The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Realizing Autonomous Valet Parking with Automotive Grade Sensors

Prasanth Jeevan, Frank Harchut, Bernhard Mueller-Bessler, and Burkhard Huhnke

Abstract— **The availability of several Advanced Driver Assistance Systems has put a correspondingly large number of inexpensive, yet capable sensors on production vehicles. By combining this reality with expertise from the DARPA Grand and Urban Challenges in building autonomous driving platforms, we were able to design and develop an Autonomous Valet Parking (AVP) system on a 2006 Volkwagen Passat Wagon TDI using automotive grade sensors. AVP provides the driver with both convenience and safety benefits - the driver can leave the vehicle at the entrance of a parking garage, allowing the vehicle to navigate the structure, find a spot, and park itself. By leveraging existing software modules from the DARPA Urban Challenge, our efforts focused on developing a parking spot detector, a localization system that did not use GPS, and a back-in parking planner. This paper focuses on describing the design and development of the last two modules.**

I. INTRODUCTION

This decade has born witness to several achievements in research in the area of autonomous driving. The 2005 DARPA Grand Challenge and 2007 DARPA Urban Challenge competitions confronted entrants with diverse, yet realistic driving scenarios in both the desert and urban environments, respectively. The vehicles developed by Volkswagen jointly with Stanford, namely Stanley [1] and Junior [2], were among the participants in these races. In parallel, Volkwagen's autonomous driving research has continued in other areas including: vehicle dynamics evaluation in 2003 with the *Steering Robot* [3], reproducible driving at the vehicle dynamics limit in 2006 with the *GTI 53+1* [4], vehicle following also in 2006 with the *Twin Car* [5], and highway driving in 2008 with the *iCar* [6].

With regards to production vehicles, the list of Advanced Driver Assistance Systems (ADAS) available to the consumer continues to grow. Closely following this trend is a corresponding increase in sensing modalities available in these vehicles. Currently, the breadth of the sensor suite includes cameras, radars, ultrasonic sensors, lidars, GPS and others. Admittedly, the stock car versions of these sensors cannot compare in performance to those used in, for example, Junior, though they are still quite capable. Realizing this fact, the latest autonomous vehicle built at Volkswagen Group of America's Electronics Research Lab, Junior 3, couples the lessons learnt from its predecessors along with a reduced sensor suite consisting of automotive grade sensors. It is built on the same 2006 Volkswagen Passat Wagon TDI platform as earlier Juniors, with similar computer systems and a modular software framework [1], [2].

In this paper we describe the key components in the design and development of an Autonomous Valet Parking (AVP) system on the Junior 3 platform. Section II will describe the AVP system. Section III will detail the new localization system developed for use in the AVP environment. Section IV will describe the additional parking planner that works in concert with the existing general planner developed for the DARPA Urban Challenge. Sections V and VI will conclude.

II. AUTONOMOUS VALET PARKING SYSTEM

The benefit to the driver from an ADAS can be divided into three areas: convenience, fun, and safety. By taking drivers out of the loop, autonomous driving systems typically maximize the convenience and safety aspects. The AVP system provides exactly these benefits to the driver by allowing him/her to leave the vehicle at the entrance of the parking garage. The vehicle, when switched into AVP mode by the driver will enter and navigate the parking garage autonomously, search for a free spot, and initiate a back-in parking maneuver upon finding one.

The system architecture represents a baseline version with the aim to determine feasibility. Analysis of the baseline version gives valuable insight into where increased complexity is actually needed to achieve better performance in future versions. Complexity leads to increased cost, development time, and risk of failure among other undesirable outcomes. As a consequence, we made the following design choices. The sensor suite in AVP is limited to automotive grade sensors consisting of stock wheel tic sensors, a stock forward facing camera, and two side scanning automotive grade lasers as shown in Fig. 2. A digital map informs the vehicle of the search path, landmark locations for the camera algorithm, and the location and dimensions of all parking spots in the parking garage. The garage itself is assumed to be without humans or other mobile vehicles. In addition to a controlled environment, algorithm development focused on deterministic approaches.

