Symmetries in World Geometry and Adaptive System Behaviour*

Robin Popplestone and Roderic A. Grupen

Laboratory for Perceptual Robotics,
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{coelho, piater, grupen}@cs.umass.edu
http://www-robotics.cs.umass.edu/

Abstract. We characterise aspects of our worlds (great and small) in formalisms that exhibit symmetry; indeed symmetry is seen as a fundamental aspect of any physical theory. These symmetries necessarily have an impact on the way systems exhibit reactive behaviour in a given world for a symmetry determines an *equivalence* between states making it appropriate for an reactive system to respond identically to equivalent states. We develop the concept of a General Transfer Function (GTF) considered as a building block for reactive systems, define the concept of *full symmetry operator* acting on a GTF, and show how such symmetries induce a quotient structure which simplifies the process of building an invertible domain model for control.

1 Introduction

This paper explores the relationship between symmetries of the world and symmetries of the *(generalised) transfer functions* which are used to characterise the response of a reactive system. It is written from the perspective of Artificial Intelligence to the extent that we consider how some principles of automating problem reductions which might in many cases be "obvious" to humans.

A symmetry of a physical theory is an invertible mapping of the space in which the theory is expressed to itself under which the theory is invariant. For example, the symmetries of Newtonian mechanics are drawn from the *Galilean group* of symmetries of space. This consists of translations and rotations and uniform translatory motion (but not of rotatory motion).

In characterising a reactive system it is adequate to take a special case of a physical theory — for example we characterise the gravitational field as uniform rather than using the full Newtonian formulation of gravitation. However, a reactive (or adaptive) system does not sense the world directly, but only through its sensors so that taking such a limited view has advantage for the explanation

* This work was supported in part by the NSF (CDA-9703217), by AFRL/IFTD (F30602-97-2-0032), and by DARPA/ITO/SDR DABT63-99-1-0022.

of the behaviour of *adaptive* biological systems: the world is modelled at a level closer to what may be perceived. In general, sensor space is not isomorphic to world-space. The question then naturally arises, how do world-symmetries relate to symmetries of transfer functions which are used to characterise reactive systems? In particular, is there a useful concept of the symmetry of a controller and of a plant?

The disposition of matter in space has the effect of reducing the symmetry of the world from the point of view of reactive systems existing therein. Nevertheless, residual symmetry frequently remains and is important for determining the possible behaviour of an reactive system. When we say that an object or world feature has a symmetry we are in essence identifying an invertible mapping from the object (or feature) to itself under which it is invariant. In the case of rigid bodies, all symmetries are members of the Special Euclidean group of translations and rotations in 3-space.

Why should we be interested in symmetric controllers? The advantage lies in the possibility of being able to handle discrete event dynamic systems by combining a repetoire of controllers. We can regard a controller as establishing a property (such as maintaining stability in a gravitational field). If a controller establishes just that property and no other, then it may be combined with another controller which establishes another property (forward locomotion, say). Thus a space of behaviours can be spanned by combining elementary controllers.

But with a given property may be associated world-transformations that preserve the property. A controller that maintains stability in a gravitational field should be invariant with respect to the symmetries of that field. In the case of a linear controller, our concept of *input symmetry* can be related to that of the null-space of the controller. Our group-theoretic formulation has the advantage of being a generalisation to systems that may be non-linear, non-differentiable or even non-continuous.

2 Previous Work

The idea of classifying physical theories in terms of symmetry groups is due to Noether[8]. Noether's Theorem is a very general result which shows that any physical theory couched in variational terms necessarily has conservation laws related to the symmetries of the space in which the theory is expressed. While this has applications to the more exotic groups associated with modern physical theories, the historical development of physics can also be seen as a progression from theories of more restricted symmetry to those of less restricted symmetry. Thus Newtonian mechanics, characterised by the Galilean group, provides a more symmetric world-view than the Aristotelian.

In the 1980's one of us, cognisant of the importance of group theory in physics, sought to apply it to robotics — specifically to the characterisation of spatial relationships between body features established during assembly [12]. Subsequently Liu[7] demonstrated the practicality of this approach by developing a computationally tractable representation of subgroups of the Euclidean group, providing

a software implementation thereof together with theoretical justification. Earlier Zahnd and Nair [13] had provided a more limited approach. It should be noted that what needs to be represented is not a *member* of the Euclidean group but a subgroup embedded in it.

