A Motion Controller for Compliant Four-Wheel-Steering Robots

Pedro F. Santana, Carlos Cândido, and Vasco Santos

IntRoSys, S.A.
FCT-UNL, 2829-516 - Portugal
{pfs,cpc,vps}@uninova.pt

José Barata
UNINOVA
FCT-UNL, 2829-516 - Portugal
jab@uninova.pt

Abstract—This paper proposes a behaviour-based approach for the motion control problem of sustainable and compliant Four-Wheel-Steering robots. Each wheel is considered to be an independent entity that must react according to some local rules. Motivated by the Virtual Components Approach, behaviours implement these local rules by means of a set of virtual elements with a physical counterpart (e.g. springs, potential fields) to describe the desired behaviour a part of the robot has relative to another one, or to the environment. The local emplacement of the virtual components allows an intuitive design of motion control systems rendering to a fast development phase. An instance for a four-wheeler steering robot is proposed and tested in simulation, demonstrating a good compliance w.r.t. the environment in order to reduce power consumption and mechanical stress.

Index Terms—Mobile robots, four wheel steering vehicles, behaviour-based motion control

I. INTRODUCTION

In addition to the cognitive capabilities, it is also paramount to consider energetic sustainability and long-lasting capabilities of everyday robots.

These problems have been widely explored in the humanoids research community for the last few years, where energetic autonomy is hard to achieve and control systems have to comply with a complex body embedded in a complex dynamic environment. Servo-based approaches (e.g. [1]) have shown to be highly accurate but they require an explicit trajectory plan for each joint. In addition, robots following this approach tend to show rigid interactions with the environment, which results in significant impacts and crisp behaviour. Conversely, the Virtual Components approach [2] focuses on the dynamical behaviour each joint is desired to have relative to other elements of the robot, by computing forces or torques accordingly. This latter approach avoids the need for explicit planning, and at the same time it also allows the robot to naturally cope with unexpected situations with low stiffness. On the other hand, in servo-based solutions, a discrepancy between the plan and the actual robot-environment state uses to be considered an exception and replanning is required.

A Four-Wheel-Steering Robot (FWSR) is also a machine with several joints. Although they can be considered as independent entities interacting with the world (i.e. robots by themselves), they have to co-operate (not necessarily explicitly) in order to maintain the Ackerman geometry. Maintaining the Ackerman geometry is essential to avoid wheel slippage, which usually induces mechanical stress and extra energy consumption. The environment (e.g. terrain irregularities) and mechanical problems (e.g. mechanical slack) can project some unexpected forces onto the robot. Instead of pushing the motors up-to their saturation levels in order to keep Ackerman geometry whilst reducing the reference error (i.e. the desired turning radius), the robot should comply with the environment in a dynamical way. Hence, instead of defining a desired behaviour which when unfeasible triggers an error recovery strategy, one designs the way both environment and robot interact in a dynamical way.

FWSRs are attracting considerable attention from the research community, given their tremendous manoeuvrability (see for instance the Nomad robot for planetary exploration [3]). Usually, linear control techniques (e.g. [4]) are applied for the control of this type of robots. However, the compliance requirements set above hardly cope with the use of linear models around a nominal state. Moreover, gain matrices usually employed in linear control techniques in order to shape the control loop are not intuitive due to their lack of operational semantics. On the other hand, local interactions based design more easily attaches meaning to control loop shaping parameters.

II. ROBOT KINEMATIC MODEL

A typical FWSR schematic is illustrated in figure 1. Let us now analyse the kinematic model of the vehicle under the light of the work presented in [5].
FWSRs with independent steering and driven wheels (from now on it will always be assumed this configuration) can generate four different locomotion modes: in **Ackerman mode** the Ackerman geometry has to be continuously maintained and the turning radius is measured from the geometrical centre of the vehicle to the Instant Rotating Centre (IRC); in **point turning mode** the robot will have the possibility to rotate over its own geometrical centre without any lateral displacement by superposing its IRC with the vehicle’s geometrical centre; in **lateral displacement mode** the four wheels will be aligned, allowing the robot to move along a linear trajectory; and in **differential mode** both linear and angular velocities are set by injecting different speeds in the left and right wheels pack maintaining the steering angles unchanged, which though being simpler to control, it induces significant mechanical stress.

