if any obstacle is in their path, they can quickly take advantage of this to push in front of the AV and take priority. If this becomes common knowledge across a whole city, AVs will make little or no progress because they will be forced to yield at every single interaction.

Understanding and predicting other agents' behavior, especially when that includes understanding and predict-

[1]Author affiliations: *a* Ibex Automation Ltd, UK; , *b* School of Computer Science, University of Lincoln, UK; *c* Institute for Transport Studies, University of Leeds, UK. This project has received funding from the European Union's Horizon 2020 R&I programme under grant agreement InterACT No 723395.

ing their understandings and predictions of oneself, and of one's understanding and predictions of them, *ad infinitum,* is a massively more complex problem than inferring locations and maps. It may even be formally uncomputable as it requires predictions of and actions based on one's future behavior (via the other party's models of it), which are well known in Computer Science to lead to paradox and uncomputability as in Gödel's theorem and the Halting problem [17].

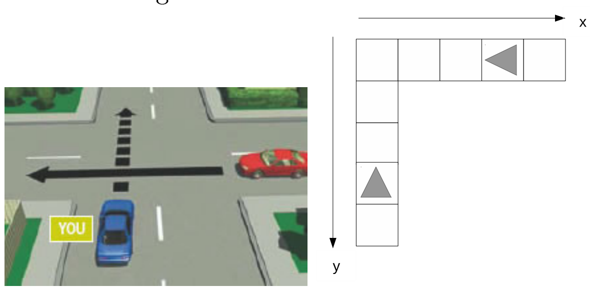
Game Theory provides some framework for managing such self-referential decisions, but appears to be incomplete as a prescriptive theory when multiple equilibria are present. Solutions may be formally computationally intractable (NP-hard) in some cases [13]. And when multiple equilibria are present, it is not clear whether game theory or any other rational line of argument will ever be able to even formulate the problems, let alone solve them. A simple example of game theory is the classic game 'Chicken', in which two cars each drive *straight* towards each other at speed or *swerve* away, and the nominal loser (the 'chicken') is the one to swerve, but both players are much bigger losers if they both do not swerve, and collide. The classic formal Chicken model makes the strong and unrealistic assumption that the straight/swerve decision is made as a single action (a) selection, simultaneously by both players (X and Y) so that the payoff values (v) given the actions can be represented as a 2×2 matrix of pairs $(v^Y_{a_Y,a_X}, v^X_{a_Y,a_X})$:

$Y \mid X$	$a_X =swerve$	$a_X =straight$
$a_Y =swerve$	(0,0)	(-1, +1)
$a_Y =straight$	(+1, -1)	(-100,-100)

The central concept of game theory is *equilibrium* [11] which for a 2×2 matrix game as above describes any pair of strategies for the two players such that if either player knew the other's they would not change their own. Conceptually, if equilibria exist then one can usually be found via 'fictitious play', i.e. each player simulates her and her opponent's behaviour in a virtual world where they know each other's strategies, until they converge. Strategies are probability distributions over actions. If a game has only one equilibrium then it is optimal for both players to play its strategies. When there are multiple equilibria, the question of what to do is less clear. Historically, Game Theory has focused on description of observed behaviors (how *do* people act? Why *did* the chicken cross the road?) and mechanism design (how can government make them act?) rather than prescription (when *should* we act? When *should* the chicken cross the road?) of real-time action selection by agents. This is due to its roots in economics and mathematics rather than robotics. This distinction becomes crucial when multiple equilibria are present in a game. The descriptive school would say that *any* of the equilibria are valid descriptions of what might be observed in human behavior data [4]. The 'mechanism design' school typically argues [12] that the problem of equilibrium selection is 'unpleasant' and should be avoided by changing the rules of the game to produce a single unambiguous equilibrium which all players can usefully use. For cases where the equilibrium selection problem cannot be avoided in these ways, many different solution concepts (aka. 'equilibrium refinements') have been proposed and debated [2] for choosing between them. Two of these are widely acknowledged to be 'rational' when applied in order: *1. Dominance* - Clearly, if an equilibrium is worse for all players than at least one other, then we discard it without question*; 2. Evolutionary stability / symmetry* (ESS) - Assume everyone in the world was to use the same equilibrium as me. Discard equilibria where this would not work. These concepts help to reduce the number of potential multiple equilibria but do not guarantee reduction to a unique one. To discard further equilibria, more controversial concepts have been debated [2] including: *1. Trembling hand stability* - Assume other player has a small epsilon probability of making a mistake; prefer equilibria that we converge back to if that happens; *2. Basin size* (aka. 'risk dominance'). Assume both players use fictitious play starting from maximum entropy strategies. Consider which equilibrium attracts from the most start points, and this thus most likely to occur. *3. Social maximum* (aka. 'payoff dominance') Choose the equilibrium with the largest sum of payoffs to all players (even if I am worse off than in others). *4. Other arbitrary conventions***.** Such as using the action with the first letter in the alphabet. These work only if all players agree to use them in advance or can be argued to possess social knowledge to make them confident that others will choose the same ones as them. This appears to be the point where the mathematics of Game Theory ends, and philosophical debate about the meaning of 'rationality' takes its place [2]. However, as autonomous vehicle engineers building real-time control systems, we *must* make some action selection in these situations, somehow.