In developing software on the Junior 3 platform, we leverage the existing modular software framework used by Stanley and Junior in the DARPA Challenges. In this framework, software modules run concurrently across multiple CPUs and communicate with one another through an asynchronous, publish/subscribe message passing protocol implemented using the Inter Process Communications Toolkit [7]. The modular framework allows us to reuse a majority of the software developed for the Urban Challenge with minor modifications,

This work was supported by Volkswagen Group of America Electronics Research Lab.

P. Jeevan, B. Mueller-Bessler, and B. Huhnke are with Volkswagen Group of America Electronics Research Lab, 4005 Miranda Ave., Suite 100, Palo Alto, CA 94304. {Prasanth.Jeevan, Bernhard.Mueller, Burkhard.Huhnke}@vw.com

F. Harchut is with Volkswagen AG Group Research, Brieffach 17770, Wolfsburg, Germany 38436 Frank.Harchut@volkswagen.de

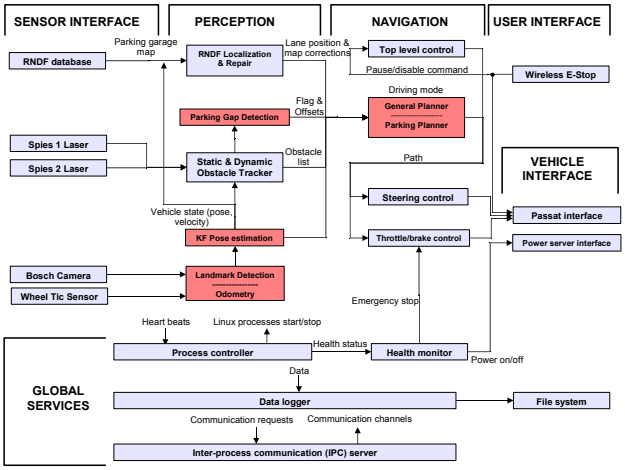

Fig. 1. Major software modules and interconnections in Junior 3. Additions and modifications specific to the AVP system are highlighted in red.

Fig. 2. Locations of automotive grade sensors used in the AVP system.

while easily incorporating new modules specific to the AVP system (See Fig. 1). The design choices allow us to focus on the localization, parking path planning, and parking spot detection aspects for the initial version of the system. These additional modules are described in the following sections. The parking spot detection module is described in [8].

III. Localization

For AVP, Junior 3 requires an always available localization system that functions within a parking garage using automotive grade sensors. Our solution combines the absolute pose output of a camera-based landmark detection system and the relative output of odometry through a Kalman filter. In the Kalman filter, odometry generates the prediction, while the camera-based landmark detection provides the observation update. The system is initialized by the observation of a landmark from the start position, e.g. at the entrance of the parking garage. We describe the two components as well as the calibration procedure for each.

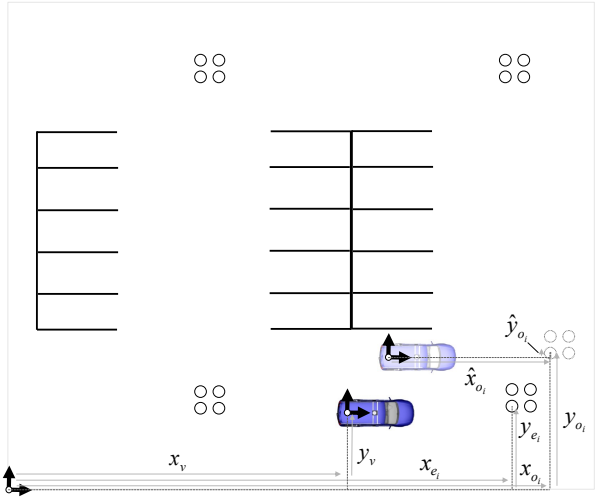

Fig. 3. The camera algorithm provides an observation update (opaque) upon registering a detected landmark with the digital map.

