Heidelberg University
Ziti – Institute of Computer Engineering
Automation Laboratory

TITLE

Master Thesis
submitted by

Elena Elenkova

Matrikelnummer 328 00 12

conducted at
Automation Laboratory,
Heidelberg University

Date: 24th May 2016

Supervisor:	Prof. Dr. sc. techn. Essameddin Badreddin
Second marker:	Prof. Dr. Peter Fischer

Declaration of Authorship

according to the § 16 (2) „Prüfungsordnung der Universität Heidelberg für den Master-Studiengang Angewandte Informatik " from 20 July 2010.

I have written the following thesis completely on my own without using any other sources and help materials other than the ones explicitly specified.

Mannheim, 24th May 2016

Elena Elenkova

Abstract

Acknowledgement

I would like to thank ....

Contents

List of Figures

1 Introduction

1.1 Motivation

1.2 Goals

1.3 State of the art

The first hybrid vehicle was built in 1900 by Ferdinand Porsche. The car was called Lohner-Porsche Mixte Hybrid. It was a serial hybrid car with two wheel motors, it had a 5.5 liter 18kW engine, a battery and an electric generator. The first mass produced hybrid car was the Toyota Prius from 1997 manufactured in Japan.

The first research papers about hybrid vehicles date back from the 1970s. LaFrance and Schult (1973) give an overview of one serial configuration and two parallel configurations - with a single and with a dual motor. They compare the suitability of different electric motor designs for various driving conditions.

Although there has been a lot of research in hybrid vehicles in the last two decades, there is a limited number of approaches which consider the power management problem from a game-theoretical point of view.

First of all, Gielniak and Shen (2004) describe a fuel cell hybrid electric vehicle and solve the power distribution problem as a two-player non-cooperative game. The vehicle has a fuel cell, battery, ultra capacitor and two 35 kW motors. The fuel cell tries to maintain a target 60% State of Charge (SOC) of the Energy Storage Subsystem (ESS) or the battery. Power is firstly taken from the fuel cell and then the rest is taken from the ESS. The voltage of the ESS depends on the SOC and it is also being cooled.

The Ultra Capacitor (UC) is a lumped energy storage device, its SOC depends on the UC voltage and it is also air cooled. The motor has a maximum bus power limit. In addition, there is a transmission model and accessory loads. The accessories such as air conditioner and power steering are powered from the electric motor.

The game-theoretical approach involves two players - all power supplying components as the first player and all power consuming components as the second player. The Master Power Management Controller (MPMC) governs all components in the power-train and calculates the solution. The objective of the game is to save fuel and to accelerate fast. The decisions of the players are how much power to supply at a given moment. The payoff of each player is represented by utility functions - efficiency, performance and composite utilities. The efficiency utility is a function of the strategy of the opponent - how much power they contribute. The performance utility is in this case acceleration. The composite utility is then computed for a time moment after all components have been mapped to an efficiency or performance function. The authors Gielniak and Shen (2004), however, do not provide any insights on how exactly the game-theoretical solution was computed.

Regarding simulation three drive-cycles were examined - Federal Test Procedure (FTP), US06 and Constant 60 Miles Per Hour. A simulation tool called ADVISOR (Burch, Cuddy and Markel, 1999) is used. A comparison is drawn between a Basic control strategy without game theory and a game-theoretical control strategy. In the US06 cycle, the Game Theory control achieves more miles per gallon than the Basic control. For acceleration the Basic control is able to accelerate from 0 to 85 miles in 17.48s, whereas the Game Theory control needs only 11.77s.

A further approach towards power control is presented by Chin and Jafari (2010). The hybrid is based on the Toyota Prius and has a gasoline engine and an electric motor which are the two players in a bimatrix game. The vehicle configuration consists of a Gasoline Engine Controller, Electric Motor Controller and a Power Controller as the main unit for the computation of the solution. On the one hand, the Gasoline Engine Controller manages the fuel and air injection. A three-way catalyst system is utilized for the gas emissions of HC, CO_2, CO and NO_x. On the other hand, the Electric Motor Controller includes also battery and capacitor units. Batteries have high energy capacity but cannot provide much power, whereas capacitors have less energy capacity, but can produce a lot of power. Capacitors help improving battery lifetime and are also useful for providing short bursts of power during acceleration. The Power Controller is responsible for computing the game theory solution. It takes the requested torque from the driver and determines the optimal strategies for the engine and the electric motor. It outputs the torque for each of the two power sources. In addition, it charges the battery when the SOC is low. The goal of the Power Controller is to minimize fuel usage and at the same time maximize torque and reduce gas emissions. This hybrid configuration also has different modes of operation - Engine only, Motor only, Mixed and Battery Charge.

The game-theoretical solution consists of a non-cooperative bimatrix game. The payoff matrices for both players - the engine and the electric motor, are of size $M \times N$. These denote the strategies of player 1 and 2. The game is solved by a Nash Equilibrium (Nash, 1951). When player 1 chooses strategy $i \in M$ and player 2 chooses strategy $j \in N$ the payoff is (a_{ij}, b_{ij}) and it constitutes the Nash Equilibrium if this pair contains the optimal strategies for both players. Pure strategies are extended to mixed by specifying the strategies as a vector of probabilities over all pure strategies. The strategies are how much torque to contribute, measured from 0 to 6000rpm. Each payoff entry in the matrix is a function of fuel consumption, gas emissions, engine temperature, SOC deviation, extra weight and driver's demands. The Nash Equilibrium of the game is computed by the Lemke-Howson algorithm (Lemke and Howson, 1964) in the Power Controller. However, a major weakness is that no simulation results are presented.