These four modes provide the robot with excellent mobility. This work will focus on the Ackerman mode though it should be generic enough to cover the entire locomotion envelope. In order to avoid ambiguities, from now on, we consider the linear velocity, the steering actuator and desired IRC, \( \text{IRC}_{\text{des}} \), which is modulated by a higher level system.

Let us define \( FL = 1, FR = 2, RL = 3, \) and \( RR = 4 \) as a way of referring to front-left, front-right, rear-left, and rear-right wheels, respectively. \( \text{IRC}_a \) refers the IRC of wheel \( a \in A, \) where \( A = \{ FR, FL, RL, RR \} \). When the Ackerman geometry is respected, \( \text{IRC}_{FL} = \text{IRC}_{FR} = \text{IRC}_{RL} = \text{IRC}_{RR} \), holds. \( \text{IRC}_a \) can be computed based on the steering angle \( \beta_a \) by making \( \text{IRC}_a = \frac{h \cdot \cos \beta_a}{\tan \beta_a} + \left( -1 \right)^{a-1} \frac{d}{2} \), where \( h \) and \( d \) are the height and width of the robot, respectively, and \( \Lambda_a = 1 \) if \( \{ a \in \{ FL, FR \} \} \) or \( \Lambda_a = -1 \) if \( \{ a \in \{ RL, RR \} \} \).

Each wheel encompasses one steering actuator and one traction actuator. In order to avoid ambiguities, from now on, only the terms ‘steering actuator’ and ‘traction actuator’ will be employed, leaving behind the term ‘wheel’.

We define \( \hat{e}_{a,b} \) as the **Ackerman error** between two steering actuators \( a \) and \( b \). The Ackerman error refers to the angle the steering actuator \( a \) has to rotate so that \( \text{IRC}_a = \text{IRC}_b \) holds. If the Ackerman error was to be computed directly over IRC, then it would describe a non-linear curve, which is highly undesirable for control purposes. Hence, \( \hat{e}_{a,b} = \Lambda_a \cdot \arctan \left( \frac{\text{IRC}_b + \left( -1 \right)^{a-1} \frac{d}{2}}{\text{IRC}_a} \right) - \beta_a \).

This error function allows the tracking of transient and static relative Ackerman errors. However, it is also necessary to track the error relative to the desired IRC, \( \hat{e}_{a, \text{IRC}_{\text{des}}} \). This latter error source is related to the angular distance a steering actuator has to travel so that \( \text{IRC}_a = \text{IRC}_{\text{des}} \) holds. Formally, \( \hat{e}_{a, \text{IRC}_{\text{des}}} = \arctan \left( \frac{\text{IRC}_{\text{des}} + \left( -1 \right)^{a-1} \frac{d}{2}}{\text{IRC}_a} \right) - \beta_a \).

As it will be shown, traction and steering control are tackled separately, i.e. with decoupled control loops. Hence, considering the current steering angle \( \beta_a \) and robot’s desired linear velocity, \( v_{\text{des}} \), the angular velocity of each traction actuator is given by \( \dot{\beta}_a = \frac{v_{\text{des}} \cdot \text{IRC}_a + \left( -1 \right)^{a-1} \frac{d}{2}}{\text{IRC}_a \cos \beta_a} \). A PID controller is responsible for maintaining the desired traction speed. This option is based on the fact that the traction dynamics is significantly faster than the steering one, which discards the need for special synchronisation mechanisms.

### III. Schema Based Control

As aforementioned, the control problem is approached using a local perspective, by specifying a set of interaction rules between key components of the system. Then, the overall behaviour emerges as time unfolds. The Virtual Components approach [2] instantiates this methodology with passive virtual components, like springs and dampers. In this paper a simpler approach is followed. In particular, a 1-D version of the potential fields method is used. This, as will be shown below, allows an easier implementation and at the same time enables the (almost) direct application of the schema-based behavioural architecture.

