ELEC4410

Control Systems Design

Lecture 3, Part 2: Introduction to Affine Parametrisation

School of Electrical Engineering and Computer Science
The University of Newcastle

Outline

- Here we develop a novel way of expressing a control transfer function.

Outline

- Here we develop a novel way of expressing a control transfer function.
- We will see that this novel parametrisation leads to deep insights into control system design and reinforces, from an alternative perspective, ideas that have been previously studied.

Outline

- Here we develop a novel way of expressing a control transfer function.
- We will see that this novel parametrisation leads to deep insights into control system design and reinforces, from an alternative perspective, ideas that have been previously studied.
- The key feature of this parametrisation is that it renders the closed loop sensitivity functions linear (or more correctly, affine) in a design variable.

Outline

- Here we develop a novel way of expressing a control transfer function.
- We will see that this novel parametrisation leads to deep insights into control system design and reinforces, from an alternative perspective, ideas that have been previously studied.
- The key feature of this parametrisation is that it renders the closed loop sensitivity functions linear (or more correctly, affine) in a design variable.
- We thus call this ‘Affine Parametrisation’.

Outline

The main ideas to be presented include:

- Motivation for the affine parametrisation from the idea of open loop inversion.

Outline

The main ideas to be presented include:

- Motivation for the affine parametrisation from the idea of open loop inversion.
- Affine parametrisation and Internal Model Control.

Outline

The main ideas to be presented include:

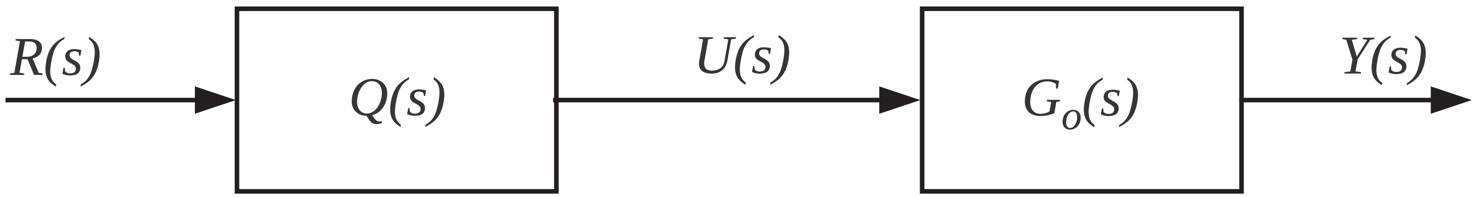
- Motivation for the affine parametrisation from the idea of open loop inversion.
- Affine parametrisation and Internal Model Control.
- Affine parametrisation and performance specifications.

Open Loop Inversion Revisited

- Recall that control implicitly and explicitly depends on plant model inversion. This is best seen in the case of open loop control.

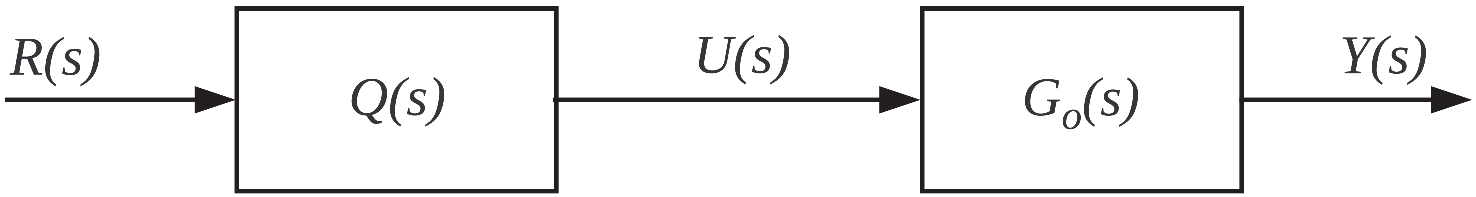
Open Loop Inversion Revisited

- Recall that control implicitly and explicitly depends on plant model inversion. This is best seen in the case of open loop control.
- In open loop control the input, $U(s)$, is generated from the reference signal $R(s)$, by a transfer function $Q(s)$, i.e. $U(s) = Q(s)R(s)$.



Open Loop Inversion Revisited

- Recall that control implicitly and explicitly depends on plant model inversion. This is best seen in the case of open loop control.
- In open loop control the input, $U(s)$, is generated from the reference signal $R(s)$, by a transfer function $Q(s)$, i.e. $U(s) = Q(s)R(s)$.