A. Camera-based Landmark Detection

Junior 3 uses an automotive grade camera as the sensor for its absolute positioning system. It provides 640 by 480 12-bit grayscale images at 25 Hz. The camera is forward-facing and mounted near the rearview mirror. Typically production vehicles use cameras in this location to warn the driver if the vehicle is approaching the lane edge, for traffic sign recognition, for automatically controlling the high beam, and other applications.

Junior 3 uses the camera to detect the relative position of landmarks in the parking lot (See Fig. 3). Artificial landmarks on the road surface are used instead of ”natural” landmarks in the camera algorithm in order to increase the applicability of the system to existing parking garages, while introducing only minimal additional cost to the parking garage operation with regards to initial application and maintenaince. The landmarks are sets of four circular markers in a rectangular configuration, laid flat on the road surface, and covered in a retro-reflective material for high contrast. The markers' locations are georeferenced offline and known to Junior 3 via a map file before it enters the parking structure. By determining its own pose relative to a landmark, Junior 3 can localize itself within the parking structure.

For detecting the markers, which appear as ellipses on the camera's image plane, we use OpenCV, an open source computer vision library. As a first step, we use the morphological operators dilate and subtract to create an intermediary image as shown in Fig. 4. In this image, the solid ellipses become elliptical rings whose brightness represents the difference in brightness of the marker against the surrounding road in the original image. We chose a suitable threshold based on the expected edge contrast to create a binary image for the OpenCV ellipse detector.

Before we relate the detected markers to those in the map, their pixel coordinates must be converted to physical coordinates in Junior 3's coordinate frame. This relation is

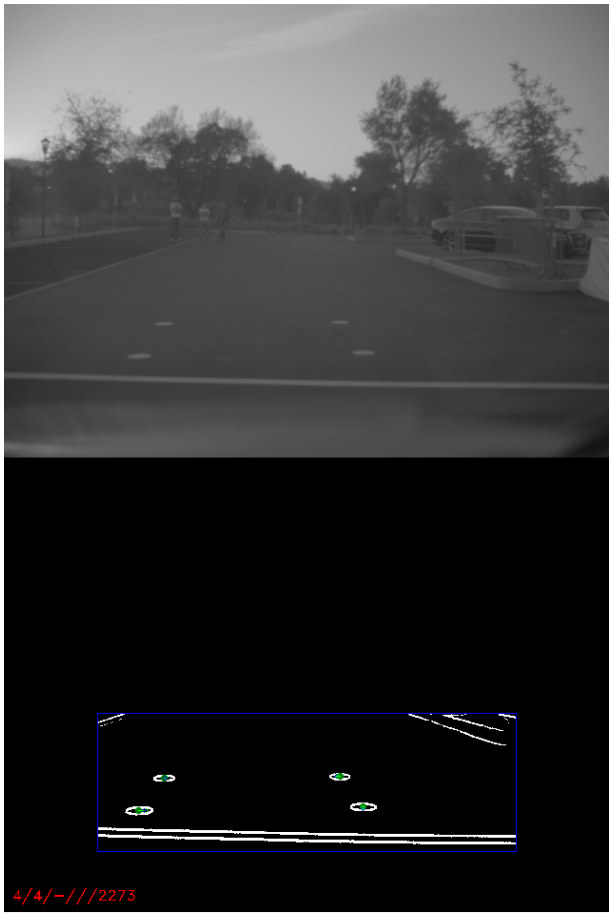

Fig. 4. The raw camera image (above) is pre-processed with morphological operators (below). An ellipse detector recognizes markers from this image.

known as a homography (see Fig. 5), an invertible transformation in which points lying on the same planar surface in the scene can be related to image coordinates of a camera viewing that surface. It is expected that irregularities in the road surface will lead to errors in the estimation of relative marker position. Additionally, pitch, roll and height changes of Junior 3 will induce errors in estimation of the markers' positions in Junior 3's coordinate frame. Camera nonlinearities or distortions, which can be mostly attributed to the lens, can be sufficiently overcome by calibration techniques. In typical parking lot environments we found that the magnitude of the transformation errors to be acceptable for the task of navigating a parking lot and parking in a parking spot.