Chicken is intended as a simple educational example game and not as a model of real vehicles. This study modifies it into a general and more realistic vehicle interaction problem, where an AV competes for priority with another vehicle or a pedestrian stepping out in front of it, and allowing them to negotiate with one another by observing each others' behaviour over time. Initially we

Figure 1: Scenario and model.



consider the simplest possible model of this class of problems, of an AV and another *similar vehicle* approaching an unmarked intersection at speed as in fig. 1(left). This would also apply to the case of two pedestrians meeting each other and negotiating for space. We then extend it to cases where the two players have *asymmetric penalties* in the event of collision – modeling an AV encountering a potential jaywalker stepping into the road, or an AV encountering a physically larger or smaller vehicle such as an armored SUV or cyclist at an intersection. The model proposed here is intended to be the simplest possible which captures the dynamics of interest common to these cases, but which can also serve as a foundation for many more complex ones.

Game theory is used extensively in macroscopic traffic modeling via Wardrop equilibrium in flow networks [3] with focus on route selection in large, economy-like, markets of many road users rather than microscopic pairwise interactions. Where game theory has been applied to pairwise traffic decisions, it has mostly been at the level of simple single-shot games as reviewed in [5]. In a few cases such as lane-changing [10, 7] and merging [8] it has been extended to sequential games as used here, but not for AV-pedestrian interactions as here. The meta-strategy convergence concept used here is novel to our knowledge, as is the use of the sequential model as a foundation for AV-human intersection and jaywalking control.

2 Methods

Turn-taking model. In the simplest possible model we set up two symmetric agents (vehicles or pedestrians) approaching an intersection as a chess-like, discrete space, discrete speed, and discrete time, turn-taking game, as in fig. 1(right). This does not yet use any Game Theory because the players' decisions are not made at the same time. We will use it to introduce notation and as a base to grow more detailed models. We assume a (1 meter) grid world, with two straight roads at right angles meeting at the intersection. Assume one vehicle on each road, labeled[2] Y and X. (We will later consider one player to be a pedestrian or different types of vehicle.) Assume discrete alternating turns one per second, in which one vehicle can choose either a 1m/s or 2m/s speed, i.e. move either one box forward or two boxes forward. (This ensures a finite game, because the vehicles are guaranteed to move closer to the intersection at each step. Formally, y and x are decreasing variants.) 2m/s is a slow real world speed but is chosen to coincide with simple integer movements of 1 or 2 boxes per turn. (The reader may wish to multiply all distances and speeds by 10 if they wish to think in more real-world units.) Write y for Y's distance in meters to the intersection, x for X's distance in meters. Assume a crash occurs if the vehicles are in the same square or if one vehicle is in the intersection square and the other moves *through* it in a single step. (This may be implemented by treating $(y, x) \in \{(0, 0), (1, 1)\}$ as crash states, and all other states with one agent at 0 or 1 as non-crash endgame states, avoiding the need to model negative positions beyond the intersection.) We assign (negative) utility U_{crash} to each player for a crash; otherwise $-U_{time}T$ where T is the number of seconds it takes to reach the intersection from the start of the game, and U_{time} is the (positive) value of saving one second of travel time. Assume that both players have identical utility functions, and know this to be the case. This game can be played, for example, as a board game between two human players. The turn-taking model can be solved by a standard [14] backward induction max-max tree search as in algorithm 1, where the boolean b represents which player's turn is current, t is time elapsed, y and x are the two players' positions, and the results are expected value pairs for the two players, $v_{y,x,t,b} = (v^Y_{y,x,t,b}, v^X_{y,x,t,b})$.