In psychology, Michael Leyton [5] is a pioneer in the use of group theory as a basis for understanding perception. His view is that the mind perceives a shape in terms of a causal history of how it was formed, so that a deformed can is perceived as resulting from the act of denting it. Such deformations are not members of the Euclidean Group; indeed they are drawn from a group of diffeomorphisms which is much larger than the Euclidean group and therefore more challenging to represent computationally.

By relating the study of shape to the study of symmetry, Leyton is able to argue that symmetry is crucial to cognitive processing. Thus perception is seen by Leyton as the creation of a causal history which explains the sense-data in terms of a process that extends over time — what we would call a general transfer function. Thus, in our terms, Leyton sees a primary skill of the human mind as being the synthesis of general transfer functions. Crucial to this skill is the understanding of the characteristic symmetries of such functions. We perceive a pot in terms of the rotational symmetry induced by the potter's wheel.

In this paper we shall draw, upon the concept of "naive physics" expounded by Hayes[4] in the general sense that it can be desirable to make use of a simplified physics for understanding the functioning of biological adaptive systems. Such simplified physical systems may in general be characterised by having more restricted symmetry groups than do the standard models of physics.

While our formalism does not in general require that mappings be differentiable, important examples are differentiable and characterisable by differential equations. In that case invariance under groups of symmetries is recognised to be an important characteristic of a set of equations, see [2]. Our treatment of *output symmetries* is related to the topological concept of *homotopy*.

A discussion of the Missionaries and Cannibals problem is found in Amarel[1]; our treatment of the quotient GTF of this problem is closely related to his.

Over discrete domains, our work has a strong relationship with *model checking*. Chapter 14 of [3] entitled "Symmetry" contains a definition of an automorphism group of a Kripke structure, and develops the concept of a quotient structure. This is closely related to our discussion of the symmetries of a GTF. One view of our work is that it is an approach building a synthesis of classical Control Theory (over a continuous domain) with model checking (over a discrete domain).

3 Notation: Operators and Groups

By an *operator* σ we mean an entity drawn from an arbitrary set Σ (which may be infinite). A *multiplication* is defined on operators. If σ_1 and σ_2 are operators, then their product is written $\sigma_1\sigma_2$.

This product is associative, that is

$$(\sigma_1\sigma_2)\sigma_3 = \sigma_1(\sigma_2\sigma_3)$$

The *identity operator* denoted by 1 has the property that $1\sigma = \sigma = \sigma 1$ for all operators σ.

If σ is an operator it may have an inverse, written writen σ^{-1} with the property that $\sigma\sigma^{-1} = \sigma^{-1}\sigma = 1$.

A set Σ of operators for which every operator has an inverse, which includes the identity operator, and which is closed under multiplication and inverse is called a *group* of operators.

Given a set U and a set Σ of operators, we say that the operators of Σ *act* on U if each operator of Σ is associated with a mapping from U to U. If σ is an operator, we write the effect of applying σ to $u \in U$ by $u.\sigma$. We require that

- the identity operator acts as the identity mapping, that is $u.1 = u$.
- the product of operators acts as the product of the corresponding mappings, so that $u.(\sigma_1\sigma_2) = (u.\sigma_1).\sigma_2$.
- if an operator σ has an inverse, then mapping corresponding to σ is invertable, and the mapping $u \mapsto u.(\sigma^{-1})$ is the mapping inverse to $u \mapsto u.\sigma$, that is $(u.\sigma^{-1}).\sigma = u$.

If U is any finite set, then we denote the *symmetric group* of all permutations of the members of U by S_U[1]. We shall use S_U as an operator set.

4 General Transfer Functions and their Symmetries

Transfer functions have long been used by control theorists to characterise the behaviour of systems. Essentially, a transfer function characterises the input-output relationship of a system that may have internal state (for example the charge on the capacitor of an integrator). As such, they are necessarily *functionals* or *higher order functions*, mapping from a specification of how the input to a (sub)system evolves over time to a specification of how its output evolves over time. A specification of initial state is also required.

It should be noted that we regard a Finite State Automaton (FSA) with outputs as a generalised transfer function, so that we are not restricting ourselves to mappings that are differentiable or even continuous. We may regard the evolution of a system over time as discrete or continuous (but not, in our current formulation, hybrid).

Definition 41 *We use $\mathcal{I}$ to denote an index-set used to characterise the passage of time. $\mathcal{I}$ will be the real numbers ≥ 0 (for a continuous system) or the integers ≥ 0 (for a discrete system).*

In general, the index set will be totally ordered, with a least element written as 0.

[1] Group theorists identify symmetric groups by their isomorphism class, and speak, for example, of S_3 as the group of all permutations on 3 elements. This identification is inappropriate for our purposes.