- This leads to an input-output transfer function of the following form:

$$T_o(s) = G_o(s)Q(s).$$

Open Loop Inversion Revisited

- This simple formula highlights the fundamental importance of inversion, as $T_o(j\omega)$ will be 1 only at those frequencies where $Q(j\omega)$ inverts the model. Note that this is consistent with the prototype solution to the control problem described in earlier lectures.

Open Loop Inversion Revisited

- This simple formula highlights the fundamental importance of inversion, as $T_o(j\omega)$ will be 1 only at those frequencies where $Q(j\omega)$ inverts the model. Note that this is consistent with the prototype solution to the control problem described in earlier lectures.
- A key point is that $T_o(s) = G_o(s)Q(s)$ is affine in $Q(s)$.

Open Loop Inversion Revisited

- This simple formula highlights the fundamental importance of inversion, as $T_o(j\omega)$ will be 1 only at those frequencies where $Q(j\omega)$ inverts the model. Note that this is consistent with the prototype solution to the control problem described in earlier lectures.
- A key point is that $T_o(s) = G_o(s)Q(s)$ is affine in $Q(s)$.
- On the other hand, with a conventional feedback controller, $C(s)$, the closed loop transfer function has the form

$$T_o(s) = \frac{G_o(s)C(s)}{1 + G_o(s)C(s)}.$$

Open Loop Inversion Revisited

- This simple formula highlights the fundamental importance of inversion, as $T_o(j\omega)$ will be 1 only at those frequencies where $Q(j\omega)$ inverts the model. Note that this is consistent with the prototype solution to the control problem described in earlier lectures.
- A key point is that $T_o(s) = G_o(s)Q(s)$ is affine in $Q(s)$.
- On the other hand, with a conventional feedback controller, $C(s)$, the closed loop transfer function has the form

$$T_o(s) = \frac{G_o(s)C(s)}{1 + G_o(s)C(s)}.$$

- The above expression is nonlinear in $C(s)$.

Open Loop Inversion Revisited

- Comparing the two previous equations, we see that the former affine relationship holds if we simply parameterise $C(s)$ in the following fashion:

$$Q(s) = \frac{C(s)}{1 + G_o(s)C(s)}.$$

Open Loop Inversion Revisited

- Comparing the two previous equations, we see that the former affine relationship holds if we simply parameterise $C(s)$ in the following fashion:

$$Q(s) = \frac{C(s)}{1 + G_o(s)C(s)}.$$

- This is the essence of affine parametrisation.

Affine Parametrisation. The Stable Case

- We can invert the relationship given on the previous slide to express $C(s)$ in terms of $Q(s)$ and $G_o(s)$:

$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

Affine Parametrisation. The Stable Case

- We can invert the relationship given on the previous slide to express $C(s)$ in terms of $Q(s)$ and $G_o(s)$:

$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

- We will then work with $Q(s)$ as the design variable rather than the original $C(s)$.

Affine Parametrisation. The Stable Case

- We can invert the relationship given on the previous slide to express $C(s)$ in terms of $Q(s)$ and $G_o(s)$:

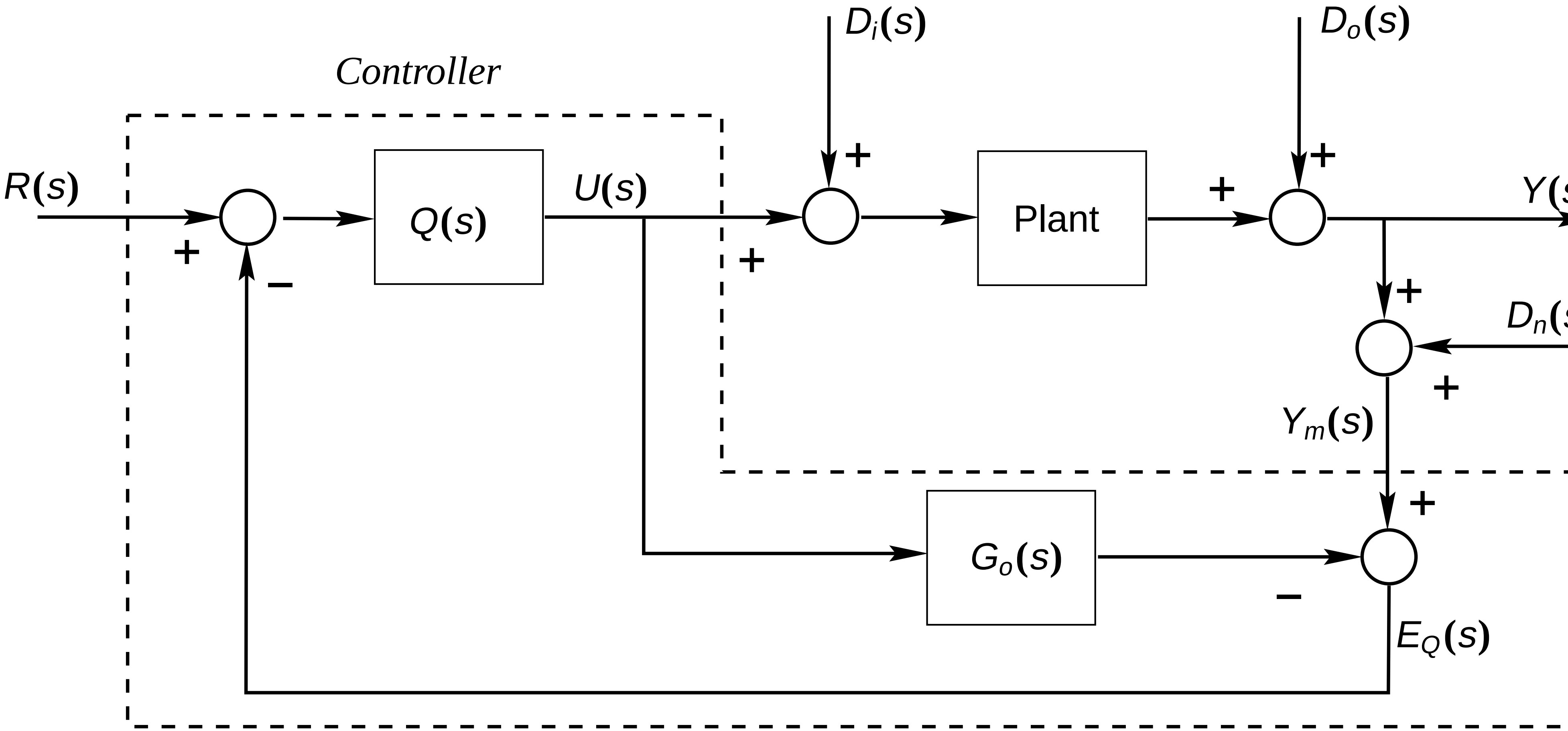
$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

- We will then work with $Q(s)$ as the design variable rather than the original $C(s)$.

- Note that the relationship between $C(s)$ and $Q(s)$ is one-to-one and thus there is no loss of generality in working with $Q(s)$.

Youla's parametrisation of all stabilising controllers for stable plants

This particular form of the controller, i.e. $C(s) = \frac{Q(s)}{1-Q(s)G_o(s)}$, can be drawn schematically as:



Stability

- Actually a very hard question to answer is:

  Given a stable transfer function $G_o(s)$, describe all controllers, $C(s)$, that stabilise this nominal plant.

Stability

- Actually a very hard question to answer is:

  Given a stable transfer function $G_o(s)$, describe all controllers, $C(s)$, that stabilise this nominal plant.

- However, it turns out that, in the $Q(s)$ form this question has a very simple answer, namely all that is required is that $Q(s)$ be stable.

Stability

Lemma. (*Affine parametrisation for stable systems*). Consider a plant having a stable nominal model $G_o(s)$ controlled in a one d.o.f. feedback architecture with a proper controller. Then the nominal loop is internally stable if and only if $Q(s)$ is any stable proper transfer function when the controller transfer function $C(s)$ is parameterised as:

$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

Stability

Proof. We note that the four sensitivity functions can be written as

$$
\begin{aligned}
T_o(s) &= Q(s)G_o(s) \\
S_o(s) &= 1 - Q(s)G_o(s) \\
S_{io}(s) &= (1 - Q(s)G_o(s))\, G_o(s) \\
S_{uo}(s) &= Q(s)
\end{aligned}
$$

We are for the moment only considering the case when $G_o(s)$ is stable. Then, we see that all of the above transfer functions are stable if and only if $Q(s)$ is stable.

Nominal Design

- For the nominal design case (i.e. no modelling errors) we recall that:

$$T_o(s) = Q(s)G_o(s)$$

$$S_o(s) = 1 - Q(s)G_o(s)$$

$$S_{io}(s) = (1 - Q(s)G_o(s))\, G_o(s)$$

$$S_{uo}(s) = Q(s).$$

Nominal Design

- For the nominal design case (i.e. no modelling errors) we recall that:

$$T_o(s) = Q(s)G_o(s)$$
$$S_o(s) = 1 - Q(s)G_o(s)$$
$$S_{io}(s) = (1 - Q(s)G_o(s))\, G_o(s)$$
$$S_{uo}(s) = Q(s).$$

- All of these equations are affine in $Q(s)$.