Potential Fields approaches have been proposed to produce smooth trajectories for both mobile robots and manipulators [6], [7]. In these approaches obstacles exert repulsive forces onto the robot, whereas targets apply attractive forces onto the robot. Then, the sum of all the forces, the resulting force, determines the direction and speed of travel. Later, Arkin developed the schema-based behaviour-based architecture [8], which adapted the concept of potential fields for fast and sensor-based control. It is an agent-based architecture that allows each element to be instantiated at any time, according to the task at hand and environment state, which endows the architecture with an interesting run-time flexibility.

Each behaviour contributing for the displayed behaviour is implemented as a motor-schema, which acts according to the information provided by a set of perceptual-schemas. The output of each motor schema is an action vector, which provides the desired direction and speed for the robot. Then, the output of all active behaviours cooperate to find the best action at a given moment. The cooperation is based on weighted vectorial sum, where weights allow one to distinguish motor-schemas in terms of priority. Modularity is provided by its agent-based nature and standard vectorial form for motor-schemas output. In this line, a set of motor-schemas and respective coordination nodes can be aggregated into a single motor-schema, also called of assemblage behaviour. Behavioural assemblages become active according to the current task and environment’s state. From the above considerations is possible to realise that this mechanism is simple and consequently with low computational requirements. Its simplicity also extends to the design phase, where behaviours are highly decoupled and their integration is made by tuning the gains specifying how much each motor-schema contributes for the actual action.

Figure 2 illustrates the proposed schema-based instance for the FWSR control problem. This architecture controls one steering actuator and so it has to be replicated for the other three. This means that, each actuator is seen as an independent entity which has to be controlled with a set of local rules so as to react to the environment (i.e. physical world and other
steering wheels), pretty much like a mobile robot in a multi-robot system. Three motor-schemas have been defined: (1) the turning radius control motor-schema, (2) the Ackerman error control motor-schema, and (3) the stiffness control motor-schema. Each of these motor-schemas contributes, partially, for the steering actuator angular velocity.

Formally, for each steering actuator \( a \), the motion controller must determine the steering and traction angular velocity tuple, \((\beta_a[n], \alpha_a[n])\) (discrete time index \( n \) will be discarded for the sake of clarity). Once more, bear in mind that the following considerations are concerning a set of motor-schemas controlling a single steering actuator. Then, this set of motor-schemas is replicated for the remaining three wheels. Hence, the synchronisation between the four wheels is made via the world and not through any sort of special signalling. Following Brooks’ line of thought [9], there is no need for any internal representation or explicit signalling, synchronisation between behaviours is made via the world in an implicit way.

A. Potential Fields Space

In this work, each motor-schema creates its own 1-D potential fields space, carefully designed to produce the desired behaviour. A point in the potential fields space corresponds to an angular distance to be travelled by the steering actuator. According to sensory information, goals, etc., the motor-schema \( m \) populates its potential fields space, \( S^m \), with potential fields. These potential fields can either attract or repel the steering actuator. The superpositioning of all potential fields over position zero in the potential fields space, produces a “force” which will generate a proportional steering angular speed.

This means that the control system design is done in three major steps: (1) decompose the control problem into sub-problems (i.e. motor-schemas), (2) specify each motor-schema (i.e. which potential fields to add where), and (3) to transform the superposed potential field into a angular velocity.

Figure 3 illustrates the major components composing a potential fields space. In this example, a motor-schema adds two, one attractive and another one repulsive. The superposition of both potential fields over position zero represents the “force” applied to the steering actuator, which will induce the steering actuator to rotate anti-clockwise. Hence, specifying these local interactions one can modulate actuators to behave as desired.

The resulting vector, \( f^m \), induced by motor-schema \( m \) via a set of potential fields \( P^m \), is given by

\[
\mathbf{f}^m = \sum_{p \in P^m} f^m_p \cdot \mathbf{u}^m_p,
\]

where \( f^m_p \) is the force generated by the potential field \( p \) over the steering actuator, and \( \mathbf{u}^m_p \) is the relative weight given to the potential field in question (i.e. \( \sum_{p \in P^m} u^m_p = 1 \)).

Intuitively, \( f^m \) represents how fast the steering actuator should rotate from the current angular position according to the potential fields distribution. And potential fields represent how much the motor-schema wants steering actuators to have a certain steering angle.