4.1 Domains of values

Control systems have traditionally been defined to act on real valued variables or vectors thereof. Here we use the concept of a *domain* of values as a general set on which components of a control system may act. In particular, elements of domains of values are not necessarily real numbers. We can think of discrete domains as supporting the concept of *logical sensors* and *logical behaviours* abstracted from actual sensors and behaviours by software layers.

We shall generally use U or U_{in} to denote a domain of inputs for a given transfer function. We may use X or U_{out} to denote a domain which is an outputs of a given transfer function.

We extend operators over a domain to act on functions over that domain. Thus if U is a domain, and σ acts on U, and $\mathbf{u} : \mathcal{I} \to U$ then $\mathbf{u}.\sigma$ is defined by $(\mathbf{u}.\sigma)(t) = \mathbf{u}(t).\sigma$

4.2 Initialiser Domains

A general transfer function may have an *initial state*, which must be specified. For example, in a classical control system, integrators may be given an initial value determined by the system designer. Likewise a robot may be activated in one of many possible initial states. We use P to denote a set of initialiser values, referring to P as the *initialiser domain.*

In order to support the *cascading* of generalised transfer functions, initialiser values need to be finite sequences, possibly empty, possibly of length one. Cascading general transfer functions will involve concatenation of their initialiser values, which we'll write as a product $p_1 p_2$. For a given initialiser domain P the sequences must be all of the same length.

Definition 42 *Let U and X be domains which we will call the* input domain *and the* output domain *respectively. Let P be an initialiser domain. Then a general transfer function T is a mapping in $T : ((\mathcal{I} \to U) \times P) \to (\mathcal{I} \to X)$.*

We will use the abbreviation "GTF" for "general transfer function". The idea is that a GTF maps an input function (of time) whose values range over an input-space U to an output function (of time) whose values range over an output space X. However this mapping may also depend on initialisation determined by a member of P.

Definition 43 *A GTF T with preset domain P and output domain X is said to be* presettable *if*

$$T(\mathbf{u}, p)(0) = p$$

Necessarily $P \subset X$; thus the initial output of a presettable GTF can be directly specified to be $p \in X$.

Definition 44 *An invertible operator σ acting on input-space U is said to be an input-symmetry of a transfer-function T if $T(\mathbf{u}.\sigma, x_0) = T(\mathbf{u}, x_0)$ for all $\mathbf{u} \in \mathbf{U}$.*

The idea being captured here is that there may be transformations of the input space to which a given transfer function is insensitive. This can amount to saying that there is state which is hidden, at least from the given function. Whether this is a problem depends on whether what is hidden is relevant. A more positive view of input symmetries is that they provide a way of defining properties of the input space of a transfer function which may be kept invariant.

For example, suppose we have a robot with a sideways pointing range sensor. Suppose the range sensor is sensing a planar wall. Then the transfer function for this sensor (mapping from robot coordinates to a real-valued distance) has an input symmetry with respect to translation parallel to the wall, thereby enabling wall-following behaviour.

Definition 45 *An output-symmetry of a presettable transfer-function T with output domain X is an invertable operator σ which acts on X and for which*

$$T(\mathbf{u}, x_0.\sigma) = T(\mathbf{u}, x_0).\sigma$$

5 Examples of GTF's

Typically, the GTF that represents the electro-mechanics of a robot will be a represented as a presettable transfer function if we regard it as being possible to initialise its position. On the other hand, incorporating an output-transducer as complex as a digitised TV camera leads to a system that not presettable: we can define images, (that is luminance functions defined on an image plane) which are not images of any scene realisable in a given world; to define $P \subset X$ would characterise the system as one which could be set up with an *exactly determined* image on the image plane, something we can't do in practice for we don't have an exact model of the imaging process.

A mobile robot Consider, for example, a mobile robot that is placed, initially at rest, on an infinite plane. Its input space is the cartesian product $U_{MR} = \mathcal{R} \times \mathcal{R}$. We shall write (u_a, u_c) for a typical member of U_{MR}. u_a is acceleration, and u_c is path-curvature, determined by a conventional steering mechanism. Its output space is $X_{MR} = \mathcal{R} \times \mathcal{R} \times \mathcal{R}$. We shall write (x, y, θ) for a typical member of X_{MR}, where (x, y) is the position of the robot in the plane, and θ is its orientation. All of these are functions of time.