Nominal Design

- For the nominal design case (i.e. no modelling errors) we recall that:

$$T_o(s) = Q(s)G_o(s)$$
$$S_o(s) = 1 - Q(s)G_o(s)$$
$$S_{io}(s) = (1 - Q(s)G_o(s))\, G_o(s)$$
$$S_{uo}(s) = Q(s).$$

- All of these equations are affine in $Q(s)$.

- This makes design particularly straightforward.

Prototype Control Solution

- Specifically, if we look at $T_o(s)$,

$$T_o(s) = Q(s)G_o(s)$$

Prototype Control Solution

- Specifically, if we look at $T_o(s)$,

$$T_o(s) = Q(s)G_o(s)$$

- We recall that a reasonable design goal is to have $T_o(s)$ near 1 since this implies that the system output exactly follows the reference signal.

Prototype Control Solution

- Specifically, if we look at $T_o(s)$,

$$T_o(s) = Q(s)G_o(s)$$

- We recall that a reasonable design goal is to have $T_o(s)$ near 1 since this implies that the system output exactly follows the reference signal.
- Thus a prototype controller would seem to be to simply choose:

$$Q(s) = (G_o(s))^{-1} .$$

Prototype Control Solution

- Specifically, if we look at $T_o(s)$,

$$T_o(s) = Q(s)G_o(s)$$

- We recall that a reasonable design goal is to have $T_o(s)$ near 1 since this implies that the system output exactly follows the reference signal.
- Thus a prototype controller would seem to be to simply choose:

$$Q(s) = (G_o(s))^{-1}\,.$$

- Unfortunately, $(G_o(s))^{-1}$ is most likely to be improper in practice.

Design Considerations

- Hence we introduce a filter $F_Q(s)$ to keep $Q(s)$ proper.

Design Considerations

- Hence we introduce a filter $F_Q(s)$ to keep $Q(s)$ proper.
- It thus seems that a reasonable choice for $Q(s)$ might be:

$$Q(s) = F_Q(s)\,(G_o(s))^{-1}$$

where $(G_o(s))^{-1}$ is the exact inverse of $G_o(s)$.

Design Considerations

- Hence we introduce a filter $F_Q(s)$ to keep $Q(s)$ proper.
- It thus seems that a reasonable choice for $Q(s)$ might be:

$$Q(s) = F_Q(s)\,(G_o(s))^{-1}$$

  where $(G_o(s))^{-1}$ is the exact inverse of $G_o(s)$.

- Not unexpectedly, we see that inversion plays a central role in this prototype solution.

Design Considerations

- Hence we introduce a filter $F_Q(s)$ to keep $Q(s)$ proper.
- It thus seems that a reasonable choice for $Q(s)$ might be:

$$Q(s) = F_Q(s)\,(G_o(s))^{-1}$$

  where $(G_o(s))^{-1}$ is the exact inverse of $G_o(s)$.

- Not unexpectedly, we see that inversion plays a central role in this prototype solution.
- NOTE: In this case:
  $T_o(s) = Q(s)G_o(s) = F_Q(s)\,(G_o(s))^{-1}\,G_o(s) = F_Q(s)$.

Design Considerations

- Although the design proposed above is a useful starting point it will usually have to be refined to accommodate more detailed design considerations.
- In particular, we will investigate the following issues:
  1. Non-minimum phase zeros
  2. Model relative degree
  3. Disturbance rejection
  4. Control effort
  5. Robustness

1. Non-minimum Phase Zeros

- Recall that, provided $G_o(s)$ is stable, then $Q(s)$ only needs to be stable to ensure closed loop stability.

1. Non-minimum Phase Zeros

- Recall that, provided $G_o(s)$ is stable, then $Q(s)$ only needs to be stable to ensure closed loop stability.

- This implies that, if $G_o(s)$ contains NMP zeros, then they cannot be included in $(G_o(s))^{-1}$.

1. Non-minimum Phase Zeros

- Recall that, provided $G_o(s)$ is stable, then $Q(s)$ only needs to be stable to ensure closed loop stability.

- This implies that, if $G_o(s)$ contains NMP zeros, then they cannot be included in $(G_o(s))^{-1}$.

- One might therefore think of replacing the previous equation by:

$$Q(s) = F_Q(s)\,(G_o(s))^i$$

where $(G_o(s))^i$ is a stable approximation to $(G_o(s))^{-1}$.