Dextreit and Kolmanovsky (2014) use the hybrid configuration of the Jaguar Land Rover Freelancer2. It has a diesel engine, two electric motors and a six-speed dual clutch transmission. The first electric motor is attached to the engine Crankshaft Integrated Starter Generator (CISG), while the second is attached to the rear wheels and is called the Electric Rear Axle Drive (ERAD). There are five driving modes. The EV mode is when only the ERAD provides the torque, the Engine-only mode is when only the engine supplies the torque. The parallel mode means that both the engine and the motors provide the torque. There is a charging mode in which the engine produces the driving torque and the CISC torque. The last mode is a boosting mode, where the engine supplies the additionally needed driving torque.

The game-theoretical approach Dextreit and Kolmanovsky (2014) adopt is a finite-horizon non-cooperative game, where the two players are the driver and the powertrain. The cost function penalizes fuel consumption, NO_x emissions and the battery SOC deviation. In a typical dynamic programming approach the cost is a function of a state vector, a control vector and a vector of operating conditions. The driver chooses the operating conditions - requested wheel speed and wheel torque. The powertrain chooses the control variables and the state vector is the SOC of the battery. The game is solved using a feedback Stackelberg equilibrium (Von Stackelberg, 1952). Firstly, the game is solved statically where the first player is the leader, who maximizes a function $J(w, t)$ by selecting the powertrain operating demands $w(t) \in W$. The second player (the follower) is able to observe the first player's decision and depending on that it selects its control vector $u(t) \in U$. It is assumed that both players make rational decisions. The pair (w^*, u^*) is the Stackelberg equilibrium. Next, the dynamic game is described which consists of $(T-1)$ stages, also called the horizon of the game. Given the initial state vector $x(0)$ the follower chooses their move $w(0)$. Then the follower selects their move $u(0)$. Thus, after the first stage the state is $x(1) = f(x(0), u(0), w(0))$. This continues until the last $(T-1)$ stage of the game. The GT controller Dextreit and Kolmanovsky (2014) implemented computes the state, control and operating values offline in three modules - GT Maps, Mode Arbitration and Mapping to torque demand. In the GT Maps module the wheel torque and speed, gear and battery SOC are discretized. The wheel torque is in the range of 0 to 1000 Nm. The wheel speed is between 0 and 100 rad/s. The battery SOC is between 40% and 70%. As output the module produces two modes to choose from. The Mode Arbitrator chooses one of these, for example the EV or the parallel mode. The aim of the Mapping to torque demand module is to distribute the torque between the engine and the motors.

Dextreit and Kolmanovsky (2014) present a baseline controller solved using Dynamic Programming and compare it with the game theory controller. Measurements are conducted over three drive-cycles for CO_2 emissions, equivalent to fuel consumption, NO_x emissions and deviation of SOC. These show that the GT controller outperforms the baseline controller.

There is another similar approach to the one of Dextreit and Kolmanovsky (2014). The idea is extended by Chen et al. (2014) who describe a single-leader multiple-follower game, where the follower is not only one (the battery) like in Dextreit and Kolmanovsky (2014), but also include other auxiliaries. The vehicle is a heavy-duty truck with an electric refrigerated semi-trailer and a battery. The aim is to manage the battery power and the refrigerated semi-trailer power. The vehicle model consists of the internal combustion engine. The total power which is requested from it is a sum from the driving power and the power for all auxiliaries. The semi-trailer model considers the air transfer from inside and outside and the cooling power. The problem is formulated as minimization of fuel consumption over a time interval. Energy balance is sought so that the battery energy and the air temperature in the semi-trailer at the end of the drive cycle are the same as in the beginning.

The game-theoretic approach assumes a single leader - the driver and multiple followers - all auxiliaries. The leader selects their action $w(t) \in W$ - the demanded wheel torque and speed. Then each follower f_i chooses their action $u_i(t)$. The game is split in two levels. The first level considers N two-player games between the leader and each of the N followers. It solves the games offline with a Stackelberg equilibrium where the driver maximizes and the leader minimizes the cost function. The strategies are stored in a lookup table for each follower. The second stage is computed online and it combines all followers who play their Stakelberg strategies as stored in the lookup table. They all have to reach a mutual decision and this can be solved by a Nash Equilibrium. The central computation happens in the Energy Management System Operator (EMSO). It gathers the actions from all followers and sends them back the computed overall strategy for this stage. This happens until an equilibrium is reached. For simulation Chen et al. (2014) compare two control strategies - optimizing only the battery and optimizing both the battery and the semi-trailer power with the game theory approach. The optimized battery and trailer achieve better fuel consumption by 0.08%.

After exploring the literature, it can be concluded that all game-theoretical approaches assume a non-cooperative game and solve it either by a Nash Equilibrium or by a Stackelberg Equilibrium if the players are regarded as a leader and a follower. Mostly, two players are involved in the game methods proposed. In the leader-follower concept the players are the driver and the powertrain, as described in two of the approaches above. In another approach the two players are divided into the power-consuming and power-supplying devices in the powertrain. In another case Chin and Jafari (2010) described the two players as the electric motor and the engine. The most crucial point to bear in mind is that that no solution approach applies a cooperative game.

1.4 Content

2 Fundamentals

This chapter firstly describes the cooperative game and specifically explains all required mathematical definitions. It then presents six different game-theoretical solution approaches and how they are applied to solve the game.

2.1 Game definition

Firstly, the generic game definition is given. Additional definitions for each of the solution approaches are defined in the corresponding subsections of each solution.