Let us now specify the potential fields. For this work, two linear and discontinuous attractive and repulsive potential fields have been defined (both adapted from [10]). Notice that, the concept of potential field is an abstraction, to create attractive and repulsive “forces”, which does not intend to fully commit with any physical counterpart.

1) Attractive Potential Field: The attraction force generated by an attractive potential field \( p \) onto the steering actuator is defined in terms of its module and signal:

\[
|f_p| = \begin{cases} 
0 & \text{if } (p_x < R) \\
\frac{|p_x| - R}{S - R} & \text{if } (S \geq |p_x| \geq R) \\
1 & \text{if } (|p_x| > S) 
\end{cases}
\]

\[
f_p = \text{sign}(p_x) \cdot |f_p| 
\]

where \( R \) allows to discard the attractor below a given threshold, \( S \) is the sphere of influence of the attractor, \( p_x \) is the position of the potential field generator in \( S^m \), and \( \text{sign}(x) = 1 \) if \( x > 0 \) and \( \text{sign}(x) = 0 \) otherwise.

Although simple, these local rules will be shown to suffice for the task in hand. Nevertheless, other could be selected, like non-linear attractors defined by differential equations. Moreover, other state information could be associated to potential field generators, like the current steering angular velocity.

2) Repulsive Potential Field: Similarly to the attractive potential field, the repulsive counterpart is defined as follows:

\[
|f_p| = \begin{cases} 
0 & \text{if } (p_x < -R) \\
\frac{|p_x| - (-R)}{S - (-R)} & \text{if } (-S \geq |p_x| \geq -R) \\
1 & \text{if } (|p_x| > S) 
\end{cases}
\]

\[
f_p = \text{sign}(p_x) \cdot |f_p| 
\]
\[ |f_p| = \begin{cases} 
\frac{1}{s-p_x} & \text{if } (|p_x| < R) \\
\frac{2-s-p_x}{s-R} & \text{if } (S \geq |p_x| \geq R) \\
0 & \text{if } |p_x| > S 
\end{cases} \] (3)

\[ f_p = -\text{sign}(p_x) \cdot |f_p| \] (4)

B. Motor-Schemas Fusion

Now it is necessary that all motor-schemas contribute, in different amounts, for the resulting vector, \( O^n \), to be applied onto steering actuator \( a \). This is done by making a weighed vector summation, i.e. \( O^n = \sum_{m \in M} f^m \cdot g^m \), where \( M \) is the set of active motor-schemas and \( g^m \) is the relative gain of motor-schema \( m \) (i.e. \( \sum_{m \in M} g^m = 1 \)).

These gains in addition to those related to potential fields can be changed in run-time as task and environment change. Hence, the controller has inherent plasticity, which is easily manageable due to the semantics associated to all free parameters. Moreover, since the output of each motor-schema is normalised before being merged with all the remaining motor-schemas, potential field weights and motor-schema gains are pretty much decoupled.

C. Action Production

In the previous sections it has been shown how a motor-schema produces its output and how all motor-schema outputs cooperate to generate a resulting vector. Now it is necessary to compute the actual angular velocity applied to the steering actuator \( a \). \( \beta_a = \Omega (K \cdot O^n) \), where \( K \) scales the resulting vector so as to generate an angular velocity and \( \Omega \) maintains \( \beta_a \) within \([\beta_{max}, \beta_{max}]\). \( \beta_{max} \) corresponds to the maximum angular velocity attainable by the steering actuator. It is assumed the existence of a PID controller (i.e. an inner loop) to maintain the desired angular velocities. Thus, the inner loop contains an integrative component necessary to remove any static error.

IV. AN INSTANCE FOR STEERING CONTROL

This section introduces each of the motor-schemas implemented for the FWSR control. The design rationale is to decompose the motion controller into a set of simple decoupled motor-schemas, each one responsible for attaining a sub-goal (e.g. to reduce Ackerman error) with all the already mentioned advantages.

Remember that motor-schemas specification is done by adding potential field generators into the potential fields space according to the desired dynamical behaviour. The set of active motor-schemas is \( M = \{a, t, s\} \), where \( a \), \( t \), and \( s \) refer to Ackerman error control motor-schema, turning radius control motor-schema, and stiffness control motor-schema, respectively.