1. Non-minimum Phase Zeros

- For example, if one factors $G_o(s)$ as:

$$G_o(s) = \frac{B_{os}(s)B_{ou}(s)}{A_o(s)}$$

where $B_{os}(s)$ and $B_{ou}(s)$ are the stable and unstable factors in the numerator, respectively, with $B_{ou}(0) = 1$.

1. Non-minimum Phase Zeros

- For example, if one factors $G_o(s)$ as:

$$G_o(s) = \frac{B_{os}(s)B_{ou}(s)}{A_o(s)}$$

where $B_{os}(s)$ and $B_{ou}(s)$ are the stable and unstable factors in the numerator, respectively, with $B_{ou}(0) = 1$.

- A suitable choice for $(G_o(s))^i$ would be

$$(G_o(s))^i = \frac{A_o(s)}{B_{os}(s)}.$$

2. Model Relative Degree

- To have a proper controller it is necessary that $Q(s)$ be proper.

2. Model Relative Degree

- To have a proper controller it is necessary that $Q(s)$ be proper.
- Thus it is necessary that the shaping filter, $F_Q(s)$, have a relative degree at least equal to the relative degree of $(G_o(s))^i$.

2. Model Relative Degree

- To have a proper controller it is necessary that $Q(s)$ be proper.
- Thus it is necessary that the shaping filter, $F_Q(s)$, have a relative degree at least equal to the relative degree of $(G_o(s))^i$.
- Conceptually, this can be achieved by including factors of the form $(\tau s + 1)^{n_d}$ where $\tau \in \mathbb{R}^+$ in the denominator.

3. Disturbance Rejection

- Recall, again, the following expressions for the closed loop sensitivity functions in terms of $Q(s)$:

$$T_o(s) = Q(s)G_o(s)$$
$$S_o(s) = 1 - Q(s)G_o(s)$$
$$S_{io}(s) = (1 - Q(s)G_o(s))\, G_o(s)$$
$$S_{uo}(s) = Q(s).$$

3. Disturbance Rejection

- Recall, again, the following expressions for the closed loop sensitivity functions in terms of $Q(s)$:

$$T_o(s) = Q(s)G_o(s)$$
$$S_o(s) = 1 - Q(s)G_o(s)$$
$$S_{io}(s) = (1 - Q(s)G_o(s))\, G_o(s)$$
$$S_{uo}(s) = Q(s).$$

- It would seem that to achieve perfect disturbance rejection at frequency ω_i simply requires that QG_o be 1 at ω_i.

3. Disturbance Rejection

- Recall, again, the following expressions for the closed loop sensitivity functions in terms of $Q(s)$:

$$T_o(s) = Q(s)G_o(s)$$
$$S_o(s) = 1 - Q(s)G_o(s)$$
$$S_{io}(s) = (1 - Q(s)G_o(s))\, G_o(s)$$
$$S_{uo}(s) = Q(s).$$

- It would seem that to achieve perfect disturbance rejection at frequency ω_i simply requires that QG_o be 1 at ω_i.
- For example, rejection of a d.c. disturbance requires $Q(0)G_o(0) = 1$.

3. Disturbance Rejection

- Once we have found one value of $Q(s)$ (say we call it $Q_a(s)$) that satisfies $G_o(0)Q_a(0) = 1$, then all possible controllers giving constant disturbance rejection can be described.

3. Disturbance Rejection

- Once we have found one value of $Q(s)$ (say we call it $Q_a(s)$) that satisfies $G_o(0)Q_a(0) = 1$, then all possible controllers giving constant disturbance rejection can be described.

- Consider a stable model $G_o(s)$ with input and/or output disturbance at zero frequency. Then, a one d.o.f. control loop, giving zero steady state tracking error, is stable if and only if the controller $C(s)$ can be expressed in the affine form where $Q(s)$ satisfies:

$$Q(s) = s\overline{Q}(s) + (G_o(s))^{-1} Q_a(s)$$

and $\overline{Q}(s)$ is any stable transfer function, and $Q_a(s)$ is any stable transfer function which satisfies $Q_a(0) = 1$.

3. Disturbance Rejection

- Once we have found one value of $Q(s)$ (say we call it $Q_a(s)$) that satisfies $G_o(0)Q_a(0) = 1$, then all possible controllers giving constant disturbance rejection can be described.