Sequential chicken model. If we replace turn-taking by simultaneous action selection by both players at each discrete 1s turn, the model transforms into a sequence of Game Theoretic matrix games ("sub-games"). This is equivalent to a board game where both players write down their speed choice (1 or 2) in secret then reveal them and make the moves together rather than in turns. The payoffs of any sub-game at state $(y > 1, x > 1, t)$ become recursive functions of the next states, $(y - a_Y, x - a_X, t + 1)$,

[2] We use the convention of writing Y before X, and the orientation of the grid world of fig. 1, to match (row,column) matrix notation.

Algorithm 1

```
world_value((y,x,t,b)) =
  if y==x==0  or y==x==1
  or (y==1 and x==0) or (y==0 and x==1)
  or (y==0 and x==-1) or (y==-1 and x==0):
    (-100,-100)             #crash
  if y<=0 or x<=0:
    (-t-y/2 , -t-x/2)       #someone arrives
  if b:                     #recursion
    mymax(world_value((y-2,x,t+1,False)),
          world_value((y-1,x,t+1,False)))
  if not b:                 #recursion
    yourmax(world_value((y, x-2, t+1, True)),
            world_value((y, x-1, t+1, True)))
```

where $a_Y, a_X \in \{1, 2\}$ are the speed selection actions for speeds of 1m/s and 2m/s. As in the turn-taking model, these inductive values are based on the endgame states when one or both vehicles have reached the intersection (considered to occur at square 0 or 1). Consider the value $v_{y,x,t} = (v^Y_{y,x,t}, v^X_{y,x,t})$ of the sub-game when the game is in state (y, x, t). The induction relation for this sub-game's payoff matrix is,

$$v_{y,x,t} = v(\begin{bmatrix} v(y-1, x-1, t+1) & v(y-1, x-2, t+1) \\ v(y-2, x-1, t+1) & v(y-2, x-2, t+1) \end{bmatrix})$$

and is computable via standard matrix Game Theory. Optimal mixed strategies, where they exist, and resulting state probabilities given an initial start state, for this model are shown in fig. 2.

Asymmetric utility model. A final model asks what happens if the two players have different collision utilities. This occurs for example if one player has a heavier/safer car than the other, such as an SUV (Sports Utility Vehicle). Or if one player is a weaker road user such as a cyclist, or a pedestrian negotiating to cross the road in a conflict zone in front of our AV. Will even a small change to these utilities break the symmetry of the sequential chicken model and tip the balance of who yields? If so, this would give a rational justification for the purchase of heavy vehicles such as SUVs: the intent of such purchases is not to actually get into collisions and benefit from reduced damage, but rather to maintain the possibility, however remote, of such a disaster, and exploit the backward induction from it to obtain concessions in more benign possible worlds, namely of the other player yielding. Purchase of an SUV would then be rational, reducing the cost of time delays to the owner. (There is of course then a higher level game when the other road users can buy similar vehicles, beyond the scope of our present model.) For simplicity we retain the assumption that both players have the same time delay utilities, as in the previous model. We assume the original collision-to-time value scaling of $(U_{crash,}, U_{crash})$, and consider ratios where one player is stronger than the other, as (U_{crash}, rU_{crash}), for $r \in [1, 100]$. When utilities become asymmetric it is possible that more equilibria will be present, so we switch to numerical computation of them using the Lemke-Howson algorithm.