According to Holler and Illing (2006) a game $G = (N, S, u)$ is defined by:

- set of players $N = \{1, 2\}$, in this case two players, where player 1 is the Engine and player 2 is the Motor.
- strategy space S, which is the set of all possible strategy combinations $s = (s_1, s_2)$ for each player, where $s \in S$. The strategy space contains two sets with all strategy combinations of that player.
- utility function $u = (u_1, u_2)$, where $u_i(s)$ for $i = 1, 2$ gives the utility (or also called payoff) for that player when the strategy combination s is played.
- utility space (payoff space) which is the set of all possible utility combinations: $P = \{u(s)|s \in S\} = \{(u_1(s), u_2(s), \forall s \in S\}$

Let us denote the number of pure strategies of each player with m and n. Therefore, these constitute a bimatrix game, meaning that the payoffs of the game can be represented in two matrices of size $m \times n$. Let the payoff matrices A and B denote the payoffs for the first player, the engine, and the second player, the motor respectively, where

$A = (a_{ij} : i \in \{1, ..., m\}, j \in \{1, ..., n\})$ and $B = (b_{ij} : i \in \{1, ..., m\}, j \in \{1, ..., n\})$. It should be noted that $u_1 = a,\ u_2 = b$ and a and b notations have been introduced for simplicity. Sometimes player 1 is called the row player and player 2 is called the column player, since their strategies vary along the rows or along the columns of the matrices. The game is a non-zero sum game, since the sum of the two payoff matrices is not 0, $A + B \neq 0$.

As opposed to a non-cooperative game, in cooperative games the players are allowed to make binding agreements among themselves in order to achieve a better payoff, that is, they can form coalitions. We distinguish between the grand coalition of all players N or $\{1, 2\}$, where they all cooperate, and the individual coalitions $\{i\}, \forall i \in N$ which are $\{1\}$ and $\{2\}$. There is also the empty coalition, where neither cooperates, but this coalition is irrelevant to our purposes. Each coalition has a value associated with it. The grand coalition forms its value as a sum of the payoffs of the engine and motor multiplied element-wise by a matrix with weights. The weights are distributed according to the torque deviation which the engine and motor produce. The torque deviation is defined as the difference between the required and the actual torque at the current time step. When the deviation is 0, the payoffs are weighted by 0.99, when it is between 0-10% of the required torque it is weighted by 0.991, when between 10-20% weight is 0.992 and so on up to 1.0 (the full sum of engine and motor torque).

The goal of the game is to save fuel and to maintain low gas emissions while achieving the required torque at any moment in time. Therefore, the payoffs are penalties as opposed to benefits and they have to be minimized. All of the solutions have been defined by taking into account that this is a minimization problem. Most of the solutions applied in the literature work with maximizing payoffs, but for the purpose of this thesis their definitions and implementations have been modified to minimize the two payoff functions instead.

There exist two types of cooperative games - with transferable and with non-transferable utility. In transferable utility (TU) games the payoff of one player can be transferred to another player without any loss. In contrast, in non-transferable utility (NTU) games the payoffs of each player cannot be redistributed among the other players. In this thesis the payoffs of the engine and the motor are not interchangeable, since decreasing the payoff of one player does not mean increasing the payoff of the other player at the same time. Therefore, their payoff functions are thought to be independent from each other.

The payoff functions are constructed in the following way. The engine payoff is:

$$\begin{aligned} a_{ij} = w_1 \times fuelConsumptionRate + w_2 \times |requiredTorque - actualTorque| + \\ w_3 \times HCemissions + w_4 \times COemissions + w_5 \times NOXemissions + \\ w_6 \times fuelConsumed \end{aligned} \tag{2.1}$$

Where the fuel consumption rate is in grams per second (gps), the difference between required and actual torque is in Newton meters (Nm), the gas emissions are all in gps and the consumed fuel from the beginning of the simulation up to this time step is in liters. The motor payoff is:

$$\begin{aligned} b_{ij} = w_2 \times |requiredTorque - actualTorque| + w_7 \times powerConsumed \\ w_8 \times SOCdeviation \end{aligned} \tag{2.2}$$

Where the consumed power is in kW from the beginning of the simulation and SOC deviation is the difference between the target SOC of 70% and the current SOC of the battery.

To summarize, the game defined is a two-player cooperative bimatrix non-zero-sum game with non-transferable utility.

2.2 Game-theoretical solution approaches

This thesis examines a variety of solution approaches for cooperative games which are later applied and simulated as described. The following section describes the theoretical and mathematical definitions of the six solution concepts. These are Pareto Optimality, Nash Equilibrium, Nash bargaining solution, Kalai-Smorodinsky bargaining solution, the Core and the Shapley value.

2.2.1 Pareto Efficiency and Pareto Optimality

In the literature two terms are often used to refer to the same concept - Pareto Efficiency and Pareto Optimality. However, they are used with distinct meanings throughout this thesis. Pareto Efficiency denotes an allocation of resources such that no player can improve their outcome without impairing another player. In the strategy space of the game there can exist more than one Pareto efficient outcome. Therefore, we define a Pareto optimal outcome, or Pareto Optimality, as the single best outcome from the set of all Pareto efficient outcomes. The criteria for determining the best outcome is as follows. All Pareto efficient points are compared by their torque deviation and the point with the least torque deviation is taken as the Pareto optimal point. Torque deviation is defined as the absolute difference between the required and the actual torque at this stage of the game. If more than one outcomes have the same torque deviation the outcome with the smallest fuel consumption rate is taken as a second criterion. Similarly, if more than one outcome have the same fuel consumption rate the one with the smallest power consumed by the motor is taken. This is the third and last criterion.