Its GTF, T_{MR}, can be characterised by the differential equations

$$\frac{dx}{dt} = v(t)\cos\theta, \frac{dy}{dt} = v(t)\sin\theta$$

$$\frac{d\theta}{dt} = v(t)u_c(t), \frac{dv}{dt} = u_a(t)$$

For example, if we apply T_{MR} to the simple input function defined by $u_{simple}(t) = (0.1, 0)$, then

$$T_{MR}(u_{simple}, (10, 5, 0))(t) = (0.05t^2 + 10, 5, 0)$$

that is uniform acceleration along a straight line through the starting point (10,5,0) and parallel to the X-axis.

The guards and prisoners problem Three guards and three prisoners are on the left bank of a river, and need to cross over to the right. A boat is available on the left bank. It holds two people. Prisoners must not be allowed to outnumber guards. How can the party cross the river?[2]

To express the problem formally[3] we will need the following notation:

- if s is a finite sequence (or tuple), then we use $s_{i \leftarrow v}$ to mean that finite sequence which differs from s only at index i, where it has the value v. This is extended to multiple successive modifications. For example $s_{i \leftarrow v, j \leftarrow w}$ is $s'_{j \leftarrow w}$ where $s' = s_{i \leftarrow v}$. We write () for the empty sequence.
- There is a set of 2 boolean values $\{\mathbf{t}, \mathbf{f}\}$.
- There is a set of 3 *guards* $\{\mathbf{g1}, \mathbf{g2}, \mathbf{g3}\}$.
- There is a set of 3 *prisoners* $\{\mathbf{p1}, \mathbf{p2}, \mathbf{p3}\}$.
- An *occupant* can be EITHER a guard OR a prisoner OR **n** (indicating that a place is unoccupied). The first 3 places on each bank will, if occupied, be occupied by a guard. The remaining 3 places will, if occupied, be occupied by a prisoner.
- A *bank* is a sextuple of occupants.
- A *state* can be EITHER
  - A triple $(c, bank_1, bank_2)$ where the condition c is a boolean indicating whether the boat is on the left bank $c = \mathbf{t}$ or the right bank $c = \mathbf{f}$ and $bank_1$ and $bank_2$ each specify the occupants of the left and right banks respectively.
  - OR **b** indicating a *bad state* resulting from a physically impossible transition such as trying to move the occupant of an empty location. Other states are referred to as "good".

  We denote the set of states[4] by U_{GP}

To specify a move, we select one or two "places", that is indices into of the state vector, whose occupants may cross the river to the opposite bank. If any of the selected places is unoccupied, the move will be deemed physically impossible.

$$U_{move} = \{(i|i \in 1 \ldots 6)\} \cup \{(i, j|i, j \in 1 \ldots 6, i \neq j)\}$$

[2] Following Amarel, we are deliberately not using a concise representation of the state-space, for we wish to discuss how state-space can be contracted by the recognition of its symmetries. Our representation is arguably a natural one for a graphical presentation of the problem which is to be solved by a human interacting with a computer.

[3] This formalisation was guided by a definition of aspects of the problem written in the SML language [6].

[4] Good states and bad states are specified by a discriminated union in the SML formulation. The two cases given correspond to the two cases in the **datatype** declaration in the program

We may now define the presettable GTF which characterises the physically possible moves of the problem. We'll call it T_{moveGP}. To define it, we need a $Move$ function, which moves a single occupant from one bank to the other. Here $Move(i,x) = x'$ means that the output state x' is obtained from the output-state x by moving the occupant $o = b_i$ to the other side of the river, where b is the appropriate bank, indexed by i.

There are three main cases

- Case 1: $x = \mathbf{b}$ In this case $Move(i,x) = \mathbf{b}$. In other words, once a state is bad, it remains so.
- Case 2: $x = (\mathbf{t}, b, b')$ where b, b' are banks. So the boat is on the left bank since the first component of the state is $\mathbf{t}$.
  Let $o = b_i$. Let $o' = b'_i$ There are three sub-cases
  - Case 2.1: $o = \mathbf{n}$. To move a non-existent occupant is physically impossible, so $Move(i,x) = \mathbf{b}$.
  - Case 2.2: $o' \neq \mathbf{n}$. To move an occupant into an occupied location is impossible[5], so $Move(i,x) = \mathbf{b}$
  - Case 2.3 $o \neq \mathbf{n}$ $Move(i,x) = (\mathbf{f}, b_{i \leftarrow \mathbf{n}}, b_{i \leftarrow o})$
- Case 3: $x = (\mathbf{f}, b, b')$ where b, b' are banks.
  Let $o = b_i$. Let $o' = b'_i$ There are three sub-cases
  - Case 3.1: $o' = \mathbf{n}$. To move a non-existent occupant is physically impossible, so $Move(i,x) = \mathbf{b}$.
  - Case 3.2: $o \neq \mathbf{n}$. To move an occupant into an occupied location is impossible, so $Move(i,x) = \mathbf{b}$
  - Case 3.3 $o' \neq \mathbf{n}$ $Move(i,x) = (\mathbf{f}, b_{i \leftarrow o}, b_{i \leftarrow \mathbf{n}})$