Meta-strategy convergence. The asymmetric chicken model may have multiple equilibria which are not fully disambiguated by dominance and ESS solutions concepts. We propose a novel (to our knowledge) solution concept for use in solving this and other models, which we call *meta-strategy convergence*. This is based on everything we currently know about the 'rational' process of equilibrium pruning and selection, including the *absence* of information in some cases, and on a temporal ordering of rational reasoning. After removing dominated and non-ESS equilibria, we know of no good remaining solution concepts under the assumption that the other player is also rational. (Trembling hand, basins etc. make a different assumption about a fallible opponent, but for AVs which will eventually interact with other autonomous vehicles, we want all players to be completely mechanized and rational.) In the absence of any other way to select from the remaining equilibria's strategies, we form a new meta-strategy which chooses one of them from a flat (maximum entropy) prior. By symmetry, there is no way to prefer any over any other, hence their selection probabilities must be equal, given this state of knowledge. Curiously, this is equivalent to a new strategy which averages the action probabilities from each remaining strategy, yet is not itself in that set, because it is an average between them. Hence, it is not a member of any equilibrium and cannot be an optimal strategy itself. However, we have derived it step-by-step over time in a completely rational way. It is our best solution so far at this new point in time. As it is our best rational solution at this time, the other player will also compute that we have reached it. We then consider, as in standard fictitious play, what the other player will do next. They will apply fictitious play to modify their strategy in response. Then we will modify ours, and they will modify theirs again. This will iterate until we converge, unambiguously, onto a specific and uniquely defined one of the original equilibria. This

Figure 2: Strategy selection (1) and backward induction state probability (2-3) equations.

$$P(a_Y = 1|y,x,t) = \frac{v^Y_{y-1,x-2,t+1} - v^Y_{y-2,x-2,t+1}}{v^Y_{y-1,x-2,t+1} + v^Y_{y-2,x-1,t+1} - v^Y_{y-1,x-1,t+1} - v^Y_{y-2,x-2,t+1}}$$

$$P(y_{init}, x_{init}, t_{init}) = 1, P(y_{init}, x \neq x_{init}, t_{init}) = 0, P(y \neq y_{init}, x_{init}, t_{init}) = 0$$

$$P(y,x,t) = \sum_{\Delta y \in 1,2} \sum_{\Delta x \in 1,2} P(a_Y = \Delta y, a_X = \Delta x | y - \Delta y, x - \Delta x, t-1) P(y - \Delta y, x - \Delta x, t-1)$$

contains the rational strategies for both players, and can be reached deterministically by both of them without the need for any pre-established conventions or communications. Like all solution concepts, this is something of a philosophical rather than purely mathematical argument. It is the best argument currently known to us so we consider it to be rational for the AV control tasks.

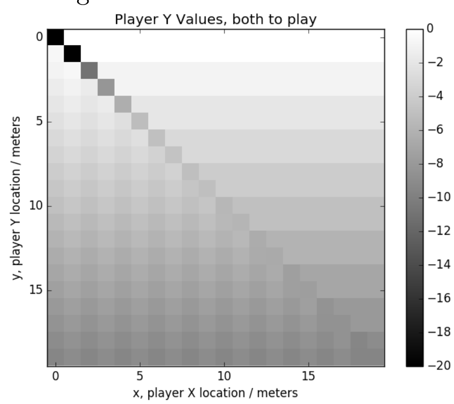
Temporal gauge invariance. The state values of the game theoretic models above are presented as functions of (y, x, t) but our current implementation makes use of an approximation to reduce the number of sub-games to be solved and thus the computation time. Because both player's utilities are linear function of time, we may (up to a small change in the ratio of crash to delay utilities) choose different gauges to measure time, such as considering the time of every turn game to be $t = 0$. When the first player, say Y, reaches the intersection, we assign values $(v^Y_{0,x}, v^X_{0,x}) = (0, -U_{time}x/2)$ as it will take X a further $x/2$ seconds to reach the intersection at maximum speed 2m/s now the road is clear. (Also $(v^Y_{1,x}, v^X_{1,x}) = (0, -U_{time}(x-1)/2)$ to handle the other required end states in the same way.) This simplifies all state values and functions of them to be functions only of (y, x). Removing dependence on t also makes it simpler to visualize results as 2D (y, x) matrices.