- Consider a stable model $G_o(s)$ with input and/or output disturbance at zero frequency. Then, a one d.o.f. control loop, giving zero steady state tracking error, is stable if and only if the controller $C(s)$ can be expressed in the affine form where $Q(s)$ satisfies:

$$Q(s) = s\overline{Q}(s) + (G_o(s))^{-1} Q_a(s)$$

  and $\overline{Q}(s)$ is any stable transfer function, and $Q_a(s)$ is any stable transfer function which satisfies $Q_a(0) = 1$.

- The above idea can be readily extended to cover rejection of disturbances at any frequency ω_i.

4. Control Effort

- We see that if we achieve $S_0 = 0$ at a given frequency, i.e. $QG_0 = 1$, then we have infinite gain in the controller C at the same frequency, i.e.

$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

4. Control Effort

- We see that if we achieve $S_0 = 0$ at a given frequency, i.e. $QG_0 = 1$, then we have infinite gain in the controller C at the same frequency, i.e.

$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

- For example, say the plant is minimum phase, then we could choose $(G_o(s))^i = (G_o(s))^{-1}$.

4. Control Effort

- We see that if we achieve $S_0 = 0$ at a given frequency, i.e. $QG_0 = 1$, then we have infinite gain in the controller C at the same frequency, i.e.

$$C(s) = \frac{Q(s)}{1 - Q(s)G_o(s)}.$$

- For example, say the plant is minimum phase, then we could choose $(G_o(s))^i = (G_o(s))^{-1}$.

- This gives the controller,

$$C(s) = \frac{F_Q(s)\,(G_o(s))^i}{1 - F_Q(s)}.$$

4. Control Effort

- By way of illustration, say that we choose

$$F_Q(s) = \frac{1}{(\tau s + 1)^r}$$

then, the high frequency gain of the controller, K_{hfc}, and the high frequency gain of the model, K_{hfg}, are related by:

$$K_{hfc} = \frac{1}{\tau^r K_{hfg}}$$

4. Control Effort

- By way of illustration, say that we choose

$$F_Q(s) = \frac{1}{(\tau s + 1)^r}$$

then, the high frequency gain of the controller, K_{hfc}, and the high frequency gain of the model, K_{hfg}, are related by:

$$K_{hfc} = \frac{1}{\tau^r K_{hfg}}$$

- Thus, as we make $F_Q(s)$ faster, i.e. τ becomes smaller, we see that K_{hfc} increases. This, in turn, implies that the control energy will increase.

4. Control Effort

- By way of illustration, say that we choose

$$F_Q(s) = \frac{1}{(\tau s + 1)^r}$$

then, the high frequency gain of the controller, K_{hfc}, and the high frequency gain of the model, K_{hfg}, are related by:

$$K_{hfc} = \frac{1}{\tau^r K_{hfg}}$$

- Thus, as we make $F_Q(s)$ faster, i.e. τ becomes smaller, we see that K_{hfc} increases. This, in turn, implies that the control energy will increase.
- This consequence can be appreciated from the fact that, under the assumption $G_o(s)$ is minimum phase and stable, we have

$$S_{uo}(s) = Q(s) = \frac{(G_o(s))^{-1}}{(\tau s + 1)^r}$$

5. Robustness

- Finally, we turn to the issue of robustness in choosing $Q(s)$.

5. Robustness

- Finally, we turn to the issue of robustness in choosing $Q(s)$.
- We recall that a fundamental result is that, in order to ensure robustness, the closed loop bandwidth should be such that the frequency response $|T_o(j\omega)|$ rolls off before the effects of modelling errors become significant.

5. Robustness

- Finally, we turn to the issue of robustness in choosing $Q(s)$.
- We recall that a fundamental result is that, in order to ensure robustness, the closed loop bandwidth should be such that the frequency response $|T_o(j\omega)|$ rolls off before the effects of modelling errors become significant.
- Thus, in the framework of the affine parametrisation under discussion here, the robustness requirement can be satisfied if $F_Q(s)$ reduces the gain of $T_o(j\omega)$ at high frequencies.

5. Robustness

- Finally, we turn to the issue of robustness in choosing $Q(s)$.
- We recall that a fundamental result is that, in order to ensure robustness, the closed loop bandwidth should be such that the frequency response $|T_o(j\omega)|$ rolls off before the effects of modelling errors become significant.
- Thus, in the framework of the affine parametrisation under discussion here, the robustness requirement can be satisfied if $F_Q(s)$ reduces the gain of $T_o(j\omega)$ at high frequencies.
- This is usually achieved by including appropriate poles in $F_Q(s)$.