Now let's define $MoveAll$ which operates on the members of U_{move}.

$$MoveAll((),x) = x, \quad MoveAll((i,u_1 \ldots u_n,x) = Move(i, MoveAll((u_1 \ldots u_n),x))$$

We can now define the GTF T_{moveGP} with index-set the non-negative integers as follows:

$$T_{moveGP}(\mathbf{u},(p_1,p_2,p_3,p_4,p_5,p_6))(0) = (\mathbf{t},(p_1,p_2,p_3,p_4,p_5,p_6)(\mathbf{n},\mathbf{n},\mathbf{n},\mathbf{n},\mathbf{n},\mathbf{n}))$$

Provided that $p_1 \ldots p_3 \in Guards$ and $p_4 \ldots p_6 \in Prisoners$ — otherwise $T_{moveGP}(\mathbf{u},(p_1,p_2,p_3,p_4,p_5,p_6))(0) = \mathbf{b}$ That is, initially everybody is on the left bank.

For $t > 0$ let's suppose that $x = T_{moveGP}(\mathbf{u},p)(t-1)$.

$$T_GP(\mathbf{u},p)(t) = MoveAll(\mathbf{u}(t-1),x)$$

To meet the conditions of the problem we can classify a state as either legal $\mathbf{l}$ (so that neither on the left bank nor on the right bank are the guards outnumbered) or illegal $\mathbf{i}$ (in which the guards are outnumbered on the left bank or on

[5] With a standard initial condition in which everybody is on one bank it's not possible to move an occupant into an occupied location.

the right bank), or terminal $\mathbf{t}$ which is a legal state with everybody on the right bank.

The domain

$$U_{eval} = \{\mathbf{l}, \mathbf{i}, \mathbf{t}\}$$

will be the output domain of a GTF which classifies a given situation as either legal, illegal or terminal.

We can also define the GTF T_{evalGP} which evaluates a state arising from a move in the guards-and-prisoners problem.

To evaluate $T_{evalGP}(\mathbf{x}, p)(t)$, let $(c, b, b') = \mathbf{x}(t)$. Then

- $T_{evalGP}(\mathbf{x}, p)(t) = \mathbf{t}$ if $b = \emptyset$
- $T_{evalGP}(\mathbf{x}, p)(t) = \mathbf{i}$ if $OutNumbered(b) = \mathbf{t}$
- $T_{evalGP}(\mathbf{x}, p)(t) = \mathbf{l}$ otherwise

The $OutNumbered$ function applied to a bank b evaluates to $\mathbf{t}$ if there are guards on b, and they are fewer in number than the prisoners on b.

Input symmetries of T_{evalGP} Suppose our set of operators Σ contains the symmetric group S_{Guards}, which acts on the space U_{GP} by permuting the guards on each bank. Then S_{Guards} is a group of input symmetries of T_{evalGP}. Likewise if Σ contains the symmetric group $S_{Prisoners}$ then $S_{Prisoners}$ is a group of input symmetries of T_{evalGP}.

That is to say, if we take any member of $U_{stateGP}$ and permute the prisoners and/or the guards, that state will receive the same evaluation under T_{evalGP}.

5.1 Full Symmetries

We have seen so far symmetries of the inputs and of the outputs of GTF's. However it is frequently the case that a GTF has a symmetry that affects both input and output. For example, any device with a bilateral symmetry will have a symmetry operator that, in some sense, interchanges left and right. Applying such an operator on the input space (so that left and right are interchanged on input) will give rise to an output that has left and right interchanged. A left-right interchange on the preset specification will also be needed.

For example, in the problem of balancing an inverted pendulum, a left-right interchange will involve changing the sign of the input command to the motor driving the system. Provided the preset (the initial position and velocity of the pendulum) is also mapped by a left-right interchange, the output behaviour will also exhibit a left-right interchange.

In our mobile robot example the transformation $u_c \mapsto -u_c$ is a left-right symmetry operator applied to the input. The output-space, which is also the preset-space, may be mapped by $y \mapsto -y$, $\theta \mapsto -\theta$. If we apply these mappings to the input-space and the ouput-space the behaviour of the system is described by the same transfer function. This is not the only such symmetry operator — any reflection operator on the output-space gives rise to a left-right symmetry, provided we correctly map θ.