3 Results

Assume $U_{crash} = -20$ and $U_{time} = 1$ throughout. (This values a crash as being equally bad as a 100 second delay reaching the intersection. In the real world the crash penalty would be much larger, but smaller ones produce more easily visualizable results for our present purpose.)

Turn-taking model. The value function for Y in the turn-taking model is shown in fig. 3. The game is symmetric so X has the same function when the player's names are swapped. The turn taking model is fully deterministic, because full information is available to each player when it is their turn. Fig. 4 and 5 show simulated runs beginning at asymmetric $(y \neq x)$ and symmetric $(y = x)$ starting states.When the vehicles start with very different differences, e.g. $(y = 12, x = 8)$ they both proceed at full speed (2m/s) and avoid each other. When started at identical distances, such as $(y = 10, x = 10)$, the initial turn-taking advantage becomes the tie-breaker, in both parties interests. *Collisions never occur in the turn-taking model* due to its determinism.

Figure 3: State values for turn-taking game.



Sequential chicken model. Fig. *6* shows the value matrix for games with vehicles at up to 20m from the intersection, and fig. 7 show the optimal strategies.

Fig. 8 shows the state space probabilities and fig. 9a stochastic sample, starting with large time (2 seconds) gap between the vehicles. All probabilities in the state

Figure 4: Simulation of turn taking game, differing starts.

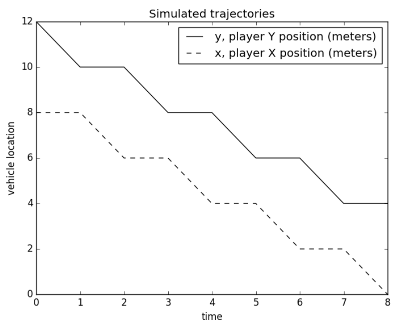


Figure 5: Simulation of turn taking game, equal starts.

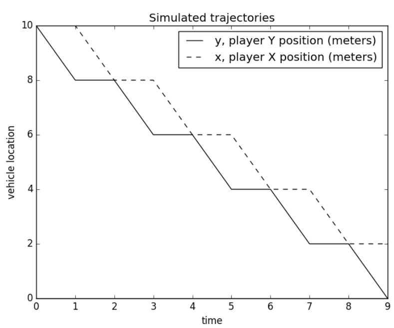


Figure 6: Sequential chicken state values.

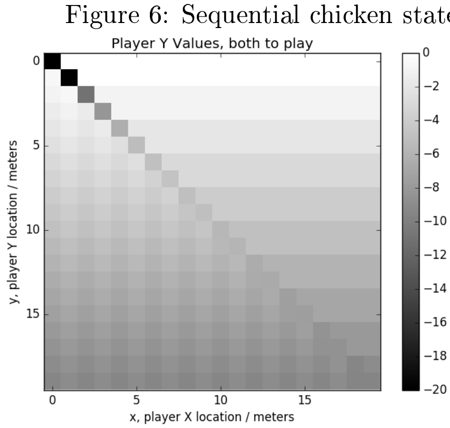


Figure 7: Sequential chicken optimal strategy.

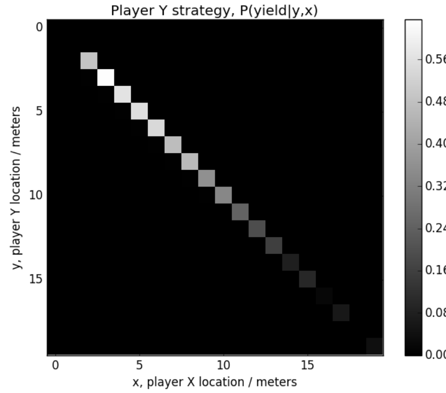
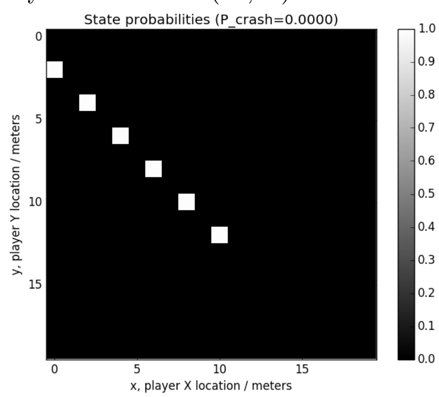