2.2.2 Nash Bargaining solution

The Nash Bargaining solution or shortly the Nash solution was defined by Nash (1950b). It considers the game as a bargaining game where each player requests some portion of the existing good, in this case torque. A bargaining game G is defined by specifying a conflict point $c = (c_1, c_2)$ in addition to the payoff space P, comprising of the two payoff vectors as defined above $u = (u_1, u_2)$. The pair is denoted as (P, c) as in Holler and Illing (2006). The conflict point is the point where both players do not cooperate with each other. There exists a common misinterpretation because the word "conflict" point itself implies a point where both players face a disagreement and hence their payoffs are the worst. A Nash Equilibrium does not present such a worst case and therefore the term conflict point may seem misleading, but for the sake of consistency in the literature, the term will be kept as it is. Since the conflict point in the literature is often taken to be the Nash Equilibrium of the corresponding non-cooperative game, the next subsection deals with an algorithm to find a Nash Equilibrium.

Taking these into account, a Nash Bargaining solution is the vector u^* from the set P:

$$NP^* = max(c_1 - u_1^*)(c_2 - u_2^*) \tag{2.3}$$

so that $u^* = (u_1^*, u_2^*) \in P$ and $u_i^* < c_i$ for $i = 1, 2$, meaning that the Nash solution point u^* payoff is less than the conflict point payoff. The main idea of the Nash solution expressed graphically as in Figure 2.1 is to maximize the product of the differences between the conflict point and the Nash solution point coordinates in 2D.

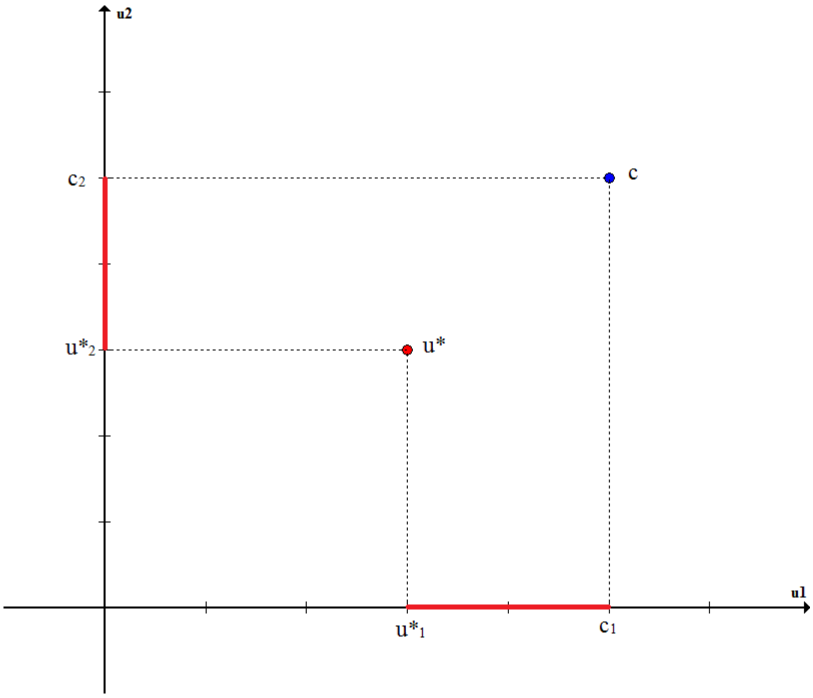

Figure 2.1.: *Maximized Nash Product*

The Nash solution is characterized by four axioms as defined in Holler and Illing (2006).

The first axiom is called Invariance with Respect to Affine Transformations of Utility. Given a bargaining game (P, c) and two random real numbers $a_i > 0$ and b_i, where $i = 1, 2$ for both players, it is true that $f_i(P', c') = a_i f_i(P, c) + b_i$. This holds if (P', c') is a game resulting from a linear transformation which preserves the order of all points u and c from P so that $y_i = a_i x_i + b_i$ and $c'_i = a_i c_i + b_i$, where $y \in P'$ and $c' \in P'$. In other words such a linear transformation of the game space does not affect the solution of the bargaining game.

The second axiom is Symmetry. If (P, c) is a symmetric bargaining game then $f_1(P, c) = f_2(P, c)$. A game is symmetric if $c_1 = c_2$ and if (u_1, u_2) and (u_2, u_1) are both in the payoff space P. This results in a conflict point lying on the 45°line from the origin of the coordinate system. P is symmetric with regard to this line. This axiom tells that if the payoffs of the two players can be interchanged and the game does not change as a consequence of that, then its solution should also not distinguish between the two players.

The third axiom is Independence of Irrelevant Alternatives. It states that $f(P, c) = f(Q, c)$ if two games (P, c) and (Q, c) have the same conflict point c, $P \subseteq Q$ and $f(Q, c) \in P$. Therefore, only the conflict point is relevant, because if the payoff space of the game is extended to a superset or it is shrunk to a subset, then the solution itself does not change.

The last axiom is the Pareto Optimality. For a bargaining game (P, c) there is no $x \neq f(P, c)$ in P such that $x_1 \leq f_1(P, c)$ and $x_2 \leq f_2(P, c)$, which is an example of group rationality.

2.2.3 Nash Equilibrium and Lemke-Howson algorithm

This subsection firstly describes formally the concept of Nash Equilibrium and then presents the Lemke-Howson algorithm for finding one Nash Equilibrium.