Solving for the vehicle pose amounts to relating the observed marker coordinates in the vehicle centered coordinate frame to the corresponding expected markers in the parking lot coordinate frame given by the map. Fig. 3 provides an illustration of this relationship. The relation between the ith observed marker with coordinates (x_{o_i}, y_{o_i}) and the corresponding expected marker (x_{e_i}, y_{e_i}) is:

$$x_{e_i} = x_{o_i} + x_v, \quad (1)$$

$$y_{e_i} = y_{o_i} + y_v, \quad (2)$$

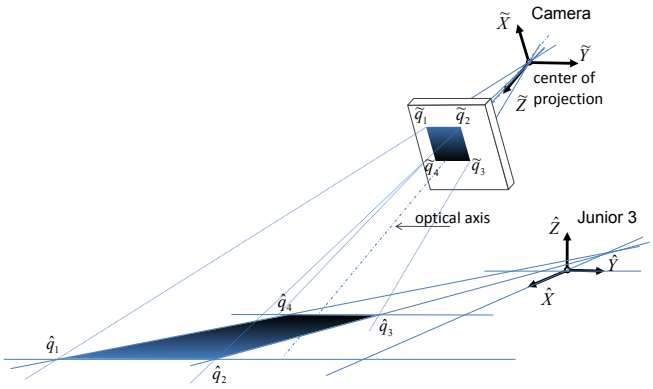

Fig. 5. A homography matrix is used to map pixel coordinates on the image plane to road surface coordinates relative to Junior 3.

where,

$$x_{o_i} = [\cos(\psi_v) * \hat{x}_{o_i}] - [\sin(\psi_v) * \hat{y}_{o_i}], \quad (3)$$

$$y_{o_i} = [\sin(\psi_v) * \hat{x}_{o_i}] + [\cos(\psi_v) * \hat{y}_{o_i}]. \quad (4)$$

The two observed marker coordinates, $(\hat{x}_{o_i}, \hat{y}_{o_i})$ and (x_{o_i}, y_{o_i}), are related by a rotation from the vehicle centered coordinate frame to a parking lot coordinate frame. Thus, x_v, y_v, and ψ_v are the parameters to be solved, corresponding to the Junior 3's position and heading, respectively.

We developed an algorithm based on the linear least squares (LLS) approach to provide a vehicle pose solution that minimized the distances between observed and expected marker positions. To make the problem tractable for LLS we needed to remove two sources of nonlinearity. The first was the vehicle heading. We limited the LLS algorithm to estimate only the vehicle's position based on an array of possible vehicle headings. This allowed for the separation of the coordinate frame rotation in (3) from the LLS minimization. The array of vehicle headings was limited to an interval of $\pm 5^\circ$ centered around the previous heading estimate with a granularity of 0.05°. In practice, the actual error in vehicle heading never exceeded these limits so the limited interval size was acceptable.

The second source of nonlinearity was the $L2$ or $Euclidean$ distance metric defining the distance between observed and expected markers. In our approach, the LLS algorithm chooses a vehicle position minimizing the $L1$ or $Manhattan$ distance between observed and expected markers instead of the optimal $L2$ distance. This led to the following expression for the residual:

$$R(x_v, y_v) = \sum_{i=0}^{n} [x_{e_i} - x_{o_i} - x_v]^2 + [y_{e_i} - y_{o_i} - y_v]^2, \quad (5)$$