Figure 8: Sequential chicken state probabilities, from asymmetric start (12,10).



space are very close to 0 or 1, so the outcome is almost deterministic as in the turn-taking model of the same setting.

Fig. 10 shows state space probabilities when the vehicles start at identical distances $y = x = 10$. In this case, the outcome is different from the turn-taking model, because the game is fully symmetric but the symmetry is no longer broken by turn-taking. This means that both players must employ a policy consisting of mixed strategies until the symmetry is broken by one of them. The optimal policy is to yield with an increasing probability as distance to collision decreases, as seen in fig. 8. Fig. 11 is a typical sample simulation drawn from the above state probabilities. The most common outcome is for one vehicle to begin to yield at a random time, with yield

Figure 9: Sequential chicken simulation, from asymmetric start (12,10).

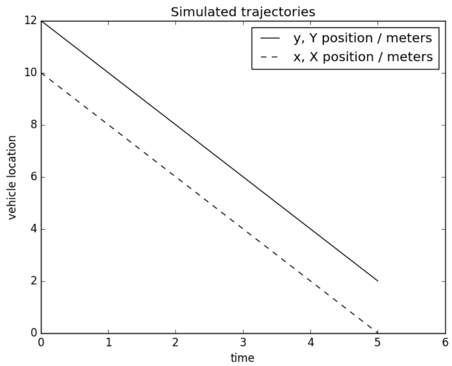
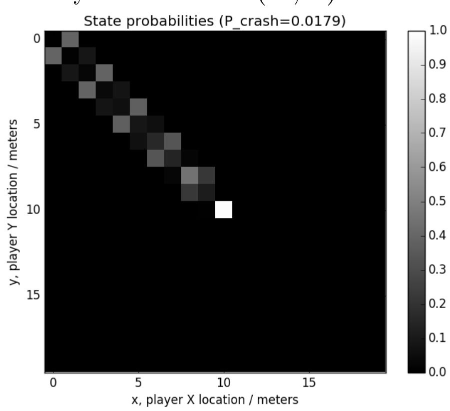


Figure 10: Sequential chicken state probabilities, from symmetric start (10,10).



probability increasing as the vehicles draw closer.

Occasionally, as in fig. 12, both players choose to yield at the same time, prolonging the conflict and costing them both a delay. Very occasionally a collision will occur as a result of this process, when the players fail to negotiate priority before both reaching the intersection together. This is rare, but must have a non-zero probability, computed and denoted as P_{crash}.

The collision probability with $U_{crash} = -20$ is 1.79%, and with $U_{crash} = -100$ it lowers to 0.7%. It is interesting to test how choice of this scaling (versus the fixed U_{time}=1) affects the residual collision probability, because if it has a large effect then any realistic model will require scaling calibration against some empirical data.

Figure 11: Sequential chicken typical simulation, from symmetric start (10,10).

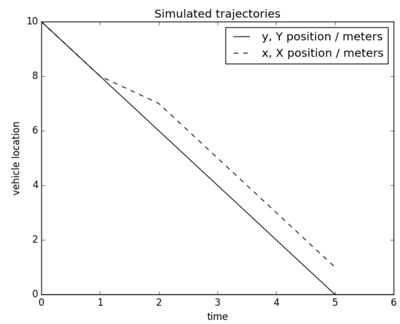


Figure 12: Sequential chicken atypical simulation, from symmetric start (10,10).