1) Ackerman Error Control Motor-Schema: Figure 4 illustrates the operation of the Ackerman error control motor-schema concerning the front-left wheel, \( FL \).

First, the motor-schema determines the Ackerman error that the steering actuator \( FL \) has relative to all others. For instance, relative to \( FR \), the error is given by \( \hat{e}_{FL,FR} \). This value refers to the number of degrees the \( FL \) has to turn so as to guarantee that there is no Ackerman error between both steering actuators. In the bottom of the figure it is shown the potential fields space for \( FL \).

Then, still for the \( FL \) case, an attractive potential field \( p_{FL}^{\hat{e}_{FL,FR}} \) is added to the potential fields space in the position defined by \( \hat{e}_{FL,FR} \). The same procedure is iterated for all steering actuators. The potential fields have \( R = 0 \), and \( S = 30 \) (empirically determined). Each of the potential fields induce an attractive force onto the steering actuator so as to induce it to approach the steering wheel generating the potential field and as a result to reduce their Ackerman error. The attraction to one of the steering actuator is then weighed against the attraction to the other steering actuators as previously explained. Since all other wheels are implemented likewise, steering actuators will cooperate implicitly.

Weights are defined empirically but their explicit semantic allows one to distinguish them clearly so as to refine the displayed behaviour, even in qualitative terms, as it will be shown by the experimental results. At the first sight one could conclude that potential field weights in this motor-schema have to be equal. However, giving more weight to the potential field associated to Ackerman error relative to the adjacent wheel (e.g. if the being controlled is the front-left then the adjacent one would be the front-right), \( p_{a,adj}^{\hat{e}_{FL,FR}} \), then to the other two potential fields, \( p_{other1}^{\hat{e}_{FL,FR}} \) and \( p_{other2}^{\hat{e}_{FL,FR}} \), the Ackerman error would reduce at a faster pace between wheels located side-by-side. The true impact of this ability is to be studied in terms of mechanical stress.

2) Turning Radius Control Motor-Schema: This turning radius control motor-schema (see figure 5) generates the goal-oriented behaviour by inducing the robot to turn respective to the desired IRC, \( IRC_{des} \). Similar to the previous motor-schema, herein the angular distance that allows \( IRC_a = IRC_{des} \), i.e. \( \hat{e}_{a,IRC_{des}} \), is calculated and then a potential field \( p_{IRC}^{\hat{e}_{a,IRC_{des}}} \) is added to the potential field space according to \( \hat{e}_{a,IRC_{des}} \). The potential field is empirically parameterised.
with $S = 30$ and $R = 0$.

3) Stiffness Control Motor-Schema: The previous two motor-schemas endow the robot with the ability of keeping a desired IRC while respecting the Ackerman geometry as much as possible. This solution is dynamic and so it is able to provide a control solution even when unpredictable situations occur (i.e. generalisation capability). The potential field space is linear and so the control laws remain reasonably faithful in the entire motion envelop.

This section introduces a third motor-schema, the one responsible for controlling the stiffness of the robot; in other words, to control up to which degree the robot will try to impose itself against the environment. The stiffer the system is, the less error it has, however the more current it consumes and the more mechanical stress it suffers. Thus, having a way of controlling the stiffness of the system allows the selection between small errors and robot sustainability, in both mechanical and energetic terms.

It is assumed the ability to measure the current being consumed by each steering motor. It is also assumed that there is a "reference" current consumption, and a threshold from which one may start to consider abnormal resistance. Figure 6 illustrates how this motor-schema operates.

In short, if the steering actuator has positive angular velocity and it is feeling resistance above a certain threshold, which is described in terms of the consumed average current\(^1\), then a repulsive potential field, $p^e_{rep}$, is added as shown in figure 6. If the angular velocity was instead anti-clockwise, the repulsive potential field generator would be positioned in a symmetric position. The weight associated to the potential field is multiplied by the resistance level. Hence, the greater the weight the stiffer the system is. The resulting behaviour is that this motor-schema will contradict the other two leading the steering actuator to slow down according to the applied resistance.