and taking partial derivatives gives us the optimal vehicle

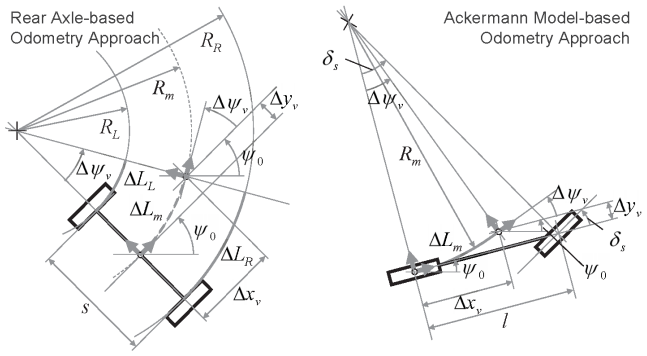

Fig. 6. The underlying vehicle models used in the odometry approaches.

position in the $L1$-sense:

$$x_v = \frac{1}{n}\sum_{i=0}^{n} x_{e_i} - x_{o_i}, \quad (6)$$

$$y_v = \frac{1}{n}\sum_{i=0}^{n} y_{e_i} - y_{o_i}. \quad (7)$$

The algorithm chooses the new vehicle position and heading estimate based on the LLS instance with the least residual. The output of the camera-based component of the localization system is an absolute pose of the vehicle in the parking lot coordinate frame whenever a landmark is visible. To provide a pose estimate at all times, a relative positioning system based on odometry is fused with camera pose estimates.

B. Odometry

Odometry is a relative positioning system and thus is prone to drifting. It is, however, always available unlike the camera-based system and therefore a complimentary method. There exists two major approaches to odometry: one based on the rear axle model and the other based on the Ackermann model, which is itself a bicycle lane model with infinite cornering stiffness [9],[10]. Both approaches use the wheel tic sensors mounted on the left and right rear wheels, denoted ΔL_L and ΔL_R, respectively. The Ackermann model approach additionally uses the steering wheel angle δ_s.

In the following description, the reader is referred to the illustration in Fig. 6 for variable definitions. Both approaches are similar in how they use estimated heading change and the rear wheel traveled distances to update the vehicle's position:

$$\Delta x_v = R_m \sin(\Delta\psi_v), \quad (8)$$

$$\Delta y_v = R_m(1 - \cos(\Delta\psi_v)), \quad (9)$$

for a nonzero heading change and,

$$\Delta x_v = \Delta L_m, \quad (10)$$

$$\Delta y_v = 0, \quad (11)$$

when the vehicle is traveling straight. ΔL_m is computed as the average of the traveled distances of the rear wheels. Vehicle turning radius R_m is calculated by:

$$R_m = \frac{\Delta L_m}{\Delta\psi_v}. \quad (12)$$

The difference between the approaches is the manner in which heading change is calculated. The rear axle approach is based on wheel travel distance and track width s:

$$\Delta\psi_v = \frac{\Delta L_L + \Delta L_R}{s}. \quad (13)$$

The Ackermann model approach uses steering wheel angle and the wheel base l:

$$\Delta\psi_v = \frac{\Delta L_m}{l}\tan(\delta_s * i), \quad (14)$$

where

$$i = \frac{\delta_s}{\delta_w}. \quad (15)$$

The parameter i relates the steering wheel angle to the wheel angle δ_w through a transfer function.

In our experience with the rear axle approach, we found that due to quantization noise in the wheel tic sensor, the vehicle heading output contains a significant amount of noise. The vehicle position is also impacted by the noise in the heading change estimate. In Junior 3 this manifests itself as vibrations in the steering wheel as the controller tries to compensate for the errors in vehicle heading and position. We decided against filtering the output since it would introduce errors in the overall pose estimate. Instead, we chose the Ackermann model approach without any significant impact on performance. For odometry, the main cause for error is typically uneven pavement.