Definition 51 *Let T be a GTF. Let σ be an invertible operator which acts on the input domain U_{in} of T, the output domain U_{out} of T, and the preset domain P of T. We say that σ is a* full symmetry *of T if*

$$T(\mathbf{u}.\sigma, p.\sigma) = T(\mathbf{u}, p).\sigma$$

Proposition 52 *Let T be a GTF with input domain U_{in} and output domain U_{out} and preset domain P. Let Σ be a set of operators acting on these domains. Then the set of full symmetry operators on T form a group.*

Proof: Let $\sigma_1, \sigma_2 \in \Sigma$ be full symmetry operators on T. Then

$$\begin{aligned} T(\mathbf{u}.(\sigma_1\sigma_2), p.(\sigma_1\sigma_2)) &= T((\mathbf{u}.\sigma_1).\sigma_2, (p.\sigma_1).\sigma_2) = T(\mathbf{u}.\sigma_1, p.\sigma_1).\sigma_2 \\ &= T(\mathbf{u}, p).\sigma_1).\sigma_2 = T(\mathbf{u}, p).(\sigma_1\sigma_2) \end{aligned}$$

Hence $\sigma_1\sigma_2$ is a full symmetry operator on T.
Let σ be a full symmetry operator on T

$$T(\mathbf{u}.\sigma^{-1}.\sigma, p.\sigma^{-1}.\sigma) = T(\mathbf{u}.\sigma^{-1}, p.\sigma^{-1}).\sigma$$

Now consider

$$\begin{aligned} T(\mathbf{u}.\sigma^{-1}.\sigma, p.\sigma^{-1}.\sigma).\sigma^{-1} &= T(\mathbf{u}.\sigma^{-1}, p.\sigma^{-1}).\sigma.\sigma^{-1} = T(\mathbf{u}.\sigma^{-1}, p.\sigma^{-1}).(\sigma\sigma^{-1}) \\ &= T(\mathbf{u}.\sigma^{-1}, p.\sigma^{-1}) \end{aligned}$$

However

$$\begin{aligned} T(\mathbf{u}.\sigma^{-1}.\sigma, p.\sigma^{-1}.\sigma).\sigma^{-1} &= T(\mathbf{u}.(\sigma^{-1}\sigma), p.(\sigma^{-1}\sigma)).\sigma^{-1} = T(\mathbf{u}.1, p.1).\sigma^{-1} \\ &= T(\mathbf{u}, p).\sigma^{-1} \end{aligned}$$

So we've shown that

$$T(\mathbf{u}.\sigma^{-1}, p.\sigma^{-1}) = T(\mathbf{u}, p).\sigma^{-1}$$

Hence σ^{-1} is a full symmetry operator on T.
We will write $\mathcal{G}_\Sigma T$ for the group of full symmetries of T.

We can regard input symmetries of a GTF as a special case of full symmetries in which the operator acts as the identity operation on the output and preset spaces. Likewise we can regard output symmetries of a presettable GTF as a special case of full symmetries in which the operator acts as the identity operation on the input space.

In the Prisoners and Guards world, $\Sigma = S_{Guards} \cup S_{Prisoners} \cup S_{\{1\ldots3\}} \cup S_{\{4\ldots6\}}$ so that Σ includes operators permutating the guards, the prisoners, the guards' places and the prisoners' places. All three of the operator subgroups above naturally map the domains U_{GP}, U_{move} and (trivially) U_{eval}. Each is a group of full symmetries of the GTF's T_{moveGP} and T_{evalGP}.

6 Cascaded GTF's

Definition 61 *A transfer function T is said to be a* transducer *if there is a function f for which $T(\mathbf{u}, x_0)(t) = f(\mathbf{u})$.*

Thus a transducer (in our sense) is a transfer function whose output depends only on the instantaneous value of its input.

Definition 62 *The identity transducer is the map defined by*

$$I(\mathbf{u}, ()) = \mathbf{u}$$

Definition 63 *Let T_1, T_2 be general transfer functions. Then the product T_1T_2 is defined by*

$$(T_1T_2)(\mathbf{u}, p_1p_2) = T_1(T_2(\mathbf{u}, p_2), p_1)$$

Note that the factorisation of a sequence p into p_1p_2 is unique because the sequence-length in a given initialiser domain is fixed. Clearly, the product of GTFs is a GTF.