On the other hand, an attractive potential field, $p^a_i$, is added per each steering actuator, $i$ but itself, whose weight is also affected by their consumed average current. Thus, these attractors will induce the steering actuator to match those with higher resistance. That is to say that the motor-schema induces the Ackerman error to be reduced around steering actuators affected by some resistance.

The resulting net of the contribution of all these potential fields is that the steering actuator will slow down as resistance increases, and at the same time will try to reduce the Ackerman error around steering actuators affected by some resistance. Once more, the generalisation capability of the system guarantees that a solution is generated even if more than one actuator is feeling some resistance to rotation.

V. Experimental Results

This section presents some simulation results obtained from a dynamical model of the robot. By default, the gains of the Ackerman error control, turning radius control, and stiffness control motor-schemas are $g^a = 0.3$, $g^t = 0.1$, $g^s = 0.6$, respectively. In the same line, the weight associated to the potential field for Ackerman error reduction relative to the adjacent actuator, $p^a_{adj}$, and the weight for each of the other two actuators, $p^a_{other1}$ and $p^a_{other2}$, are set to $0.33$. Finally, the weight associated to repulsive potential fields added by the stiffness control motor-schema, $p^e_{rep}$, is 0.1, whereas the one associated to each attractive potential field, also introduced by the same motor-schema, $p^a_i$, is 0.3.

Figure 7 plots the steering angles and respective average Ackerman error against time. At the outset, actuators are set to angles completely disregarding Ackerman geometry. As a consequence, actuators will move so as to reduce this initial error at the same time they aim to reduce turning radius error, which is set to 2 m. It is possible to depict that actuators momentarily increase the turning radius error so as to reduce Ackerman error. After one second the robot is asked to move with a turning radius of $-2$ m, and then back to $2$ m after two seconds.

The Ackerman error for the steering actuator $a$, has been defined by $e_{ack} = \sum_{b \in A} \frac{|e_{a,b}|}{4}$. As it is possible to see from figure 7(b), the Ackerman error is kept low. It could be even lower if the gains were selected otherwise. One of the causes for some considerable

\[^1\]Current is averaged so as to accommodate their natural variability; in addition, current is normalised to be within $[0, 1]$.\n
---

\[\begin{align*}
\text{Fig. 5. Turning radius control motor-schema.} \\
\text{Fig. 6. Stiffness control motor-schema. Dotted arrows represent the influence of steering actuators power consumption in each potential field. Curved solid arrows represent the steering direction of each steering actuator.}
\end{align*}\]
error is that the robot has been asked to change of IRC in a significant amount in a short period of time, which makes the robot to be much more responsive to the IRC error than to the Ackerman one. Making the IRC to change more slowly would reduce this effect.

In a second run (see figure 8) some mechanical stress has been simulated. In particular, high current sink has been simulated for FL between 1 s and 1.5 s, which means that there is some resistance imposed by the environment or a mechanical problem during that period.

The net effect is that the affected actuator stops rotating towards the goal and all the other try to keep Ackerman error as small as possible. Some static error emerges due to the contradictory potential fields. Nevertheless, the error is kept small. Then, between 2.5 s and 3 s another type of resistance is tested. Instead of injecting simulated current into an actuator, the RL steering angle is forced to change of position instantaneously (this corresponds to a strong and sudden impact projected onto the wheel). Once more, the system is capable of adapting to the new situation.

VI. CONCLUSIONS AND FUTURE WORK

A novel motion controller for compliant and sustainable FWSRs was presented. The schema-based behavioural architecture is well established in mobile robot reactive navigation, but to our knowledge it has never been applied to actuator coordination. This methodology allows the designer to focus on intuitive local interactions.

Other attractor types will be investigated, specially those based on virtual components, such as springs and dampers. These and others will introduce non-linearities, integrative, and derivative components in a physically intuitive way.

Finally, it will be necessary to test and validate this promising approach in a physical robot running the four identified locomotion modes, which is expected to happen in the short-term.

ACKNOWLEDGEMENTS

The authors of the paper would like to thank Professor Luís Correia for inspiring discussions and reviewing this paper.

REFERENCES