C. Calibration

As described in Section III-A, we employed the homography matrix in the localization system as a mapping between pixels on the image plane to vehicle relative positions of markers on the road. To compute the homography matrix, we first laid out the markers in front of Junior 3 on a flat road surface and physically measured their coordinates relative to the vehicle. We then reused the ellipse detection algorithm described in Section III-A to compute the pixel coordinates of the markers. Finally, we input all the pixel and real world coordinate pairs for each marker into a RANSAC-based algorithm for robustly fitting a homography matrix to the data [11]. Estimates of the variance for use in the Kalman filter were computed by viewing landmarks at different regions of the image plane and comparing the camera output against a differential GPS (dGPS) reference system. When Junior 3 runs in AVP mode, the variance for a landmark detected at

a particular location on the image plane is interpolated from the reference points measured offline.

Our Ackermann model based approach for odometry uses Junior 3's wheelbase, the wheel circumferences of the rear wheels, and a steering wheel angle to curvature transfer function as parameters. The wheelbase is from the specifications for our 2006 Volkswagen Passat Wagon TDI. Wheel circumferences were measured by dividing the traveled distance when driving straight on a flat road surface by the number of wheel revolutions indicated by the wheel tic sensors. The traveled distance was measured using dGPS. Lastly, to compute the transfer function, we needed data relating curvature to the entire range of steering wheel angles. To accomplish this we drove spirals with random start positions and recorded the maneuvers using dGPS. The recordings were analysed to produce many steering wheel angle-curvature pairs across the entire range, allowing us to calculate the transfer function. We calculated the variance estimate for odometry by comparing many short segments of odometry localization to a dGPS reference.

D. Localization Results

Fig. 7 shows a representative run of the AVP system. The system is initialized with an observation of the first landmark and, along the search path, Junior 3 observes four additional landmarks with the camera. An Applanix POS LV 420 differential GPS (dGPS) is used in conjunction with a radio linked base station as a reference localization system. The accuracy of the dGPS is stated to be 2 cm RMS for position and 0.02 degrees RMS for heading.

To analyze the performance of the camera algorithm, we look at the error compared to dGPS of the final fused output of camera and odometry after each cluster of landmark observations. This is sufficient because the low variance attributed to the camera algorithm output causes the fused output of the Kalman filter to closely follow the camera algorithm's output. We see that inaccuracy in position can be as high as 60 centimeters, though typically it is around 20 centimeters. Heading accuracy is typically within 2 degrees. We found the larger errors in position and heading to be primarily due to shifts in the pitch, roll, and height of Junior 3 while driving. The predominant causes of this are uneven road surfaces and sharp turns initiated by the controller. The latter effect can be seen in Fig. 7 as Junior 3 observes the fifth landmark. To mitigate this effect we place landmarks before turns in the vehicle path instead of during or shortly after them.

The performance of odometry is most important during the parking maneuver due to the narrow spaces. While searching for a parking spot, the driving path is generally wide enough to allow as much as a meter of error. Moreover, when executing the parking maneuver, the localization uses only odometry to avoid any sudden shifts in pose due to the camera during the maneuver. We measured the final error in Junior 3's pose in the parking spot, accumulated over the course of the parking maneuver, to be on average 40 cm laterally, 11 cm in depth, and 3.5 degrees in heading.

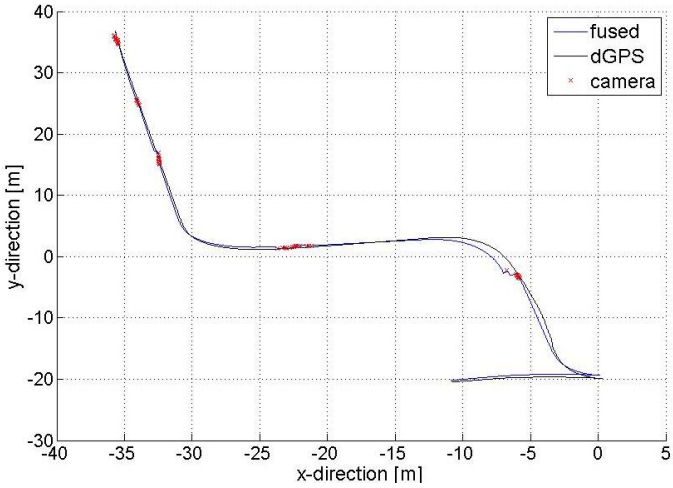

Fig. 7. Camera-Odometry comparison to dGPS reference for an AVP run.