The notion of an equilibrium was firstly presented by Nash (1950a) in his one-page paper where he defined an equilibrium as a self-countering n-tuple of strategies. An n-tuple is said to counter another n-tuple if the strategy of each player in the first n-tuple gives him the largest possible payoff against the n-1 strategies of the other players in the second n-tuple (Nash, 1950a).

Let s^* be a strategy combination where each player chooses an optimal strategy s_i^* assuming that all other players have also chosen their optimal strategies. A strategy of a player i is called optimal when i cannot achieve a better payoff given the current decisions (strategies) of the other players denoted as $-i$. Then, a Nash Equilibrium is formally defined as a strategy combination such that (Holler and Illing, 2006):

$$u_i(s_i^*, s_{-i}^*) \leq u_i(s_i, s_{-i}^*), \ \forall i, \forall s_i \in S_i \tag{2.4}$$

Therefore, in an Nash Equilibrium state no player has the incentive to deviate from his strategy assuming that all other players also keep their strategies unaltered. Each player has to guess which strategy his opponent will play and based on that to choose his own best strategy. Nevertheless, there can be several best responses and this means several Nash Equilibria. Nash (1950a) classifies the equilibria into three types - solutions, strong solutions and sub-solutions. It is possible that a non-cooperative game does not have a solution. However, if it has, then the solution must be unique which is what a strong solution is. A sub-solution is not necessarily unique. Holler and Illing (2006) present a theorem for the existence of a Nash Equilibrium in a game

$G(N, S, u)$ with the following characteristics. If the strategy space is compact and convex $S_i \subset R^m, \forall i \in N$ and it is true that $u_i(s)$ is continuous and bounded in $s \in S$ and quasi-concave in s_i, then there exists a Nash Equilibrium. However, a lot of functions do not fulfil these prerequisites for continuity and quasi-concavity and hence have no Nash Equilibrium in pure strategies. In this case the game can be extended to mixed strategies by assigning probabilities to each pure strategy or also called randomization. By randomizing the strategies of matrix games the strategy space S_i can be transformed to a convex and compact space and the payoff $u_i(s)$ to quasi-concave.

One of the most popular algorithms for finding a Nash Equilibrium for bimatrix non-zero-sum games is the Lemke-Howson algorithm (Lemke and Howson, 1964). The Matlab implementation developed by Katzwer (2014) was utilized to find one Nash Equilibrium in mixed strategies. This function contains a parameter, which affects the final result of the algorithm. Changing the parameter yields different Nash equilibria as output. This parameter k is the initial pivot, a number between 1 and $n+m$, where n and m are the number of strategies of player 1 and player 2 respectively. The general idea of the the algorithm is that it works on two graphs containing nodes and edges, one graph per player. It starts at the (0,0) point. It selects a k, the pivot, or the label of the graph, containing the strategy to be dropped first when traversing the graph. From there a path to the end is followed in order to find a Nash Equilibrium.

There a number of reasons for the different Nash Equilibrium solutions that the algorithm produces. According to Lemke and Howson (1964) there exist an odd number of Nash Equilibria in any non-degenerate game. A non-degenerate game is a game where no mixed strategy with a support of size k has more than k pure strategies (Nisan et al., 2007). Moreover, a support of a mixed strategy is defined as the set of pure strategies with positive probabilities. Since there is an odd number of Nash Equilibria, there must be at least one Equilibrium, which proves that the algorithm will always find one solution in mixed strategies. However, which of all Nash Equilibria is found depends on which strategy label is dropped first. The initial pivot label which the algorithm drops can belong to either of the two players and be any of their n or m strategies.

As discussed in Shapley (1974) the Lemke-Howson algorithm possesses a significant weakness, namely that it is neither guaranteed that the algorithm will find all possible solutions, nor that it will tell if there are any unfound solutions.

The fundamentals of the Lemke-Howson algorithm are described next. Let us assume a scenario of a 2-player bimatrix game as the one used throughout this thesis, where the players 1 and 2 each have n and m number of pure strategies and their payoffs are in the matrices $A = (a_{ij} : i \in \{1, ..., n\}, j \in \{n+1, ..., n+m\})$ and $B = (b_{ij} : i \in \{1, ..., n\}, j \in \{n+1, ..., n+m\})$ respectively. The mixed strategies are the vectors $s = (s_1, s_2 ..., s_n)$ and $t = (t_{n+1}, t_{n+2}, ..., t_{n+m})$ where $S = \{s \geq 0; \sum_{i=1}^{n} s_i = 1\}$ and $T = \{t \geq 0; \sum_{j=n+1}^{n+m} t_i = 1\}$ are the sets for the mixed strategies spaces. Then, the payoff for player 1 is $\sum_{i=1}^{n} \sum_{j=n+1}^{n+m} a_{ij} s_i t_j$ and the payoff for player 2 is $\sum_{i=1}^{n} \sum_{j=n+1}^{n+m} b_{ij} s_i t_j$. An equilibrium is a pair of strategies (s^*, t^*) satisfying:

$$\sum_{i=1}^{n} \sum_{j=n+1}^{n+m} a_{ij} s_i^* t_j^* = \max_{s \in S} \sum_{i=1}^{n} \sum_{j=n+1}^{n+m} a_{ij} s_i t_j^* \tag{2.5}$$

$$\sum_{i=1}^{n} \sum_{j=n+1}^{n+m} b_{ij} s_i^* t_j^* = \max_{t \in T} \sum_{i=1}^{n} \sum_{j=n+1}^{n+m} b_{ij} s_i^* t_j \tag{2.6}$$