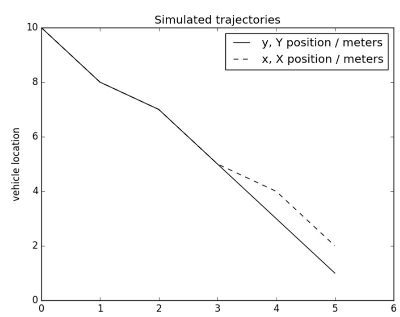


Figure 13:

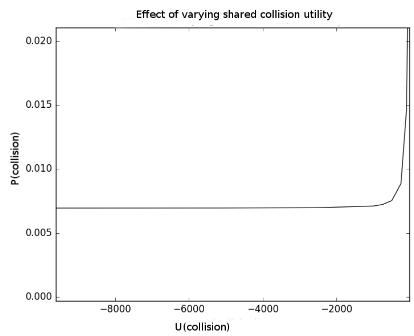
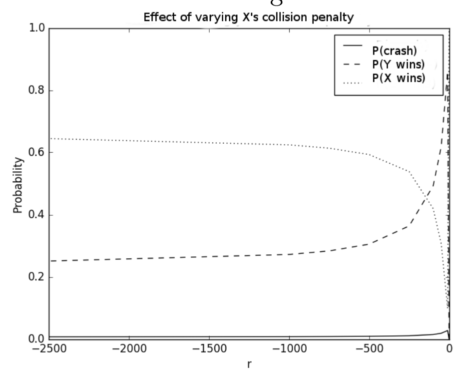


Figure 14:



Assuming that both players continue to share the same utility function, and know this to be the case, fig. 13 shows the effect of varying the collision utility scale while keeping the time utilities fixed.

This suggests that choices in range [-1000,0] do have large effects on the collision probability, but penalties worse than -1000 are much the same as each other in this effect. In real life, the cost of crashing a car (even very slightly) is almost always vastly larger than that of losing a few seconds journey time. This graph suggests that for these realistic penalties, the precise choice of shared collision penalty values is unimportant as long as it is over 1000 times worse than a 1 second time delay penalty.

Asymmetric utility model. As hypothesized, asymmetric collision penalties (such as our AV encountering a strong SUV or a weak cyclist or pedestrian) have a large effect on who must yield, and with a very small change in the probability of actual collision. This small change is the key to breaking symmetry and ensuring strong probability of the weaker player yielding fig. 14.

4 Conclusion

In all the models, when one agent has any small advantage it is usually – but probabilistically – optimal for both agents for the strong one to take the priority and the weak one to yield.

It is essential that there is some small but strictly non-zero probability of collision being allowed to occur as a consequence of both sides' optimal strategies. It is impossible for an AV to make any progress at all if this is not the case, because given this knowledge, every single other road user could dominate them in any conflict - even pedestrians jumping out in front of them for fun as seen in real-world trials [9]. Under these models, it is essential that AVs are programmed with a non-zero probability of deliberately causing a collision. This may be legally difficult, as such programming may be argued to constitute not only manslaughter but also murder, being rationally pre-meditated by the software engineer.

These results are interesting as they suggests that purchasing SUVs, or armoring our autonomous vehicle like an SUV, is a very rational strategy, not in order to better survive the rare collisions that do occur, but to ensure a high probability of other vehicles getting out of our way to save our time and money on delays. By adding armor to our own vehicle we can make the optimal strategy for the other player yielding tend towards certainty at every encounter.This also models what will happen when our AV encounters a pedestrian. The larger cost of collision to the pedestrian than to our AV gives us a strong position, from which we can act aggressively and be confident that the pedestrian will yield. The answer to 'When should the chicken cross then road?' is 'quite rarely if there is a car coming, but with non-zero probability'.

We found that the way in which the models quantize time is important. The turn-taking model artificially removed most of the subtly of game theory by breaking its symmetry via the turn taking mechanism itself. This suggests that such a quantization is not a good model for the real world, it hides the main problem of the scenario from the start. A related modeling issue around time relates to Zeno's Paradox. In the models presented here, time ticks are discrete and of equal length. It might be argued that two Zeno-like players could choose to define each of their ticks to have half the duration of the previous one, and thus create an infinite number of ticks which would be certain to eventually lead to an asymmetric yield and avoidance of collision. Formal mathematical analysis of this claim could form future work, though in practice, any human or machine compute system has some finite limit on its computation speed.