The accompanying standard deviations were an order of magnitude lower. The main cause for the systematic errors in the odometry were high steering angles in the parking maneuver. The low variances allowed us to introduce an offset in pose for the parking maneuver to remove the systematic error. The remaining error is then largely due to the error at the start of the maneuver. This error is managed through strategic placement of markers along the search path.

IV. Path Planning

The existing general planner module used in the DARPA Urban Challenge handles traffic signs, right of way, and among many other things, also parking [2]. It considers detected obstacles as well and ultimately creates dynamic paths for a dynamic environment. For navigating the parking lot in the AVP application, the existing planner easily handles the task. The path planning for parking, however, satisfies only Urban Challenge competition rules and is not sufficient to our task. Two main limitations are that the algorithm allows the car to make multiple attempts in finding a path into the spot with the only constraint being obstacle avoidance. Most importantly, it does not constrain the vehicle to park aligned and accurately within the spot.

We briefly introduce the parking spot detector here, which is further described in [8]. The detection occurs with the aid of two single plane lidar scanners mounted on the side of the vehicle. At the time the lidar unit passes abeam (see Fig. 8) of a free spot, a message is sent via IPC triggering a transition from the general planner to the parking planner.

A. Parking Planner

The goals of the parking planner are to provide a reliable, predictable, path that guided the vehicle to the spot in one attempt. Unlike the general planner, it does not consider obstacles in its path planning. The vehicle's pose relative to the target parking spot at the point of free spot detection is the only variable input. Parameters used to create a drivable path are the vehicle's minimum turning radius, maximum speed of the steering wheel actuator, and vehicle speed.

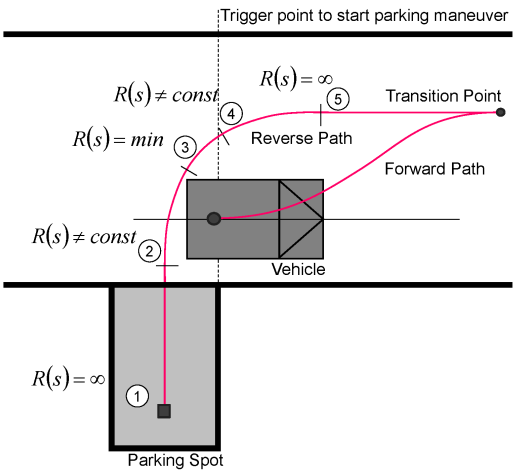

Fig. 8. Components of the parking planner's forward and reverse path.

The parking planner plans a fixed reverse path leading into the spot as well as a dynamic forward path linking the vehicle's start pose to the start of the reverse path as shown in 8. The forward path is defined by a cubic spline connecting the start pose to the point of transition to the reverse path. Since a cubic spline is not guaranteed to be drivable, an iterative process shifts the end point of the forward path, relaxing the spline until the car parameter constraints are met. The reverse path is generated in five segments *from* the car parameters using an Ackermann model (Fig. 6) described in (8-12, 14-15). The length of the segment 1 is determined by the dimensions of the parking spot and the vehicle. Segments 2 and 4 bring the vehicle to minimum and maximum turning radius, respectively, with the motion definied by the vehicle speed and the steering actuator speed. Vehicle motion in segment 3 is defined by the speed at the minimum turning radius. Lastly, segment 5's length is defined by the distance to the transition point. By constraining the orientaiton of the this segment to be perpendicular to the spot we largely avoid the intersecting the path with another spot.