Choice of Q. Summary for the case of Stable Open Loop Poles

- We have seen that a prototype choice for $Q(s)$ is simply the inverse of the open loop plant transfer function $G_o(s)$.

Choice of Q. Summary for the case of Stable Open Loop Poles

- We have seen that a prototype choice for $Q(s)$ is simply the inverse of the open loop plant transfer function $G_o(s)$.
- However, this ideal solution needs to be modified in practice to account for the following:

Choice of Q. Summary for the case of Stable Open Loop Poles

- We have seen that a prototype choice for $Q(s)$ is simply the inverse of the open loop plant transfer function $G_o(s)$.
- However, this ideal solution needs to be modified in practice to account for the following:
  - **Non-minimum phase zeros**. Internal stability precludes the cancellation of these zeros. They must therefore appear in $T_o(s)$. This implies that the gain of $Q(s)$ must be reduced at these frequencies for robustness reasons.

Choice of Q. Summary for the case of Stable Open Loop Poles

- We have seen that a prototype choice for $Q(s)$ is simply the inverse of the open loop plant transfer function $G_o(s)$.
- However, this ideal solution needs to be modified in practice to account for the following:
  - **Non-minimum phase zeros**. Internal stability precludes the cancellation of these zeros. They must therefore appear in $T_o(s)$. This implies that the gain of $Q(s)$ must be reduced at these frequencies for robustness reasons.
  - **Relative degree**. Excess poles in the model must necessarily appear as a lower bound for the relative degree of $T_o(s)$, since $Q(s)$ must be proper to ensure that the controller $C(s)$ is proper.

Choice of Q. Summary for the case of Stable Open Loop Poles

- (cont.)

Choice of Q. Summary for the case of Stable Open Loop Poles

- (cont.)
  - **Disturbance trade-offs**. Whenever we roll $T_o(s)$ off to satisfy measurement noise rejection, we necessarily increase sensitivity to output disturbances at that frequency. Also, slow open loop poles must either appear as poles of $S_{io}(s)$ or as zeros of $S_o(s)$, and in either case there is a performance penalty.

Choice of Q. Summary for the case of Stable Open Loop Poles

- (cont.)
  - **Disturbance trade-offs**. Whenever we roll $T_o(s)$ off to satisfy measurement noise rejection, we necessarily increase sensitivity to output disturbances at that frequency. Also, slow open loop poles must either appear as poles of $S_{io}(s)$ or as zeros of $S_o(s)$, and in either case there is a performance penalty.
  - **Control energy**. All plants are typically low pass. Hence, any attempt to make $Q(s)$ close to the model inverse necessarily gives a high pass transfer function from $D_o(s)$ to $U(s)$. This will lead to large input signals and may lead to controller saturation.

Choice of Q. Summary for the case of Stable Open Loop Poles

- (cont.)
  - **Disturbance trade-offs**. Whenever we roll $T_o(s)$ off to satisfy measurement noise rejection, we necessarily increase sensitivity to output disturbances at that frequency. Also, slow open loop poles must either appear as poles of $S_{io}(s)$ or as zeros of $S_o(s)$, and in either case there is a performance penalty.
  - **Control energy**. All plants are typically low pass. Hence, any attempt to make $Q(s)$ close to the model inverse necessarily gives a high pass transfer function from $D_o(s)$ to $U(s)$. This will lead to large input signals and may lead to controller saturation.
  - **Robustness**. Modeling errors usually become significant at high frequencies, and hence to retain robustness it is necessary to attenuate T_o, and hence Q, at these frequencies.

Summary of results for stable systems

- $C(s) = Q(s)\,(1 - Q(s)G_o(s))^{-1}$, where the design is carried out by designing the transfer function $Q(s)$.

Summary of results for stable systems

- $C(s) = Q(s)\,(1 - Q(s)G_o(s))^{-1}$, where the design is carried out by designing the transfer function $Q(s)$.
- Nominal sensitivities:

$$
\begin{aligned}
T_o(s) &= Q(s)G_o(s) \\
S_o(s) &= 1 - Q(s)G_o(s) \\
S_{io}(s) &= (1 - Q(s)G_o(s))\,G_o(s) \\
S_{uo}(s) &= Q(s)
\end{aligned}
$$

Design Methodology

- Determine the design specifications. (e.g. What do you want the B.W. of $T_o(s)$ to be?)

Design Methodology

- Determine the design specifications. (e.g. What do you want the B.W. of $T_o(s)$ to be?)
- Invert the nominal model $G_o(s)$. (Do you need to take an approximate inverse?)