Proposition 64 *The product of transfer functions is associative, with the identity transducer as its identity.*

Proof:

$$(T_1(T_2T_3))(\mathbf{u}, p_1p_2p_3)) = T_1((T_2T_3)(\mathbf{u}, p_2p_3), p_1) = T_1(T_2(T_3(\mathbf{u}, p_3), p_2), p_1)$$

while

$$((T_1T_2)T_3)(\mathbf{u}, p_1p_2p_3)) = (T_1T_2)(T_3(\mathbf{u}, p_3), p_1p_2) = T_1(T_2(T_3(\mathbf{u}, p_3), p_2), p_1)$$

Moreover $(IT)(\mathbf{u}, p) = I(T(\mathbf{u}, p), ()) = T(\mathbf{u}, p), \qquad (TI)(\mathbf{u}, p) = T(I(\mathbf{u}, ()), p) = T(u, p)$.

Proposition 65 *Let T_1, T_2 be GTF's for which U_1 is the input space of T_1, U_2 is the output space of T_1 and the input space of T_2, while U_3 is the output space of T_2. Let Σ_1 be a group of full symmetries of T_1, while Σ_2 is a group of full symmetries of T_2. Then $\Sigma_1 \cap \Sigma_2$ is a group of full symmetries of T_2T_1.*

Proof: Let $\sigma \in \Sigma_1 \cap \Sigma_2$. Consider

$$\begin{aligned}(T_2T_1)(\mathbf{u}.\sigma, (p_1p_2).\sigma) &= T_2(T_1(\mathbf{u}.\sigma, p_1.\sigma), p_2.\sigma) = T_2(T_1(\mathbf{u}, p_1).\sigma, p_2.\sigma) \\ &= T_2(T_1(\mathbf{u}, p_1), p_2).\sigma = (T_2T_1)(\mathbf{u}, p_1p_2).\sigma\end{aligned}$$

Thus σ is a full symmetry of T_1T_2.

For example consider a mobile robot, equipped with a camera, on an infinite plane on which a straight line is painted. A symmetry group of the whole system is the intersection of the output-symmetries of the mechanics with the input symmetries of the camera as it views the line.

7 Taking the Quotient Simplifies Transfer Functions

Proposition 71 *Any group of operators Σ' defines an equivalence relationship on a domain on which it operates.*

$$u \equiv u' \Leftrightarrow \exists \sigma \in \Sigma',\ \ u' = u.\sigma$$

Definition 72 *Let U be a domain, P be a preset domain, $\Sigma' \subset \Sigma$ a group of operators on U. Then we write U/Σ' for the set of classes of members of U equivalent under Σ'. Also we write $(U \times P)/\Sigma'$ for the set of classes of members of $U \times P$ equivalent under Σ'*[6].

We write $\bar{u}$ for the equivalence class corresponding to u The mapping $u \mapsto \bar{u}$ is treated as the operator $1/\Sigma'$.

For finite domains, the advantage of going to quotient domains is that the size of the search space required to invert a transfer function for the purpose of creating a regulator or an open-loop controller is reduced. In [11] being able to take the quotient of the Special Euclidean Group by the group of translations proved a useful way of simplifying the robotic assembly problems studied in that paper.

Proposition 73 *Let $\Sigma' \subset \Sigma$ be a group of full symmetries of a transfer function T whose input space is U, whose output space is X and whose preset space is P. Let $\theta = 1/\Sigma'$. Then the function T/Σ' defined by $(T/\Sigma')(\mathbf{u}.\theta, p.\theta) = T(\mathbf{u}, p).\theta$ is a GTF.*

Proof: The only thing we have to show is that the mapping is well defined. Suppose $(\mathbf{u}, p) \equiv (\mathbf{u}', p')$. Then $\mathbf{u}' = \mathbf{u}.\sigma, p' = p.\sigma$

$$(T/\Sigma')(\mathbf{u}'.\theta, p'.\theta) = T(\mathbf{u}', p').\theta = T(\mathbf{u}.\sigma, p.\sigma).\theta = T(\mathbf{u}, p).\sigma.\theta = T(\mathbf{u}, p).\theta$$

7.1 The Quotient of The Guards and Prisoners Problem

In taking the quotient domain U_{move}/Σ' the guards' places are equivalent and so are the prisoners' places. Thus the possible moves are $(\bar{1})$ meaning "1 guard crosses", $(\bar{1}, \bar{1})$ meaning "2 guards cross", $(\bar{1}, \bar{4})$ and $(\bar{4}, \bar{1})$ meaning "1 guard and one prisoner cross", $(\bar{4}$ meaning "one prisoner crosses" and $(\bar{4}, \bar{4})$ meaning "two prisoners cross". Thus the size of the input space is reduced from 36 for T_{moveGP} to 6 for T_{moveGP}/Σ'. [7]

Reducing the size of the input domain makes a significant reduction in the size of the search-space for a sequence of inputs that will produce the desired terminal output $\mathbf{t}$ — the fan-out is divided by 6 at each stage.