Extended models should handle speed more realistically. Rather than just two discrete speeds, a continuum of speeds should be available, including stopping at a complete halt. Continuous speeds may require sampling approximations to compute over, while complete halts allow for potentially infinite time games which require further consideration to model. Human drivers when faced with, for example, a busy motorway merge, may gradually slow down towards a halt at the end of the slip-road, while drivers in their path may do the same. Perhaps under a continuous speed model this behavior can be shown to converge safely as everyone slows down towards a halt and reduces both the probability and penalty of collisions. Nevertheless, the underlying logic must still hold - that there must be a credible threat of a non-zero probability of causing some collision, in order that the other party cannot take advantage of the AV every time. Future models should add further realistic details to the framework. Real drivers do not know each other's utility functions and must infer them in an information game during the interaction. This could include giving and reading signals about utility such as the model, age, colour and cleanliness of their cars, their lateral positions on the road, their facial expressions and hand gestures as well as more formal car signaling via light flashing and horn usage. Real drivers may not have Markovian time delay utilities and more detailed models should allow for time dimensioned value functions $v(y, x, t)$ rather than than simpler $v(y, x)$ used here. Traffic regulations and conventions such as legally binding and non-binding signs and lights, and the cost of public humiliation or legal action for being seen or recorded breaking them should be added to modify utilities. Such models suggest new signaling conventions for autonomous vehicles, such as use of V2V radio communications and virtual currencies to aid negotiations.

References

[1] JB Bidwell, RS Cataldo, and RM Van House. Chassis and control details of Firebird III. Technical report, SAE Technical Paper, 1959.

[2] Ken Binmore. *Playing for real: a text on game theory*. Oxford university press, 2007.

[3] JD Bolland, MD Hall, D Van Vliet, and LG Willumsen. Saturn: Simulation and assignment of traffic in urban road networks. In *Proceedings of the International Symposium on Traffic Control Systems, Berkeley, Calif. D*, volume 2, pages 99–114, 1979.

[4] Federico Ciliberto and Elie Tamer. Market structure and multiple equilibria in airline markets. *Econometrica*, 77(6):1791–1828, 2009.

[5] Rune Elvik. A review of game-theoretic models of road user behaviour. *Accident Analysis & Prevention*, 62:388–396, 2014.

[6] Shinpei Kato, Eijiro Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.

[7] Changwon Kim and Reza Langari. Game theory based autonomous vehicles operation. *International Journal of Vehicle Design*, 65(4):360–383, 2014.

[8] Hideyuki Kita. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. *Transportation Research Part A: Policy and Practice*, 33(3):305–312, 1999.

[9] Ruth Madigan, Tyron Louw, Marc Dziennus, Tatiana Graindorge, Erik Ortega, Matthieu Graindorge, and Natasha Merat. Acceptance of automated road transport systems (ARTS): an adaptation of the UTAUT model. *Transportation Research Procedia*, 14:2217–2226, 2016.

[10] Fanlin Meng, Jinya su, Cunjia Liu, and Wen-Hua Chen. Dynamic decision making in lane change: Game theory with receding horizon. In *Conference: 11th UKACC International Conference on Control*, 08 2016.

[11] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[12] Thomas R Palfrey. Implementation in Bayesian equilibrium: the multiple equilibrium problem in mechanism design. 1990.

[13] Christos H Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 82–91. Society for Industrial and Applied Mathematics, 2005.

[14] Elaine Rich and Kevin Knight. Artificial intelligence. *McGraw-Hill, New*, 1991.

[15] Mauricio Soto-Alvarez and Petri Honkamaa. Multiple hypotheses data association propagation for robust monocular-based slam algorithms. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6543–6547. IEEE, 2014.

[16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[17] K Vela Velupillai. Uncomputability and undecidability in economic theory. *Applied Mathematics and Computation*, 215(4):1404–1416, 2009.