If we also define the sets:

$$\tilde{S} = S \cup \{s \geq 0 : \sum_{i=1}^{n} s_i \leq 1 \text{ and } \prod_{i=1}^{n} s_i = 0\} \tag{2.7}$$

$$\tilde{T} = T \cup \{t \geq 0 : \sum_{j=n}^{n+m} t_i \leq 1 \text{ and } \prod_{j=n}^{n+m} t_i = 0\} \tag{2.8}$$

this allows us to define closed convex polyhedral regions S^i and S^j which together form $S^k \in \tilde{S}$:

$$S^i = \{s \in \tilde{S} : s_i = 0\} \text{ for } i \in \{1, ..., n\} \tag{2.9}$$

$$S^j = \{s \in S : \sum_{i=1}^{n} b_{ij} s_i = \max_{l \in \{n+1, ..., n+m\}} \sum_{i=1}^{n} b_{il} s_i\} \text{ for } j \in \{n+1, n+2, ..., m\} \tag{2.10}$$

S^i contains all $\tilde{S} - S$ and S^j contains the mixed strategies for player 1 and the pure strategy j of player 2 which is his best outcome. S^i and S^j both make S^k and cover the whole set $\tilde{S}$. The same can be applied to define the regions $T^k \in \tilde{T}$:

$$T^i = \{t \in T : \sum_{j=n+1}^{n+m} a_{ij} t_j = \max_{l \in \{1,...,n\}} \sum_{j=n+1}^{n+m} a_{lj} t_j\} \text{ for } i \in \{1, ..., n\} \tag{2.11}$$

$$T^j = \{t \in \tilde{T} : t_j = 0\} \text{ for } j \in \{n+1, ..., n+m\} \tag{2.12}$$

The Lemke-Howson algorithm represents the strategies of both players in two graphs with nodes and edges. The already mentioned definitions are required in order to define a labelling for the graphs. A labelling of a node consists of all of the labels of all surrounding regions of this node. Let the nonempty label of $s \in \tilde{S}$ be $L'(s) = \{k : s \in S^k\}$ and similarly the nonempty label of $t \in \tilde{T}$ be $L''(t) = \{k : t \in T^k\}$ and the label of the node pair with pure strategies $(s,t) \in \tilde{S} \times \tilde{T}$ be $L(s,t) = L'(s) \cup L''(t)$. A node pair (s,t) is completely labelled whenever $L(s,t) = K$, meaning that it contains the labels for all regions $k \in K$. A node pair is almost completely labelled if $L(s,t) = K - \{k\}$ for some $k \in K$.

Then, as in Shapley (1974) a node pair $(s,t) \in S \times T$ is an equilibrium point of (A,B) if and only if (s,t) is completely labelled. Let the two graphs be $G' \in \tilde{S}$ and $G'' \in \tilde{T}$. Two nodes are adjacent if they are on the two ends of the same edge which means that their labels differ in exactly one element. The set of k almost completely labelled nodes in the graphs and their edges are the disjoint paths of the graph and cycles. The equilibria of the game are always located at the end of these paths. Also, the starting node, by default (0,0) is called an artificial equilibria and it is also located at the end of a path.

The algorithm works by starting at $(s,t) = (0,0) \in G' \times G''$. Then a label k, which is to be dropped from (s,t), is chosen (initial pivot). This k can belong to either n or m. Let this node be (s,t) and its new label (after dropping k) be l. Then, if $l = k$ a Nash Equilibrium is reached. If $l \neq k$, then the algorithm continues by dropping another label and continuing until a completely labelled node has been reached, which is an equilibrium point. The starting label to drop is the pivot parameter as in the Matlab implementation. Since there are $n + m$ labels (strategies) which can be dropped first and they can end up in a different Nash Equilibrium node in the graph, it is inefficient to experiment with all different starting strategies. Experiments were done with 1,

$\frac{n+m}{4}$, $\frac{n+m}{2}$, $\frac{3(n+m)}{4}$ and $n+m$ as the first strategy to be dropped. In order to choose which point to use, the Matlab implementation NPG from Chatterjee (2010) was used the Lemke-Howson solution with the k strategy, which was closest to the NPG solution was chosen. This k was $\frac{n+m}{2}$, which was passed to the Lemke-Howson algorithm as an input.

2.2.4 Kalai-Smorodinsky Bargaining Solution

The Nash Bargaining solution has the significant disadvantage that it is not monotonic. The third of the four axioms defined above Independence of Irrelevant Alternatives was criticized as in Kalai and Smorodinsky (1975). Therefore, Kalai and Smorodinsky (1975) propose another unique bargaining solution called the Kalai-Smorodinsky solution. They replace the third axiom with another axiom for Individual Monotonicity, which the Nash solution does not conform to.

The axiom states that if for two bargaining games (P, c) and (R, c) such that $P \subset R$ the equation $m_i(P) = m_i(R)$ holds for player i, then for player $j \neq i$ it is true that $f_j(R, c) \geq f_j(P, c)$.