The path generated will park Junior 3 into the parking spot in one attempt. When localizing with dGPS, Junior 3 parked with a standard deviation of below 1.5 cm for both lateral and depth error. The standard deviation of heading was 0.5 degrees. Contributions to these errors include: GPS inaccuracy and variabilities in the execution of controller module commands to Junior 3's actuators. The final pose error in the parking spot is a sum of the localization error at the point of free spot detection, localization error accumulated by odometry during the parking maneuver, and controller performance during the parking maneuver.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper we described the software development of a modified 2006 Volkswagen Passat TDI, also known as Junior 3, repurposed from its use in the DARPA Urban Challenge to enable an Autonomous Valet Parking system. The design and development centered in three areas: parking spot detection, a localization system that can function in a parking garage, and a path planner for parking. The localization system fused the absolute pose output of a camera-based marker detection algorithm with the relative pose update given by odometry through a Kalman filter. The parking planner augmented the existing general planner by taking over at the moment a free spot is received from the parking spot detector, guiding Junior 3 to the parking spot in a single attempt. A successful demonstration of the system took place on October 24th, 2009 at the dedication event for the Volkswagen Automotive Innovation Lab at Stanford.

B. Future Works

Future work will be directed at creating a more generalized parking path planner that considers the external environment and also at characterizing the performance of other methods of localization in a parking garage.

VI. ACKNOWLEDGMENTS

The authors would like to thank the members of the *Autonomous Driving Team* at the Electronics Research Lab: Jose Acain, Simon Herrmann, Dirk Langer, Lorenz Laubinger, Mark Malhotra, Robert Maclellan, Peter Mirwaldt, Ugur Oezdemir, Tim Schneider, Ganymed Stanek, and David Quintero. Their diverse skills and persistence made the completion of such a large project possible.

Additionally the authors would like to thank Jan Becker and Charles Duhadway from Bosch for their support in utilizing the camera hardware and software.

REFERENCES

[1] M. Montemerlo et. al., "Winning the DARPA Grand Challenge with an AI robot", *Proceedings of the AAAI National Conference on Artificial Intelligence*, Boston, MA, 2006.

[2] M. Montemerlo et. al., "Junior: The Stanford Entry in the Urban Challenge", *Journal of Field Robotics*, Vol. 25, Issue 9, 2008, pp 569-597.

[3] B. Müller-Bessler et. al., Reproducible Transverse Dynamics Vehicle Evaluation in the Double Lane Change, *ATZ - Automobiltechnische Zeitschrift*, pp 44-49, April 2008.

[4] B. Müller-Bessler et. al., Customer oriented Safety and Handling Evaluation via adjusted Driver Model using Real Vehicle, *Proceedings of FISITA*, München, Germany, 2008.

[5] D. Stüker, C. Brenneke, Sichere Software-Entwicklung für Fahrerassistenzsysteme, *Proceedings of AAET - Automation, Assistance and Embedded Real Time Platforms for Transport*, Braunschweig, Germany, 2006.

[6] A. Weiser et. al., Intelligent Car, Teilautomatisches Fa.ren auf der Autobahn, *Proceedings of AAET - Automation, Assistance and Embedded Real Time Platforms for Transport*, Braunschweig, Germany, 2009.

[7] R. Simmons and D. Apfelbaum. "A task description language for robot control", *Proceedings of the Conference on Intelligent Robotics and Systems*, October, 1998.

[8] G. Stanek et. al., "JUNIOR 3: A Test Platform for Advanced Driver Assistance Systems", *Proceedings of the IEEE Intelligent Vehicles Symposium*, June, 2010.

[9] M. Mischke and H. WallenTowitz. *Dynamik der Kraftfahrzeuge 4th ed.*, Berlin Heidelberg:Springer Verlag, 2004.

[10] J. Borenstein, H.R. Everett, and L. Feng, "Where am I? Sensors and Methods for Mobile Robot Positioning", *University of Michigan Technical Report*, April, 1996.

[11] P. D. Kovesi. "MATLAB and Octave Functions for Computer Vision and Image Processing", *The University of Western Australia*, 2000 <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.