Design Methodology

- Determine the design specifications. (e.g. What do you want the B.W. of $T_o(s)$ to be?)
- Invert the nominal model $G_o(s)$. (Do you need to take an approximate inverse?)
- Recall $Q(s) = F_Q(s)G_o^{-i}$. Determine the relative degree of $F_Q(s)$ to ensure $Q(s)$ is proper.

Design Methodology

- Determine the design specifications. (e.g. What do you want the B.W. of $T_o(s)$ to be?)
- Invert the nominal model $G_o(s)$. (Do you need to take an approximate inverse?)
- Recall $Q(s) = F_Q(s)G_o^{-i}$. Determine the relative degree of $F_Q(s)$ to ensure $Q(s)$ is proper.
- Specify the parameters of $F_Q(s)$ to satisfy the design specification.

Design Methodology

- Determine the design specifications. (e.g. What do you want the B.W. of $T_o(s)$ to be?)
- Invert the nominal model $G_o(s)$. (Do you need to take an approximate inverse?)
- Recall $Q(s) = F_Q(s)G_o^{-i}$. Determine the relative degree of $F_Q(s)$ to ensure $Q(s)$ is proper.
- Specify the parameters of $F_Q(s)$ to satisfy the design specification.
- Design $Q(s)$.

Design Methodology

- Determine the design specifications. (e.g. What do you want the B.W. of $T_o(s)$ to be?)
- Invert the nominal model $G_o(s)$. (Do you need to take an approximate inverse?)
- Recall $Q(s) = F_Q(s)G_o^{-i}$. Determine the relative degree of $F_Q(s)$ to ensure $Q(s)$ is proper.
- Specify the parameters of $F_Q(s)$ to satisfy the design specification.
- Design $Q(s)$.
- Convert to $C(s)$ if required.

Summary of results for stable systems

- Observe the following advantages of the affine parametrisation:

Summary of results for stable systems

- Observe the following advantages of the affine parametrisation:
  - Nominal stability is explicit.

Summary of results for stable systems

- Observe the following advantages of the affine parametrisation:
  - Nominal stability is explicit.
  - The known quantity G_o and the quantity sought by the control engineer (Q) occur in the highly insightful relation $T_o(s) = Q(s)G_o(s)$ (multiplicative in the frequency domain); whether a designer chooses to work in this quantity from the beginning or prefers to start with a synthesis technique and then convert, the simple multiplicative relation $Q(s)G_o(s)$ provides deep insights into the trade-offs of a particular problem and provides a very direct means of pushing the design by shaping Q.

Summary of results for stable systems

- Observe the following advantages of the affine parametrisation:
  - Nominal stability is explicit.
  - The known quantity G_o and the quantity sought by the control engineer (Q) occur in the highly insightful relation $T_o(s) = Q(s)G_o(s)$ (multiplicative in the frequency domain); whether a designer chooses to work in this quantity from the beginning or prefers to start with a synthesis technique and then convert, the simple multiplicative relation $Q(s)G_o(s)$ provides deep insights into the trade-offs of a particular problem and provides a very direct means of pushing the design by shaping Q.
  - The sensitivities are affine in Q, which is a great advantage for synthesis techniques relying on numerical minimisation of a criterion.

Summary of results for stable systems

- The following points are important to avoid some common misconceptions:

Summary of results for stable systems

- The following points are important to avoid some common misconceptions:
  - The associated trade-offs are not a consequence of the affine parametrisation: they are general and hold for any linear time invariant controller including PID, pole placement, LQR, H_∞, etc.

Summary of results for stable systems

- The following points are important to avoid some common misconceptions:
  - The associated trade-offs are not a consequence of the affine parametrisation: they are general and hold for any linear time invariant controller including PID, pole placement, LQR, H_∞, etc.
  - Affine parametrisation makes the general trade-offs more visible and provides a direct means for the control engineer to make trade-off decisions; this should not be confused with synthesis techniques that make particular choices in the affine parametrisation to synthesise a controller.

Summary of results for stable systems

- The following points are important to avoid some common misconceptions:
  - The associated trade-offs are not a consequence of the affine parametrisation: they are general and hold for any linear time invariant controller including PID, pole placement, LQR, H_∞, etc.
  - Affine parametrisation makes the general trade-offs more visible and provides a direct means for the control engineer to make trade-off decisions; this should not be confused with synthesis techniques that make particular choices in the affine parametrisation to synthesise a controller.
  - The fact that Q must approximate the inverse of the model at frequencies where the sensitivity is meant to be small is perfectly general and highlights the fundamental importance of inversion in control.