$m_i(P, c)$ and $m_i(R, c)$ denote the minimum payoffs for players $i = 1, 2$. They are defined as $m_i(P) = min(u_i|(u_1, u_2) \in P$. The point $m(P) = (m_1(P), m_2(P))$ is called the ideal point of the game, because that is where both players achieve minimum payoffs; hence, the ideal outcome of the game. However, often this point is not feasible. As in the Nash solution, the conflict point is also crucial in the Kalai-Smorodinsky solution. A further term is required and this is the Utility Boundary of the utility space P of a game:

Consequently, the Kalai-Smorodinsky solution is defined as follows:

$$\frac{c_2 - u_2}{c_1 - u_1} = \frac{c_2 - m_2}{c_1 - m_1} \tag{2.13}$$

so that

$$u_i \leq v_i, \frac{c_2 - v_2}{c_1 - v_1} = \frac{c_2 - m_2}{c_1 - m_1} \tag{2.14}$$

where $(v_1, v_2) \in P$. The pair (u_1, u_2) is the Kalai-Smorodinsky solution. As Kalai and Smorodinsky (1975) point out, this solution is graphically represented as the intersection point of the line $L(c, m)$ and the utility (payoff) boundary $H(P)$ as shown in Figure 2.2. $L(c, m)$ is the line connecting the conflict point c and the ideal point m. The utility boundary $H(P)$ or also called the Pareto frontier is defined as the set of all Pareto efficient points. For all random payoffs of one player the boundary gives the minimal possible payoff of the other player. If the Pareto frontier only contains one Pareto efficient point then the Kalai-Smorodinsky solution is the ideal point itself $u* = m(P)$.

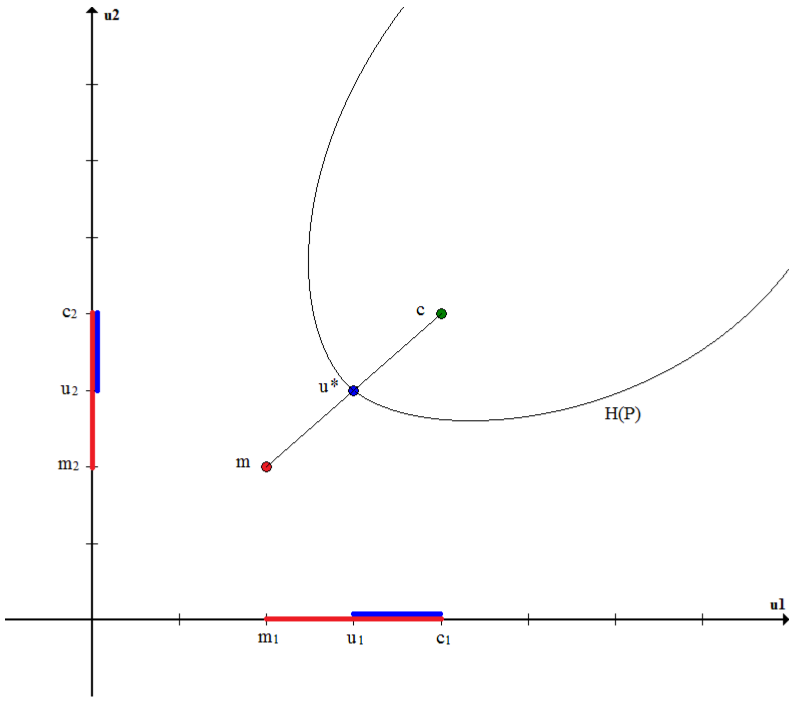

Figure 2.2.: *Kalai-Smorodinsky Solution*

2.2.5 Core

As opposed to the previous four solution approaches, which treated the game as individually cooperative, the next two approaches the Core and the Shapley value regard the game as a coalitional game. The difference is that the previous approaches were applicable to 2-player individually cooperative games only, whereas the Core and the Shapley value can additionally be extended to coalitional n-player games. However, this thesis only deals with 2-player games and therefore does not need this extension. Moreover, there is a crucial distinction between the Core and the Shapley value because the Core produces a set of points as a solution, while the Shapley value is always only one single point. For this reason the Core is similar to the Pareto efficiency concept because both can contain more than one point. A single point from this set needs to be chosen and the criteria for choosing a single solution from the Core is the same as in the case of Pareto efficiency - torque deviation, fuel consumption rate and power consumed by the motor are taken into account.

To define the core of a game, a new term called imputation is introduced. An imputation is such an allocation of resources that is both individually rational and group rational (or Pareto efficient). For transferable utility (TU) games individual rationality means that each player in the coalition receives at most the payoff that they would receive on their own expressed as $u_i \leq v(\{i\})$. Group rationality, which is equivalent to efficiency and in our case also to Pareto efficiency, means that the sum of the payoffs of all players in the grand coalition is the value of the game, $\sum_{i=1}^{N} u_i = v(N)$. For non-transferable utility (NTU) games such as ours a payoff vector u is an imputation when there is no vector u' which strictly dominates it. This is expressed by $u' \in V(N)$ such that $u'_i < u_i, \forall i \in N$. According to Holler and Illing (2006) a vector u' is said to dominate another vector u with regard to the coalition K if $u'_i \leq u_i, \forall i \in K$ and for at least one $i \in K$ the inequality $u'_i < u_i$ holds when $u' \in V(K), u \in V(K)$.

After describing the notion of imputation, the core C of a game G is the defined as the set of all non-dominated imputations:

$$C(G) = \{u \mid v(K) - \sum u_i \leq 0,\ \forall i \in K,\ \forall K \in N\} \tag{2.15}$$

If an imputation x is in the core $x \in C(G)$ then for all coalitions $K \in N$ there is no other coalition K for which another imputation y dominates x denoted as y *dom* x *via* K (Holler and Illing, 2006). Therefore, no coalition can make its participants better by substituting x with y; hence, x is thought to be coalitionally rational. There are two different possibilities when this can be true. There is either no coalition at all which can realize y, or y is worse than x meaning that it is not true that $y_i < x_i$ for at least one $i \in K$ and $y_i \leq x_i$ for all $i \in K$.