[6] We are extending the operator set to act on the cartesian product in the obvious way

[7] Further condensation of the input space is possible if we note that permutation of the entries in an input-tuple is an input symmetry of T_{moveGP}

Moreover the state-space is also shrunk by taking the quotient. In the original formulation there were $2 \times (6 \times 2^3)^2 = 4608$ states that could be reached from possible initial states. These are shrunk down to $3 * 3 * 2 = 18$ possible states in the quotient domain, for any two states that have the same number of guards on the left bank and have the same number of prisoners on the left bank and have the boat in the same place will be equivalent under Σ'.

8 Summary — Future Work

In this paper we have developed the concept of a generalised transfer function, illustrating how the concept encompasses both discrete and continuous systems. We have developed the concept of symmetry of a GTF which, in its most general form, tells us how modifications of the input (and preset) of a GTF will affect its output. Symmetry thus has the potential to play the role of differentiation in classical control theory.

A strong motivation for generalisation is that biological adaptive systems appear to operate both in continuous and discrete domains. While there is a continuum of configurations of the human body, human communication takes place in a discretised vocabulary of words which are used *inter alia* to discuss actions and which arguably may characterise aspects of the mental processes underlying actions.

A potential bridge between continuous and discrete domains is the *quotient* operation, since it supports the collapse of a continuous domain into a discrete one, for example on the basis of topological invariants.

The value of the *generalisation* depends on whether it can be used as the basis of synthesis and analysis of reactive systems. We might wish to analyse a GTF from and *extensive* or *intensive* point of view. The extensive (or "white box") approach supposes we have a definition of a GTF that is open for inspection, so that its symmetries can be inferred from its definition. The intensive (or "black box") approach requires the formulation of a characterisation of a GTF by observing its behaviour.

One question that has been left unexplored is "Of what class of inputs is a given GTF a symmetry operator?" This is crucial to the understanding of linear systems whose analysis depends on the observation that a linear GTF is a scale symmetry of the exponential function over the complex domain.

References

1. Amarel, S. "On Representations of Problems of Reasoning About Actions", Machine Intelligence 3, Michie (ed), Edinburgh University Press, 1968; reprinted in "Readings in Artificial Intelligence", by B.L.Webber and N.J.Nilsson, Tioga, 1981.
2. Arnold,V.I. [1988] "Geometrical Methods in the Theory of Ordinary Differential Equations" (in English translation) Springer Verlag.
3. Clarke E.M., Grumberg O, Peled D.A.[1999] "Model Checking", MIT Press.
4. Hayes P,[1979] "The Naive Physics Manifesto" in D. Michie (Ed.) Expert Systems. Edinburgh U. Press.

5. Leyton M [1992] "Symmetry, Causality, Mind", MIT Press.
6. Milner R., Tofte M., Harper R., MacQueen D [1997] The Definition of Standard ML (Revised)
7. Liu Y., Popplestone R.[1994] "A Group Theoretic Formalization of Surface Contact", I.J.R.R. Vol. 13 No. 2.
8. E. Noether [1918], "Invariante Variationsprobleme", Nachr. d. König. Gesellsch. d. Wiss. zu Gottingen, Math-phys. Klasse, 235-257.
9. Popplestone R.J., Liu Y., and Weiss R.[1990] "A Group Theoretic Approach to Assembly Planning." *Artificial Intelligence Magazine*, Spring 1990, Vol. 11, No. 1, pp. 82-97.
10. Y.Liu, "Symmetry Groups in Robotic Assembly Planning", *COINS Technical Report 90-83* Computer and Information Sciece Dept, University of Massachusetts.
11. Popplestone,R.J., Ambler A.P., and Bellos, I. 1980, "An Interpreter for a Language for Describing Assemblies", *Artificial Intelligence* Vol. 14 No. 1, pp 79-107
12. Popplestone,R.J, 1984, Group Theory and Robotics, Robotics Research: The First International Symposium, (eds Brady,M. and Paul,R.), MIT Press, Cambridge MA and London.
13. Zahnd,A., Nair,S. and Popplestone, R.J. [1989] "Symmetry Inference in Planning Assembly", Proc. 5th ISRR, Tokyo Japan.