Since all imputations in the core are not dominated by any other imputation, the core is internally stable. The core of a game can be empty or can contain many imputations. This comes from the fact that the relation *dom* is not transitive; hence, if it is true that an imputation x dominates y and y dominates z, this does not imply that x dominates z.

2.2.6 Shapley Value

The Shapley value of a game with transferable utility according to Holler and Illing (2006) is defined as:

$$\Phi_i(v) = \sum_{i \in K, K \subset N} \frac{(k-1)!(n-k)!}{n!}[v(K) - v(K - \{i\})], \tag{2.16}$$

$$\sum \Phi_i(v) = v(N),\ \Phi(v) = (\Phi_i(v)) \tag{2.17}$$

k denotes the number of players in coalition K and n is the total number of players. $v(K)$ is the value of coalition K, whereas $v(K - \{i\})$ is the value of coalition K without the player i. The sum of the Shapley values of all players must equal the total value of the game, which is also the value of the grand coalition N.

The Shapley value was introduced by Shapley (1952) where it is described that the game must be represented by its characteristic function $v : 2^N \to \mathbb{R}$. In addition, the Shapley value is the single point which satisfies the following three axioms.

The first axiom is Group Rationality or Efficiency, which is equivalent to Pareto efficiency. This axiom was already described in the subsection of Nash bargaining solution. The second axiom is Symmetry and it states that if the players change their order then the values of the players change accordingly. Therefore, the number, or the i, of the player does not have an influence on the value which that player receives. The third axiom is based on the assumption $\Phi_i(v + w) = \Phi_i(v) + \Phi_i(w)$ where $v + w$ is the combination of two games v and w.

In general the Shapley value is about permutations of the players and especially which player arrives first, meaning which player is able to make the decision first. The order of the players is irrelevant and all of the various orders have the same probability. Since there are $n!$ permutations of the players, the probability of each of them is $\frac{1}{n!}$ as shown in Equation 2.16. Additionally, the probability of player i being at the k-th place is $\frac{(k-1)!(n-k)!}{n!}$. On the one hand, a player who has an influence on the value of the coalition is called a crucial or pivot player. A player i is a pivot player whenever K is a winning coalition and without him the coalition $K - \{i\}$ is a losing coalition. On the other hand, a player is called a dummy player when a coalition K has the same value regardless whether i is participating in it or not - $[v(K) - v(K - \{i\})] = 0$.

3 Application

4 Simulation

5 Implementation

6 Experiments

7 Discussion

Bibliography

Burch, Steve, Matt Cuddy and T Markel (1999). 'ADVISOR 2.1 Documentation'. In: *National Renewable Energy Laboratory*.

Chatterjee, Bapi (2010). *n-person game*. URL: http://www.mathworks.com/matlabcentral/fileexchange/27837-n-person-game/content/npg/npg.m (visited on 12/05/2016).

Chen, H et al. (2014). 'Game-theoretic approach for complete vehicle energy management'. In: *Vehicle Power and Propulsion Conference (VPPC), 2014 IEEE*. IEEE, pp. 1–6.

Chin, Hubert H and Ayat A Jafari (2010). 'Design of power controller for hybrid vehicle'. In: *System Theory (SSST), 2010 42nd Southeastern Symposium on*. IEEE, pp. 165–170.

Dextreit, Clement and Ilya V Kolmanovsky (2014). 'Game theory controller for hybrid electric vehicles'. In: *Control Systems Technology, IEEE Transactions on* 22.2, pp. 652–663.

Gielniak, Michael J and Z John Shen (2004). 'Power management strategy based on game theory for fuel cell hybrid electric vehicles'. In: *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*. Vol. 6. IEEE, pp. 4422–4426.

Holler, Manfred J and Gerhard Illing (2006). *EinfÃ¼hrung in die Spieltheorie*. Springer-Verlag.

Kalai, Ehud and Meir Smorodinsky (1975). 'Other solutions to Nash's bargaining problem'. In: *Econometrica: Journal of the Econometric Society*, pp. 513–518.

Katzwer, Richard (2014). *Lemke-Howson Algorithm for 2-Player Games*. URL: http://www.mathworks.com/matlabcentral/fileexchange/44279-lemke-howson-algorithm-for-2-player-games (visited on 08/03/2016).

LaFrance, RC and RW Schult (1973). 'Electrical systems for hybrid vehicles'. In: *Vehicular Technology, IEEE Transactions on* 22.1, pp. 13–19.

Lemke, Carlton E and Joseph T Howson Jr (1964). 'Equilibrium points of bimatrix games'. In: *Journal of the Society for Industrial and Applied Mathematics* 12.2, pp. 413–423.

Nash, John F (1950a). 'Equilibrium points in n-person games'. In: *Proceedings of the National Academy of Sciences of the United States of America* 36, pp. 48–49.

– (1950b). 'The bargaining problem'. In: *Econometrica: Journal of the Econometric Society*, pp. 155–162.

– (1951). 'Non-cooperative games'. In: *Annals of mathematics*, pp. 286–295.

Nisan, Noam et al. (2007). *Algorithmic game theory*. Vol. 1. Cambridge University Press Cambridge.

Shapley, Lloyd S (1952). *A value for n-person games*. Tech. rep. DTIC Document.

– (1974). *A note on the Lemke-Howson algorithm*. Springer.

Von Stackelberg, Heinrich (1952). *The theory of the market economy*. Oxford University Press.

A Appendix A